Published on *CRISMA catalogue*

(**https://crisma-cat.ait.ac.at**)

Home > CRISMA Catalogue book

CRISMA Catalogue book

Title: Catalogue of CRISMA Applications, Framework Building Block Specifications and Software Implementations

Editors: **Havlik, D.**, **Dihé, P.**, **Frings, S.**, **Steinnocher, K.**, **Aubrecht, C.**

Authors: Denis Havlik, Pascal Dihé, Sandra Frings, Klaus Steinnocher, Christoph Aubrecht, Miguel Almeida, Armonie Cossalter, Mehdi Pierre Daou, Oren Deri, Maria Egly, Alexander Garcia, Marianne Grisel, Arnaud De Groof, Hanna Honkavuo, Hermann Huber, Heinrich Humer, Markus Jähi, Frank Jonat, Tomer Kaplan, Kosonen, Susanna Kunttu, Peter Kutschera, Stefano Nardone, Maria Polese, Chaim Rafalowski, Minna Raikkonen, Kalev Rannat, Ari Kosonen, Valeria Reva, Luis Mario Ribeiro, Johannes Sautter, Friederike Schneider, Martin Scholl, Stanislav Vasilyev, Domingos Viegas, Manuel Warum, Sven Wirth, Jussi Yliaho

Publisher: CRISMA consortium, 2015
Permalink: **https://crisma-cat.ait.ac.at/content/crisma-catalogue-book**
Printable version: **https://crisma-cat.ait.ac.at/print/book/export/html/109**

Note: The tag cloud above is more than just an illustration. Click on one of the terms to search for relevant pages (can be slow!).

Management Summary

From the technical point of view, the main result of the **CRISMA project (http://www.crismaproject.eu/)** is a software **framework** which faciliates development of the decision support applications. The framework is defined in terms of the **Building Blocks** specifications and the reference implementations thereof (**Components**) have been developed, integrated in Reference Applications and tested in CRISMA Pilots.

CRISMA Catalogue provides the most accurate and up-to-date information on all CRISMA Applications, **Building Block**, Models and Components which have been specified, implemented or inherited from previous projects. Thanks to the way relations between reference applications, BB specifications and Components are presented on the site, **it** is easy to navigate back and forth between various elements and gain in-depth understanding of the way **CRISMA framework** and applications are designed.

However, the site is not well suited for sequential reading and does not provide any background information on the CRISMA project, **application** **business logic** and **architecture**. This is where the "book" comes in.

This online book (**Catalogue of CRISMA applications, framework building block specifications and software implementations)** provides the complete catalogue content in a form which is better suited for sequential reading and **referencing in the publications**. In fact, it is possible to dump a complete CRISMA book content in a single HTML or PDF document. The result is not as well formatted as a real book, but it can be stored locally and printed (e.g. to PDF) and thus allows you to read the book contents offline.

Who should read this book?

The CRISMA Catalogue book should not be mistaken for the replacement for the

, nor understood as an introduction in CRISMA.

- First two sections (Introduction and CRISMA Pilots) can be read by anyone interested in CRISMA applications.
- Third section (Reference applications) *should* be read by potential integrators, before attempting to develop own applications.
- The **rest** of the book is conceived as a reference material for the software developers and the readers are encouraged to use the overview relation tables as a way to discover the interesting elements, rather than reading the "Building Block" and "Software Components" sections sequentially.

Please note that the videos and online demo applications can only be used online. Finally, the relations between reference applications, building blocks, and the components are presented in a form of a summary table, rather than "per **entity**" in the book.

Glossary

The **CRISMA** glossary entries are defined on the CRISMA **catalogue** and imported in the deliverables as needed. The latest version of the table can always be downloaded from **https://crisma-cat.ait.ac.at/glossary/deliverableglossary?field_used_in_tid=971**

Name	Term description
Application	See **CRISMA application**
Application architecture	An **Application Architecture** provides a specification of application-specific **Simulation** Cases in accordance to the Integrated **System** Viewpoint of the **Conceptual Business Logic** of the **CRISMA Framework** Architecture.
Application programming interface	An **application programming interface** (**API**) is a protocol intended to be used as an interface by software components to communicate with each other. See **http**://en.wikipedia.org/wiki/API
Architecture	In computer science and engineering, computer **architecture** is the art that specifies the relations and parts of a computer **system**. See **https://en.wikipedia.org/wiki/Computer_architecture**
Capability	**Capability** can be viewed as the ability to perform actions.
CRISMA federate	Any component that connects to the Middleware **Infrastructure** of the **CRISMA Framework** and is able to exchange Control and Communication Information with the Middleware Infrastructure. More specifically, a **CRISMA Federate** has to be aware of the **API** of the **ICMM**.
CRISMA federation	A number of **CRISMA** Federates that act together as a unit. A **CRISMA Federation** is a subset of a **CRISMA Application**.
CRISMA framework	A **framework** composed of ready-to-use Building Blocks and supporting tools that can be connected together to form a **CRISMA Application**.
CRISMA system	In the perception of the **architecture**, the **CRISMA System** is the overall project results consisting of all CRISMA Applications.

Name	Term description
Damage classification	**Evaluation** and recording of damage to structures, facilities, or objects according to three (or more) categories, such as: 1. "severe damage" which precludes further use of the structure, facility, or **object** for its intended purpose. 2. "moderate damage" or the degree of damage to principal members, which precludes effective use of the structure, facility, or object for its intended purpose, unless major repairs are made short of complete reconstruction. 3. "light damage" such as broken windows, slight damage to roofing and siding, interior partitions blown down, and cracked walls; the damage is not severe enough to preclude use of the installation for the purpose for which was intended. (OFDA) (**Sahana** – Glossary)
Disaster	A serious disruption of the functioning of a community or a society involving widespread human, material, economic or environmental **losses** and impacts, which exceeds the ability of the affected community or society to cope using its own resources. (2009 UN-ISDR Terminology on **Disaster** **Risk** Reduction)
Disaster risk management	The systematic process of using administrative directives, organizations, and operational skills and capacities to implement strategies, policies and improved coping capacities in order to lessen the adverse impacts of hazards and the possibility of **disaster**. **Disaster risk management** aims to avoid, lessen or transfer the adverse effects of hazards through activities and measures for **prevention**, **mitigation** and **preparedness**. (2009 UN-ISDR)
Elements at risk	Population, buildings and engineering works, **infrastructure**, environmental features, cultural values and economic activities in an area potentially affected by an **event** (e.g. landslide).
Emergency	Any **incident**, whether natural, technological, or human-caused, that requires responsive action to protect life or property (FEMA Glossary, 2013)
Evacuation	1- **Emergency** **evacuation**: Removal of persons from a dangerous place due to a **disaster** (e.g. in the context of **pilot** D) 2- Casualty movement: the procedure for moving a casualty from its initial location to an ambulance (e.g. in the context of pilot C) (Wikipedia, 2013)
Evaluation	**Evaluation** is "systematic investigation of the worth or merit of an **object**." (Frechtling 2011). Evaluation is a valuable source of information on how a project is being implemented, specifically, what works and what should be modified.

Name	Term description
Federate	see **CRISMA federate**
Hazard	A “dangerous phenomenon, substance, human activity or condition” (UNISDR, 2009) - characterized by its location, intensity, frequency and probability - that may cause adverse impacts on a social (e.g, loss of life, injury or other health impacts, property damage, social and economic services disruption) or environmental (e.g., ecological damages) **system** (e.g., Pelling et al., 2004; Birkmann et al., 2013; Dewan, 2013)
Hazard model	**Hazard** models are a piece of software and/or related data to simulate hazardous events.
Hypertext Transfer Protocol	**Hypertext Transfer Protocol** (**HTTP**) is **standard** protocol for data transport in the **world** wide web (WWW). See (RFC_1945, 1996) and (RFC_2616, 1999)
Incident	An occurrence, natural or human-caused, that requires a **response** to protect life or property. Incidents can, for example, include major disasters, emergencies, terrorist attacks, terrorist threats, civil unrest, wildland and urban fires, floods, hazardous materials spills, nuclear accidents, aircraft accidents, earthquakes, hurricanes, tornadoes, tropical storms, tsunamis, war-related disasters, public health and medical emergencies, and other occurrences requiring an **emergency** response (FEMA Glossary, 2013).
Integrated component	A component that takes part in an interaction of **CRISMA** Federates but is not itself a **CRISMA Federate**. **It** is not CRISMA-aware and thus does not interact with the CRISMA Middleware **Infrastructure**. An **Integrated Component** may be a member of a **CRISMA Application** but not of a **CRISMA Federation**.
Integrated Crisis Management Middleware	The **ICMM** (**Integrated Crisis Management Middleware**) is a central **Building Block** in every **CRISMA Application**. **It** connects Crisis Management Simulations with the Analyis and Decisison Support functionality of CRISMA by providing a central repository for harmonized **world state** and **indicator** information. The ICMM is fed by simulations providing the basic information to be used for world state analysis and decision support Building Blocks.

Name	Term description
Interface	In the context of **IT**, a named set of operations that characterise the behaviour of an **entity**. The aggregation of operations in an **interface**, and the definition of the interface, shall be for the purpose of software re-usability. The specification of an interface shall include a static portion that includes definition of the operations. The specification of an interface shall include a dynamic portion that includes any restrictions on the order of invoking the operations. (ISO_19119, 2003)
Mashup platform	A component that allows to combine smaller components (widgets) providing specific functionality into a complete **graphical user interface** (**GUI**) to provide all the functionality needed
Mock-up	Painted (by pencil or by computer drawing tool) representation of a human-computer-interaction dialogue. A special variant is a paper-**mock-up** (done by pencil and paper).
Model	A **model** is a hypothetical simplified description of a complex **entity** or process (Sterling & Taveter, 2009). A model can be considered as "an abstract representation of a **system** or process" (Carson, 2005). A model is a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process that has been designed for a specific purpose (NATO, 2010). Stachowiak (1973) describes a model using three features: the mapping **feature** (reproduction of the original), the reduction feature (abstraction of the original) and the pragmatic feature (addressing a purpose for its user).
Open Geospatial Consortium	The **Open Geospatial Consortium** (**OGC**) is an international industry consortium of 480 companies, government agencies and universities participating in a consensus process to develop publicly available **interface** standards. OGC® Standards support interoperable solutions that "geo-enable" the Web, wireless and location-based services and mainstream **IT**. The standards empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications. (OGC, 2013)
OpenId	A **standard** for user identification. (see **http**://**openid**.net/)
Pilot demo phase	The actual testing phase of a **CRISMA application** building, modelling and **Simulation** by a the **pilot**.

Name	Term description
Point of interest	A specific point location that someone may find useful or interesting (see **http**://en.wikipedia.org/wiki/Point_of_interest). In **CRISMA** context, the **Point of Interest** (POI) is a map **object**, usually visualized with a marker/icon on the map. POI is an object with properties, like geographic location, and/or some properties characterizing this object. The POI can represent both passive (like buildings, bridges, etc.) and active objects (e.g. agents as representatives of some physical or virtual entities) on the map.
Preparedness	(1) The knowledge and capacities developed by governments, professional **response** and **recovery** organizations, communities and individuals to effectively anticipate, respond to, and recover from, the impacts of likely, imminent or current **hazard** events or conditions. (2) One of the phases of the **Crisis Management Cycle**
Reference Application	A **Reference Application** is a completely integrated and functional **demonstration** of how **CRISMA** goals may be attained, in one or more particular domain(s). Hence, a reference application is the transferable “wiring together” of building blocks without any **pilot**-specific logic. Reference Applications are not necessarily industry-grade, production-mature software suites; rather, their purpose is to demonstrate how CRISMA concepts can be (not necessarily should be) implemented in their particular domain. So, reference applications demonstrate how the CRISMA models and building blocks can be used to build various applications pertinent to **crisis management**. The presentation targets developers and integrators rather than end users.
Reference architecture	A **reference architecture** "is an architectural design pattern that indicates how an abstract set of mechanisms and relationships realizes a predetermined set of requirements. One or more reference architectures may be derived from a common **reference model**, to address different purposes/usages to which the Reference Model may be targeted." (**SOA-RM**, 2006)
Reference model	An abstract **framework** for understanding significant relationships among the entities of some environment. **It** enables the development of specific reference or concrete architectures using consistent standards or specifications supporting that environment. A **reference model** consists of a minimal set of unifying concepts, axioms and relationships within a particular problem domain, and is independent of specific standards, technologies, implementations, or other concrete details. (**SOA-RM**, 2006)
Resource	**Crisis management** context: Resources are deployed in the scope of crisis management activities. Resources may include material (e.g. sandbags, medical products, oxygen tank), personnel (e.g. medical officer, ambulance **driver**, crisis manager), vehicles (e.g. fire trucks), protection **infrastructure** and facilities (e.g. hospital, shelters), installations (e.g. weirs). **IT**-context: **Resource** is every possible data **object** as part of the common **CRISMA** meta **information model**.

Note: the glossary of terms relevant for this deliverable should appear in a table above automatically. Currently set-up to show the D35.1 glossary.

1. Introduction

This online book provides an alternative view to the contents of the **CRISMA** **catalogue**. While the catalogue is meant to be a highly interactive site with organically interconnected contents, the "book" is organised in a linear way.

The main purpose of the "book" presentation is to be downloaded and read off-line or incorporated in CRISMA deliverables and other documents. Unfortunately, the formatting of the resulting material is far below the **standard** of the other CRISMA documents. However, the experience with development of various CRISMA deliverables deliverables shows that the effort required for re-formatting the "book" and incorporating the information in nicely formatted documents is relatively modest. In any case, the book is provided **free of charge**, "as is", and in the hope that **it** might be useful, but with absolutely no warranty.

CRISMA project and its applictions

The **CRISMA** project is a European Union funded project focusing on **simulation** of multi-sectorial large scale **crisis** scenarios with multi-dimensional effects on the society. According to **CRISMA framework architecture, (http://www.crismaproject.eu/deliverables.htm)** the aspects of the **World** we are interested in must be modelled as a set of representative **World State** data (WSs) which develops in discrete steps (World Transitions), depending on the inherent characteristics of the problem at hand as well as on the users' decisions (Decision Chain).

Depending on the selected **use case**, new WSs may be generated at regular time intervals, following the users decisions (e.g. "evacuate"), or to capture the results of a specific **model** execution (e.g. "ambulance has arrived to the scene"). In most CRISMA applications, the user can **test** the effects of alternative decisions. This results in distinct Scenarios, i.e. sets of WSs corresponding to unique Decision Chains. World States and the whole Scenarios can be compared and assessed using a set of Indicators relevant to a problem at hand. In addition to indicators, CRISMA also supports **Criteria** – a normalized form of the Indicators which take into account the desired values of the Indicators (e.g. "evacuating the critical patients in less than 30 minutes is good"). This simplifies the **task** of analysing the results and ranking the different Scenarios. However, the Criteria are both subjective and extremely sensitive to small changes in the problem definition. Consequently, the criteria must be defined *by end users* separately for each training or planning session, e.g. as a part of the training setup.

CRISMA decision support **concept** can thus be summarized as: help the users to assess the results of their decisions, but do not suggest a course of action nor impose any particular solutions.

The software and methodology developed in the CRISMA project thus simplifies the task of building decision support applications for the **crisis management**. More precisely, **it** supports the task of building training and planning applications which:

1. Combine the model calculations with the users' decisions to simulate the development of the "World" in a crisis **situation**.
2. Use indicators to depict the state of the World during and at the end of the training or planning **exercise**.
3. Allow users to rank different exercise outcomes

How to cite this book?

Licensing

Acknowledgements / Citations

The number of CRISMA team members who contributed materials to CRISMA catalogue is huge, so please use the editors names when citing the whole book. Depending on the citation style, the result might look like this:

- Havlik, D., Dihé, P., Frings, S., Steinnocher, K., Aubrecht, C. (eds.). *Catalogue of CRISMA decision support applications, **framework** **building block** specifications and software implementations*. CRISMA consortium, **https://crisma-cat**.**ait**.ac.at/content/crisma-catalogue-book (2015)

You can add the "Retrieved at" date if appropriate, but the book content will not significantly change after the end of 2014.

Individual articles (should) provide Author names. They also provide the explicit "last changed" date. This means that they can be cited as:

- <AUTHORS>: <ARTICLE NAME>. in Havlik, D., Dihé, P., Frings, S., Steinnocher, K., Aubrecht, C. (eds.): *Catalogue of CRISMA decision support applications, framework building block specifications and software implementations*. CRISMA consortium, Retrieved from <URL> (2015)

Depending on the citation format, you may need to use "Retrieved on <DATE> from <URL>" form. However, please note that each article states the date of last edit and that no major changes to articles are expected in 2015.

For example, **this article** should be cited as:

- Oren Deri, Denis Havlik, Chaim Rafalowski: Accidental spillage from a container at large city port (Israel). In Havlik, D., Dihé, P., Frings, S., Steinnocher, K., Aubrecht, C. (eds.): *Catalogue of CRISMA decision support applications, framework building block specifications and software implementations*. CRISMA consortium, Retrieved from **https://crisma-cat**.**ait**.ac.at/print/188 (2015)

Notice that we are using the print link. You can also use the link to the online article (**https://crisma-cat.ait.ac.at/ref-app/Accidental spillage from a container at large city port (Israel)**), but that link might break if someone decides to slightly change the title. /node/188 is the unique key of this article in our DB and therefore linking to it is more robust.

Disclaimer

Mapping of building blocks to applications

CRISMA applications are modular. This table shows the relations between reference applications (section 3) and the building blocks (section 4) which were used to build these reference applicaitons.
Please note that the relations are at the level of specifications, not at the level of software. There can be more than one implementation for each of the building blocks. Likewise, some of the building blocks were realized by combining two or more existing software tools.

Relation between applications and BBS

Applications	Constituents

Applications	Constituents
Reference Application for Exercise Support This is the **Reference Application** for **Exercise** Support. **It** represents a tool capable of supporting relief organisations in the execution and analysis (esp. debriefing) of field exercises. An IT-based gathering and exploitation of relevant data from the exercise and data visualisations for the exercise debriefing are provided to users of the application. A reasonable presentation of the exercise results allows to identify deficiencies and thus to identify measures for improvement. After the tool was used for gathering relevant data before and during the exercise, it provides essential insights in form of a debriefing view, which comprises not only a general overview on the exercise results, but also more detailed information on essential aspects related to the medical care of patients and the organisation of the operation (Bracker et al., 2014). The usability of the tool was evaluated within two practical exercises organized by the German and Bavarian Red Cross.	**Integrated Crisis Management Middleware BB** **Data Integration Publish Subscribe Context Broker BB** **Indicator Building Block** **UI Mashup Platform** **Integrated Planning View** **UI Integration Platform BB**

Applications	Constituents
Reference Application for Resource Planning This is the **Reference Application** for **Resource** Planning. The main objective of this Reference Application is to provide what and how much resources are needed to provide the best **response** in case of a mass casualty **incident**. **It** represents local-specific constraints that exist, such as the density of hospitals nearby, what their surgery capacities are like, the distance between the local rescue bases and the incident scene as well as the level of **preparedness** of first responder.	**Indicator Building Block** **Publish Subscribe Context Broker BB** **UI Integration Platform BB** **Integrated Crisis Management Middleware BB** **Scenario Analysis and Comparison View** **Multi Criteria Analysis and Decision Support View** **Agent Oriented Simulation Models** **OOI World State Repository BB** **OOI Management View** **Resource Allocation Tactic Model**

Applications	Constituents
Reference Application for the Coastal Submersion Domain This **Reference Application** is based on the requirements of a storm surge induced Coastal Submersion **use case**. The main field of application is **crisis** **preparedness** planning and **evaluation** of **mitigation** solutions. Thereby, different storm surge events and their impacts are analysed under different circumstances (e.g. summer or winter, day or night) and under different intensities of events (e.g. stronger winds, concomitance with high tidal level). Then, the effects of different mitigation solutions are evaluated, and, eventually, also several decisions at different times.	**Worldstate View** **Integrated Crisis Management Middleware BB** **GIS View Building Block** **Coastal Submersion Model** **Population exposure model** **Dikes vulnerability model** **Simulation Model Integration BB** **Multi Criteria Analysis and Decision Support View** **Data Integration** **Simulation Model Interaction View** **Publish Subscribe Context Broker BB** **Evacuation model for coastal submersion**

Applications	Constituents
Reference Application for the Earthquake and Forest Fire Domains	**Indicator Building Block** **Time Dependent Vulnerability model (TDV)** **GIS View Building Block** **Cascade Events Configuration and Interaction View** **Simulation Model Integration BB** **Road network vulnerability model (RNV)** **Publish Subscribe Context Broker BB** **Cascading Effects Model** **Worldstate View** **Forest fire behaviour model** **Simulation Model Interaction View** **Scenario Analysis and Comparison View** **UI Integration Platform BB** **Multi Criteria Analysis and Decision Support**

General Description

This is the **Reference Application** for the Earthquake and Forest Fire Domains. **It** represents the transferable wiring together of **CRISMA Framework** Building Blocks and **Simulation** Models (Software Components, respectively) without any **pilot**-specific logic. The main objective of this Reference Application is to provide a customizable, extensible and transferable example of a **CRISMA Application** for the simulation of natural **disaster** (Earthquake and Forest Fire) with irreversible damages that enables comparing scenarios in a multi-**risk** framework including cascading events simulation and the assessment of **decision making** choices and possible consequences in each foreseen evolving **scenario**.

Applications	Constituents
	View **Building impact model** **Integrated Crisis Management Middleware BB** **Earthquake casualty model** **Data Integration** **Population exposure model**

Applications	Constituents
Reference Application for the Nordic Winter Storm Domain The **Reference Application** for the Nordic Winter Storm Domain serves as a technological example for one class of the applications which can be implemented by wiring the **CRISMA** Components (reference implementations of the **CRISMA Framework** Building Blocks and Models) together. **It** provides the software functionality which is required in the CRISMA **Pilot** A "Electricity outage in the far North" and similar scenarios. The reference application demonstrates the usage of **agent**-based simulations with appropriate models describing cooling of buildings and living conditions (life quality) in selected geo-cells within the **crisis** area, during electric power limitations and extreme low temperatures. The application has facilities for driving the simulations, demonstrating planning of **evacuation** and **evaluation** of economic impacts. Choosing different setups and **simulation** scenarios, the operator can find an optimal solution for both planning and **response**.	**VTT House model** **Economic impacts model** **Economic impacts analysis view** **OOI Management View** **OOI World State Repository BB** **Agent Oriented Simulation Models** **Preparedness Plan BB** **Worldstate View** **Scenario Analysis and Comparison View** **Evacuation Resources Simulation Model** **Integrated Crisis Management Middleware BB**

Applications	Constituents
Reference Application for the Resource Management Training Support This **reference application** showcases the use of **Mashup platform** + widgets, Indicators and analysis services, Agents platform and **World State** repository in teaching of the **crisis** managers and assessing their knowledge. **It** demonstrates how interaction of users with a software **system** where various resources and population are represented by **agent** models can be used to assess and improve the reasoning of the decision makers in simulated (desktop training) **emergency** situations. The "educational" use of the application relies on the principle of trying out different strategies for resolving of a well-defined crisis and comparing the outcomes. in **Crisis Management** (deskop) Training Scenarios and provides the technological platform for the Israeli **CRISMA** **validation** application (**Pilot** C).	**UI Integration Platform BB** **Indicator Building Block** **Agent Oriented Simulation Models** **Resource Management Training Simulation Scenario Setup View** **Resource Management Training Dispatch and Monitor View** **Resource Management Training Indicators and Statistics View** **OOI World State Repository BB** **OOI Management View** **Integrated Crisis Management Middleware BB** **Publish Subscribe Context Broker BB** **Resource Management Tactical Training BB**

Applications	Constituents
	Patients model **Integrated Planning View**

Reference implementations of the Building Blocks

The following table shows which software was used as reference implementation of the various building blocks. The table also lists the names of reference applications which use the corresponding building blocks. For short description of the reference applications, please see previous section.

Software used in CRISMA is very heterogeneous mixture: open source and proprietary; developed within project and inherited; fully supported and provided as is; owned by one of the project partners and third party..

The table therefore provides this type of information for all software components which were used of developed in CRISMA, as well as their relation to building block specifications and to reference applications.

Building Block	Use in applications	Components

Building Block	**Use in applications**	**Components**
	Reference Application for the Nordic Winter Storm Domain	
Agent Oriented Simulation Models The **Agent**-Oriented **Simulation** Models **Building Block** serves for the development of dynamic maps – specific (individual-based) simulation models composed of interacting software agents situated in some environment. This Building Block comprises a collection of generic agents and interaction templates for dynamic map construction, for describing, defining and specifying points, areas and layers of interest. **It** provides furthermore a dynamic-map-based user **interface** for interaction and visualization. Thus, it can be considered both an Integration and User Interaction Building Block.	**Cross Border Emergency Crisis (Finland)** **Reference Application for Resource Planning** **Reference Application for the Resource Management Training Support**	*Dynamic Map Agents* Owner: **TTU** License: **Proprietary**, **using Google Maps API for non-commercial purposes** Development Context: **Extended**
BB to be defined This is just a place-holder to be used in **application** descriptions. The idea is to refer to this **BB** and explain what is missing as a TODO reminder. We should occasionally check for applications still declaring to use this BB as a part of QA.		*Component to be defined* Owner: **AIT** License: **To be defined** Development Context: **New**

Building Block	**Use in applications**	**Components**
Building impact model **Model** for the assessment of expected damage on building classes due to earthquakes	**CRISMA Technology Demonstrator** **Reference Application for the Earthquake and Forest Fire Domains**	*PostgreSQL stored procedure* Owner: **PLINIVS** License: **Proprietary** Development Context: **Used**
Cascade Events Configuration and Interaction View The **Cascade Effects** View is a User interaction Building that allows a user to configure and run a Cascade Effects **Scenario**. The user can select a triggering **event** (for example, an earthquake) and provide may either specify the characterisation of the event (**Simulation Control Parameter**) and thus initiate a new **Simulation Model** Run for this particular event, or select (if available) the output of a past event or an event already simulated.	**Reference Application for the Earthquake and Forest Fire Domains** **CRISMA Technology Demonstrator**	*Cascade Events Configuration Widget* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **New**
Cascading Effects Model The cascading effect **model** for dynamic **scenario** assessment calculates the probability of attainment of cascading events scenarios, given an initial triggering **event**, or estimate consequence paths given the occurrence of selected scenarios.	**Reference Application for the Earthquake and Forest Fire Domains**	*TDV Python package* Owner: **AMRA** License: **To be defined** Development Context: **New**
Coastal Submersion Model The Costal Submersion **Model** is a 2D-hydrodynamic model based on the **open source** TELEMAC-MASCARET **system**. In TELEMAC-MASCARET system, we use mainly TELEMAC2D to calculate the time and space dependent hydrodynamic characteristics such as water levels, velocities, discharges.	**Reference Application for the Coastal Submersion Domain**	*TELEMAC MASCARET System* Owner: **Eléctricité De France** License: **GPL (controlled OS)**, **LGPL (permissive OS)** Development Context: **Used**

Building Block	**Use in applications**	**Components**
Data Integration The Data Integration **Building Block** provides components that can be used to easily serve data in a **CRISMA**-compliant (**OGC** open **standard** compatible) way so that other Building Blocks may use them for further processing like viewing or editing. That way data can be made accessible for CRISMA components.	**Reference Application for Exercise Support** **Reference Application for the Coastal Submersion Domain** **CRISMA Technology Demonstrator** **Reference Application for the Earthquake and Forest Fire Domains**	*MapServer* Owner: **Open Source Geospatial Foundation** License: **MIT (permissive OS)** Development Context: **Used** *GeoServer* Owner: **community-driven project** License: **GPL v2 (controlled OS)** Development Context: **Used**
Dikes vulnerability model The dikes **vulnerability model** is a model programmed in python which allow to calculate the potential statistical **impact** on dikes depending on their status.The model is based on the damage probability matrix. In view of results of dike vulnerability model, the user could make an informed choice on break or failure dikes for local **simulation**.	**Reference Application for the Coastal Submersion Domain**	*PyShp* Owner: **Geospatialpython** License: **MIT License** Development Context: **Used**

Building Block	**Use in applications**	**Components**
Earthquake casualty model **Model** for the assessment of expected number of injured and deaths due to an earthquake	**Reference Application for the Earthquake and Forest Fire Domains** **CRISMA Technology Demonstrator**	*PostgreSQL stored procedure* Owner: **PLINIVS** License: **Proprietary** Development Context: **Used**
Economic impacts analysis view This **Building Block** is an **economic evaluation** tool to support **crisis management** and to be used in the **preparedness** phase for planning and training purposes. The main objective of an economic evaluation in **CRISMA** is: - to present the economic impacts arising from crises (ex post performance) and - to assess different **mitigation** proposals and their costs/benefits (ex ante planning).	**Reference Application for the Nordic Winter Storm Domain** **Cross Border Emergency Crisis (Finland)**	*CRISECON Service* Owner: **VTT** License: **Proprietary** Development Context: **New** *CRISECON GUI* Owner: **VTT** License: **Proprietary** Development Context: **New**
Economic impacts model **Model** for: • presenting economic impacts arising from crises (ex post performance) and • assessing different **mitigation** proposals and their costs/benefits (ex ante planning).	**Reference Application for the Nordic Winter Storm Domain** **Cross Border Emergency Crisis (Finland)**	*CRISECON Service* Owner: **VTT** License: **Proprietary** Development Context: **New**

Building Block	**Use in applications**	**Components**
Evacuation Model The **Evacuation** **Model** is a prototypical model that represents how the population can be evacuated from the **hazard** area(s) to the safety zone(s). The purpose of the evacuation model is to determine how fast and in which health condition(s) the population is able to leave the hazard area(s).		*Evacuation model (NetLogo)* Owner: **TTU** License: **GNU** Development Context: **New**
Evacuation model for coastal submersion The **evacuation** **model** developed for coastal submersion in Charente-Maritime (France) used the software LSM2D. This model cover the Rivedoux-Plage area in Ré Island as well as the area on the coast from Yves to Chatellaion. Different scenarios of evacuation could be simulated. The main results of this model are: • the estimated time to evacuate the population • the estimated closed roads • the estimated casualties on population • the estimated impacts on buildings due to the flood	**Reference Application for the Coastal Submersion Domain**	*Life Safety Model 2D* Owner: **HR Wallingford and BC Hydro** License: **Proprietary** Development Context: **Used**
Evacuation Resources Simulation Model **Evacuation** Resources **Simulation Model** calcuates the **impact** of **resource** allocations chosen by the user (from the proposals based on the **preparedness** plan) to mitigate the **situation**.	**Reference Application for the Nordic Winter Storm Domain**	*Insta EvacSim* Owner: **Insta** License: **Proprietary** Development Context: **New**
Forest fire behaviour model The Forest Fire Behaviour **model** is a deterministic integrated **system**, based on the Model of Rothermel (1972), for the spatial **simulation** of forest fire behaviour over complex topography and wind flows in areas with heterogeneous vegetation cover. Its main components are the fire behaviour predictions at local scale and wind field prediction at local and large scale taking into account different thermal and recirculation effects. Firestation also has the **capability** to simulate smoke dispersion and particles concentration over the area afected by the simualted fire.	**Reference Application for the Earthquake and Forest Fire Domains**	*Component to be defined* Owner: **AIT** License: **To be defined** Development Context: **New**

Building Block	**Use in applications**	**Components**
GIS View Building Block The **GIS** View is a User Interaction **Building Block** that enables the visualisation and manipulation of geospatial data. Geospatial data plays a predominant **role** in all **crisis management** related applications, because most if not all information playing a role in crisis management has a geospatial component.	**Reference Application for the Coastal Submersion Domain** **Reference Application for the Earthquake and Forest Fire Domains** **CRISMA Technology Demonstrator**	*cismap* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **Extended** *Leaflet.js* Owner: **Vladimir Agafonkin** License: **BSD (permissive OS)** Development Context: **Used** *OpenLayers* Owner: **OpenLayers Community** License: **BSD (permissive OS)** Development Context: **Used** *Wirecloud OOI GIS Map* Owner: **AIT** License: **AGPL v3 (controlled OS)**, **Negociable** Development Context: **New**

Building Block	**Use in applications**	**Components**
Indicator Building Block The Indicators **Building Block** (formerly Algebraic **Evaluation**) is a component that allows definition, storage and evaluation of "simple" algebraic models in order to evaluate consequences of decisions made by **CRISMA** users during a training session. In this context "simple" means that there is no large number crunching needed. Usage is for calculation of key performance indicators, heuristic models and to some extend the implementation of models actually running in interactive **GIS** environments in order to make them usable as services in the **CRISMA framework**.	**Reference Application for Exercise Support** **Reference Application for Resource Planning** **Reference Application for the Earthquake and Forest Fire Domains** **CRISMA Technology Demonstrator** **Reference Application for the Resource Management Training Support**	*BasicIndicators* Owner: **AIT** License: **GPL v2 (controlled OS)** Development Context: **New** *Emikat* Owner: **AIT** License: **Proprietary** Development Context: **Extended**

Building Block	**Use in applications**	**Components**
Integrated Crisis Management Middleware BB The **ICMM** is a central **Building Block** in every **CRISMA Application**. **It** connects **Crisis Management** Simulations with the Analysis and Decision Support functionality of CRISMA by providing a central repository for harmonized **world state** and **indicator** information. The ICMM is fed by simulations providing the basic information to be used for world state analysis and decision support Building Blocks. On an ICT conceptual level the ICMM is a generic distributed **resource**-oriented Control and Communication Information Management **System**. Thereby it is important to note, that the term 'resource-oriented' in the ICMM refers to the **concept** of generic resources as used in the context of the Resource Oriented **Architecture** (ROA).	**Reference Application for Exercise Support** **Reference Application for the Coastal Submersion Domain** **Reference Application for the Earthquake and Forest Fire Domains** **CRISMA Technology Demonstrator** **Reference Application for the Resource Management Training Support** **Reference Application for the Nordic Winter Storm Domain** **Reference Application for**	*cids System* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **Extended**

Building Block	Use in applications	Components
	Resource Planning Cross Border Emergency Crisis (Finland)	

Building Block	**Use in applications**	**Components**
Integrated Planning View The Integrated Planning View **BB** is a generic integrated view for the configuration and inspection of arbitrary **crisis management** scenarios in planning situations. The **CRISMA** planning use cases (most use cases of CRISMA) intend to realize functionalities that are related to opening **world** states, changing them and storing them. Further running simulations, inspecting them and comparing results is addressed in all of them and the user work for planning purposes in a calm office environment.	**Reference Application for Exercise Support** **Reference Application for the Resource Management Training Support**	*Wirecloud OOI GIS Map* Owner: **AIT** License: **AGPL v3 (controlled OS)**, **Negociable** Development Context: **New** *Exercise Worldstate Data Chart Widgets (Wirecloud)* Owner: **Cassidian** License: **To be defined** Development Context: **New** *Configuration Component* Owner: **Fraunhofer IAO** License: **Open source** Development Context: **New** *Wirecloud Application Mashup Platform* Owner: **UPM** License: **AGPL v3 (controlled OS)** Development Context: **Used** *Wirecloud Charts* Owner: **AIT** License: **AGPL v3 (controlled OS)**, **Negociable** Development Context: **New**

Building Block	**Use in applications**	**Components**
Multi Criteria Analysis and Decision Support View The Multi **Criteria** Analysis View and Decision Support View is a User Interaction **Building Block** that allows performing a ranking of different **Crisis Management** Scenarios with respect to specific Criteria.	**Reference Application for the Earthquake and Forest Fire Domains** **CRISMA Technology Demonstrator** **Reference Application for Resource Planning** **Reference Application for the Coastal Submersion Domain**	*Multi Criteria Analysis and Decision Support Widget (Java)* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **New** *Multi Criteria Analysis and Decision Support Widgets (JavaScript)* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **New**
OOI Management View This **Building Block** enables to view and edit the actual data available for specific **scenario** or **simulation**. **It** enables the **system** administrator to detect the **OOI** data type's properties and based on the OOI Information Models definitions. It also enables to monitor and edit a specific snapshot context (time, user, and workflow).	**Reference Application for the Resource Management Training Support** **Reference Application for the Nordic Winter Storm Domain** **Reference Application for Resource Planning**	*OOI Management UI Component* Owner: **NICE** License: **LGPL v3 (permissive OS)** Development Context: **New**

Building Block	**Use in applications**	**Components**
OOI World State Repository BB **OOI**-WSR is a **Resource Management** related **Building Block** that enables archiving, querying and manipulation of Objects of Interest (OOI) **world state** data. This **BB** serves as a Repository **service** for OOI data that can be consumed or manipulated by other interaction or functional building blocks and resource management models.	**Reference Application for the Resource Management Training Support** **Reference Application for the Nordic Winter Storm Domain** **Cross Border Emergency Crisis (Finland)** **Reference Application for Resource Planning**	*OOI World State Repository* Owner: **NICE** License: **LGPL v3 (permissive OS)** Development Context: **New**
Patients model This **model** governs the behaviour of the patients in Israeli and German pilots. Full description shall be provided by **TTU**.	**Reference Application for the Resource Management Training Support**	*Dynamic Map Agents* Owner: **TTU** License: **Proprietary**, **using Google Maps API for non-commercial purposes** Development Context: **Extended**

Building Block	Use in applications	Components
Population exposure model **Model** for distributing population in spatial and temporal dimensions. This model uses temporal and spatial proxies in order to disaggregate the population from administrative units to spatio-temporal grids. The outcome is used in **CRISMA** as basis for time-dependent **exposure** assessment and in further steps as a base data for **evacuation** and casualty modeling (Aubrecht et al., 2014ab; Steinnocher et al., 2014).	**Reference Application for the Earthquake and Forest Fire Domains** **Reference Application for the Coastal Submersion Domain**	*Emikat* Owner: **AIT** License: **Proprietary** Development Context: **Extended**
Preparedness Plan BB The **Preparedness** Plan **Building Block** is a decision support mechanism, which helps the decision maker to take the needed actions in case of **emergency** according to plans based on analysis of threats, vulnerabilities and possible emergency scenarios.	**Reference Application for the Nordic Winter Storm Domain** **Cross Border Emergency Crisis (Finland)**	*Insta Response Preparedness Planner* Owner: **Insta** License: **Proprietary** Development Context: **Extended**

Building Block	Use in applications	Components
Publish Subscribe Context Broker BB The Publish Subscribe Context Broker **Building Block** is a cross-over between an **event** broker which accepts events and dispatches them to subscribers and an access **service** providing the information on current state of the “world”. Thereby, the Publish Subscribe Context Broker Building Block can be used to realize the event subscription and event delegation functionality of the **ICMM**. Thus, the Building Block is both suitable for dispatching events related to Control and Communication Information (**CCI**) managed by the ICMM and events related to (Worldstate) Data managed by arbitrary data access services, e.g. the **OOI** **World State** Repository **BB** or implementations of the Data Integration BB (**WMS**, **WFS**, …).	**Reference Application for the Coastal Submersion Domain** **Reference Application for the Earthquake and Forest Fire Domains** **Reference Application for the Resource Management Training Support** **Reference Application for Exercise Support** **Cross Border Emergency Crisis (Finland)** **Reference Application for Resource Planning**	*Orion Context Broker* Owner: **Telefonica** License: **AGPL v3 (controlled OS)** Development Context: **Used**

Building Block	**Use in applications**	**Components**
Resource Allocation Tactic Model This **model** replicates the decision making process of **crisis** managers in **resource management** planning applications. **It** allows us to **test** the planning scenarios with different management strategies, e.g. "prioritize transport to hospital" or "prioritize treatment on the field". It realizes a second layer above the Ambulance Model in order to being capable of running simulations with predefined strategy settings. These different strategy settings for resource planning allow users determining the best strategy in a specific **scenario**. Depending on the strategy ambulances in an idle state are assigned to certain commands before rerunning the ambulance model **simulation**.	**Reference Application for Resource Planning**	*UNDEFINED* Owner: License: Development Context:
Resource Management Model The **Resource Management** Models developed in the **CRISMA** project are built upon the **OOI** **concept** with different context dependent behavioral patterns for different **crisis** domains. Thus, there is no overall generic and all-purpose Resource (OOI) Management **Model**, but a set distinct models for different types of resources (e.g. ambulances, patients) and different situations. However, such domain and crisis specific Resource Management Models can be implemented on basis of the general **Agent**-Oriented **Simulation** Models **Building Block**, with its functionalities described in D312 .	**Cross Border Emergency Crisis (Finland)**	*Dynamic Map Agents* Owner: **TTU** License: **Proprietary**, **using Google Maps API for non-commercial purposes** Development Context: **Extended**
Resource Management Tactical Training BB **Resource Management** Tactical Training **Building Block** (RMTT **BB**) simplifies the **task** of designing the Tactical Training applications for control room operator and on-scene commanders. RMTT enables a Trainee to learn **emergency management** by assigning tasks to various resources and analysing the results in a virtual environment.	**Reference Application for the Resource Management Training Support**	*Situator Training System* Owner: **NICE** License: **Proprietary** Development Context: **Extended**

Building Block	Use in applications	Components
Resource Management Training Dispatch and Monitor View The Dispatch and Monitor view provides a high-level overview over the **resource management** **simulation**'s **world state**. Its purpose is to display one world state at a time and allow the user to distribute resources (ambulances, etc.) among different areas where the **crisis** plays out.	**Reference Application for the Resource Management Training Support**	*Wirecloud WorldState Picker* Owner: **AIT** License: **AGPL v3 (controlled OS)**, **Negociable** Development Context: **New** *Wirecloud OOI Table* Owner: **AIT** License: **AGPL v3 (controlled OS)**, **Negociable** Development Context: **New** *Wirecloud Simulation Picker* Owner: **AIT** License: **AGPL v3 (controlled OS)**, **Negociable** Development Context: **New** *Wirecloud OOI Commands* Owner: **AIT** License: **AGPL v3 (controlled OS)**, **Negociable** Development Context: **New** *Wirecloud OOI GIS Map* Owner: **AIT** License: **AGPL v3**

Building Block	**Use in applications**	**Components**
		(controlled OS), **Negociable** Development Context: **New**
Resource Management Training Indicators and Statistics View The **Resource Management** Training Indicators and Statistics View **Building Block** is a User Interaction Building Block that will be realised as Mashable Composite UI Module. **It** focuses on the visual presentation of statistics and key indicators of a given Worldstate in order to provide a quick overview of the **situation** and to allow for comparison between any two given Worldstates.	**Reference Application for the Resource Management Training Support**	*Wirecloud Charts* Owner: **AIT** License: **AGPL v3 (controlled OS)**, **Negociable** Development Context: **New** *Wirecloud WorldState Picker* Owner: **AIT** License: **AGPL v3 (controlled OS)**, **Negociable** Development Context: **New** *Wirecloud Simulation Picker* Owner: **AIT** License: **AGPL v3 (controlled OS)**, **Negociable** Development Context: **New**

Building Block	**Use in applications**	**Components**
Resource Management Training Simulation Scenario Setup View The **Resource Management** Training **Simulation** **Scenario** Setup View is a User Interaction **Building Block** that will be realised as Mashable Composite UI Module. **It** allows the creation of new resource management simulations or modification of existing ones. It allows the user - most likely a trainer - to create incidents and scenes as well as the creation and management of objects of interest (**OOI**) instances as shown in the figure below.	**Reference Application for the Resource Management Training Support**	*Wirecloud OOI Table* Owner: **AIT** License: **AGPL v3 (controlled OS)**, **Negociable** Development Context: **New** *Wirecloud Simulation Picker* Owner: **AIT** License: **AGPL v3 (controlled OS)**, **Negociable** Development Context: **New**
Road network vulnerability model (RNV) **Model** for the assessment of probability of road link interruption due to earthquakes.	**Reference Application for the Earthquake and Forest Fire Domains**	*PostgreSQL stored procedure* Owner: **PLINIVS** License: **Proprietary** Development Context: **Used**

Building Block	Use in applications	Components
Scenario Analysis and Comparison View The **Scenario** Analysis and Comparison View is able to visualise **Indicator** and **Criteria** data in a way that users are able to analyse and compare different Simulated **Crisis Management** Scenarios and ultimately come to a decision which fits the **simulation objective** best for a specific **Simulation Case**.	**Reference Application for Resource Planning** **Reference Application for the Nordic Winter Storm Domain** **Reference Application for the Earthquake and Forest Fire Domains**	*Scenario Analysis and Comparison Widgets (JavaScript)* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **New** *Scenario Analysis and Comparison Widgets (Java)* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **New**
Simulation Model Integration BB The **Simulation Model** Integration **Building Block** provides components that can be used to easily enable simulation models to participate in a **CRISMA Application**. Due to the heterogeneity of existing simulation models CRISMA has to ensure that they can be integrated in a standardized way and ultimately become a **CRISMA Federate**. That way they are made CRISMA-aware. The envisioned technique to be used is that of so called wrapping so that this Building Block provides a piece of software that can be used to make (existing) simulation models CRISMA-aware with as little effort as possible with respect to the individualities of the different simulation models.	**Reference Application for the Coastal Submersion Domain** **Reference Application for the Earthquake and Forest Fire Domains**	*PyWPS* Owner: **Intevation** License: **GPL v2 (controlled OS)** Development Context: **Used**

Building Block	**Use in applications**	**Components**
Simulation Model Interaction View The **Simulation Model** Interaction **Widget** **Building Block** is a Composite UI Module that lets end users interact with the various simulation models exposed by a Simulation Model Integration Building Block. **It** is responsible for collecting simulation model parameters, launching the respective simulation model, showing status information and retrieving simulation model results. It also constitutes the **GUI** to the Simulation Model Integration Building Block.	**Reference Application for the Coastal Submersion Domain** **Reference Application for the Earthquake and Forest Fire Domains** **CRISMA Technology Demonstrator**	*Simulation Model Interaction Widget* Owner: **Spacebel** License: **LGPL V3** Development Context: **New**
Time Dependent Vulnerability model (TDV) **Model** for the assessment of time-dependent damage on **elements at risk**.	**Reference Application for the Earthquake and Forest Fire Domains**	*TDV Python package* Owner: **AMRA** License: **To be defined** Development Context: **New**

Building Block	**Use in applications**	**Components**
	Reference Application for the Resource Management Training Support	
UI Integration Platform BB The UI Integration Platform **Building Block** is a component that is able to host Composite UI Modules. Composite UI Modules are User Interaction Building Blocks that are realised as HTML5 and JavaScript widgets. **It** constitues the Runtime Environment of the Composite UI Modules as they - by their nature - cannot be used as stand-alone applications.	**Reference Application for the Earthquake and Forest Fire Domains** **CRISMA Technology Demonstrator** **Reference Application for Exercise Support** **Reference Application for Resource Planning**	*Situator Training System* Owner: **NICE** License: **Proprietary** Development Context: **Extended** *cids Navigator* Owner: **cismet** License: **GPL v2 (controlled OS)** Development Context: **Extended**
UI Mashup Platform The **Mashup Platform** **Building Block** acts as a runtime environment for Mashable Composite UI Modules (widgets) and provides inter-**widget** communication, persistent per-widget configuration as well as proxy capabilities. The platform allows the user (in this case the person configuring a **CRISMA application**) to create a HTML5-based front-end by selecting certain user components (widgets and operations) that can be combined into a mashup. These user components are the **interface** for users to underlying back-end services. The user components themselves are provided by a 3rd party platform through a **catalogue**, from which a range of components can be installed into a mashup and connected ("wired") using a graphical editor.	**Integration of Wirecloud into native applications** **Reference Application for Exercise Support**	*Wirecloud Application Mashup Platform* Owner: **UPM** License: **AGPL v3 (controlled OS)** Development Context: **Used**

Building Block	Use in applications	Components
VTT House model **Model** for estimating the extreme cold weather related **vulnerability** curves for buildings.	**Reference Application for the Nordic Winter Storm Domain** **Cross Border Emergency Crisis (Finland)**	*VTT House service* Owner: **VTT** License: **Proprietary** Development Context: **Extended**

Building Block	Use in applications	Components
Worldstate View The Worldstate View **Building Block** is a set of generic Widgets (Mashable Composite UI Modules) that allow to visualise Control and Communication Information (**CCIM**) related to Worldstates and Worldstate Transitions. Thus, **it** operates mainly on common **meta-information** about the **world** rather than the real data of the world (**World State** Data Slots).	**CRISMA Technology Demonstrator** **Reference Application for the Nordic Winter Storm Domain** **Reference Application for the Earthquake and Forest Fire Domains** **Reference Application for the Coastal Submersion Domain**	*Worldstate Tree Widget (Java)* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **New** *Worldstate Widget (JavaScript)* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **New** *Worldstate Tree Widget (JavaScript)* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **New** *Scenario Evolution Widget (Java)* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **New** *Scenario Evolution Widget (JavaScript)* Owner: **cismet** License: **LGPL v3**

Building Block	Use in applications	Components
		(permissive OS) Development Context: **New** *Scenario List Widget (Java)* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **New** *Scenario List Widget (JavaScript)* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **New** *Worldstate Widget (Java)* Owner: **cismet** License: **LGPL v3 (permissive OS)** Development Context: **New**

2. CRISMA Pilots

This section presents the fully fledged **CRISMA** **pilot** applications which are developed in SP5. On a technology level, these applications are similar to the reference applications which are presented in the section 3 of this book. In fact, in some cases the technology may even be the same, but the difference is at the organisational level:

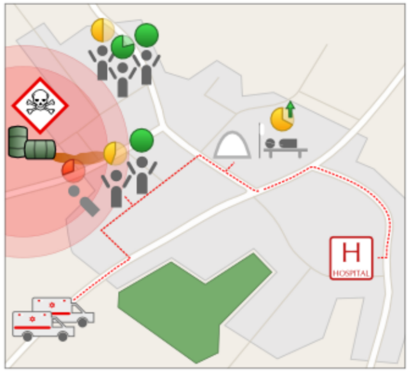
- Pilot applications demonstrate how a real life problem (e.g. "our desktop exercises aren't realistic enough" or "we don't know how to deal with extreme coastal floods") can be resolved using CRISMA. Consequently, they are user-oriented. The underlying data used in the pilot **application** is realistic and the applications are operated by domain experts => the results are relevant for end users.

- Reference applications are used as a proof of **concept** for the technical soundness and usability of the **CRISMA framework**. The underlying data therefore does not need to be realistic and the application is operated by developers => the results are only relevant for the developers.

This difference reflects in the way "pilot applications" and "reference applications" are described on a **catalogue**: pilot descriptions target end-users and the **reference application** descriptions target the developers and integrators.

Accidental spillage from a container at large city port (Israel)

Authors: Oren Deri
Denis Havlik
Chaim Rafalowski



The Israel **CRISMA** **validation** application (Pilot C) focuses on the training of commanders and **decision making** during a large scale chemical **incident**. During a **crisis** **event**, the available resources should be rapidly deployed to the most appropriate tasks. In reality, the globally optimal **resource** allocation is impossible because the Command Centre has neither a perfect **situational awareness** nor the **capacity** to compute the right decision strategy in real-time. The usual strategy for reaching the near-optimal solution in these circumstances is to rely on a combination of written **preparedness** plans, (incomplete) **situation** maps and “gut feeling” of the experienced crisis managers.

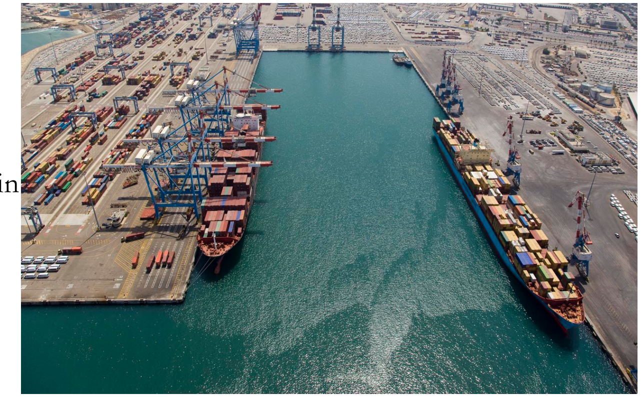
CRISMA support application provides an interactive **model** of the large scale chemical accident and assure that the **impact** of the decisions taken by the trainees is realistic in a sense that the impact is guided by the natural laws and peculiarities of the training setup.

For example:

- Only a limited number of resources are available in the neighbourhood.
- Rescuers and other resources have to rest or resupply after a while.
- The time required for all operations is realistic according to the training situation.
- State of the simulated victims changes over time. The speed and direction of these changes depends on the victim’s history (contamination, decontamination, treatment).
- Particularly badly poisoned victims can only survive if given a timely treatment.

The goal of the **exercise** is to improve the cooperation and coordination of all involved stakeholders as well as their decision making capacity. Through this, the potential ability to **response** on unexpected events will be significantly improved.

Concrete training scenario



The accident scenario starts with spilling of a container transporting bromine. The plume is affecting the port territory and, depending on the meteorological condition, could pose a threat to the population of up to 100,000 inhabitants in the city. In our concrete training scenario, the incident affects one neighbourhood with about 1000 potentially affected persons and several hundreds of victims which are suffering from Bromine related intoxication in different degrees of severity.

This CRISMA pilot application has been tested in the city of Ashdod, Israel. Due to a sheer size of its cargo port (see figure) the probability of incidental spilling of toxic chemicals, including Bromine, is non-negligible, which makes the whole scenario even more realistic. However, the application could be easily re-configured for use at any other location in the **world**.

Assessing the results

At the end of the exercise, an assessment can be made based on objective indicators such as "time required for the operation", "number of deaths", "number and types of resources used".

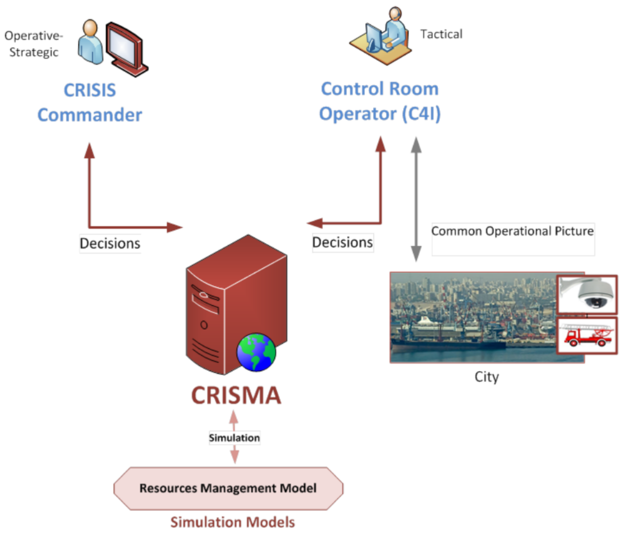
In order to facilitate in-depth analysis, trainers can also construct relevant **criteria** by assigning the good/bad/ugly ratings to values of different indicators. In conjunction with the multi-criteria analysis, the criteria can be used to rank the training results ("best in the class"). In addition, they can also be used to answer various questions concerning the trainees' performance. Some examples:

- Did the trainee manage to save more or less people than expected?
- Did the trainee use significantly more or less resources than expected?
- Did (s)he need significantly more or less time than expected to evacuate all patients which require hospital care?
- Were the established strategic zones (triage, decontamination, which hospitals) established adequate concerning the numbers and types of victims and rescuers, environment and the characteristics of the **emergency**?
- Were they optimally positioned, to facilitate the workflows?

Implementation:

In this **pilot**, the **CRISMA framework** is used for several purposes:

- **Goals**: Using models and simulations for **resource management** training purposes.
- **Stakeholders**: **Crisis** Managers at operative, strategic and tactical levels.
- **Main Resources**: Hospital-Ambulance-Patient interaction.



The Pilot applicaiton is based on the **Reference Application for Resource Management Training**. In fact, the two applicaitons are identical on the level of technology and only differ in the data and in the way the application is used:

- The reference applicaiton is used as a technology playground, and the underlying data is not based on the real information from the field.
- The **pilot** application is operated by domain experts and the underlying data is a realistic representation of the pilot area.

Constituents:

Integration level: Todo for V1
see **Reference Application for Exercise Support**

Documentation:

User Guide

Please follow the link below to watch a video walkthrough of the main **Pilot** C **scenario**.

- Pilot C Walkthrough Hands-on **http://youtu.be/Now2IZtGzJE (http://youtu.be/Now2IZtGzJE)**

Tutorial

Quick User Interface Reference Guide:

System Management View:

Used by:

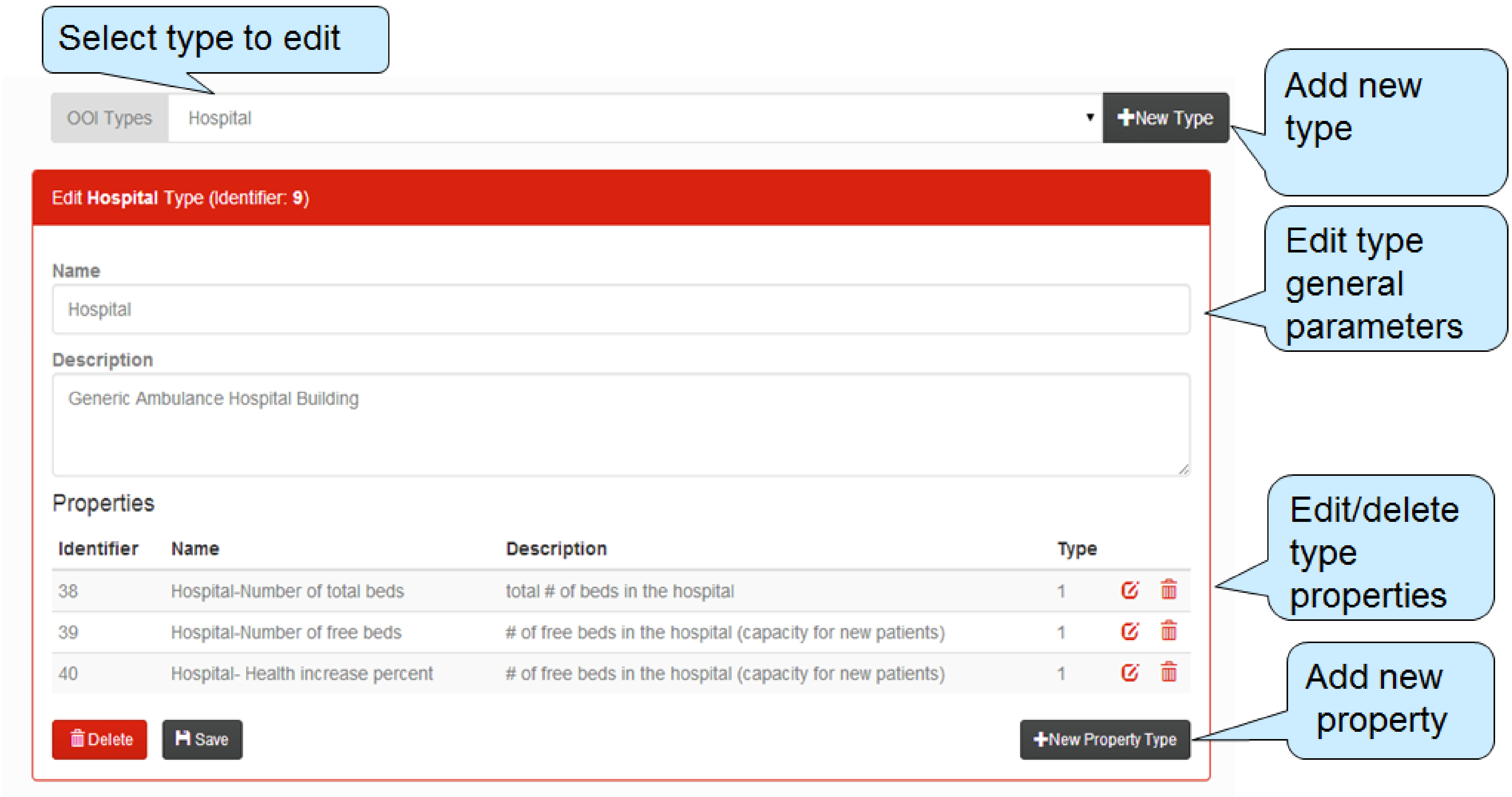
- Administrator

Enables to:

- Modify **OOI** types
- Modify OOI instances (ambulances, patients etc)
- Modify training session templates with initial **world state** data

Training session setup:

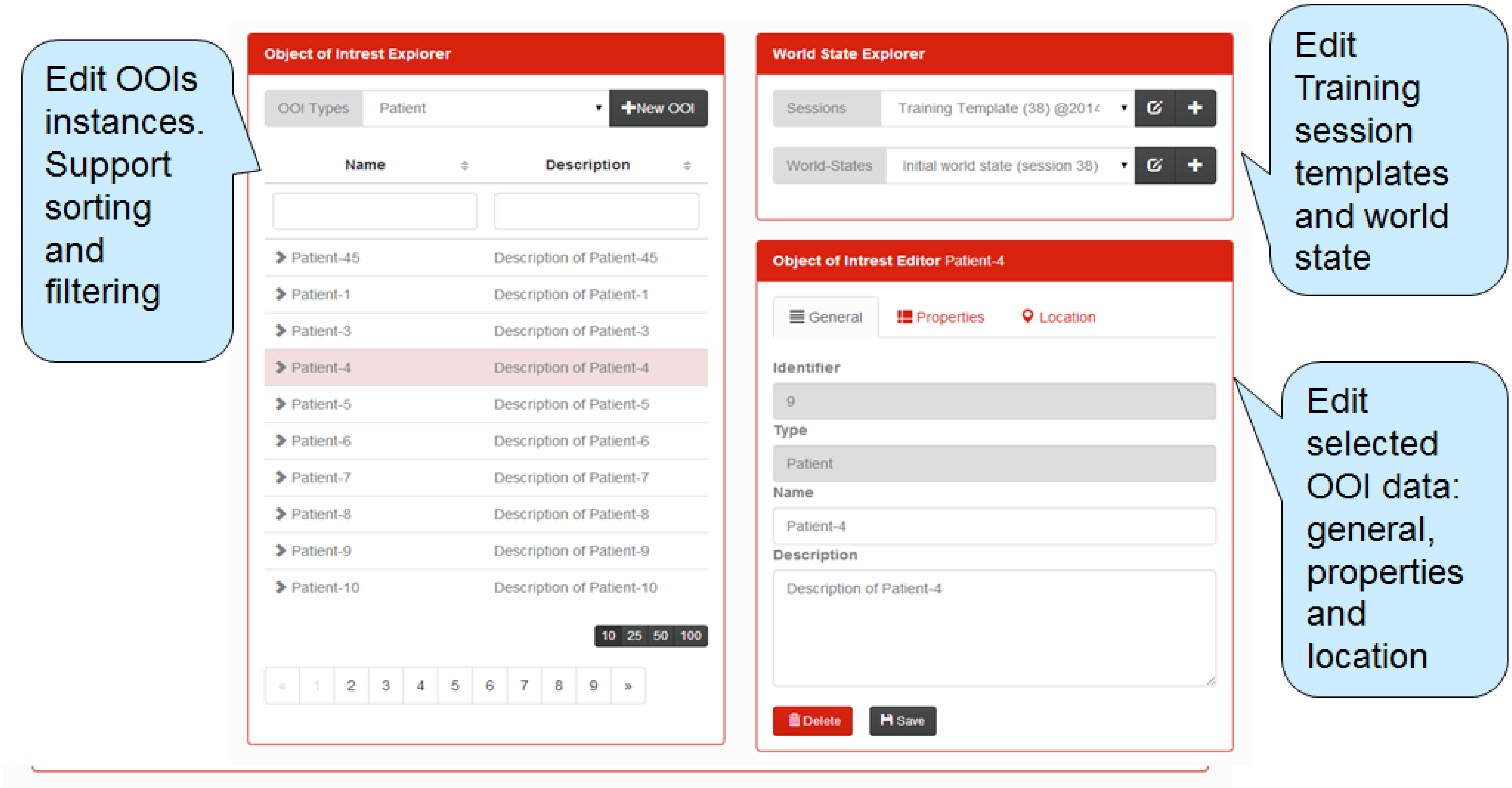
OOI Management – Manage OOI types



Used by:

- Trainer

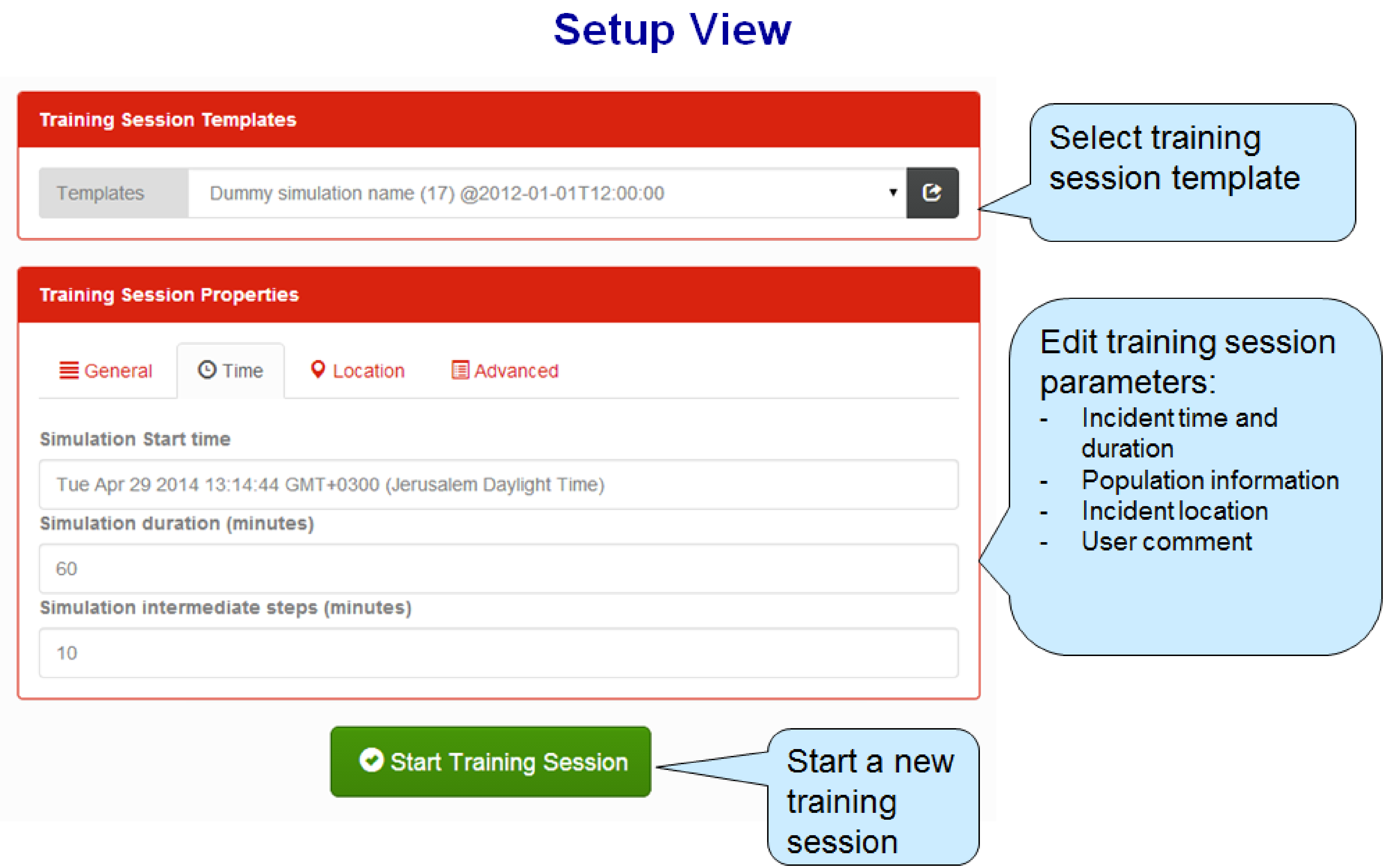
OOI Management – Manage OOI instances and Training Simulation



Enables to:

- Select training session template
- Customize training session properties (patients category, bromine plume, time and location)

- Start a new training session



Monitor and Dispatch View

Used by:

- Trainee (or CRISMA operator)

Enables to:

- Dispatch Ambulances Command by calling the RM **Model**
- Monitor training session via tabular and **GIS** display
- Navigate and visualize world state data
- Open for extensibility by adding new commands

Indicators View:

Used by:

- Evaluator (Trainer)

Enables to:

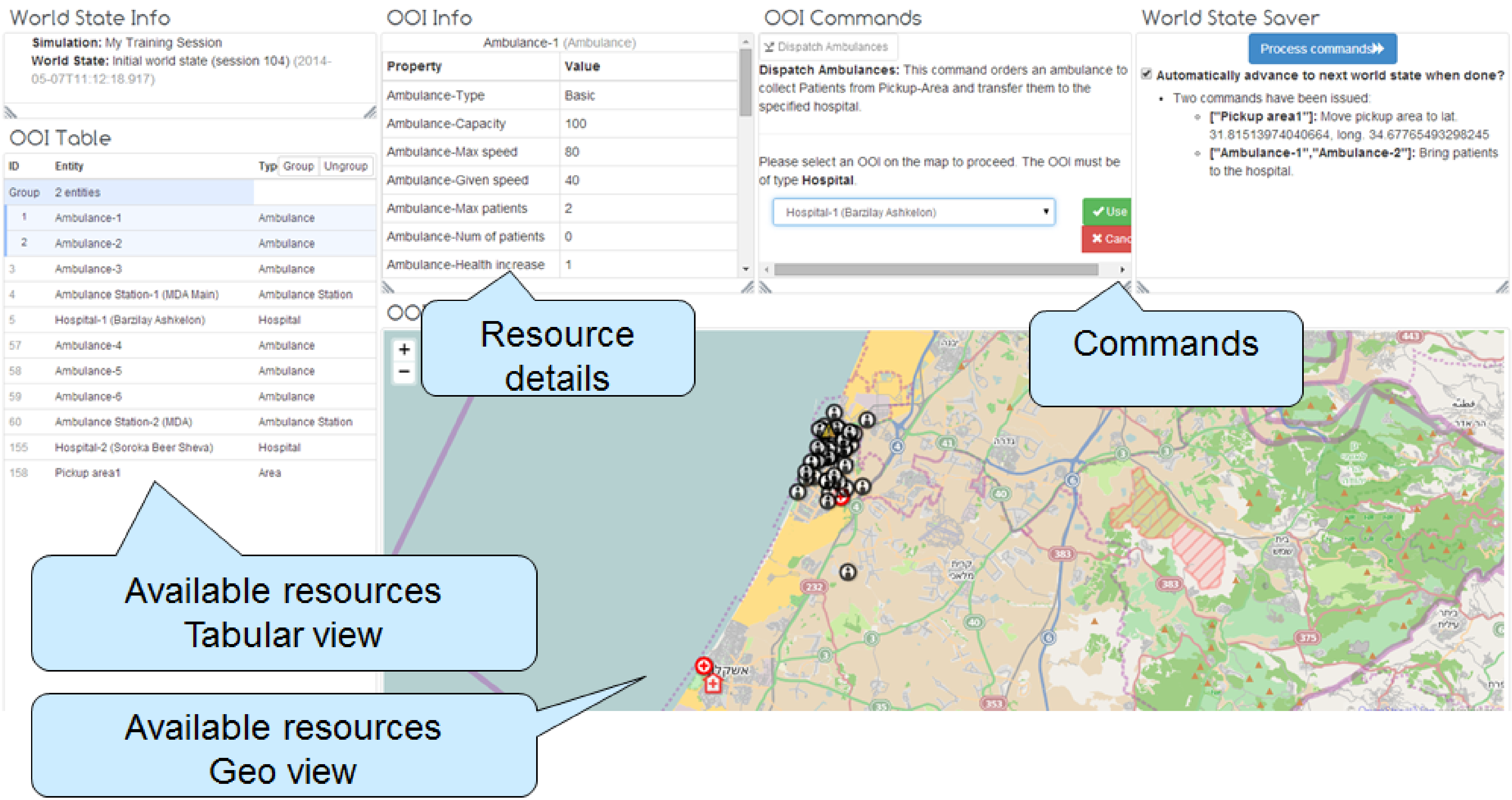
- View world states tree
- View indicators per world state
- Compare world states indicators

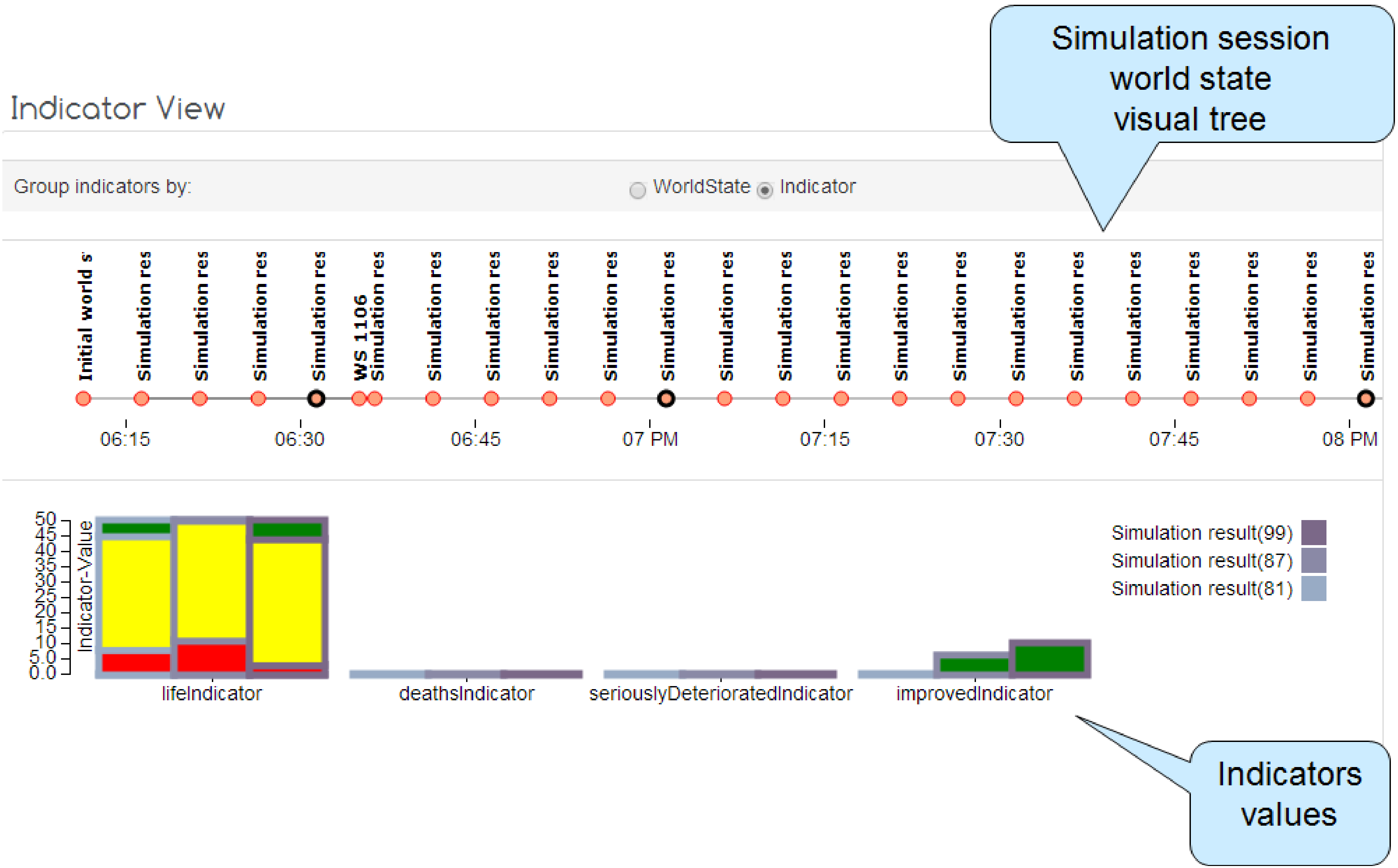
Downloads:

This video demonstrates a walkthrough of the main **Pilot** C **scenario** for v1 **application** (technology proof of **concept**). The v2 application, which will be used in **MDA** trainings, is currently in development

Please contact us for a live demo/hands-on experience with the software.

Resource Management Monitor and Dispatch View




Simulation session world state visual tree
Indicator View
Group indicators by:
WorldState
Indicator
Initial world s
WS 1106
Simulation res
06:15
06:30
06:45
07 PM
07:15
07:30
07:45
08 PM
Indicator-Value
50
45
40
35
30
25
20
15
10
5.0
0.0
lifeIndicator
deathsIndicator
seriouslyDeterioratedIndicator
improvedIndicator
Simulation result(99)
Simulation result(87)
Simulation result(81)
Indicators values

- Demonstration video on Youtube

http://wirecloud.ait.ac.at/pilotC/Setup (http://wirecloud.ait.ac.at/pilotC/Setup#view=workspace)

Team:

Data:

Hospitals

The position of the hospitals determines the time required for the **evacuation** of the patients which can't be treated on the field. The **system** take into consideration the number of free beds in each hospital. As part of the training the desision maker need to choose the prefered nearby hospital and balance the patients between the available hospitals.
In the reference applicaiton, we are simply postulating the hospital positions, but in the **pilot** **application** the real positions of the Israeli hospitals are used. This data is private and confident per Pilot C application.

Patients

Patients are humans that were affected by the Bromine plume. Each patient consists of health status that indicates his **situation**. Non treated patient will decrease their health status. Treated patients will increase their health status. Only Ambulance or Hospitals can treat patients.

Ambulances

Ambulances are vehicles associated with paramedics and treatment equipment's. Each Ambulance is associated with a specific Ambulance station. Ambulance can rescue patients from the danger zone, Treat patient at a designated treatment area or evacuate patient to hospital. There are two types of Ambulances in **Pilot** C: Basic and Advance. The Advanced type can treat Patients faster. Each Ambulance has resources that decrease as part of the treatment process. Once resources are empty the Ambulance need to go to his home Ambulance station and refill his resources.

Ashdod City GIS Map

A real geo reference map containint streets, distances and point of intrests. In **Pilot** C we are using the online Open Street Map that can viewed also from the following web site: **http**://**www.openstreetmap.org/ (http://www.openstreetmap.org/)** .

Population density

This applicaiton needs to know where the population is in order to predict how many people are endangered. In theory, we could use a realistic population density like in Italian and French pilots, but this was considered both an overkill and problematic from ethical point. As a result, the population at **risk** can be postulated by the trainer prior to start a new training session.

Treatment area

Treatment Area is a physical area that used for treating patients. **It** possible to set multiple treatment areas. The areas location is part of the decisions that a trainee needs to take. As mentioned above user need to select the specific area when dispatching ambulances to treat, evacuate or rescue patients.

Incident Area

Incident Area is a physical area that represents the center of the incident. Each incident is associated with a Plume that indicates the **affected area**. Patients that are close to the incident center will be more affected by the plume and their initial health status will be lower.

Ambulance Station

Ambulance Station represents a physical station where Ambulance park and refill. Each Ambulance is associated with a single Ambulance Station. Each station can host a limited number of ambulances.

OOIs Overview Diagram

The below diagram illustrate the relationships between **Pilot** C OOIs:

Coastal submersion defense for the Charente Maritime region (France)

In February 27-28th 2010, the Charente-Maritime was touched by the storm surge **Xynthia**. The consequences of the storm were significant : 47 people died and direct **losses** amounted to more than 2.5 billion Euros. After such **disaster**, civil protection and disaster management authorities are faced with the following questions :

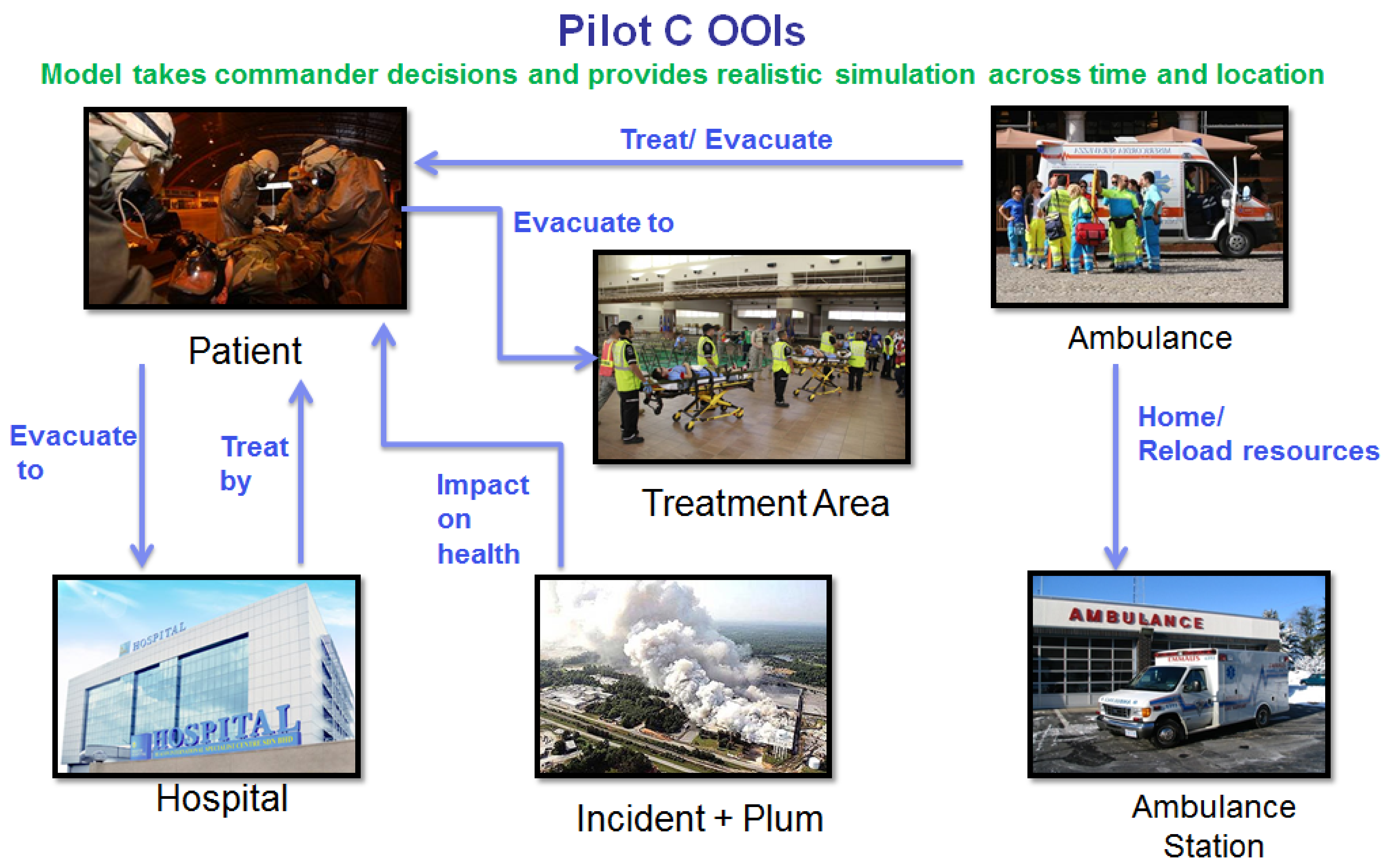
- Could we avoid the disaster?
- What are the reasons for this disaster?
- How to avoid such disaster in the future?

CRISMA **pilot** B attempts to answer these questions and investigates the consequences, impacts and damages of different coastal submersion situations depending on the various **mitigation** activities.

Charente-Maritime Coastal, Boucheleur (FRANCE)

The **Application** for the Coastal Submersion Domain demonstrates the capabilities of the **CRISMA Framework** regarding:

- **Simulation** of the submersion effects at a range of temporal and spatial scales
- **Preparedness** Planning
- Assessment of impacts depending on scenarios based on options for managing the risks
- Cost / benefit analysis



Authors: **Mehdi Pierre Daou**To do that, the main functionalities offered to users by the **Reference Application** for the Coastal Submersion Domain are:

Marianne Grisel
Armonie Cossalter

- Run coastal submersion simulation at global and local scale for different Coastal Submersion scenarii
- Evaluate the result of the simulations (water level, water velocity, water level maximum …)
- Evaluate the dike damages probability
- Modify the coastal defense (dikes) behavior (introducing breach or total failure) aiming to run new simulations and evaluate the damages
- Run **evacuation** **model**
- Compare the evolution of the coastal defenses and damages depending on scenarii of costal submersion
- Analyze indicators
- Evaluate the economic losses

Implementation:

This **application** is based on the "**Reference Application for the Coastal Submersion Domain**". In fact, these two applicaitons are identical on the technology level, and only differ in the way they are fed with data and in the way they are operated:

- The "**Reference Application for the Coastal Submersion Domain**" is used as a technical playground. That is, the scenarios and the data have very little to do with reality.
- The **Pilot** applicaiton uses the most accurate data from the local stakeholders and the scenarios are designed and executed by domain experts. Consequently, the results are considered "decision quality" and shared with the local stakeholders.

The various stages that we want to apply in Pilot B for the end purposes:

- Define the **event** characterization
- Simulate coastal submersion.
- Perform dike and building **vulnerability** assessment
- Modify the dikes **resilience**.
- Simulate the **evacuation** **Model** for coastal submersion

- Calculate indicators
- Calculate economic **losses**
- Calculate indicators for the coastal submersion
- Choice **mitigation** measures

These different stages should allow to analyze the submersion effects (in spacial and temporel scale, cost/benefit,...) , and to do **preparedness** planning with the assessment of impacts depending on scenarios.

Constituents:

Integration level: Todo for V1
See related **reference application**

Documentation:

Tutorial

TODO: please add some documentation here.
Video which explains utilisation of **application**.

Downloads:

We have a platform with the **demonstration** of the **application** V1 but **it** isn't accessible for this moment.

- Demonstation of application
  http://crisma.spacebel.be/pilotB (http://crisma.spacebel.be/pilotB)

Team:

Marc Erlich

Development roles: coordinaton
Marc coordinates the **pilot** B.

Mehdi Pierre Daou

Development roles: integration, modelling
In charge to create the coastal submersion **model**.
Help for the integration the building blocks within **Pilot** B.

Arnaud De Groof

Development roles: architecture, configuration, development, installation, integration
Responsible for development of the **Simulation Model** Interaction **Widget**.
Provides support for the integration of simulation models (Simulation Model Integration **Building Block**)

Marianne Grisel

Development roles: modelling
In charge to create the **Evacuation** **model** for coastal submersion.
she is working on the indicators.

Data:

Dams

Dam height and quality is one of the key variables determining the probability of a dam breach and subsequent coastal submersion.

Hydrodynamics model data

The evolution of the mean sea level at the marine boundary in time comes tidal measurement.
The wind velocity of area comes the weather station measurement.
The dame (or dikes) see the previous description. (used also in dikes **vulnerability model**)
Tubes comes field measurement of structure position.
Swell data comes swell model (Tomawac which is a modul of Open-Telemac- Mascaret).

LSM data

Population repartition known by work on INSEE data ("données carroyées")
Road network (data by IGN, BD TOPO)
Buildings (data by IGN, BD TOPO)
The **warning** centers (localization known by field investigation)

Earthquake and forest fire application (Italy)

The Italian **CRISMA** **validation** **application** (**pilot** D) simulates a natural **disaster** with irreversible damages in the Region of L'Aquila. The main triggering **event** is an earthquake, and a forest fire will be considered as a possible cascading event in this pilot.

The main pilot objective is to compare various **crisis management** scenarios in a multi-**risk** **framework** including the **cascading effects**.

L'Aquila region is infamous for the recent serial earthquake event with catastrophic consequences. Until now, the consequences of such "low probability/high **impact**" event were difficult to evaluate, and consequently not taken into account by the crisis managers. In the scope of Pilot D, the CRISMA team will illustrate how such events can be modelled and the resultign knowledge incorporated in the crisis management planing.

Geophysical Hazards – L'Aquila Italy (Photo AMRA)

The pilot application presents the critical issues for **decision making** during a seismic crisis and helps the decision makers by calculating the possible consequences of their decisions in a number of intervention/no-intervention scenarios, corresponding to different possible adverse event propagation (cascading effects) evaluated in a multi-risk framework.

Forest Fire, Portugal. (Photo ADAI)

Implementation:

In this **pilot**, the **CRISMA framework** is used for several purposes:

- **Preparedness** and planning – allowing the best distribution of resources
- Training – simulating intervention scenarios after an earthquake or during forest fire
- **Response** – supporting decision makers by providing alternative management policies
- Quantitative assessment of **impact** scenarios (simulated by assumption of alternative strategies) to compare
- Cost/ benefit analysis
- Multi-**criteria** analyses
- Assessment of choices (in planning and/ or **emergency**) on the basis of scenarios and Cost/ benefit Analyses.
- Compilation of information in one single tool and faster access to **it**

The Pilot applicaiton is based on the **Reference Application** for Earthquake and Forest Fire Domains. The Reference Application is filled with **Scenario** specific data and models.

Constituents:

Integration level: Todo for V1

See related **reference application**

Documentation:

User Guide

Please follow the link below to watch a video walkthrough of the main **Pilot** D **scenario**.

- A video walkthrough of the main Pilot D scenario **https://www.youtube.com/watch**

Downloads:

- Pilot D HTML Application Development Snapshot (requires authentication) **http://crisma.cismet.de/refApp (http://crisma.cismet.de/refApp)**
- Pilot D workflow demonstrator cids customisation **https://github.com/crismaproject/cids-custom-crisma-pilot-d**

Team:

Data:

List of Dataitems used in Pilot D

N°	Data category	Name	**Resource** title	Resource abstract
1	**Exposure**	REGIONE	Administrative Region Limits	Administrative limits of Abruzzo Region

2	Exposure	PROVINCE	Administrative Provinces Limits	Administrative limits of Provinces
3	Exposure	COMUNE_AQ	Administrative Municipalitiy Limits	Administrative limit of Municipality
4	Exposure	SEZ_ITA_01	Italy Census sections limits	Census area limits for data collected by "Istituto Nazionale di Statistica" (ISTAT)
5	Exposure	CLC2006	CORINE Land Cover 2006	COoRdinate INformation on the Environment - Land cover/land use map for non-commercial use provided by the European Environmental Agency (EEA)
6	Exposure	CENTRO_DEM_25	SRTM based Contour Lines (from OpenDEM)	Vector contours tile made at 25m interval for the **pilot** case zone, from a Digital Elevation **Model** (DEM) of terrain made by OpenDEM
7	Exposure	STRADE	Roads subset for the case zone	Vector lines selected subset for the road network in the case zone
8	Exposure	POWER_TOWER	Electricity distribution power lines towers	Points elements that georeference the position of powerlines towers

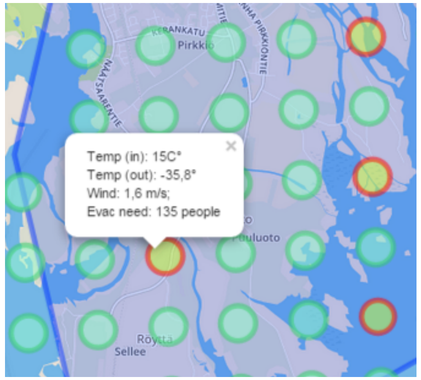
9	Exposure	EDIFICI	Buildings in the interest region	Buildings polygonal shapes in the interest region from cadastral or aero-photogrammetric recognition, with building type category
10	Exposure	GRID	Polygon layer with square cells placed over the analysis region extension	Polygon layer with square cells placed over the analysis region extension with 250x250m cells
11	Exposure	Z_C	Intersections grid cells - census areas	Zones resulting from intersecting grid cells with census areas
12	Inventory database	ISTAT_2001_B	Buildings Census data	Statistical data on buildings obtained from census (ISTAT) databases, related to census areas
13	Inventory database	ZONE_INVENT	Elementary unit of study data inventory	Data inventory on buildings, population and buildings seismic **vulnerability** classification for each elementary unit of study
14	Inventory database	ISTAT_2001_P	Population Census data	Statistical data on population obtained from census (ISTAT) databases, related to census areas
15	Inventory database	POP_TDISTR	Time related population distribution	Text tables of time related population distribuition (?)

16	Inventory database	EQV_ROADS	EQ_V distribuition along roads	Distribution of EQ vulnerability for each road segment
17	**Hazard**	EQ_CHAR	Seismic **event** characterization	EQ event(s) data characterization
18	Hazard	PGA_GRID	Shake map (PGA)	EQ shake map (PGA): Peak Ground Acceleration data referenced to each of the grid cells considered in the study region
19	Hazard	RISPOSTA_SITO	Site effects	Data in the study area for taking into account seismic local amplification effect due to the site geology to be used to modify the EQ effect.
20	Vulnerability	**DPM**	Damage Probability Matrices (DPM)	DPM Data tables stored in the model database for discrete values of Intensity
21	Exposure	FUEL_MAP_LAQUILA	Fuel map of Pilot D (L´Aquila)	Map describing fuel cover of the region o L'Aquila with a resolution of 50x50m2, produced by ArcFUEL project with whom **CRISMA** establish a collaboration protocol
22	Exposure	DEM_FF_LAQUILA	Digital terrain map of Pilot D (L´Aquila)	Digital elevation map from the pilot region derived from the Global Digital Elevation Maps produced by NASA and available at EOSDIS web site (**http**://earthdata.nasa.gov/)

23	Hazard	METEO LAQUILA	Meteorological observations	Meteorological data obtained from local meteorological stations
24	Vulnerability	DPM FF	Damage Probability Matrices (DPM)	DPM Data tables stored in the model database for discrete values of Intensity

Cross Border Emergency Crisis (Finland)

Authors: **Hanna Honkavuo**
Ari Kosonen
Markus Jähi
Kalev Rannat



The Finnish CRISMA **validation** **application** (Pilot A) focuses on contingency planning to mitigate the **impact** of a large scale power outage occuring during winter time.

The **simulation** models included in this application estimate the pace in which the buildings left without electricity are cooling below acceptable level. With the population data of the area, the **evacuation** actions can be focused on the areas with most vulnerable people in need of help. The **preparedness** planning functionality allows the user to find the optimal set of resources for evacuation actions. The alternative **mitigation** action is to ration the availablity of the power, i.e. returning electricity to impacted areas and possibly cut the electricity temporarily from areas which are more likely to survive for extended periods of time, due to more energy efficient buildings.

The application allows the decision managers to **experiment** with various **crisis management** strategies and compare the results based on factors such as economic **losses**, human suffering and use of resources. The goal is to compare different scenarios to determine the best decisions to prepare and respond to the **situation**.

The concrete **scenario** explored in this pilot is the following: The area of Kemi-Tornio (South-West Lapland, in Finland) is hit by an extreme winter storm, followed by cold temperatures, leading to increased demand for electricity. Simultaneously, the production and supply of electric energy is paralysed, leading to power outages.

Modern heating systems are vulnerable to power blackouts. Consequently, the internal temperature of buildings will drop. To avoid the worst case scenario, the decision makers will ration the energy supply, assuring the majority of buildings are heated for at least a few hours per day. Depending on the available energy supply as well as on the type of housing, this measure will either stop or at least slow down the process of house cooling. If the outage persists, the decision makers need to plan evacuation of habitants before the internal temperatures fall below a critical level, causing hypothermia for the residents. In the area of Kemi-Tornio, the help of

Sweden might be needed as the area is sparsely populated and the distances are long.

The pilot application incorporates realistic population, house cooling, **resource management** and **economic impact** models. This allows the decision makers to:

- plan the evacuation timing, based on realistic thermal cooling models for different housing types;
- plan the resource allocation for evacuation, based on realisitic assumption on resource availabiity and capabiilties; and
- compare the human and economic impacts of different crisis management strategies.

Implementation:

This **application** is based on the **Reference Application** for the Nordic Winter Storm Domain. In fact, the two applicaitons are identical on the level of technology and only differ in the data and in the way the application is used:

- The reference application is used as a technology playground and the underlying data is not based on the real information from the field.
- The **pilot** application is operated by domain experts and the underlying data is a realistic representation of the pilot area.

Constituents:

Agent Oriented Simulation Models

Integration level: Integrated

See related **reference application**

Economic impacts analysis view

Integration level: Integrated

TBD

Preparedness Plan BB

Integration level: Integrated

Preparedness Plan **BB** is a generic preparedness planning tool which can be applied to multiple scenarios with user-defined plans.

It maintains the **resource** information internally, or alternatively it can use resource information from external sources.

When the user excecutes the chosen preparedness plan, the tool provides set of resource allocation proposals, based on the rules defined in resource models.

OOI World State Repository BB

Integration level: Integrated

OOI **World State** repository is used as the centralized storage for the **Object of Interest** in this **pilot** **application**. **It** it also acts as a communication channel

between the other building blocks.
The OOIs stored to the OOI-WSR are: geocells, buildings, population

Integrated Crisis Management Middleware BB

Integration level: Todo for V2
TBD

Publish Subscribe Context Broker BB

Integration level: Optional
TBD

VTT House model

Integration level: Integrated
VTT House **model** calculates the cooling gradients of the residental builidings within the target area with given weather conditions and defined building types.

Economic impacts model

Integration level: Integrated
Economic impact **model** calculates the economic consequences of the **scenario** as well as of the **mitigation** actions.
Typical cost classes are, for example, transportation, accomodation and staff costs related to **evacuation** actions.

Resource Management Model

Integration level: Todo for V2
TBD

Integration level: Optional
Human Thermal **Model** provides information on how different groups of people (e.g. different age groups) are impacted by cool environment. E.g. elderly people may be in need of **evacuation** actions in earlier stage than younger people.

Documentation:

User Guide

TODO: add some user-level documentation, e.g. a tutorial explaining how to use the **application**.

Downloads:

TODO: As soon as teh **application** exists, please link **it** here. Could be either online, or downloadable and usable "as is" after the download. Linking the source code or somethign that's complex to get running here is a no-no, you can do that under reference apps.

fallback: maybe just a movie showing how the app is used?

Team:

Data:

Building data

In the Building Data the residental buildings are categorized according their year of construction in three age groups. The houses in the oldest age group are assumed to be thinly insulated and thus they are cooling within few hours in winter condition without electricity. The houses in the youngest age group are build according to the modern low-energy regulations and they may provide adequate shelter for several days. The third and probably the largest group is concisting of modern houses which are not build according to the latest regulations, but they still will survive at least few hours without electricity, but a long term breakdown will cause the indoor temperature falling below critical level.

The building data is based on the real estate information provided by the Population Information **System** (Finland).

Weather data

Weather data concists of temperature and wind strength values prevailing on the area.
The temporal resolution of the data depends on the **world state** step size chosen. Geographical resolution follows the size of the geocells (1 km x 1 km) but **it** can also be more coarse.

The data is based on the realistic weather information provided by the experts of **FMI**.

Population data

The population data focuses on the habitants of the residental buildings in the target area. The population is categorized into three age groups: persons below the age of 16 ("Children and adolescents"), persons between 16 and 65 years of age ("Able-bodied persons") and people above 65 years ("Elderly people"). The classification is done to estimate the need for **evacuation** or other **mitigation** actions related to the population.

The population data is based on the population information provided by the Population Information **System** (Finland).

Electric grid topology data

Topology of the electric grid in the target area. A disturbance in a given point of the electric grid is assumed to have an **impact** to all consumers in the geocells which the failing point serves.

Resource data

Resource data concists of information about resources that potentially can be used to mitigate the **crisis** **situation**.
For each resource unit, following information, for example, is provided:
- **Capability**. which the resource unit can provide (e.g. transportation of 50 persons)
- Default location
- Status (e.g. availability and related constrains)
- Cost data

Resource data may originate from external resource databases or **it** can be added and edited by the user.

Multi hazard site (Germany)

Authors: **Martin Scholl**
Denis Havlik
Johannes Sautter
Sandra Frings

The German **CRISMA** **validation** **application** (**pilot** E) focuses on **capacity** planning for the support of first responder organizations in mass accidents. The **exercise** support application developed in the scope of this pilot facilitates the planning, recording and assessment of first responders' activities during such accidents, mainly with respect to time required for various activities such as pre-triage, triage, on-site care and **evacuation** of the critically injured patients to the hospital.

The pilot application is designed with transferability in mind: **resource management** models for e.g. patients and ambulances are designed to be very generic and the site- and **experiment**- specific features, such as ambulance capabilities, type and severity of the injuries are introduced in the application as **model** parameters. These parameters are determined at exercise-planning time and entered in the **system** by the end users.

In July 2014, the first version of the mass casualty **incident** exercise support application has been tested in a bus crash **accident** **scenario** in a rural alpine region of southern Bavaria. In this exercise, 25 passengers located in the bus and inside the car had to be given adequate aid. As witnessed by two accidents which occurred within days of the CRISMA exercise, this is a rather common type of accidents which nevertheless overwhelms the local first responders:

- A train/truck **collision with 33 injured passengers near Stuttgart (http://bigstory.ap.org/article/33-injured-after-train-rams-truck-germany)** on July 4-th 2014
- A bus/bus/minibus **collision with nine death and 44 injured passengers near Dresden (http://www.dailymail.co.uk/news/article-2698233/Nine-dead-43-injured-multiple-bus-crash-eastern-Germany.html)** on July 19-th 2014

CRISMA (v1) pilot application was used to record the timing of various activities including the requests and arrival times of various resources, time needed for pre-triage, triage, on-site measures and transport of the critically injured patients.

Bus accident on a rural highway, Germany (Photo: DRK)

This information was immediately processed by the and casualty incident exercise support application and used to complement the self-reflection of the exercise participants during debriefing, as well as for the comparison of the two exercise runs.

Second exercise scenario, which will be tested in early 2015, describes an accident during a mass **event** at a stadium or congress centre. This type of events attract large number of visitors, and the exercise scenario foresees around 200 injured visitors with several nationalities (European and non-European). In this exercise, the **CRISMA application** will be also used for exercise planning and defining the operative plans for the exercise location.

Implementation:

The German **Pilot** builds on two **CRISMA** reference applications:

1. "**Exercise** support" **reference application** is used to record the timing of various activities during the exercise, assess the results and to provide information for debriefing.
2. "**Resource** planning" reference application is used in exercise planning and defining of the site-specific operative plans.

Two applications intersect at two levels. First, the various timings recorded during the exercises are used as **model** parameters for the planning application. Second, the timings predicted by the planning application are used as a baseline to measure the progress and outcome of the exercise.

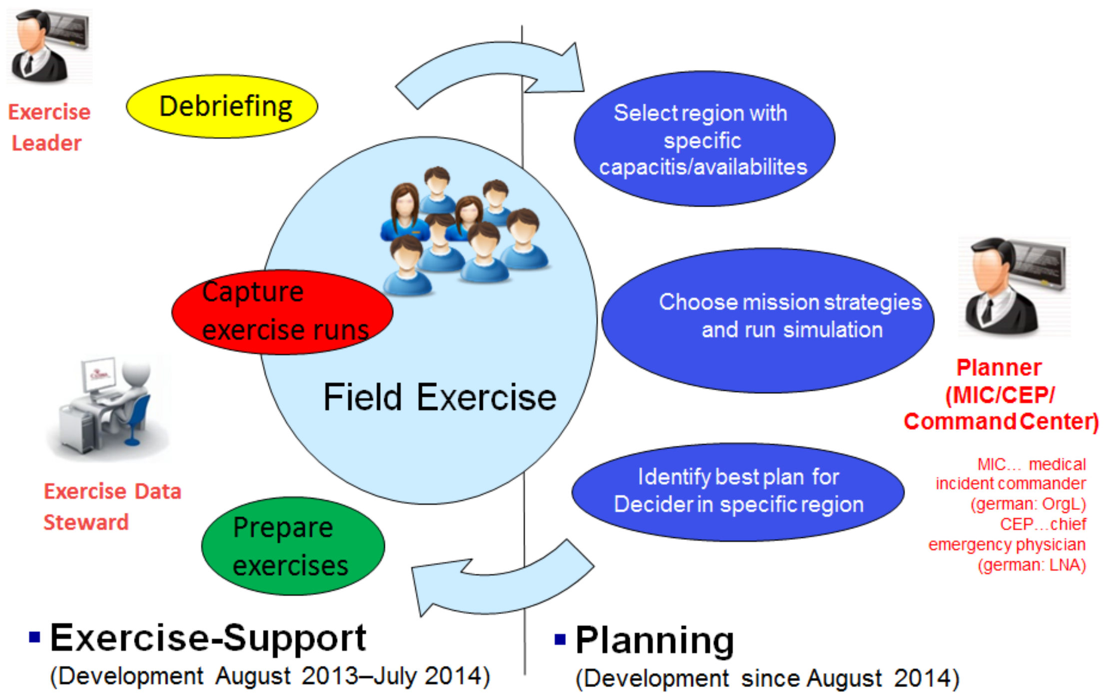


Figure1: Pilot E Applications and how they relate to each other

Our key assumption is that this approach will results in a self-correcting cycle with increasingly more realistic plans (compare Figure 1). Since the exercise is designed to be as realistic as possible (e.g. ambulances can't immediately arrive at the scene), these improved exercise plans are eventually re-used as he base for the site-specific operational plans.

Constituents:

Integration level: Integrated

See related reference applications

Documentation:

Presentation

Downloads:

TODO. AFAIK we have an online instance of the v1 **application**

TODO: AFAIK we have the **application** packed as a set of virtual images which can be installed on a lightweight server (well, **it** works on a normal dell laptop)

Team:

Friederike Schneider

Development roles: mediation, testing

Pilot Owner

Johannes Sautter

Development roles: coordinaton, mediation, requirements, specifications, testing, validation

Technical Project Manager, Requirements Engineer, User **Interface** Designer, Usability Engineer

Frank Jonat

Development roles: development, modelling, validation

Modeller working on the **Resource** Allocation **Task** (RAT) **Model**

Maria Egly

Development roles: architecture, configuration, development, installation, integration

Technical Integrator of the **Resource** Planning **Application**, User **Interface** Integration **Exercise**-Support, UI **Developer**, Installation and Integration of User Interface Platform

Martin Scholl

Development roles: architecture, coordinaton, development

Technical Integrator of the **Exercise**-Support **Application**, User **Interface** **Developer**

Holger Bracker

Development roles: integration, support

Coordination, Responsible of Data and Calibration of the **Resource** Planning **Application** using **Exercise**-Results

Data:

Background maps

Orthophoto and Streetmap of **incident** area are used mainly for illustration.

Ressource characteristics

The "planning" **application** relies on realistic **model** of the ressources which are available in the target area. Therefore the parameters such as the ambulance speed and **capacity** and times required for various activities need **to be defined** for each site.

3. Reference Applications and technology demonstrators

This section presents the reference applications as defined by SP3 team. They are closely related to the pilots, but concentrate on demonstrating the use of the **CRISMA framework** for **pilot** developers rather than on the **end user** experience and usability.

CRISMA Technology Demonstrator

The **CRISMA** **Technology Demonstrator** is a prototypical **application** for demonstrating the capabilities of the **CRISMA Framework** regarding

- Worldstate Management and Visualisation,
- **Scenario** Navigation and Analysis,

- Mulitcriteria Analysis and
- Decision Support.

The CRISMA Technology Demonstrator is not realised as **Reference Application** that can be used for the development of CRISMA Pilots. However, **it** demonstrates how Components and Building Blocks of the CRISMA Framework can be assembled and integrated to create a new **Crisis Management** **Simulation** Application. Thus, the Technology Demonstrator my be used as basis for the development of other Crisis Management Simulation Applications.The show cases demonstrated by the Technology Demonstrator is based on the Crisis Management Scenarios of **Pilot** D (Earthquake and Forest Fire) of the CRISMA project. It is prototypically implemented using the cids platform and demonstrates the main Pilot D **use case** (Earthquake **impact** and **mitigation**) as well as showcases the overall CRISMA **business logic**.The show cases demonstrated by the Technology Demonstrator is based on the Crisis Management Scenarios of Pilot D (Earthquake and Forest Fire) of the CRISMA project. It is prototypically implemented using the cids platform and demonstrates the main Pilot D use case (Earthquake impact and mitigation) as well as showcases the overall CRISMA business logic.

The following screen shots of the application provide a small overview of some of the capabilities of the CRISMA Framework regarding Mulitcriteria Analysis and Worldstate Visualisation. For a complete overview of the functionalities please refer to the demo videos of the Technology Demonstrator which can be found under documentation tab.

Worldstate Navigation

Worldstate Visualisation

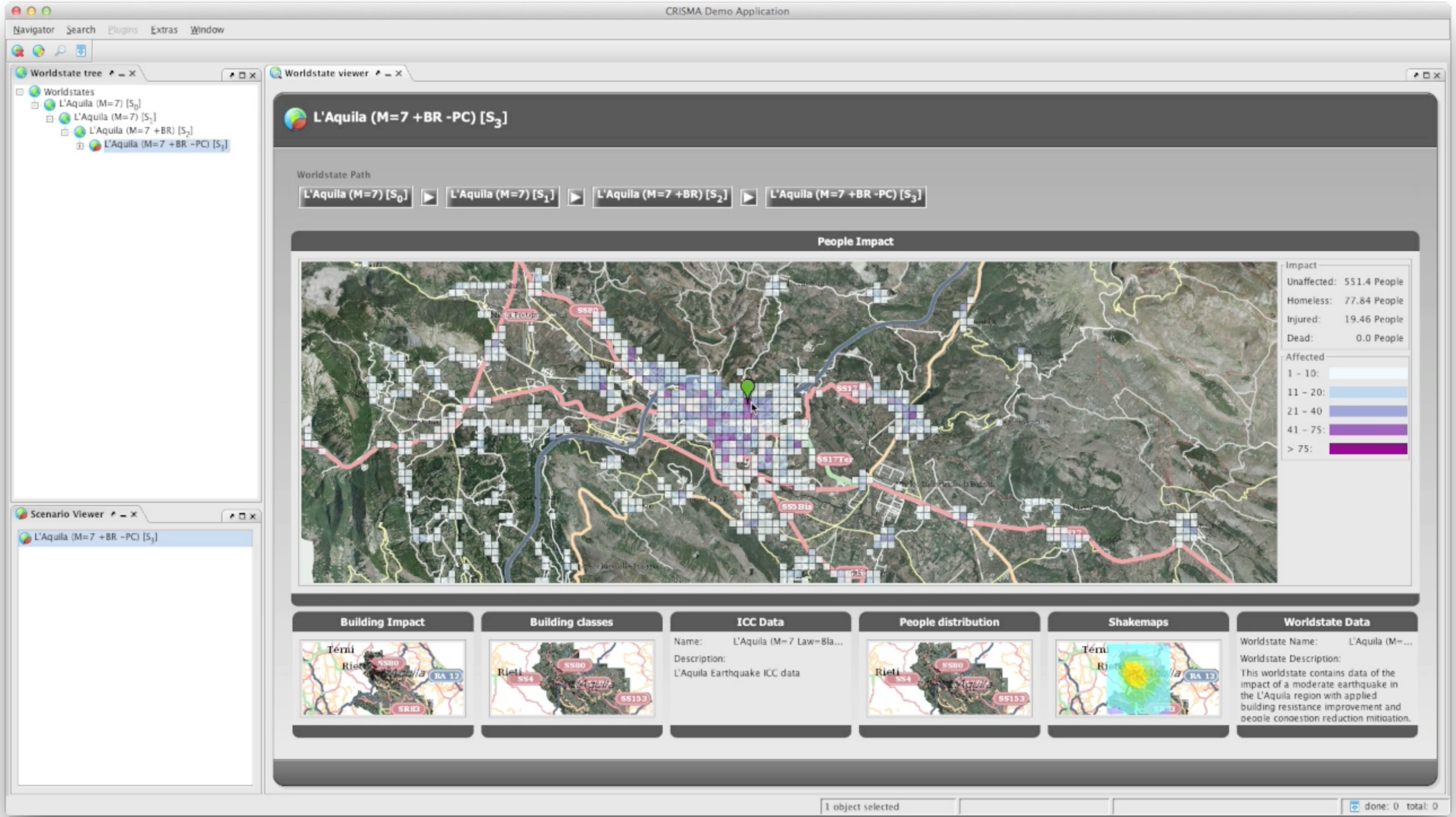
Indicator Visualisation

Multicriteria Analysis

Implementation:

Since the **Technology Demonstrator** showcases also representative use cases of **Pilot** D (Earthquake and Forest Fire), **it**'s **architecture** is nearly indentical to the architecture of the **Reference Application** for the Earthquake and Forest Fire Domains. The Technology Demonstrator is implemented in the Java programming language and makes intense use of the java-based **GUI** Components of the **CRISMA Framework**. The Reference Application for the Earthquake and Forest Fire Domains on the other hand is based entirely on JavaScript GUI Components and thus servers as the basis for the implementation of the CRISMA Eartquake and Forest Fire Application for Pilot D.

There are currently two major versions of the Technology Demonstrator available, which demonstrate the capabilities of the CRISMA Framework V1 and V2.



v1 showcases

- Worldstate management and visualisation
- Simple **scenario** comparison
- **Model** control

v2 showcases

- **Cascade effects** integration
- Scenario analysis and comparison
- Multi **criteria** analysis and decision support

Constituents:

Data Integration

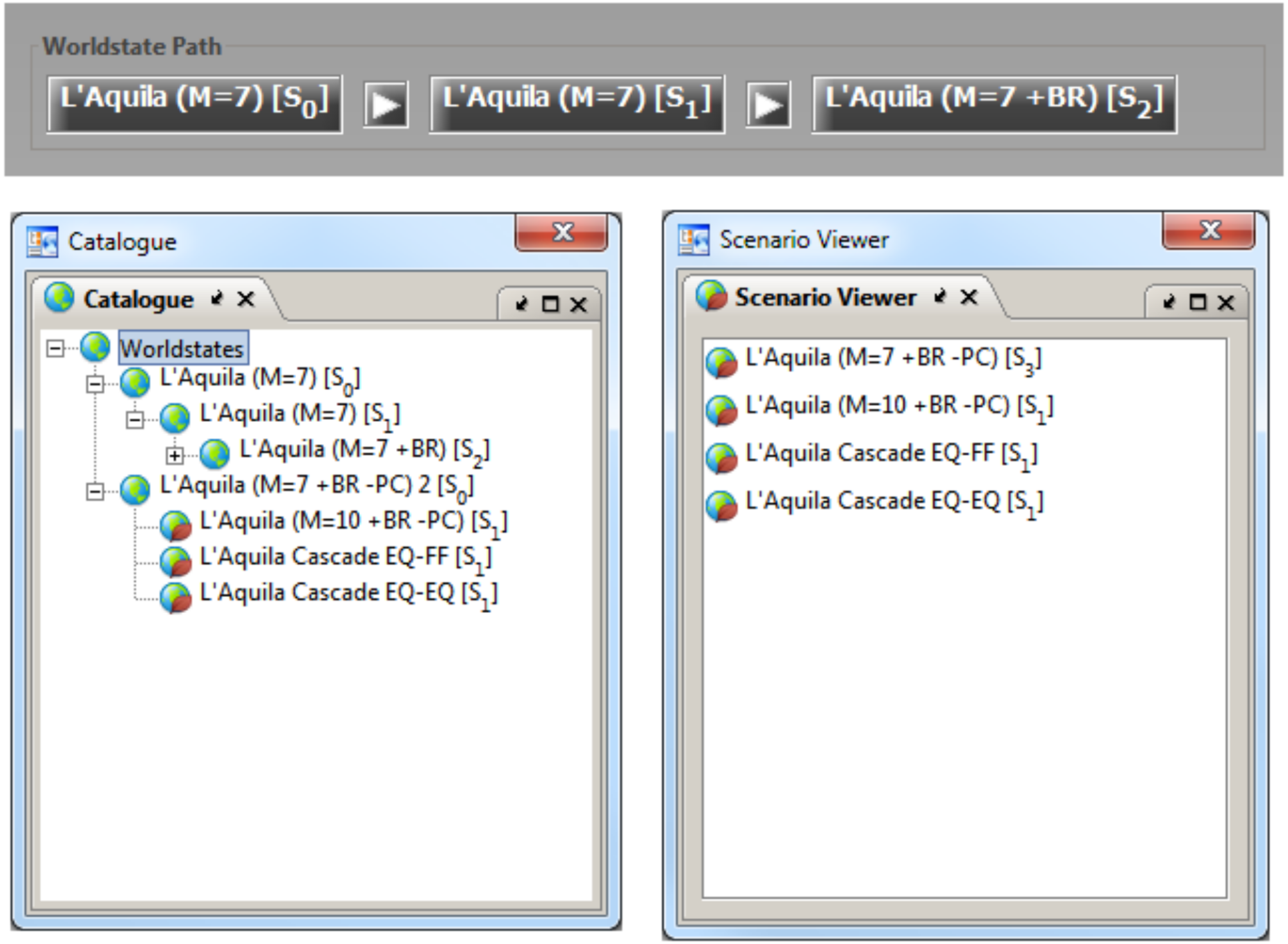
Integration level: Integrated

WMS/**WFS** Services are used to provide the relevant data.

Building impact model

Integration level: Optional

The building **impact model** is not wrapped yet thus **it** is not integrated.



GIS View Building Block

Integration level: Integrated

The Java-based implementation of the **GIS** **Widget** **BB** (cismap) is integrated into the **demonstrator**.

Integrated Crisis Management Middleware BB

Integration level: Integrated

The Java-based implementation of the **ICMM** (cids-server) is integrated into the **demonstrator**.

Worldstate View

Integration level: Integrated

The Java-based implementation of the **Scenario** List **Widget** that is part of the Worldstate View **BB** is integrated into the **demonstrator**.

The Java-based implementation of the **Scenario** **Evolution** Widget that is part of the Worldstate View BB is integrated into the demonstrator

The Java-based implementation of the Worldstate Widget that is part of the Worldstate View BB is integrated into the demonstrator.

Simulation Model Interaction View

Integration level: Integrated

A custom Java-based implementation of the **Simulation Model** Interaction **Widget** **BB** was used in the **demonstrator**.

UI Integration Platform BB

Integration level: Integrated

The cids Navigator is used as UI integration platform.

Earthquake casualty model

Integration level: Todo for V2

The earthquake casualty **model** is not wrapped yet thus **it** is not integrated.

Indicator Building Block

Integration level: Integrated

Calculation of Indicators is realised within the **Technology Demonstrator**.

Multi Criteria Analysis and Decision Support View

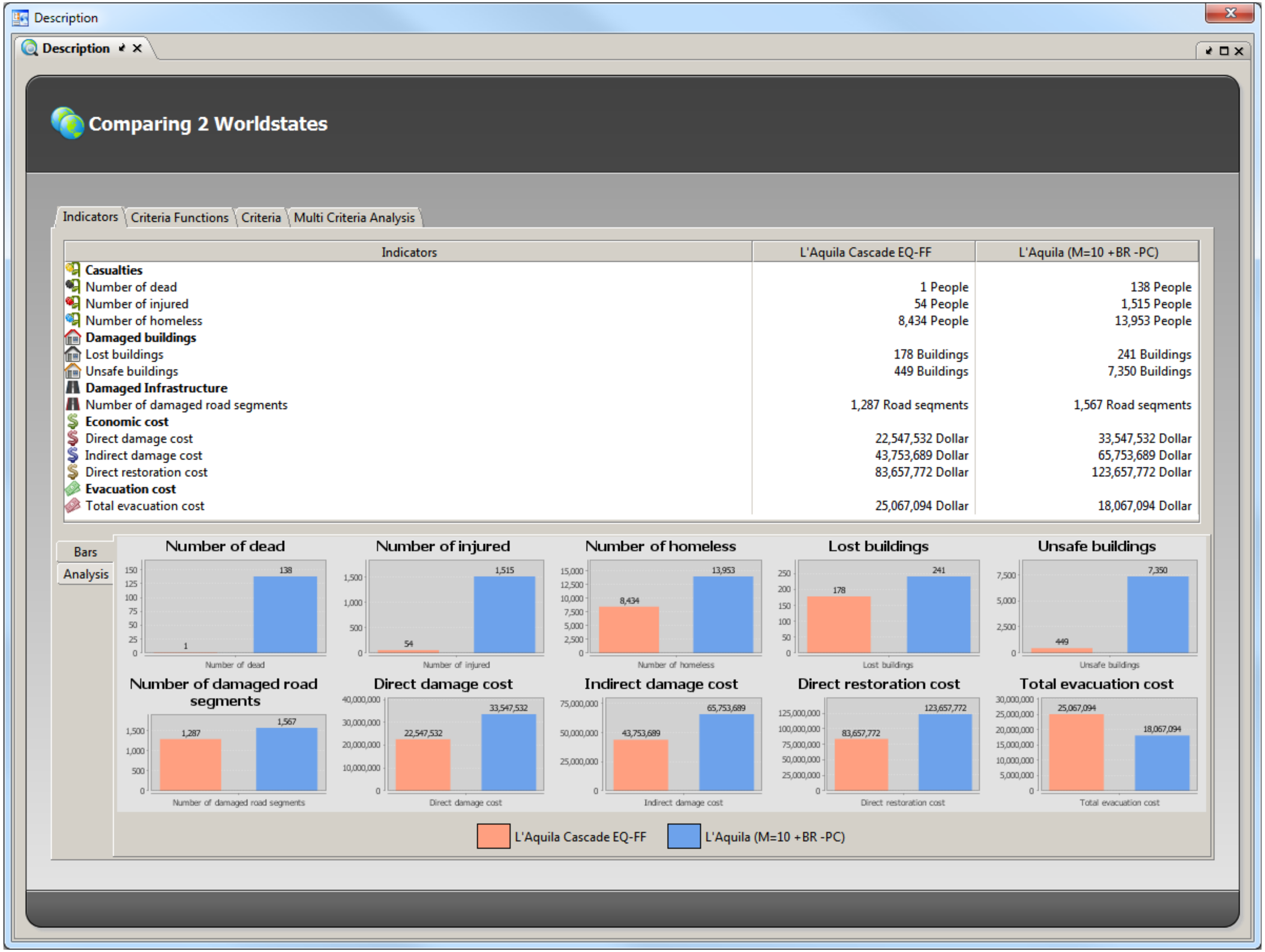
Integration level: Integrated

The Java implementation of the **OWA** based MCA and **DSS** is available.

Cascade Events Configuration and Interaction View

Integration level: Integrated

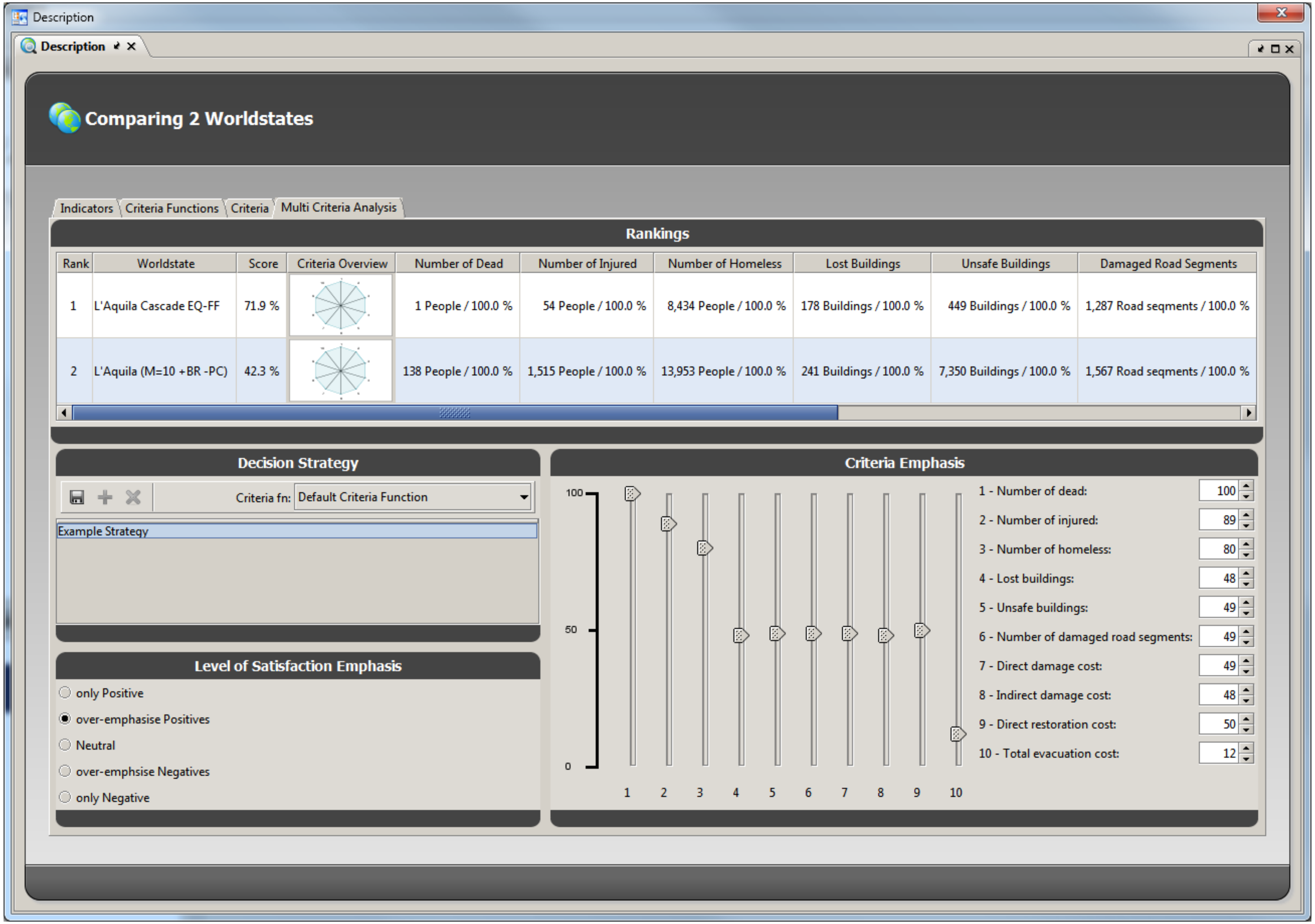
The java-based Implementation of the **Cascade Events** Configuration View is fully integrated in the **Technology Demonstrator**.



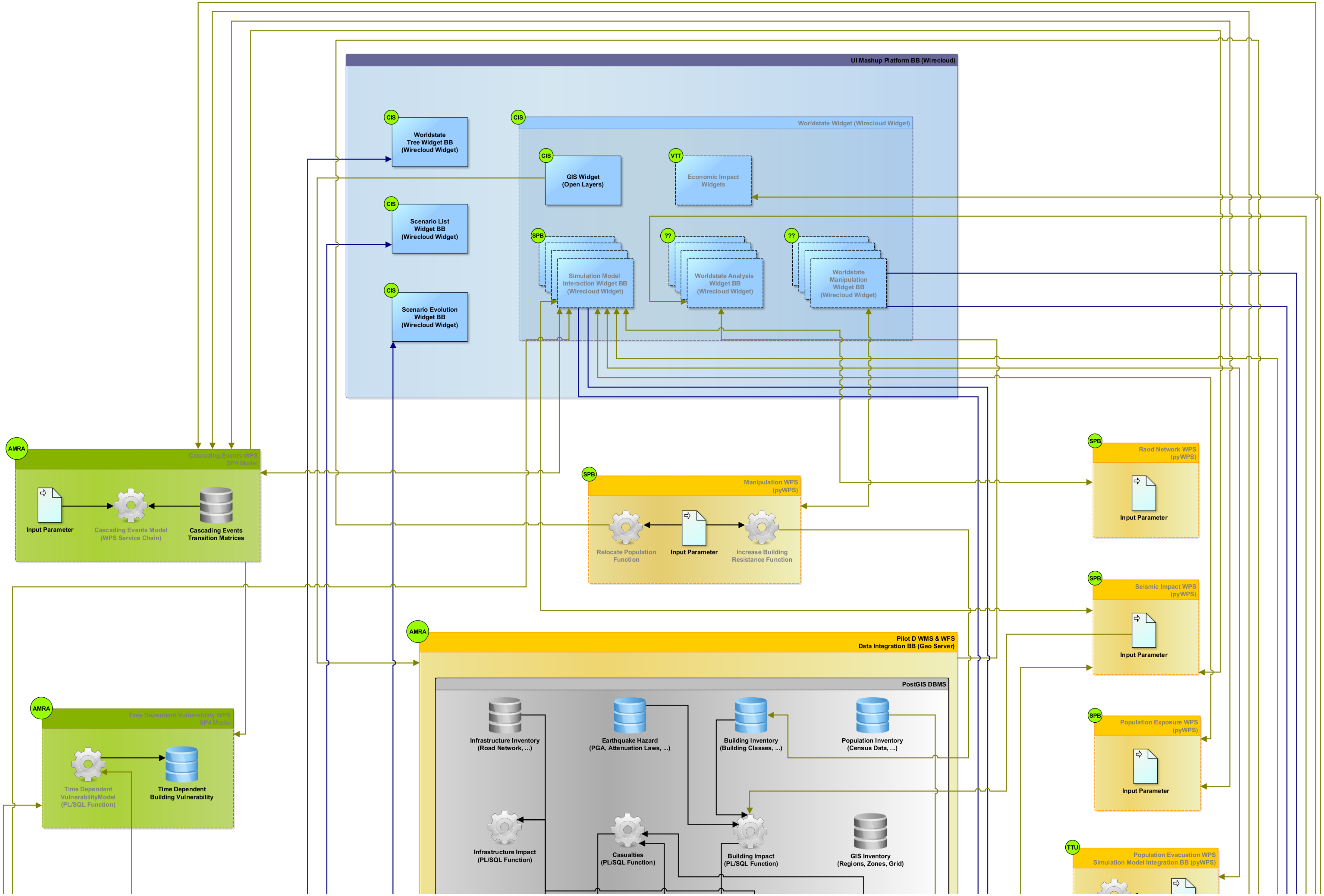
Documentation: Demo

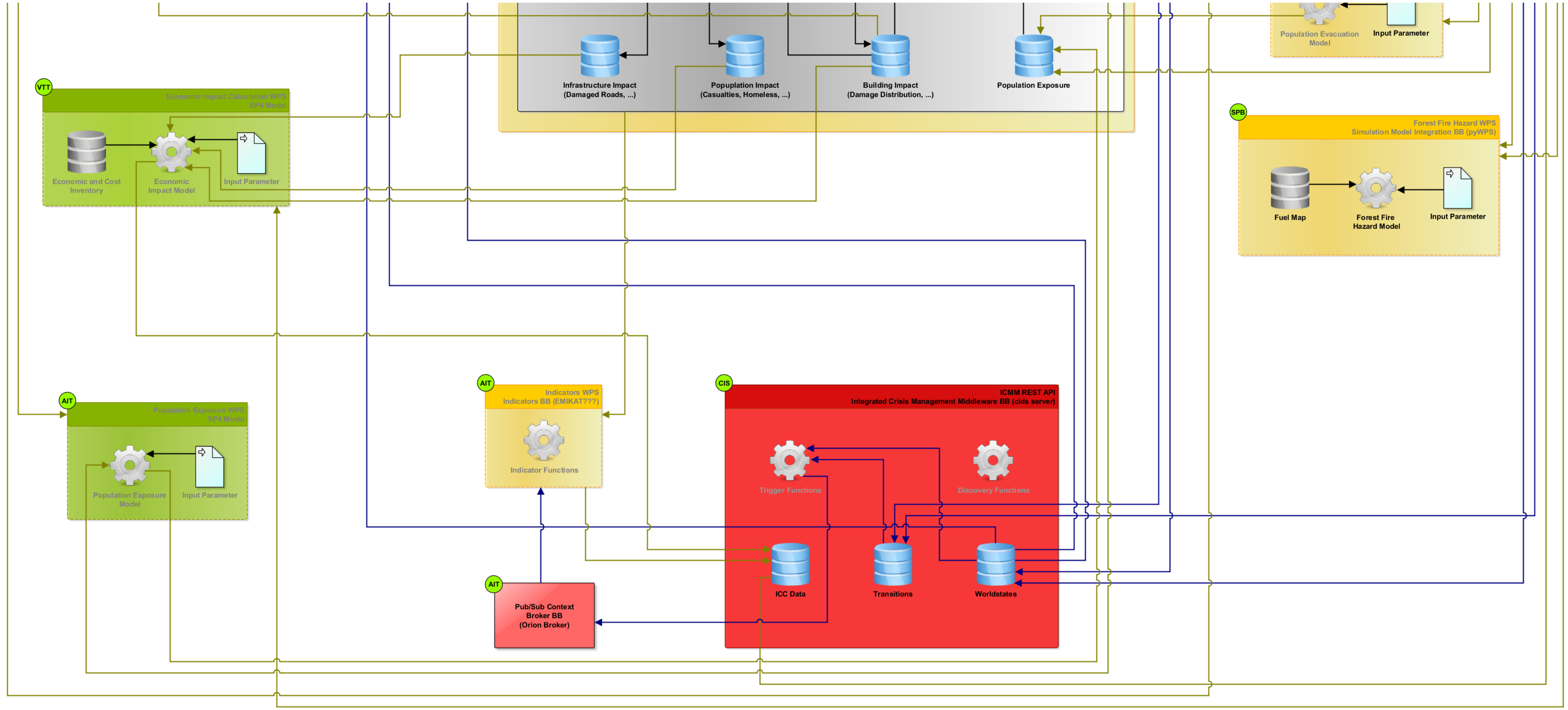
This is a **demonstrator** video of the general **mitigation use case**.

At the beginning we should have one root or initial **world state**. **It** should contain an earthquake (shake map), building classes, people **vulnerability**, road vulnerability, a proper name and a proper description what this world state describes. Then the user chooses to see what would happen if the EQ would happen like this so he chooses the Building **Impact**, People Impact and Road **Impact model** in that order. Additionally he provides a proper name and description for the result world state. The result is a world state that contains Building Impact, People Impact and Road Impact data in addition to the data that has been initially present. The user has a look at this data and additionally looks at the **ICC** data so that he gets the big picture of the overall EQ impact. Now he decides to do a mitigation, Building Resistance. He uses the initial world state again to access the **transition** wizard and chooses the Building Resistance mitigation option additionally the model he has



chosen before. He parameterizes this mitigation and again enters a proper name and description of the new world state. The result world state again contains Building Impact, People Impact, Road Impact and ICC data but with different values as the mitigation should have had an effect. The user decides to do more mitigation because the results are not satisfactory and triggers the **transition** wizard from the initial world state again. He choses the Building Resistance and the People congestion mitigation options in addition to the model from the last two transitions. The enters a meaningful name and description and again has a look at the now again different result data. To get a better idea how this mitigation would affect the impact of a stronger EQ the user chooses the last result world state and opens the transition wizard where he chooses the EQ **hazard model**, the building resistance mitigation, the people congestion mitigation, the building impact, the people impact and the road impact model. After the parameterization and another proper naming and description of the result the user can have a look at the impact and icc data again. Ultimately he decides to get a big picture of the things he did so he compares all four world states that he produced.


UI Mashup Platform BB (Wirecloud)
CIS
Worldstate Tree Widget BB (Wirecloud Widget)
CIS
Scenario List Widget BB (Wirecloud Widget)
CIS
Scenario Evolution Widget BB (Wirecloud Widget)
CIS
Worldstate Widget (Wirecloud Widget)
CIS
GIS Widget (Open Layers)
VTT
Economic Impact Widgets
SPB
Simulation Model Interaction Widget BB (Wirecloud Widget)
??
Worldstate Analysis Widget BB (Wirecloud Widget)
??
Worldstate Manipulation Widget BB (Wirecloud Widget)
AMRA
Cascading Events WPS SP4 Model
Input Parameter
Cascading Events Model (WPS Service Chain)
Cascading Events Transition Matrices
SPB
Manipulation WPS (pyWPS)
Relocate Population Function
Input Parameter
Increase Building Resistance Function
SPB
Raod Network WPS (pyWPS)
Input Parameter
SPB
Seismic Impact WPS (pyWPS)
Input Parameter
SPB
Population Exposure WPS (pyWPS)
Input Parameter
TTU
Population Evacuation WPS Simulation Model Integration BB (pyWPS)
AMRA
Time Dependent Vulnerability WPS SP4 Model
Time Dependent VulnerabilityModel (PL/SQL Function)
Time Dependent Building Vulnerability
AMRA
Pilot D WMS & WFS Data Integration BB (Geo Server)
PostGIS DBMS
Infrastructure Inventory (Road Network, ...)
Earthquake Hazard (PGA, Attenuation Laws, ...)
Building Inventory (Building Classes, ...)
Population Inventory (Census Data, ...)
Infrastructure Impact (PL/SQL Function)
Casualties (PL/SQL Function)
Building Impact (PL/SQL Function)
GIS Inventory (Regions, Zones, Grid)



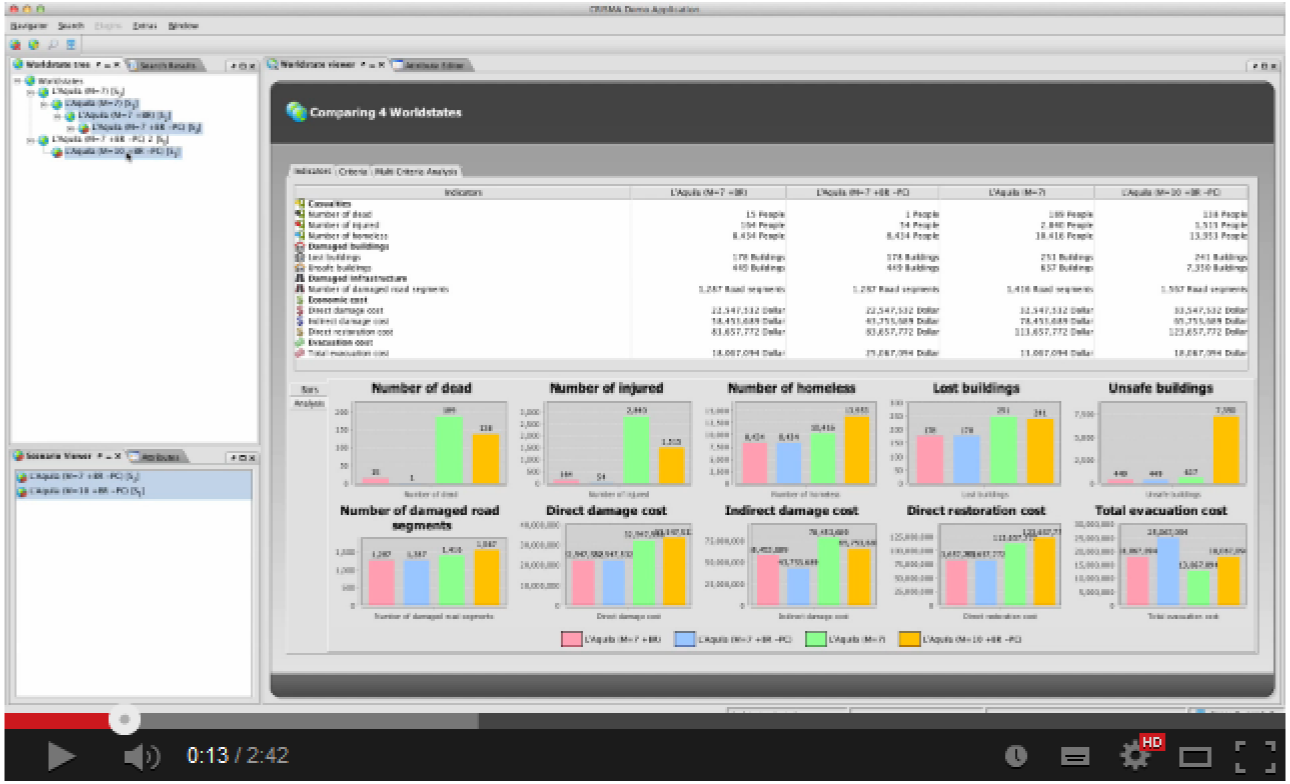
Demonstrator Video for Earthquake Mitigation
https://www.youtube.com/watch

Demo

This is a **demonstrator** video of the **Scenario** Analysis and Comparison and Multi **Criteria** Analysis and Decision Support implentation. **It** contains no sound but subtitles which should be activated by default but can simply be switched on via the caption control of the youtube player. The video is available in up to FullHD resolution.

- Demonstrator Video for Scenario Analysis and DSS
  https://www.youtube.com/watch

Downloads:



Binaries are located in the **cismet** repository.

- cismet repo
  https://repo.cismet.de
  All source code is available on github.

- GitHub repo (overall crisma custom)
  https://github.com/crismaproject/cids-custom-crisma
- GitHub repo (pilot D custom)
  https://github.com/crismaproject/cids-custom-crisma-pilot-d

Team:

Martin Scholl

Development roles: development

Sascha Schlobinski

Development roles: coordinaton

Stefano Nardone

Development roles: modelling, requirements

Sergio Guarino

Development roles: modelling, requirements

Maria Polese

Development roles: modelling, requirements

Data:

Infrastructure Inventory

Electrical power supply towers, Road Netowk, etc.

Building Inventory

Census data, People distribution and **vulnerability** classes

Infrastructure Impact

Output of the respective **impact model**: damaged roads, etc.

Popuplation Impact
Output of the respective **impact model**: Casualties, Homeless, ...

Building Impact
Output of the respective **impact model**: Damage Distribution, ...

Population Exposure
Output of the respective **exposure** **model**.

Integration of Wirecloud into native applications
Authors: **Manuel Warum**

This **technology demonstrator** is a very small and tiny **application** which examines the interactions with a Wirecloud Mashup Application when used inside a native C/C++ client.

The idea is to have a **widget** running the WireCloud Mashup and capturing the events generated by the Mashup inside the C/C++ client. The sample application "John Doe" is a Flickr-Viewer based on a keyword search. This mashup consists of 3 widgets: a keyword search, a flickr image selector, and a media viewer for the flickr images.

Each of these widgets emit some events on user action. These events are captured inside the C/C++ application as **JSON** objects for further analysis or processing by native components.

Implementation:
This **technology demonstrator** is a very small and tiny **application** which examines the interactions with a Wirecloud Mashup Application when used inside a native C/C++ client.

The idea is to have a **widget** running the WireCloud Mashup and capturing the events generated by the Mashup inside the C/C++ client. This is achieved by wrapping the actual Wirecloud application inside a Qt application running a web browser. In addition to acting as a web browser, **it** can perform automatic logins and captures events that are sent between widgets and operators of a mashup application.

The sample applcation "John Doe" is a Flickr-Viewer based on a keyword search.

This mashup consists of 3 widgets:

- a keyword search
- a flickr image selector
- a media viewer for the flickr images

Each of these widgets emit some events on user action. These events are captured inside the C/C++ application as **JSON** objects for further analysis or processing by native components.

Constituents:

UI Mashup Platform

Integration level: Optional

This **technology demonstrator** will require you to have a valid Wirecloud account and an already operational mashup.

The integration and customization needs to be done by a programmer. As the provided **application** is only a proof-of-**concept** demonstrator, **it** is only intended to show that it works and thus can at best only serve as a starting point or guide for actual implementations of native clients. Proficiency in C++ and the Qt **framework** is required.

Documentation:

Developer's Guide

The document attached below explains how to integrate the WireCloud Mashup Widgets into already existing legacy native client written in C/C++. **It**'s targeting the developers, but may also be of interest for technically inclined managers trying to figure out how to make the best out of WireCloud and existing desktop applications. Some additional information on the build requirements is described in FAQ below.

Similar principle could also be used to embed WireCloud prefabricated Mashups in web based applications. In fact this should be even easier to do, since it's just embedding HTML5 in HTML5, but we did not **test** it yet.

Downloads:

- https://crisma-cat.ait.ac.at/system/files/uid_2/Embedding%20mashups%20in%20desktop%20applicaitons%20HOWTO.docx

FAQ

Unix/Linux:

In order to build the native simple BackMeUp client **GUI** you have to install

- Qt4 **developer** libraries
- a recent C++ boost developer pack.

For a Debian **system** this should be resolved by doing

apt-get install build-essential g++ gcc cmake libqt4-dev libboost-all-dev doxygen

Now unpack the source tarball:

$ tar -xvzf wirecloud-webkit-0.2.tar.gz

Windows:

On Windows you should get this software installed:

- Visual C++ Express 2012
- Qt4 (at least V4.6)
- Doxygen
- CMake (at least V2.8)
- Boost (at least V1.49)

or alternatively

- Cygwin 1.7
- gcc (with g++ at least 4.6.3)
- Qt 4 (at least 4.6)
- Doxygen
- CMake (at least V2.8)
- Boost (at least V1.49)

How you can install this software and from where to obtain **it** is left to the reader. We strongly recommend to obtain the software as source and build it on the machine. Otherwise you'll end up with conflicting compilation references.

Demo

This video demonstrates the capabilities of the tech demo.

Downloads:

The client's source code is available on GitHub: **https://github.com/crismaproject/wirecloud-webkit-demo**

Team:

Data:

Reference Application for Exercise Support

Authors: Sven Wirth

This is the **Reference Application** for **Exercise** Support. **It** represents a tool capable of supporting relief organisations in the execution and analysis (esp. debriefing) of field exercises. An IT-based gathering and exploitation of relevant data from the exercise and data visualisations for the exercise debriefing are provided to users of the application. A reasonable presentation of the exercise results allows to identify deficiencies and thus to identify measures for improvement. After the tool was used for gathering relevant data before and during the exercise, it provides essential insights in form of a debriefing view, which comprises not only a general overview on the exercise results, but also more detailed information on essential aspects related to the medical care of patients and the organisation of the operation (Bracker et al., 2014). The usability of the tool was evaluated within two practical exercises organized by the German and Bavarian Red Cross.

System Description

The goal of exercises in the field of **emergency** **response** is to maintain and improve the capabilities of first responders in order to assure a rapid and reliable handling of an emergency **situation**. The proposed tool is to support this goal by summarizing and aggregating data gathered during the exercise in a way that allows assessing the outcome of an exercise, identifying shortcomings and deriving measures for improvement. In detail, the system supports the following activities:

a) Prior to the exercise, storing of main data related to the response **scenario**, as location of the **incident**, participating actors and the injury patterns which are to be shown by the actors playing the injured

b) During the exercise, capturing of data which constitutes the basis for exercise analysis and assessment

c) After the exercise, instantaneous processing of the captured data and presentation of results for analysis and assessment, complemented by the possibility to compare the just finished exercise with older ones

The following sections describe in more detail how the system is supporting these different phases. The description is focussing on the user perspective, i.e. on data input/output and user interfaces. Due to space limitations, we limit our description to activities mentioned in b) and c).

Data input during the exercise

The data which will be gathered during the training on the field will, in the end, reflect the performance of the involved first responders, as their activities determine the success of a rescue operation. The most important parameters should illustrate the first responders' activities including triage, treatment and **evacuation**. Consequently, the following data can be logged: a) Communication events between first responder and **headquarter** - Communication of alerts (type of **alert**, time stamp, comments if adequate) - Request of **resource** vehicles for evacuation (time stamp, number and type of vehicle requested) This data is captured by a so called data steward, who is a neutral observer of the exercise.

b) Type of care measures The actual care measures (see Figure 1) done by the first responder are indicated by a Yes/No flag. Typically, these flags are set by actors playing the injured. Depending on the setup of the data capturing process, they note the application of care measures with paper and pencil which are, at the end of the exercise, transferred into the system. At a later stage of development, mobile devices may be used to capture this data. In addition, the result of the triage respectively pre-triage process is captured in a separate **widget**.

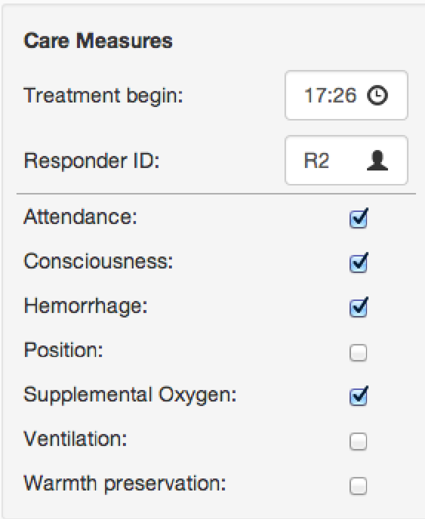


Figure 1: The care measures widget

c) Spatial planning activities Spatial planning includes the definition of the **hazard** zone, the vehicle depot and the patient depot ("tactical areas"). They are entered by dragging and dropping the appropriate symbol on a map, or alternatively, by entering a geographical position. In addition, a time stamp indicating when a tactical area was set up has to be indicated.

Provided Information After the Exercise

Once the input data is entered into the system, it can be processed and aggregated in order to produce the desired output data (Sautter et al 2014). Basically, output data is created by simple statistical means, leading to both single **parameter** values, and also time dependant readings. The latter are displayed in different graphical form (bar charts, pie charts, graphs).

The simple aggregation of the type of dataset (vehicle on site, tactical area, pre-triage, triage, evacuation) and respective timestamps results in the mission phase timeline.

On the basis of the log data about communication events, it is possible to inspect the flow of resources (vehicles) in a table, complemented by the points in time when they were requested.

The statistical processing of data about care measures gives, for every actor playing an injured person, an absolute number of applied care measures. This allows to identify if responders are able to decide which care measures to apply in which situation. The total number of classification errors occurred during the pre-triage and triage phase is another indication worth evaluating.

The data on spatial planning is used to create a chronological order the composition of the different tactical areas.

The resulting data from the exercise is presented on a debriefing view. This debriefing view consists of an overview page, which shows the essential information about the exercise. The widget element on the right hand side presents different aspects of the exercise to choose from (see Figure 2). The selection of one of these items gives a new screen visualizing key information about this particular exercise aspect (see Figure 3).

Using reference data from a previous exercise (run) is possible.

Figure 2: Debriefing view (overview)

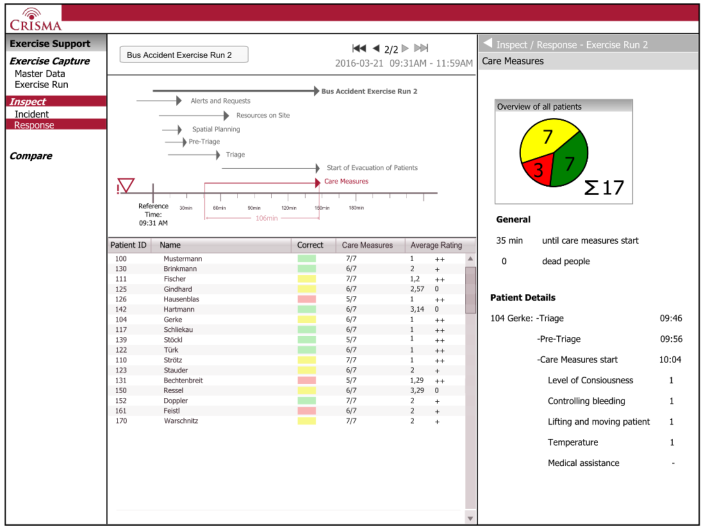


Figure 3: Debriefing view (care measures view)

Among others, to perform such a comparison, a number of indicators were defined that provide a high-level view on a particular exercise. These indicators are:

- Time until last patient is transported to the hospital
- Time until red patients are away from the incident scene
- Ratio of medical responders per patient
- Ratio of medical responders per patient per interval
- Time until all patients are pretriaged
- Time until all patients are triaged
- Number of vehicles involved
- Number of application of basic measures on scene
- Patient assessment of basic measures
- Ratio of vehicle per patient

Deployment in Exercises

Prior to an exercise the incident scenario and exercise **concept** (Geuther, A., 2013) has **to be defined**. The concept includes a detailed description of the scenario and defines parameters like number of involved patients, distribution of injury patterns (Sefrin, P., 2014) number of available responders and rescue vehicles, as well as location specification (Sautter et al 2014). Once the concept is developed, the injury pattern can be assigned to the set of (fictive) patients who are mimed by actors. This includes information on type of injury, triage category and the patient’s identity. All this information is to be registered before the exercise.

In a second step the patients and rescuers are equipped with exercise-IDs, the patients additionally receive a care measure **evaluation** sheet where they have to fill in times of (pre-) triage, care measures and evacuation as well as the kind of performed care measures. Due to limited space around the damage zone, a so-called vehicle queue area and a treatment area may be established by the rescuers. For every exercise a data steward needs to be nominated, who is in charge of collecting the sheets and typing the given information into the application forms in the software.

When data collection has been completed, the application provides key performance indicators as numerical values as well as charts that depict further data of interest. The exercise leader then has all instruments available to analyse the performance of the exercise and to discuss the results in a debriefing session together with all involved responders. The leader is enabled to detect weak points in a structured way.

Proceeding this way the developed IT-tool is not changing the elaborated structure and process of field exercises but complementing them.

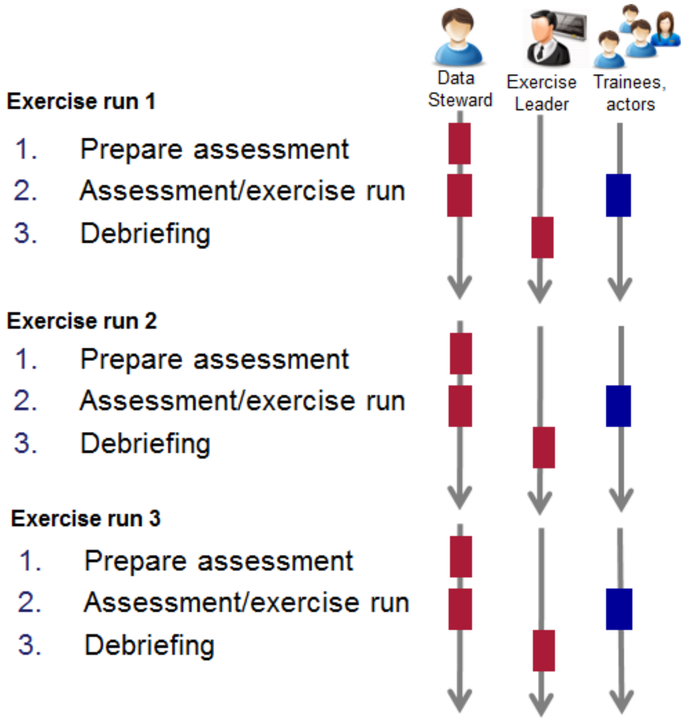


Figure 4: Typical sequence of an exercise appointment

References

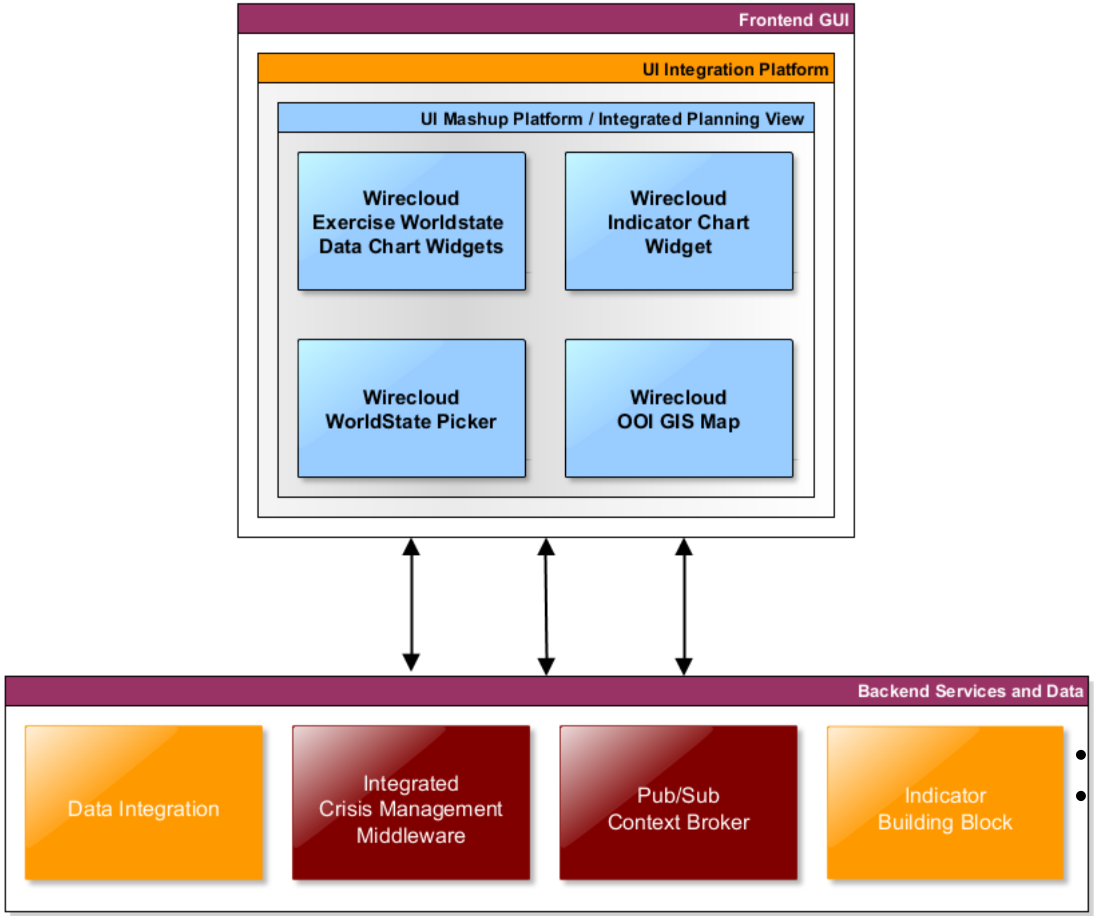
- Geuther, A. (2013): Von der Übung zum Training: Neues Trainingskonzept des BRK für Großschadensfälle, Im Einsatz, 2014; 20. Jahrgang, pp. 8-12.
- Sautter, J., Habermann, M., Frings, S., Schneider, F., Schneider, B., Bracker, H.: Übungsunterstützung für EInsatztrainings des Massenanfalls von Verletzten (MANV), Informatik2014 Conference, Stuttgart Germany, September 2014 (**http**://**www.crismaproject.eu/docs/Informatik2014_Exercise_support_v11.pdf (http://www.crismaproject.eu/docs/Informatik2014_Exercise_support_v11.pdf)**).
- Sefrin, P. (2014): Vorsichtung notwendig! Bericht von der nachfolge-Sichtungskonferenz 2013. Der Notarzt 2014; 30, pp. 85-87.
- Bracker, H. et al. (2014):An innovative approach for tool-based field Exercise-support to gain improved **preparedness** in emergency response,

(visited in March 2014)

Implementation:

Rough Architecture

A rough architecture of the **Exercise** Support **Use Case** **application** is depicted in Figure 1.



The user **interface** consists of various views realized as widgets and **widget** mashups based on the UI **Mashup Platform** **BB**. Special views are provided for defining the **scenario** setup during the preparation phase of an exercise, for capturing information like execution times, treatment deciscions, etc. during an exercise run, and for analysing and comparing exercise results in terms of CRISMA indicators. The central **building block** used behind the **GUI** frontend is the CRISMA **ICMM** where all data captured during the exercise is stored. The **Indicator** Building Block is used in this application to calculate the indicators which are required for this type of application. Based on the data stored in the ICMM during the exercise, the performance of exercise trainees at a certain point in time and the overall performance of an exercise run can be evaluated and visualized using the Indicators and Statistics view for the exercise use case.

To implement the system, a web based client-server architecture was chosen. This approach implicates a number of advantages, in particular with respect to scalability:

- The number of devices deployed for data capturing is scalable
- There is no restriction with respect to the type respectively mix of devices deployed for data capturing is (mobile devices, laptops, desktop PCs, …)

All user interfaces are designed and implemented using the Wirecloud platform (Application Mash-up Wirecloud, 2014), which is a mash-up platform enabling developers and users combing different widgets on a dashboard. The widgets themselves are coded in HTML5 and Angular **JS** (AnulgarJS, 2014). The described frontend communicates with services provided by the CRISMA **crisis management** **infrastructure**. The **incident** scenario data as well as the data captured during the exercise are stored in the **Integrated Crisis Management Middleware** (ICMM) **service**. More information about both can be found in (Dihè, P. et al., 2013).

The so far developed user interfaces are designed for deployment on a PC or laptop. **It** is planned to design user interfaces which can be used also on mobile devices. As widgets are modular components of the current web based user interfaces, it is possible to rewire them in order to obtain adequate user interfaces also for mobile devices.

References

- Application Mash-Up Wirecloud: **catalogue**.fi-ware.org/enablers/application-mashup-wirecloud
- AngularJS — Superheroic JavaScript MVW **Framework**: **https://angularjs.org/** (visited in June 2014).
- Dihé, P. et al. (2013): An architecture for integrated crisis management **simulation**, 20th International Congress on Modelling and Simulation (MODSIM2013), Adelaide, South Australia; 12/2013

Constituents:

Integrated Crisis Management Middleware BB

Integration level: Integrated

The **ICMM** is the central worldstate/**exercise** management component.

Data Integration

Integration level: Integrated

The **exercise** capture data is served by a **REST**-enabled **service** backend. Map data is served by Google (TM).

Publish Subscribe Context Broker BB

Integration level: Integrated

Used to propagate events in order to be able to calculate **indicator**/**criteria** data.

Indicator Building Block

Integration level: Integrated

Used to calculate **indicator**/**criteria** data

UI Integration Platform BB

Integration level: Todo for V1

The browser where the **application** runs in.

UI Mashup Platform

Integration level: Integrated

The platform for which the single widgets are built that constitute this **reference application**.

Integrated Planning View

Integration level: Integrated

The "**CRISMA** **PIlot** E Debriefing **Widget**" (Name in Wirecloud) aka. Configuration Component (name in **Catalogue**) realizes the core component of the Integrated planning view and is used in the **Exercise**-Support **Reference Application**.

Documentation:

Functional Description

See D51.1

- Final D51.1 deliverable (on VTT workspace) **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP51/03%20deliverables/D51.1_final/CRISMA_D51_1_final_2.doc**
- Excel file currently used by DRK Garmisch-Partenkirchen (German) **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP56/04%20Materials/Einsatztraining_anonymized.xls**
- Article describing the underlying organisational exercise concept (German) **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP56/04%20Materials/Geuther%20Proof%201_2013.pdf**
- Instructions for Patient actors (German Mimen, in German) **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP56/04%20Materials/Merkblatt%20f%C3%BCr%20Mimen%20Einsatztraining%20Vers%20%201%200.pdf**
- Analysis UML class charts (some of them also in D51.1, as MS Visio) **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP56/01%20Work%20in%20Progress/world_state_d51_1_v10.vsd**
- The most important class chart as PDF (during exercise run) **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP56/01%20Work%20in%20Progress/during_exercise_run.pdf**
- "Vision German Pilot" - including Analysis Status Quo (German, by Johannes) **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP56/01%20Work%20in%20Progress/deutscher_pilot_vision_v3.doc**

Implementation Plan

- Project Plan **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP56/01%20Work%20in%20Progress**

/CRISMA_WP56_Pilot_E_Requirements+Misc_020714.xls

Downloads:

The main github repository of this **application** is on the github and linked below.

It is currently **unclear** which source code will be hosted here, and what will be elsewhere but this kind of information will be explained in the main README file on the github.

- Exercise Support Use Case - Exercise capture view **http://wirecloud.ait.ac.at/pilotE/Exercise-Support (http://wirecloud.ait.ac.at/pilotE/Exercise-Support)**
- Pilot E capture widgets github directory **https://github.com/crismaproject/pilot-e-application/**
- Pilot E debriefing widgets github directory **https://github.com/crismaproject/pilot-e-debriefing-wirecloud-widget**

Team:

Johannes Sautter

Development roles: requirements, specifications

Coordinates the work related to concretization of the user requirenments, specifications and planning of the technical activities related to the **pilot**.

designs the UI, mockups, usability engineering, requirements engineer

Denis Havlik

Development roles: architecture, coordinaton, mediation, specifications

Mediation between users and developers with various backgrounds. Helps the development team to find a right balance between exploring the scientifically interesting "universe of possibilities" and adopting the pragmatic solutions which can be realized within the project scope.

Friederike Schneider

Development roles: requirements, specifications, validation

Owner of the **pilot** E, project manager, "cutomer"/end-user-representation

Maria Egly

Development roles: architecture, development

Responsible for integration and provides user **interface** integration platform (Wirecloud) and manages the integration testing, development of some widgets

Frank Jonat

Development roles: architecture, development

User **interface** implementation of the debriefing view. user interface **developer**

Uwe Kippnich

Development roles: validation

DRK key user

Martin Scholl

Development roles: coordinaton, development

technical coordinator and user **interface** **developer**

Data:

Structured ICMM-Data on Measures and times

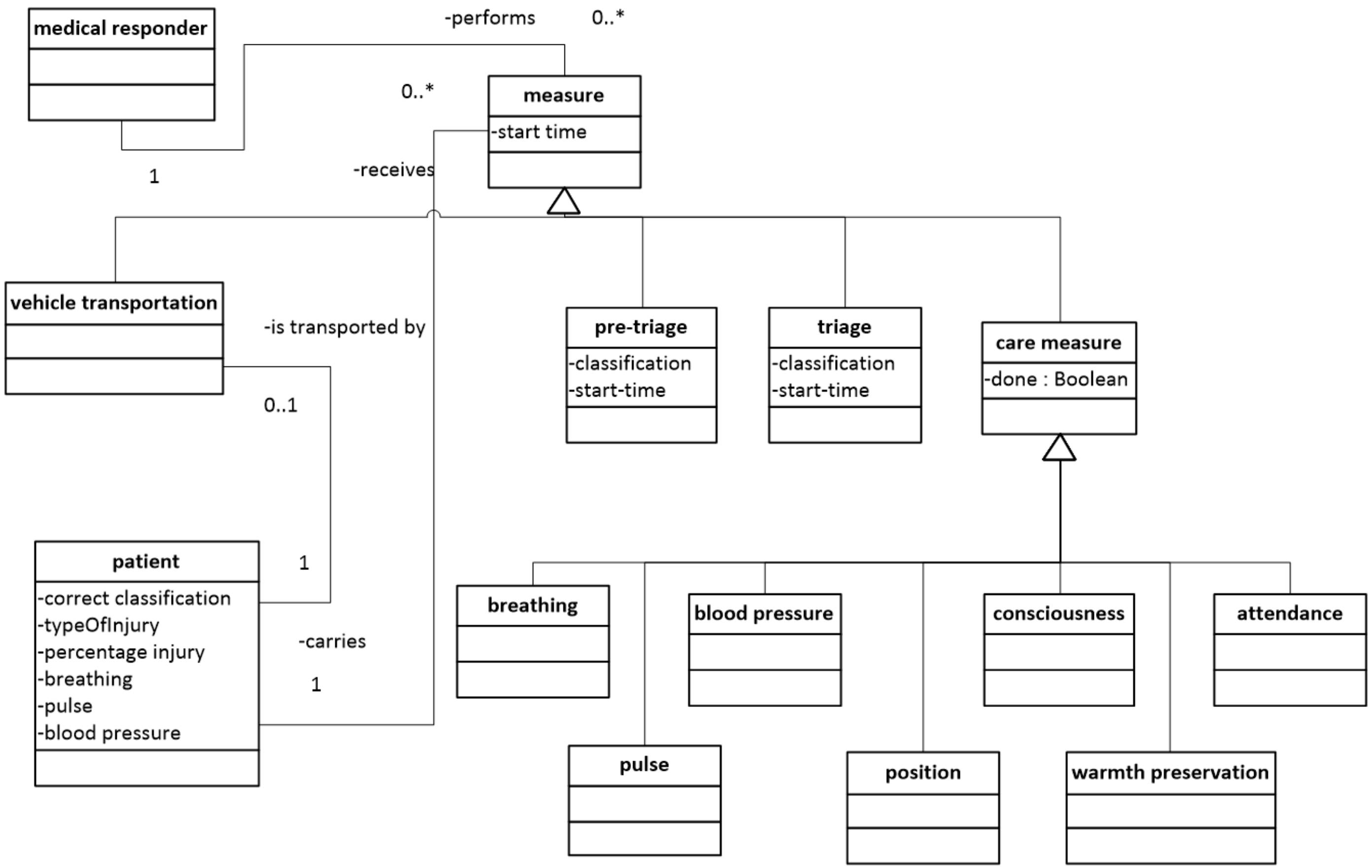
After the beginning of the **exercise** run the exercise data steward captures the locations of all tactical areas, the time of their creation and also the arrival of all rescue vehicles at the **incident** scene (as a rule from the exercise documentation). As soon as the patient mimes arrive at the virtual hospital the data from the **evaluation** sheet (care measures, time of **evacuation** and „yes or no" rating of care measures) will be added in the **system**. If necessary the patient needs to be treated locally and carried into a rescue vehicle.

Figure 1: **UML** class chart of the underlying data schema

There are four different types of measures a medical responder can perform on a patient. These are pre-triage, triage, vehicle transportation and care measures. Starting time needs to be listed for all of them. In case of (pre-)triage the classifications of the medical responders have to be saved additionally (cp. Figure 1). Concrete care measures are breathing, pulse, blood pressure, warmth preservation, consciousness, attendance and position. The patient's various attributes are: name, correct classification and type of injury (pulse, blood pressure, and breathing). Furthermore he or she can receive all measures.

Reference Application for Resource Planning

This is the **Reference Application** for **Resource** Planning. The main objective of this Reference Application is to provide what and how much resources are needed to provide the best **response** in case of a mass casualty **incident**. **It** represents local-specific constraints that exist, such as the density of hospitals nearby, what their surgery capacities are like, the distance between the local rescue bases and the incident scene as well as the level of **preparedness** of first responder.



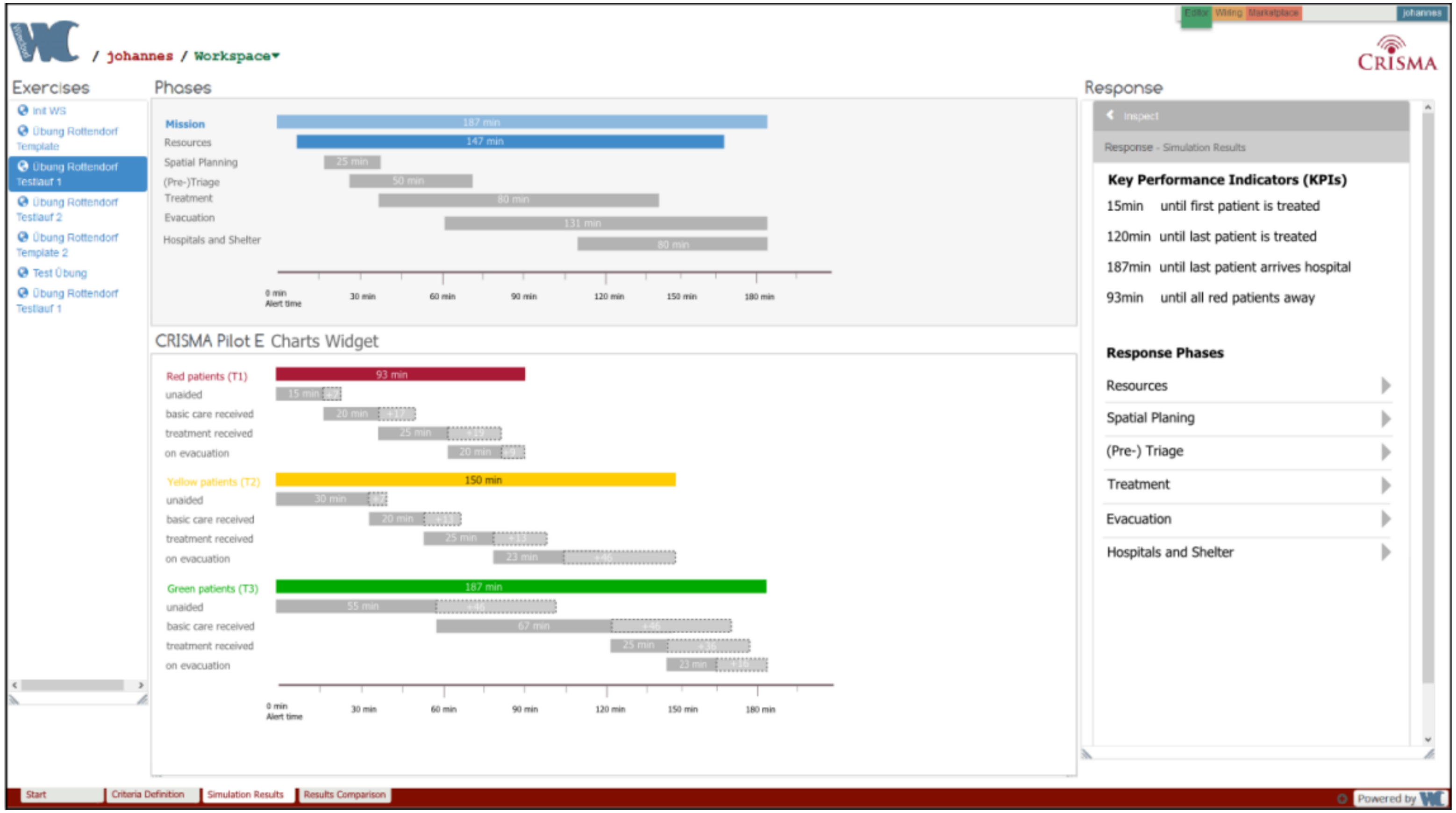
Authors: **Johannes Sautter**
Friederike Schneider
Maria Egly

A mass casualty incident (**MCI**) is an **emergency** with a large number of injured or affected persons that cannot be managed with regular emergency medical services (DIN13050 2009). In creating an efficient plan that includes how much staff would be needed and the deployment of resources for MCIs shows to be challenging due to the large amount of resources required. From practitioner's point of view, it is easy to see shortcomings regarding the awareness of the actual availability of resources during a **crisis situation**. In addition, the relation between the actual needed and available resources is often left **unclear**. Also, the medical incident commanders that are involved in ambulance organisations desire resource planning across existing administrative boundaries that allows elaborating and establishing flexible concepts for a suitable use of human and technical resources to develop realistic local concepts in cooperation with neighbouring districts (German Red Cross 2012).

Figure 1: Simulation results view showing the mission phases and a patient status overview

Figure 2: Simulation results view showing the mission phases and a resources line chart

Figure 1 and Figure 2 show the simulation results view analysing a single mass casualty mission performed by the **simulation model** using particular parameters on resources, tactical areas and **alert** times. The **Reference Scenario** and the related Simulation Cases implemented by this application are those of the **Pilot** E - mass

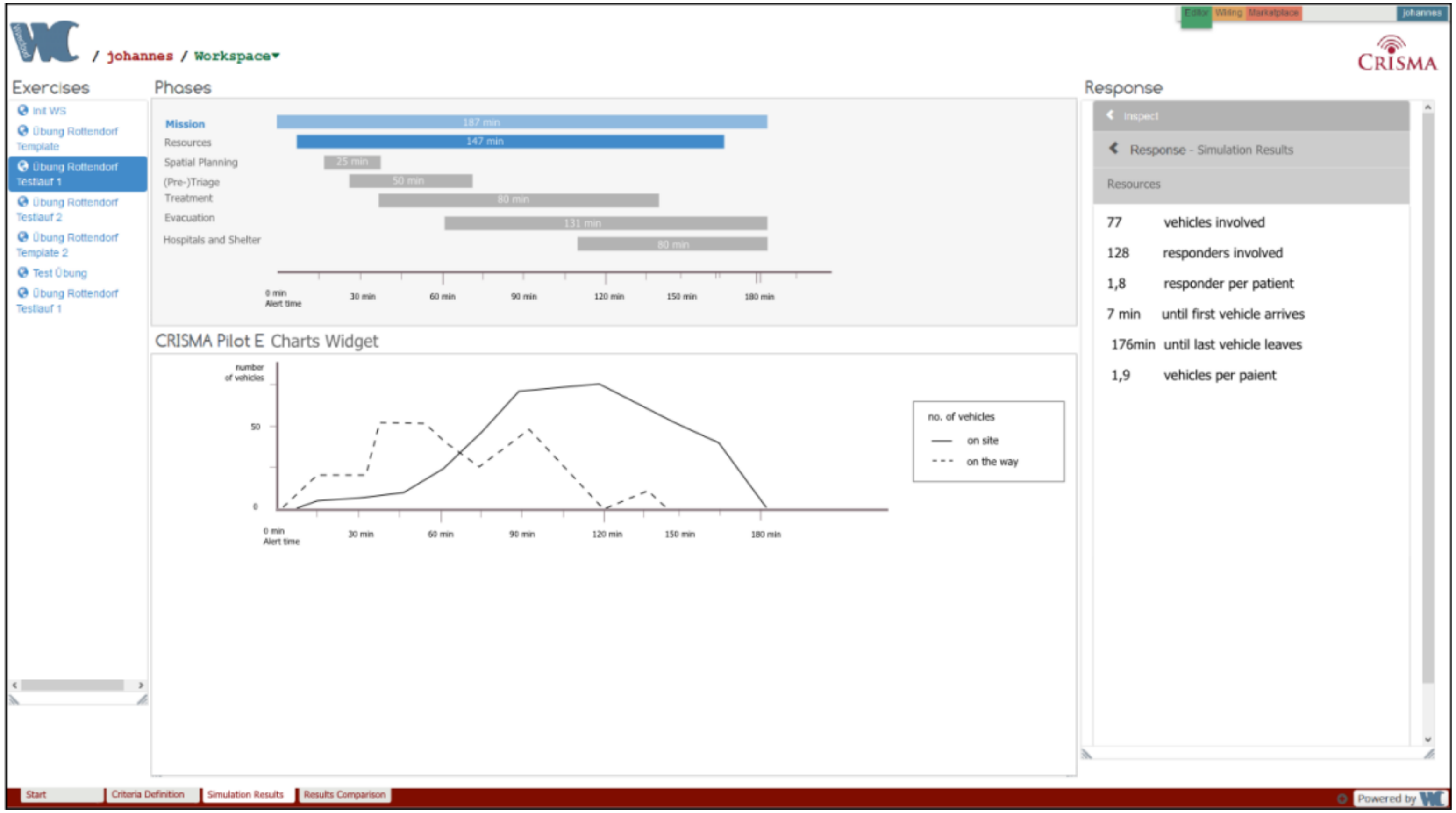


casualty incident (MCI) – (Germany). Thus, the integrated Reference Application for resource planning for MCI missions serves as basis for the development of a Pilot E **CRISMA Application** for German Red Cross mission commanders.

The Reference Application for the MCI demonstrates the capabilities of the **CRISMA Framework** regarding

- Response – supporting decision makers by providing alternative management policies
- Quantitative assessment of **impact** scenarios (simulated by assumption of alternative strategies)
- Assessment of choices (in planning and/ or emergency) on the basis of scenario

The chief emergency physician (CEP) and the medical incident commander (MIC) are mission commanders and target group at the German Red Cross. When applying the application to other countries or organisations relevant roles may differ . These two roles or other team leaders of emergency medical services organisations should elaborate local-specific strategies detached of daily operational activities in a quiet environment on a desktop computer. The user can define different response strategies (e.g. use an advanced medical post and a vehicle queue) for this specific scenario, determine the location of the advanced medical post and/or the vehicle queue as well as set the main key performance indicators (**KPI**) he/she wants to investigate by simulating the different response strategies. The result calculation can start with the first resource deployment by the user that can either be manual or automatic by choosing a response keyword of the **order of alarm and action** (OAA). Each selected response strategy is represented by a navigable timeline that visualizes the end of the response based on the previously selected main **evaluation indicator** (e.g. all patients removed from scene). The parallel display of the timelines and the focus on one indicator provides a quick overview of the effectiveness of the different strategies without an information overload. Nevertheless, results of other evaluation indicators for each strategy can be displayed separately by clicking on the related timeline (Sautter et al, 2014).



We understand **resource management** as the interactions between actors and how they affect each other including the steering and governance of crisis response actions and resources such as vehicles, personnel and equipment. Based on that, we define resource planning as the elaboration of local-specific tactical strategies in the preparedness phase. We suggest an organisational approach for resource planning by local-specific preparedness strategies. Further, we describe a simulation application that supports medical response managers in elaborating flexible and local-specific strategies for mass casualty missions. In order to generate valid results, data from Red-Cross real-life exercises are planned to be utilized for **validation** and calibration. The application is currently under development and will be evaluated within the CRISMA project.

Preparedness in local communities needs to take into account regional conditions and critical infrastructures as e.g. concert halls or train stations. Risks of potential incidents can be mitigated by choosing particular infrastructures and conditions as part of crisis scenarios for targeted trainings and exercises. In the following we suggest a further measure for a targeted improvement of local **resilience**. The organisational status quo for mass casualty missions in Germany shall be enhanced by additional functionalities of the chief emergency physician (CEP) and the medical incident commander (MIC) during the preparedness planning phase. In addition to currently performed trainings and other preparedness activities they can elaborate local-specific strategies and recommendations especially for mass casualty missions. Such strategies for tactical decisions are for instance:

- Carry the most critical 10 patients to hospitals immediately, the 11th needs to be treated on-site first
- Stabilize patients as much as possible before transportation, as ways to hospitals tend to be long

- Start pre-triage process without tactical areas in order to not bind leading personnel for each tactical area due to few immediate available personnel

As a prerequisite they need to assess regional infrastructures and perform calculations with their local resource constraints. Such regional constraints are for instance the number of beds in nearby hospitals (within 50 kilometres) and their surgery-capacities for badly injured, the number of available emergency medical services (EMS) resources, the distance of the corresponding rescue bases to the potential incident scene and the distance from the hospitals to the incident scene.

Modelling and simulation systems can provide an essential contribution to the elaboration of such strategies as they can easily perform calculations on the basis of local constraints. Virtual mass casualty missions can be assessed by so called key performance indicators (KPI). We understand key performance indicators as characteristics and relevant aspects of a mass casualty scenario. Examples are: time until all patients are pre-triaged/triaged, time until red-triaged patients are away from the incident scene, time until care measures start or time until the last patient is transported to the hospital. The consideration of these local KPIs in simulation runs allows a region-specific replication and an incorporation of the lessons learned as preparedness measures into the daily work. Thus it is possible to develop better tactical schemas that lead to improved operations, training and resource planning.

Simulation cases of the resource planning application can be summarized as:

1. Identify which tactical areas to use or not to use
2. Identify suitable location and resources for each tactical area
3. Identify other mission tactics for dedicated local districts like e.g. number of resource to be requested

References

- Sautter et al 2014: Local-specific resource planning for mass casualty incidents, **http**://
, 2014.
- DIN 13050:2009-02 (Rettungswesen) 2009:, DIN Deutsches Institut für Normung e.V. (German Institute for Standardization DIN, via **www.bbk.bund.de/DE/Servicefunktionen/Glossar/ (http://www.bbk.bund.de/DE/Servicefunktionen/Glossar/)**), Beuth-Verlag Berlin, 2009.
- German Red Cross 2012: . "Zukünftige Schwerpunkte und Forschungsbedarfe im Deutschen Roten Kreuz" ("Prospective emphases and needs for research in the German Red Cross"), Newsletter – Neues aus der Sicherheitsforschung, 2012.

Implementation:

The **Resource** planning **application** is conceptually based on the **Reference Application** for **Resource Management** Training. However, the interactive view enabling the user to manipulate during a training session the **model** behaviour ("Dispatch and Monitor View") is replaced by a "Decision Model" which undertakes the active part "during **incident**" from the user in the training case. The user interfaces relevant for the planning application remain the "**Scenario** Setup View". For Post-Incident Analysis the "**simulation results** view" reuses components of the "**exercise** debriefing view" from the Reference APplication for Execise-Support.

Further **CRISMA**-Widgets for comparing several simulation results are applied in the "Analysis and Decision Support View".

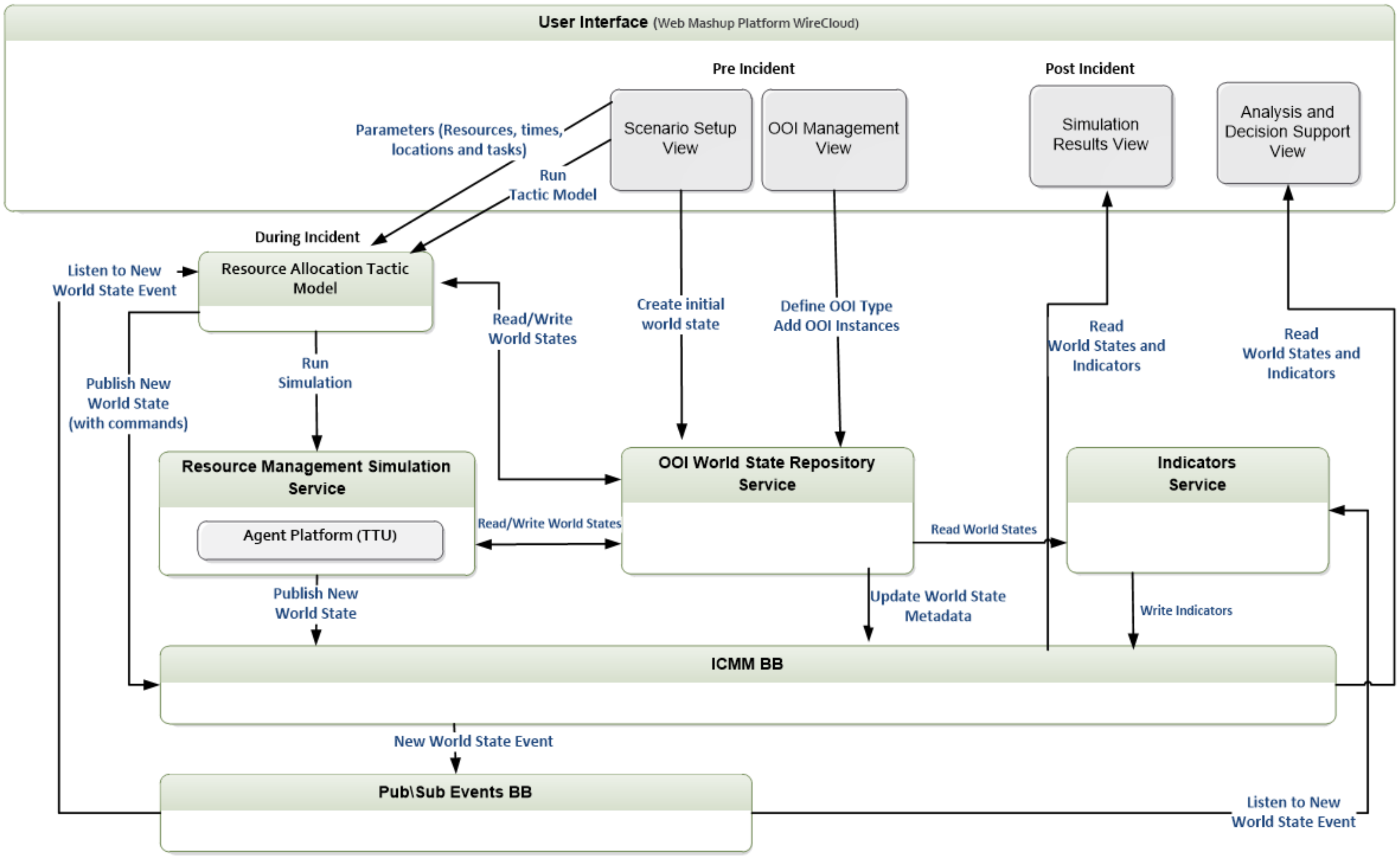


Figure 1: Implementation architecture of the resource planning application

Constituents:

Integrated Crisis Management Middleware BB

Integration level: Integrated

Core component of every **Reference Application**

Scenario Analysis and Comparison View

Integration level: Integrated

This **Reference Application** compares and analyses worldstates with the help of this view.

Multi Criteria Analysis and Decision Support View

Integration level: Integrated

This **Reference Application** integrates **DSS** with the help of this view. Further the **indicator** table from the MC & DSS View is used with in the **simulation results** view.

Agent Oriented Simulation Models

Integration level: Integrated

This **Reference Application** integrates **simulation** with the help of this view.

Integration level: Todo for V1
This **model** governs the general behaviour of the ambulances in Israeli **pilot** and is also . Modifications of this model is also used in the German pilot. Full descriptions, documentation etc. shall be provided by **TTU**.

Use in this **application**
This **simulation model** is realised using the TTUs agents platform and governs the movement of the OOIs of the "ambulance" type and their interaction with the patients. In this application, the ambulances are dispatched towards the **incident** area. The **Resource** Allocation Tactic Model assigns a **task** to each ambulance. As soon as an ambulance finishes its task, the Resource Planning Decision Model steps in again.

OOI World State Repository BB

Integration level: Todo for V1
<p>Central data repository for all OOIs used in this **application**. Currently: ambulances, citizens/patients and CDM areas. In order to simplify the **API** and decouple the **GUI** from the **model**(s), the **OOI** state is also used to encode the decisions. for instance, the "destination" **parameter** is interpreted by the model depending on the OOI type and context: empty ambulances drive to their destination ambulances with patients drive to patients destinations an area destination is assigned to patients after treatment.</p>

OOI Management View

Integration level: Todo for V1
This view is used to define the types of OOIs that are used in the training **scenario**. **It** is de-facto the management **interface** for the **OOI** **World State** Repository **BB** and normally used only by programmers and/or technically savvy administrators as a pre-requisite to set-up the training. In addition, it allows this view can be used to clean up the database from old results which aren't needed any more, as well as to set up the World State details which can not be manipulated using the more user-friendly Scenario Setup View **GUI**.

Indicator Building Block

Integration level: Integrated
The **Indicator** **Building Block** is used in this **application** to calculate all indicators needed for the ressource planning **simulation** analysis. Indicators are e.g. the duration of the triage or of the **evacuation** of all patients or the number of vehicles or responders per patient. The results of the indicators calculation are stored in the **ICMM** worldstates generated during a **simulation run**. They are visualized using the Multi **Criteria** Analysis and Decision Support View and the **Scenario** Analysis and Comparison View.

Publish Subscribe Context Broker BB

Integration level: Integrated

The Pub/Sub context broker is used by the **ICMM** to notify the **Indicator BB** that a new worldstate was created and thus triggering the indicator calculation. Furthermore, this **building block** is used by the **simulation** models platform in order to trigger new **world** **transition** calculation.

UI Integration Platform BB

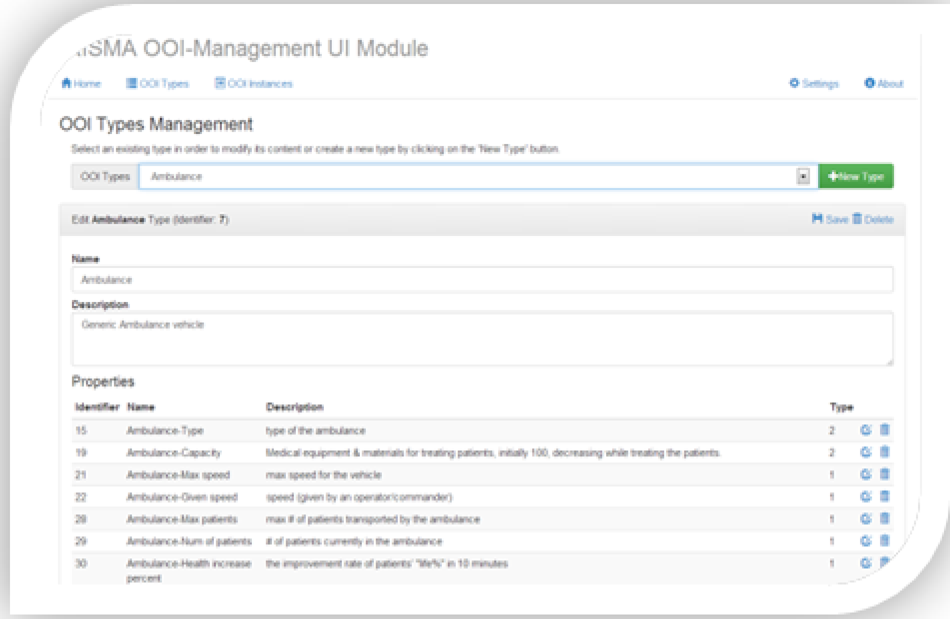
Integration level: Integrated

All user **interface** "views" defined in this **application** are realized as mashup applications running on top of the Wirecloud Application **Mashup Platform**, which is a reference implementation of the UI Integration Platform **BB**.

Resource Allocation Tactic Model

Integration level: Todo for V2

The Decision **Model** allows running a **simulation** with a predefined strategy setting and therefore is a main component of the **resource** planning **application**.



Documentation:

Functional Description

- Functional description and specification
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP56/01%20Work%20in%20Progress/V2%20and%20Harmonisation/Working%20Doc%20CRISMA_Pilot%20E_ressource%20planning_V05_DRK.doc

Implementation Plan

- Project Plan
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP56/01%20Work%20in%20Progress/CRISMA_Pilot_E_resource_planning_project_plan+Misc.xls

Downloads:

The **Application** is currently under development and will be integrated using the Wirecloud platform.

- Resource Planning Application
  https://crisma-pilote.ait.ac.at/pilotE/ResourcePlanning

Team:

Friederike Schneider

Development roles: requirements, specifications, validation

Owner of the **pilot** E, project manager, "cutomer"/end-user-representation

Martin Scholl

Development roles: development

Technical coordinator and user **interface** **developer**

Responsile for the **pilot** **architecture** and integration

develops "Excercise capture widgets" (pilot-specific adaptations of the "**OOI** Monitoring and Dispatching View")

Kalev Rannat

Development roles: modelling

Model **developer**

Support for the **OOI**-/ressource- related models; concrete ammount of work and division of work between **TTU** and CASS not yet clear.

Maria Egly

Development roles: development

User **interface** integrator

Development roles: **architecture**, development, installation, integration

provides user interface integration platform (Wirecloud) and manages the integration testing, development of some widgets

Johannes Sautter

Development roles: requirements, specifications

Coordinates the work related to concretization of the user requirenments, specifications and planning of the technical activities related to the **pilot**.

designs the UI, mockups, usability engineering, requirements engineer

Frank Jonat

Development roles: development

user **interface** **developer**

Oren Deri

Development roles: requirements

Coordination requirements and features for **resource management** training and planning

Tanel Tenso

Development roles: modelling, requirements

SP4-Steward and **model** requirements engineer

Data:

Reference Application for the Coastal Submersion Domain

Authors: **Mehdi Pierre Daou**
Marianne Grisel
Arnaud De Groof
Armonie Cossalter

This **Reference Application** is based on the requirements of a storm surge induced Coastal Submersion **use case**. The main field of application is **crisis preparedness** planning and **evaluation** of **mitigation** solutions. Thereby, different storm surge events and their impacts are analysed under different circumstances (e.g. summer or winter, day or night) and under different intensities of events (e.g. stronger winds, concomitance with high tidal level). Then, the effects of different mitigation solutions are evaluated, and, eventually, also several decisions at different times.

The Reference Application for the Coastal Submersion Domain demonstrates the capabilities of the **CRISMA Framework** regarding:

- **Simulation** of the submersion effects at a range of temporal and spatial scales
- Preparedness Planning
- Assessment of impacts depending on scenarios based on options for managing the risks
- Cost / benefit analysis

Supported **Simulation case**

1. **Define the event characterization**
   The user defines the characteristics of the storm (variation of mean sea level at the marine boundary, the atmospheric pressure, the wind conditions,…) and the status of the preventive **infrastructure** for floods protection.
2. **Simulate coastal submersion**

The user performs coastal submersion simulation. The results consist of dataset that are available for the experts and the decision makers (water level, velocities,… at different time for a specific area of interest).

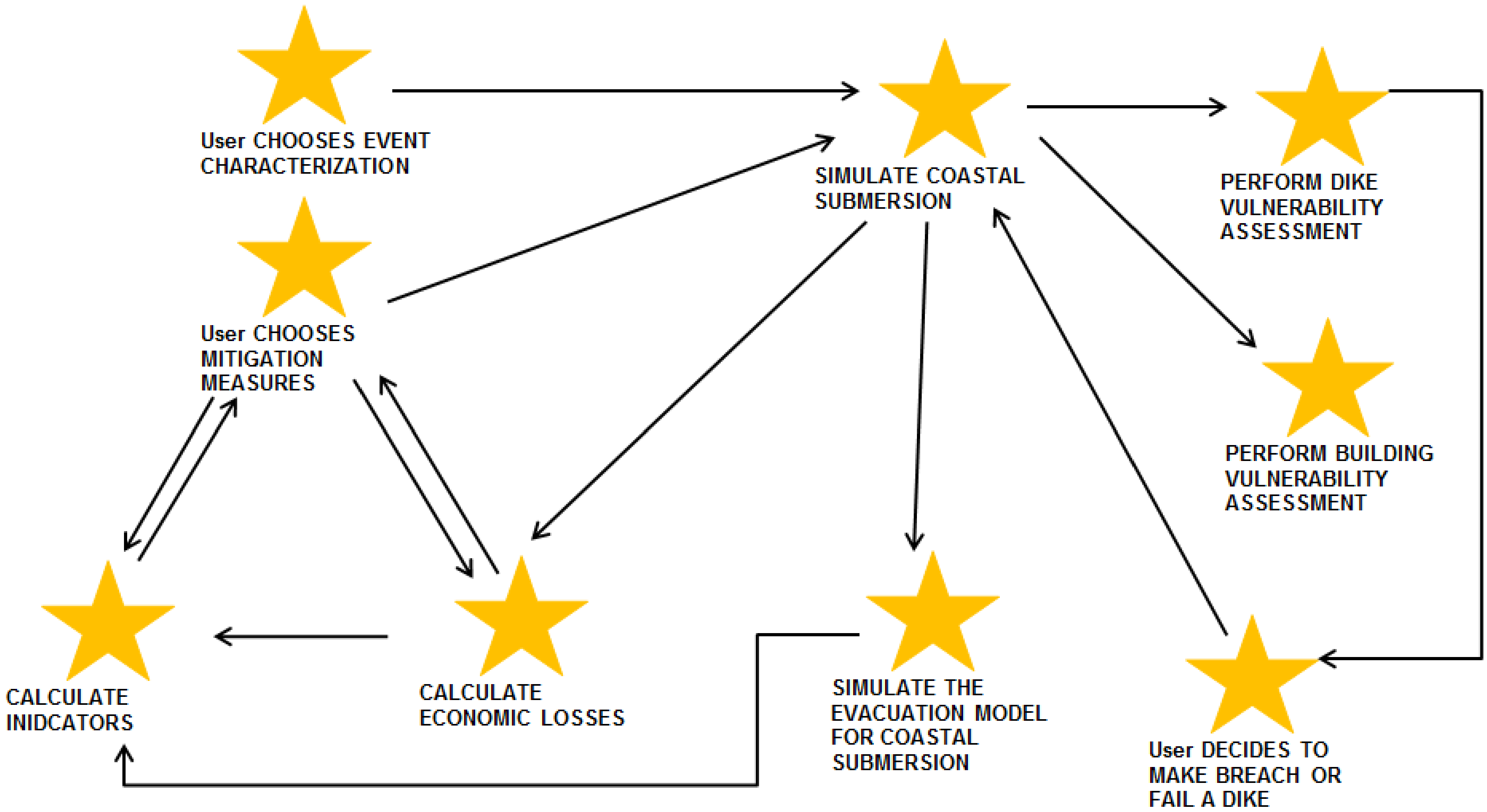
3. **Perform dike and building vulnerability assessment**
   Depending on the result of the coastal submersion simulation the user can assess the vulnerability of the dikes and the building.
4. **Modify the dikes resilience**
   The user can modify the **disaster** resistant of the dikes (introducing of failure or breach) aiming to run new simulations. The user can create different scenarios depending on the dike resilience.
5. **Simulate the evacuation Model for coastal submersion**
   The user can estimate the movement of the people, the number of safe people, people still evacuating, and dead people… by running the evacuation Model for coastal submersion.
6. **Calculate indicators**
   Calculate indicators for the coastal submersion.
7. **Calculate economic losses**
   Calculate economic losses of the coastal submersion.
8. **Calculate indicators for the coastal submersion**
   Calculate economic losses of the coastal submersion.
9. **Choice mitigation measures**
   The Decision makers can choose different options to manage the **risk** (reinforce dykes, adapt protections, adopt new **crisis management** plan,…)

Simulation case

The **reference scenario** for coastal submersion is used for long term planning.

The steps to use this application are described here-after:

1. The user visualises the area of interest on a map with different layers representing the data of the **World State** within the world description and the assets categories (for examples, the administrative boundaries, the buildings, the road network, etc...)
2. The user chooses the kind of **hazard** input and the event characterization. **It** can be a predefined event or the user can directly choose the characteristics of the event such as the date, the premium levels, the wind, and the mitigation measures (including no measure). These data are stored into the World State.
3. The user launches the hydrodynamic simulation at large scale (**pilot** B total area of interest). The integrated coastal submersion model runs at large scale and the results are stored within the World State in the hazard outputs category and a file of point set containing the maximum of each variable as **attribute** variables.
4. The user launches the time-dependant **vulnerability model** for buildings and dikes. The results are stored into the World State. (see the description of the



coastal submersion time-dependant vulnerability model for buildings and dikes)

5. The user visualises the results of the coastal submersion simulation at a large scale (water level, water velocity, duration of submersion) on a map with a layer for each variable. He can visualise the maximum of the variables or an animation to see the evolution of a variable in time. He visualises the results of the time-dependant vulnerability model. On a map, the user sees the dikes represented by polylines and the buildings represented by dots. The colour of the asset on the map is the one of the damage class with the highest probability. By clicking on the asset, the user can see all the probabilities to be in each damage class.
6. Depending on the results of the vulnerability model on dikes (probability to fail), the user can choose (or not) to simulate the same event at a local scale with

one or several dikes breaching or failing. Several local scale areas are predefined and the user chooses one of them by clicking on it. Once zoomed in, the user can choose to breach a dike or make it totally fail by clicking on the dike and choosing breach or fail.

7. The user launches the local scale simulation of the same event, considering the choices of the user on the dikes. The results are stored in the World State in the category hazard outputs and a file of point set containing the maximum of each variable as attribute variables.
8. The user can visualise the results of the local simulation (water level, water velocity, duration of submersion) on a map with a layer for each variable. He can visualise the maximum of the variables or an animation to see the evolution of a variable in time.
9. The user launches the evacuation model for coastal submersion of the chosen local area. The results are the movement of the people, the number of safe people, people still evacuating, and dead people, the closed roads, the collapsed buildings. These results are stored in the World State. The user visualises an animation of the evacuation and the submersion.
10. The indicators and economic losses are calculated with **ICC** functions and stored in the World State.
11. The user can visualise the indicators and the economic losses.
12. The user can run the **cascade effects** model.
13. The user can compare different scenarios.

Description of the coastal submersion time-dependent vulnerability model

The coastal submersion time-dependent vulnerability model is under the form of Damage Probability Matrices.

For each type of assets (buildings and dikes), vulnerability classes are defined as well as damages classes. For each vulnerability class, depending on the hazard intensity (here, water level in meters), the probability for the asset to be in the damages classes are given. (Figure below)

Medium status			
Water level above the dike	Damage level		
	No failure	Breach	Total Failure
<20 cm	99%	1%	0%
20 to 50 cm	10%	80%	10%
>50 cm	5%	15%	80%
>1 m	0.10%	4.90%	95%

To use this model, the hydrodynamic results and more particularly the water level are needed as well as the geo-localisation of the assets (contained in the World State).

Depending of the hazard intensity where the asset is localised, the results of this model is the probabilities for this asset to be in the damages classes. For example, if the water level above a dike at a medium status is 40 **cm**, the results will be 10% of chance to resist, 80% of chance to breach, 10% of chance to totally fail.

Furthermore, It's possible to simulate the several time local model. We should at each step calculate the new dike status. The new dike status depend the damage probability results and user choice (failure or break) at each simulation, so we have the evolution of dike vulnerability for each simulation.

Implementation:

The following figure shows the high level **architecture** of the **reference application** for the Coastal Submersion Domain. The user **interface** has various **GUI** elements such as the simulations **Model** interactions, the **GIS**, the worldstate management and the indicators and economic calculation setup. Different models can be used by the users and they can have the interactions between them and the others elements of the **infrastructure**.

Constituents:

Simulation Model Integration BB

Integration level: Integrated

The **Simulation Model** Integration **Building Block** is used to wrap the different models used in the **Reference Application**.

Data Integration

Integration level: Integrated

This **Building Block** is used to access geospatial data of the various Worldstates of the **Reference Application**. The geospatial results of the Coastal Submersion **Model** and the **Evacuation** Model are published thanks to Data Integration Building Block.

Simulation Model Interaction View

Integration level: Integrated

The **Simulation Model** Interaction View lists all of the available processes (model runs) of the **Reference Application**. **It** allows the user to configure, execute and monitor each Simulation Model.

Worldstate View

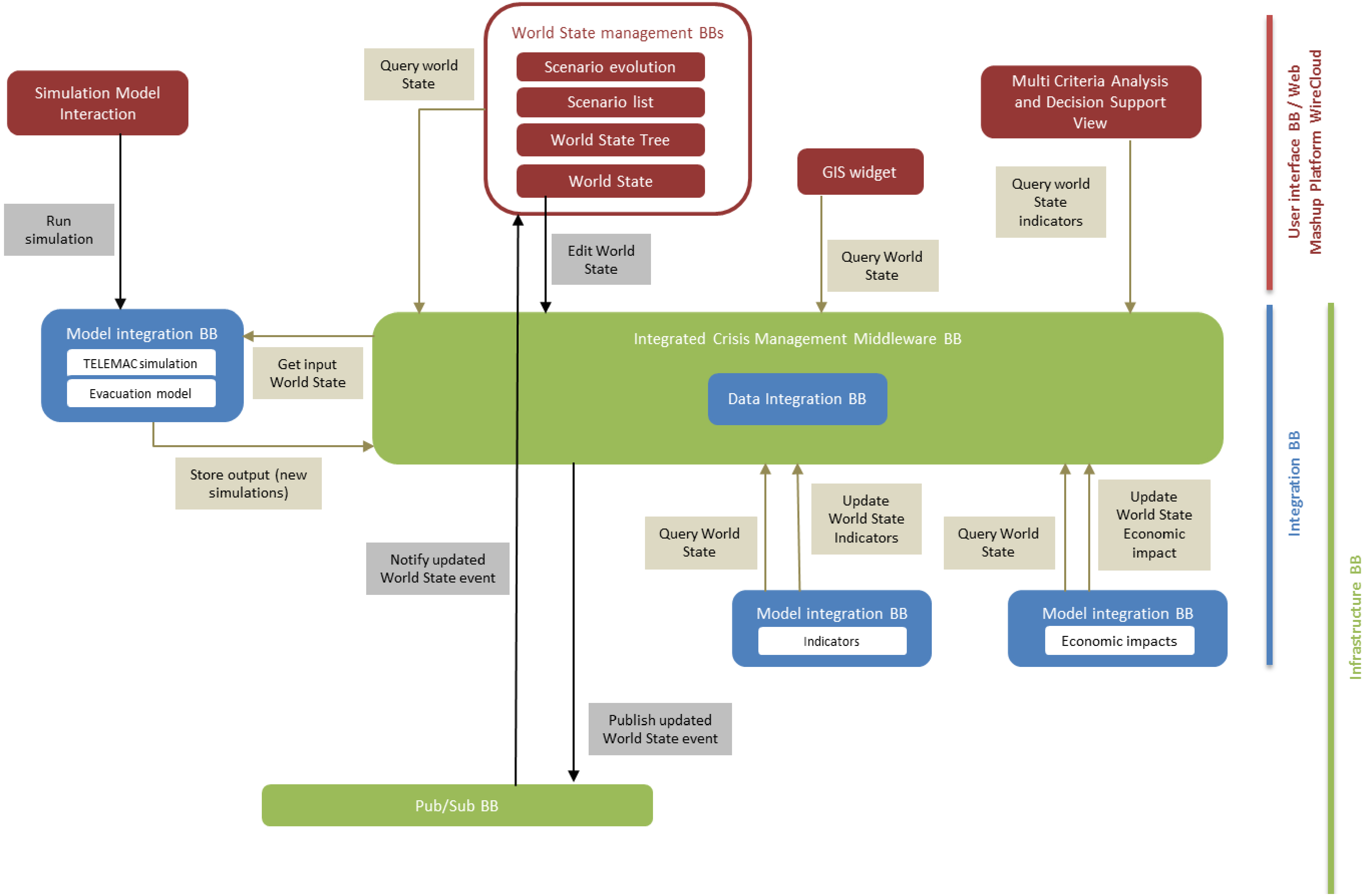
Integration level: Integrated

The Worldstate View consists of independent and generic Widgets that allow user to discover, visualize and manipulate the different Worldstates of the Coastal Submersion Scenarios.

Integrated Crisis Management Middleware BB

Integration level: Integrated

The Integrated **Crisis Management** Middelware is mandatory for each **CRISMA Federation** as specified by the **CRISMA Framework** **Architecture**. The **ICMM**


World State management BBs
Scenario evolution
Scenario list
World State Tree
World State
Query world State
Simulation Model Interaction
Run simulation
Multi Criteria Analysis and Decision Support View
GIS widget
Query world State indicators
Edit World State
Query World State
User interface BB / Web Mashup Platform WireCloud
Model integration BB
TELEMAC simulation
Evacuation model
Get input World State
Integrated Crisis Management Middleware BB
Data Integration BB
Store output (new simulations)
Integration BB
Query World State
Update World State Indicators
Query World State
Update World State Economic impact
Notify updated World State event
Model integration BB
Indicators
Model integration BB
Economic impacts
Infrastructure BB
Publish updated World State event
Pub/Sub BB

collects information regarding to data, **Simulation** Models, Worldstate Transitions, Indicators, etc. of the **Reference Application**.

Publish Subscribe Context Broker BB

Integration level: Todo for V2

The Publish Subscribe Context Broker is used to inform users when specific events are recorded in other components of the **Reference Application**. Depending on the interest of the users, **it** send messages when, for example, a new Worldstate is available.

GIS View Building Block

Integration level: Integrated

The **GIS** View is used in the **Reference Application** to display results of **Simulation** Models and for manipulating Worldstates.

SPB has developed a GIS View Software Component for the Reference Application in compliance with requirements for the Coastal Submersion **Scenario**:

- **WMS**/**WFS** with different type of geographic resources
- Popup to display dikes **vulnerability**
- Time slider
- Modify the area of interest (AOI)
- Modify the opacity of the layers
- Create new features aiming to Modify dikes' vulnerability

Coastal Submersion Model

Integration level: Integrated

The Costal Submersion **Model** is a 2D-hydrodynamic model based on the **open source** TELEMAC-MASCARET **system**. **It** allow to calculate the hydrodynamics field as flow velocity, water level, ...

One of the important is to progress in the coastal submersion model which run well but take too much time to give a result. One part of our future work will concentrate on this point with the objective to reduce computation time.

Evacuation model for coastal submersion

Integration level: Todo for V2

The **evacuation** **model** uses the software LSM2D and calculates the population displacement. The problem of this model is that we are limited by domain size, so **it** will be implemented only specific area.

Population exposure model

Integration level: Integrated

The general Population **Exposure** **Model** is adapted to the **situation** in the coastal submersion domain. **It** provides population distribution for a selected date and time.

Dikes vulnerability model

Integration level: Integrated

In view of results of dike **vulnerability model** which give the potential statistical **impact** on dikes, the user could make an informed choice on break or failure dikes for local **simulation**.

Multi Criteria Analysis and Decision Support View

Integration level: Integrated

The Multi **Criteria** Analysis and Decision Support View is used to compare specific Indicators and Criteria for different scenarios of Coastal Submersion. The Indicators that have been defined relate to the number of flooded houses (depending on the water depth) , economic **losses**, flooded area, ...

Documentation:

Demo

Video coming

Publications

A Poster was shown for international conference: Cocorisco (Connaissance et Compréhension des Risques Côtiers : Aléas, Enjeux, Représentations, Gestion).

The firsts part of poster speaks about implementation of a user defined dike behaviour with the help of Damage Probability matrix with the different **simulation** step :

1. Global scale **model**
2. Dikes **vulnerability model**
3. User choice (break or failure dikes)
4. Local scale model
5. Computation of indicators
6. Key performance indicators comparison

The second part speaks about coupling model.

- Modelling submersion on the Charente-Maritime coast **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP53/05_publication/COASTAL_RISKS_2014_3and4july/poster/Crisma_poster_BREST_20140313_V2.pdf**

Downloads:

We have a platform with the **demonstration** of the **application** V1 but **it** isn't accessible for this moment.

- Demonstation of application **http://crisma.spacebel.be/pilotB (http://crisma.spacebel.be/pilotB)**

Team:

Agnès Cabal

Development roles: coordinaton

Agnès coordinates the **pilot** B.

Mehdi Pierre Daou

Development roles: integration, modelling

In charge to create the coastal submersion **model**.

Help for the integration the building blocks within **Pilot** B.

Arnaud De Groof

Development roles: architecture, configuration, development, installation, integration

Responsible for development of the **Simulation Model** Interaction **Widget**.

Provides support for the integration of simulation models (Simulation Model Integration **Building Block**)

Marc Erlich

Development roles: coordinaton

Marc coordinates the **pilot** B.

Marianne Grisel

Development roles: modelling

In charge to create the **Evacuation** **model** for coastal submersion.

she is working on the indicators.

Data:

Hydrodynamics model data

Dikes height and quality comes field measurement. (used also in dikes **vulnerability model**)
The evolution of the mean sea level at the marine boundary in time comes tidal measurement.
The wind velocity of area comes the weather station measurement.
Tubes comes field measurement of structure position.
Swell data comes swell model (Tomawac which is a modul of Open-Telemac- Mascaret).

LSM Model data

Population repartition known by work on INSEE data ("données carroyées")
Road network (data by IGN, BD TOPO)
Buildings (data by IGN, BD TOPO)
The **warning** centers (localization known by field investigation)

indicator data

Population repartition known by work on INSEE data ("données carroyées")
Road network (data by SDIS17)
Buildings (data by SDIS17)

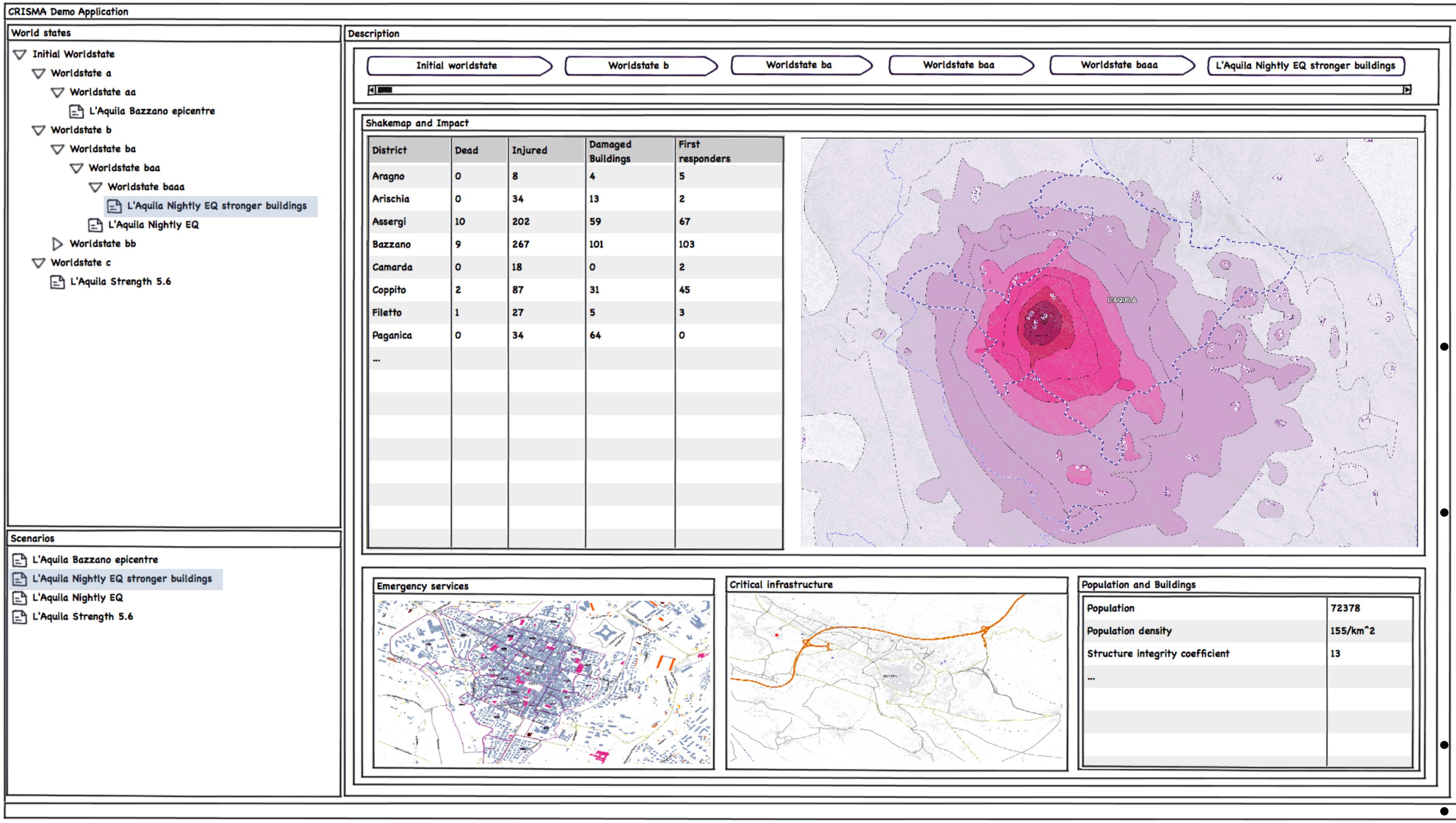
Reference Application for the Earthquake and Forest Fire Domains

General Description

This is the **Reference Application** for the Earthquake and Forest Fire Domains. **It** represents the transferable wiring together of **CRISMA Framework** Building Blocks and **Simulation** Models (Software Components, respectively) without any **pilot**-specific logic. The main objective of this Reference Application is to provide a customizable, extensible and transferable example of a **CRISMA Application** for the simulation of natural **disaster** (Earthquake and Forest Fire) with irreversible damages that enables comparing scenarios in a multi-**risk** framework including cascading events simulation and the assessment of **decision making** choices and possible consequences in each foreseen evolving **scenario**.

Mockup of the Reference Application

The **Reference Scenario** and the related Simulation Cases implemented by this application are those of the Pilot D - Geophysical hazards – L'Aquila (Italy) **use case**. Thus, the integrated Reference Application for the Earthquake and Forest Fire Domains is the basis for the development of the Pilot D CRISMA Application.



The Reference Application for the Earthquake and Forest Fire Domains demonstrates the capabilities of the CRISMA Framework regarding

- **Response** – supporting decision makers by providing alternative management policies
- Quantitative assessment of **impact** scenarios (simulated by assumption of alternative strategies) to compare
- Cost/ benefit analysis
- Multi-**criteria** analyses
- Assessment of choices (in planning and/ or **emergency**) on the basis of scenarios and Cost/ benefit Analyses.

Supported Simulation Cases

According to the current specification of the Pilot D **Architecture**, the Reference Application has to support the following types of Worldstate Transitions (**Transition** Points):

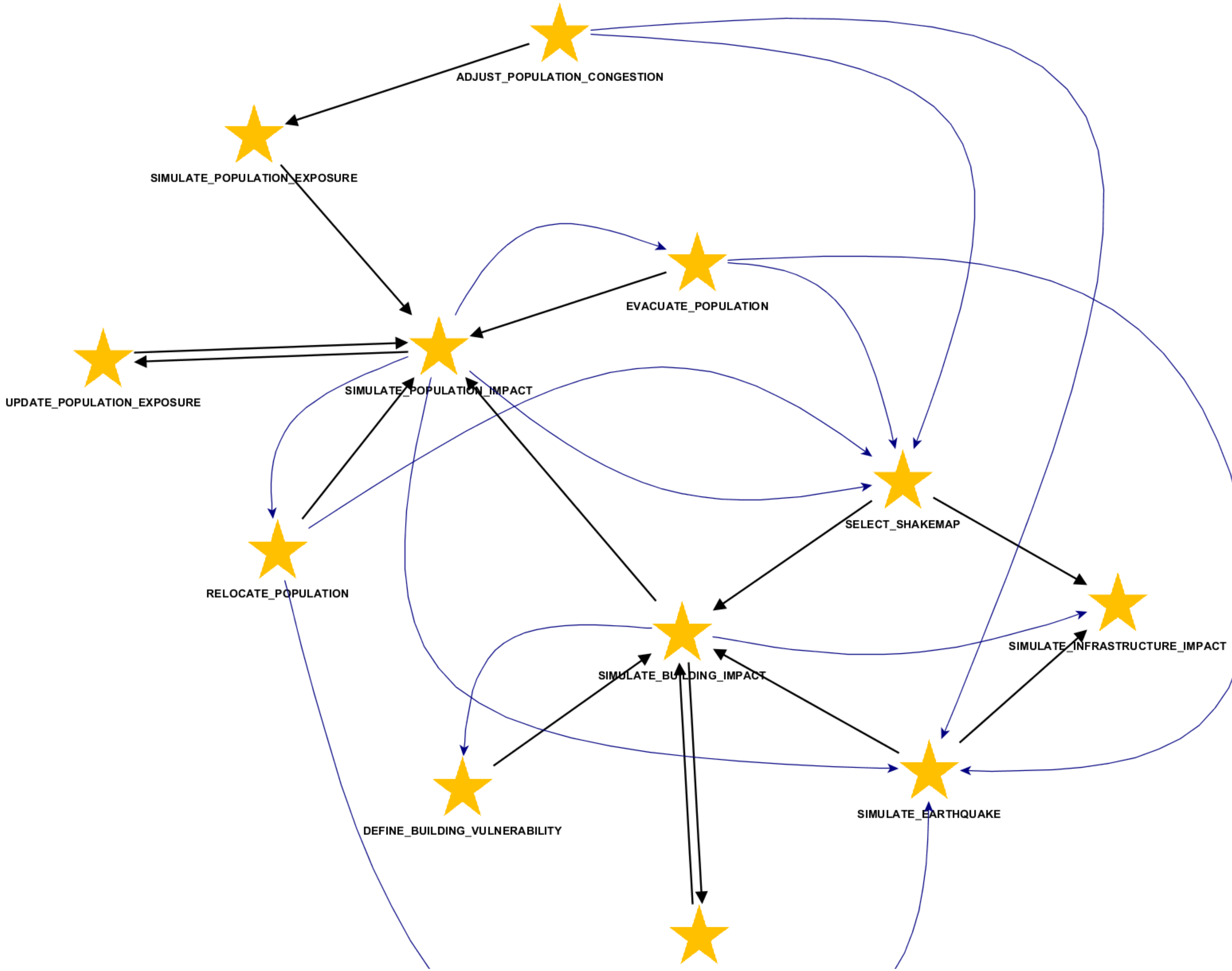
1. **Adjust the population congestion**

Allows the decision maker to reduce the congestion of population in certain areas, thus effectively **tuning** of percentage of resident population per gird cell. This is a **mitigation** option to reduce population **exposure**.

2. **Define Building Vulnerability**
   Vulnerability Classes (EQV_BUILDINGS) of the Buildings can manipulated by the user, e.g. to simulate mitigation measures (increasing building resistance). Thus, for subsequent impact simulations, the changed vulnerability classes are now used.
3. **Evacuate Population**
   Decision Make chooses the **evacuation** option. Evacuation can be performed before (**T0**) or after (T1) earthquake main shock.
4. **Relocate Population to hospitals**
   Decision Manager can choose the option to relocate injured people to hospitals
5. **Select a Shake Map**
   The user can introduce a new seismic **hazard** by selecting a static shake map (historical earthquake data).
6. **Perform building damage assessment**
   Building damage distribution for each class and for each cell is updated according to a building **impact Model**.
7. **Simulate an earthquake (Seismic Hazard Model)**
   The user defines the seismic hazard by running an Earthquake Model to calculate a Shake Map.
8. **Simulate the impact of the seismic hazard on the infrastructure**
   Simulate the impact of the seismic hazard on the infrastructure, e.g. damage to roads.
9. **Estimate the number of occupants for each building class and for each grid cell**
   Through a Simulation **GUI**, the User invokes the HUMAN_EXPOSURE model in order to estimate the number of occupants for each building class and for each grid cell (that will consider also the time of the day when hazard happens).
10. **Simulate the impact on the population (Casualties Model)**
    Given building damage distribution and population distribution the earthquake casualties model can run
11. **Update damage classes of buildings**
    Damage classes of buildings are updated according to damage caused by earthquake (T1, ...).
12. **Define human exposure**
    The Decision Maker can chooses one of the following options to directly define the human exposure: 1.) exposure after EQ casualties assessment, 2.) injured moved to hospitals; reallocation based on hospitals position (after Main Shock), or 3. Evacuation option.

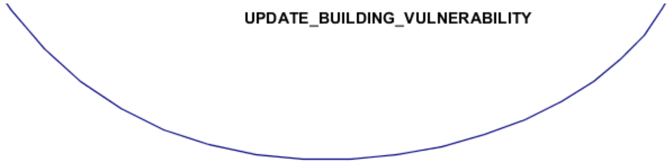
Transition Point Graph

The figure above shows an example of the (draft) Transition Point Graph of the Reference Application and the Pilot D Application, respectively. The order of some of these Transitions is directly related to the evolution of a **crisis** (black arrows). Additionally, there are many alternative paths (blue arrows) that heavily depend on the actual **Simulation Case**, e.g. whether mitigation options are performed before or after the crisis, whether **cascading effects** and/or **time dependent**



vulnerability have to be taken into account, etc.

The Transition Points define the main workflow as well as the user interactions within the Reference Application. However they do not specify the different types of Worldstate Analysis that can be performed by the user. The analyses are related to specific Simulation Case and thus are described as aprt of the Simulation Case. Currently, only two Simulation Cases have been fully specified for the Reference Application. Further Simulation Cases and Transition Points will be added when the complete



Architecture Specification of Pilot D is available.

Simulation Case 1 – "Incident Evolvement"

1. The users chooses the kind of hazard input and the **event** characterization (e.g. shake maps or values of epicentre, magnitude and depth if shake maps are not available; event characterization includes time- e.g. daytime, night time).
2. Estimate the number of occupants for each building class and for each grid cell: Through Simulation GUI, invoke the Human Exposure model in order to estimate the number of occupants for each building class and for each grid cell (that will considers also the time of the day when hazard happens).
3. Perform building damage assessment. In this step building damage distribution for each class and for each cell is updated (now considering the damage not equal to zero)
4. Given building damage distribution and population distribution the earthquake casualties model can run. **ICC** functions run to re-calculate Indicators, Criteria and Costs
5. The effects of another earthquake shall be simulated. Another shake map is needed.
6. The Decision Maker chooses one of the following exposure options 1st alternative – varied exposure after EQ casualties assessment, 2nd alternative – injured moved to hospitals ; reallocation based on hospitals position, 3rd alternative – evacuation option (% EV. MODEL)
7. Perform building damage assessment (with updated buildings EQ_V classification distribution).
8. Perform casualties assessment. ICC functions run to re-calculate Indicators, Criteria and Costs
9. **Analysis**: User compares impact scenarios in terms of casualties. User compares impact scenarios by simple visualization of maps representing casualties distribution (Data Slot: CASUALTIES_DISTRIBUTION) for each of the alternatives.
10. **Analysis**: User performs **World State** Analysis choosing Multi-criteria Analysis to compare alternatives (e.g. considering casualties (Data Slot: CASUALTIES_DISTRIBUTION) and costs (ICC Element: COSTS))

Simulation Case 2 – "Long term planning"

1. Visualization of the isoseismal map.
2. Visualization of **impact scenario** without any mitigation measures
3. Visualization of impact scenario after having increased building resistance
4. Visualization of impact scenario after a delocalization of productive activities
5. **Analysis**: Comparison of alternative scenarios with a cost/benefit analysis

Implementation:

The **Reference Application** for the Earthquake and Forest Fire Domains will either be based on the Wirecloud Platform or **it** will be developed as stand-alone Rich

Internet Application. This depends on the availability of reusable Wirecloud Widgets (Mashable Composite UI Modules) for all types of required User Interaction Building Blocks as well as on the technical feasibility of the mashup-up approach for such a complex application. Custom development of UI Widgets as well as development of Building Blocks Software components will in most cases be based on Angular **JS**.

The figure below shows a very high level view of the user **interface** of the application.

High level view of the user interface of the Reference Application

Main UI Elements are the generic Worldstate Management Views which provide an overview on the various scenarios (worldstate tree path) and allow selecting different worldstates for comparison and analysis. Most of the User Interface Building Blocks to be used in the Reference Application are still under development. Especially the **Simulation** Interaction Widgets and Worldstate Analysis Widgets are not yet available.

The figure below shows an overview on the **Architecture** of the Reference Application which is identical to the Architecture of the **Pilot** D **CRISMA Application**. It contains all Building Blocks, Simulation Models and the most important data sources (Data Slots) that are required by the Application.

Overview on the Architecture of the Reference Application

A detailed description of the usage and integration needs of the individual Building Blocks can be found under the "Constituents" tab.

Constituents:

Integrated Crisis Management Middleware BB

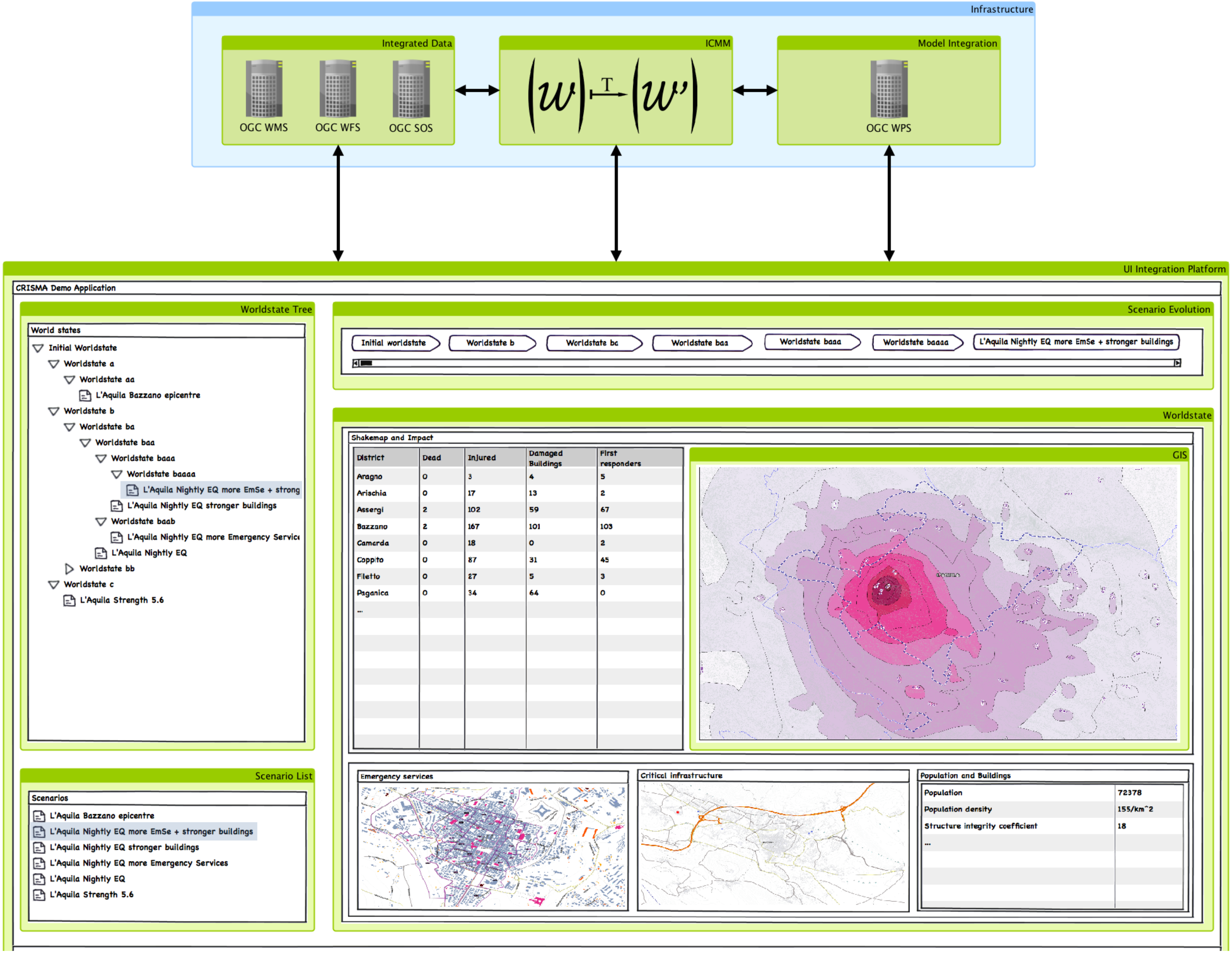
Integration level: Integrated

This **Building Block** is mandatory for each **CRISMA Federation** as specified by the **CRISMA Framework** **Architecture**. The main challenge is defining an appropriate extension of the Core **CCIM** to be able to describe the domain specific aspects of the **Application**. Furthermore detailed Information on Data Slots, **Simulation** Models and **Transition** Points are required to be able to populate the **ICMM** with the respective control and communication information. This information has been collected in the **Pilot** Architecture of Pilot D (Earthquake and Forest Fire). Integration with the **REST** **API** of the ICMM is straightforward and supported by the on-line documentation available in the **catalogue**. Installation, configuration and hosting of the ICMM is provided by **CIS**.

Data Integration

Integration level: Integrated

This **Building Block** is required for accessing respective Data Slots of the Worldstate of the **Application**. **It** is a very generic **BB** which is realised entirely by existing software (**GeoServer**, MapServer). Therefore its configuration is in general performed during the development of the respective **Pilot** Applications. Installation and Configuration of the Building Block Software Component (GeoServer) for the **Reference Application** and provision of representative **test** data is performed by **CIS**



and **AMRA**.

Indicator Building Block

Integration level: Integrated

The Indicators **Building Block** is required for calculating Indicators, including simple Economic Indicators. For this reason, no dedicated Economic Impacts **Model** is used in this **Reference Application**.

GIS View Building Block

Integration level: Integrated

The **GIS Widget Building Block** is integrated in the **Reference Application** and is responsible for visualising data items that have a spatial context.

Simulation Model Integration BB

Integration level: Integrated

The **Simulation Model** Integration **Building Block** is used to integrate various domain-specific Simulation Models of this **Reference Application**, like the Building **Impact Model**.

Publish Subscribe Context Broker BB

Integration level: Integrated

The Pub/Sub Context Broker is used in the communication of the Indicators **Building Block** with the **ICMM**.

Worldstate View

Integration level: Integrated

The Worldstate View and its related Widgets are required for this **Reference Application** to visualise the evolution of different simulated **Crisis Management** Scenarios. These Widgets are generic and need in general no specific **adaptation** for a specific **CRISMA Application**.

Simulation Model Interaction View

Integration level: Integrated

The **Simulation Model** Interaction View provides generic Simulation Model configuration, execution and monitoring capabilities.

UI Integration Platform BB

Integration level: Integrated

The **Reference Application** UI is composed of different Composite UI Modules (HTML5 & JavaScript Widgets) and thus relies on the UI Integration Platform **Building Block**.

Multi Criteria Analysis and Decision Support View

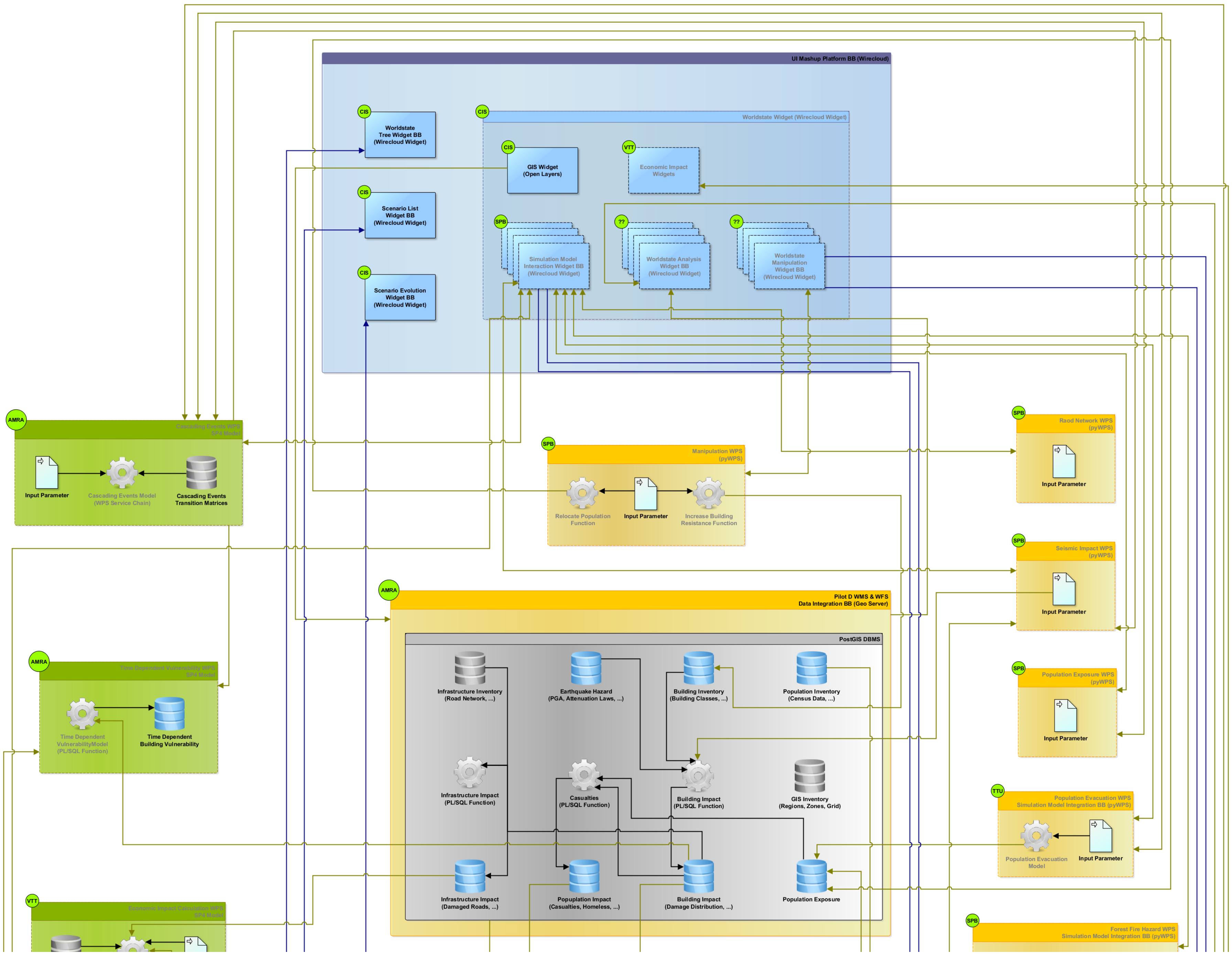
Integration level: Integrated

The **Reference Application** integrates provides advanced decision support capabilities on the basis of Multi **Criteria** Analysis and Decision Support View.

Building impact model

Integration level: Integrated

Provides Building **Impact** Maps. This domain-specific **Simulation Model** is integrated with the **Reference Application** with help of the Simulation Model Integration



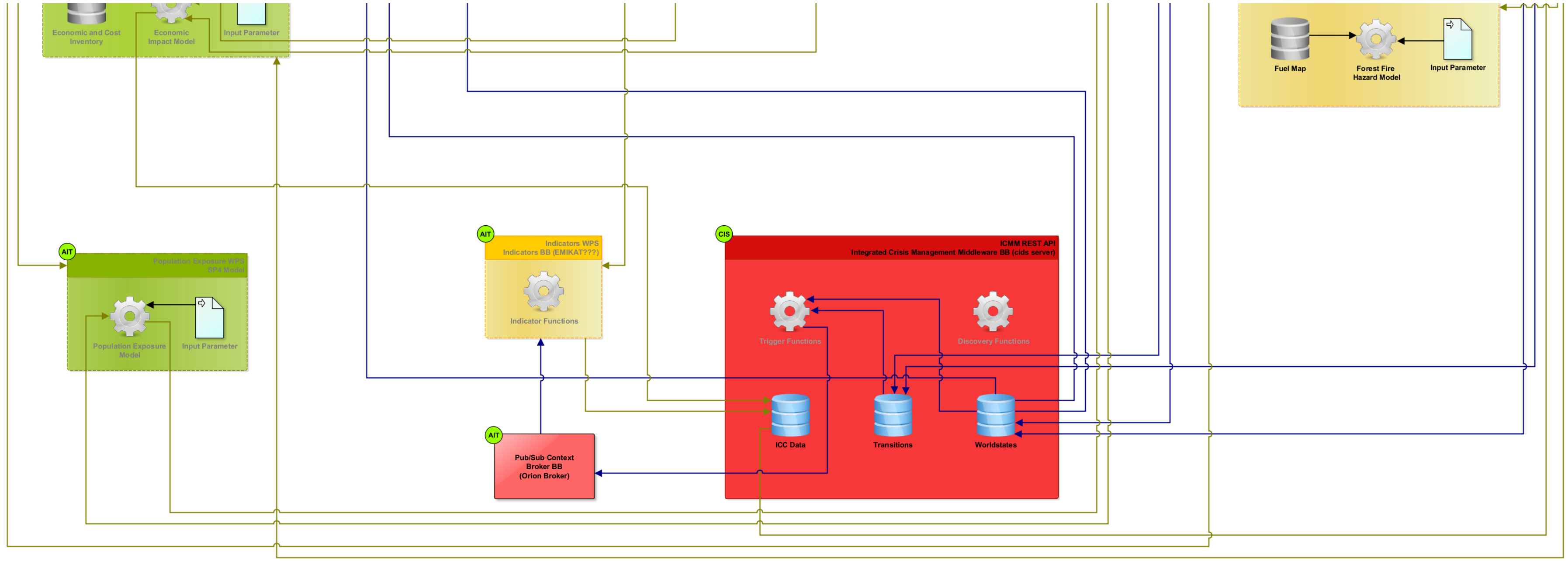
and Interaction Building Blocks.

Earthquake casualty model

Integration level: Integrated
Provides Casualty Maps. This domain-specific **Simulation Model** is integrated with the **Reference Application** with help of the Simulation Model Integration and Interaction Building Blocks.

Population exposure model

Integration level: Integrated
Used for the calculation of human **exposure**. This **CRISMA Model** is integrated with the **Reference Application** with help of the



Simulation Model Integration and Interaction Building Blocks.

Time Dependent Vulnerability model (TDV)

Integration level: Integrated

Used for the calculation of the evolvement of **vulnerability** curves. This **CRISMA** **Model** is integrated with the **Reference Application** with help of the **Simulation Model** Integration and Interaction Building Blocks.

Cascade Events Configuration and Interaction View

Integration level: Integrated

This **Building Block** allows the user to choose a **path of analysis** from a predefined **transition matrix**. This **Widget** is similar to the **Simulation Model** interaction Widget. A respective Functional Description of **Cascade Events** Building Block is described in V2 of the **CRISMA Framework** **Architecture** (D32.2).

Road network vulnerability model (RNV)

Integration level: Integrated

A **Simulation Model** for the assessment of probability of road link interruption due to earthquakes. This domain-specific Simulation Model is integrated with the **Reference Application** with help of the Simulation Model Integration and Interaction Building Blocks.

Cascading Effects Model

Integration level: Integrated

Cascading Effects is a major topic in this **Reference Application** as **it** focuses on the cascading effects of earthquake after earthquake and forest fire after earthquake.

Forest fire behaviour model

Integration level: Integrated

The Forest Fire Behaviour **Model** is used to investigate the cascading effect of a forest fire that is initiated by a preceding earthquake.

Scenario Analysis and Comparison View

Integration level: Integrated

This view enables the analysis and comparison of Worldstates on the basis of the Indicators for this **Reference Application**.

Documentation:

Specifications

This document is a draft **Application Architecture** Specification for **Pilot** D in accordance to the Application Architecture Template (**http**://goo.gl/C0XYYP). **It** has been produced on basis of the material (deliverables and presentations) currently available for Pilot D. Although still incomplete and lacking of most technical details, it may serve as example for the specification of the Application Architectures of other Pilots and as first starting point for the update of Reference Scenarios in SP2.

- Pilot D - Application Architecture
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP55/01%20work%20in%20progress/Pilot%20D%20-%20Application%20Architecture.doc
- Pilot D - Application Architecture (Diagrams)
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP55/01%20work%20in%20progress/Pilot%20D%20-%20Architecture%20(diagrams).zip
- yEd Graph Editor
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP55/01%20work%20in%20progress/yEd%20Graph%20Editor.aspx

Downloads:

- CRISMA reference application for Earthquake and Forest Fire (requires authentication)
  https://github.com/cismet/crisma-ref-app
- Development Snapshot (requires authentication)
  http://crisma.cismet.de/refApp (http://crisma.cismet.de/refApp)

Team:

Pascal Dihé

Development roles: architecture, coordinaton, development, requirements, specifications

Responsible for specification of the overall **architecture** of the **Reference Application** in compliance to the **CRISMA Framework** Architecture and specification of the **CCIM** of the Application. Assures that the specifications and developments evolve in accordance with the evolving **user requirements** and the lessons learned during development. Responsible for the development of the Reference Application and **ICMM** extensions.

Stefano Nardone

Development roles: development, modelling, requirements

Responsible for all technical aspects of **Pilot** D that drive or influence the development of the **Reference Application**. Assures that the Reference Application can be used as basis for the development of the Pilot D Application. Provides **test** data and geo services that can be used to leverage the integration. Provides support for the hosting, installation and configuration of the respective **model** **service** (**WMS**/**WFS**).

Martin Scholl

Development roles: configuration, development, installation, integration, support

Responsible for the development and integration of the **ICMM** and the Worldstate Views (Widgets). Main responsible for the deployment and configuration of the **Reference Application**.

Peter Kutschera

Development roles: integration

Provides support for the integration and usage of the Indicators **Building Block**. Provides support for the hosting, installation and configuration of the respective **service** (**WPS**).

Christoph Aubrecht

Development roles: integration

Provides support for the integration and usage of the Population **Exposure** **Model**. Provides support for the hosting, installation and configuration of the respective model **service** (**WPS**).

Arnaud De Groof

Development roles: development, installation, integration

Provides support for the integration of **simulation** models (**Simulation Model** Integration **Building Block**), develops the respective **pyWPS** wrappers and simulation model configuration user interfaces (Simulation Model Interaction Building Block). Provides support for the hosting, installation and configuration of the respective services (**WPS**).

Thorsten Hell

Development roles: development

Provides integration and development support regarding the **ICMM** and the **GIS** **Widget** **Building Block**.

Markus Jähi

Development roles: integration

Provides support for the integration and usage of the **Economic Impact** Building Blocks.

Hermann Huber

Development roles: configuration, installation

Provides and maintains an instance of the Pub/Sub Context Broker **Building Block**.

Manuel Warum

Development roles: configuration, installation

Provides and maintains an instance of the UI **Mashup Platform** **Building Block** (Wirecloud).

Data:

Infrastructure Inventory

Electrical power supply towers, Road Netowk, etc.

Building Inventory

Buildings in the study area, **vulnerability** classes, etc

Population Inventory

Census data, People distribution and **vulnerability** classes

Infrastructure Impact

Output of the respective **impact model**: damaged roads, etc.

Popuplation Impact

Output of the respective **impact model**: Casualties, Homeless, ...

Building Impact

Output of the respective **impact model**: Damage Distribution, ...

Population Exposure

Output of the respective **exposure** **model**.

Fuel Map

Inventory of the different materials within an area with information related to fire, like the fuel within a specific cell that would feed a possible fire.

Fire intensity

Fire **model** result data: grid of fire intensity data

Fire RoS

Fire **model** result data: grid of fire rate of spread data

Reference Application for the Nordic Winter Storm Domain

The **Reference Application** for the Nordic Winter Storm Domain serves as a technological example for one class of the applications which can be implemented by wiring the **CRISMA** Components (reference implementations of the **CRISMA Framework** Building Blocks and Models) together. **It** provides the software functionality which is required in the CRISMA **Pilot** A “Electricity outage in the far North" and similar scenarios.

The reference application demonstrates the usage of **agent**-based simulations with appropriate models describing cooling of buildings and living conditions (life quality) in selected geo-cells within the **crisis** area, during electric power limitations and extreme low temperatures. The application has facilities for driving the simulations, demonstrating planning of **evacuation** and **evaluation** of economic impacts. Choosing different setups and **simulation** scenarios, the operator can find an optimal solution for both planning and **response**.

The Reference Application for the Nordic Winter Storm Domain demonstrates the capabilities of the CRISMA Framework regarding:

- Simulation of the conditions and their **impact** during a long duration, multi-phased **emergency**
- **Preparedness** Planning
- Cost / benefit analysis
- Assessment of choices (both in the planning phase and during the actual emergency)

World State Transitions:

The Reference Application has to support the following types of actions, which cause a World State **Transition**:

1. Simulation: Weather conditions causing the emergency **situation**, i.e. long duration, extreme cold weather. Weather data is available e.g. with a resolution of 3 hour intervals.
   - COLD (**Exposure**)
2. Manipulation: Availability of electricity cause by cold weather or external phenomena. User in the **role** of the "Game Master" decides which areas are impacted.
   - AVAILABILITY OF ELECTRICITY (Exposure)
3. Simulation: Impact of combined extreme cold and electric power availability to the residental buildings, i.e. cooling of the houses decreasing their residental eligibility
   - COOLING (Impact)
4. User decision: what are the **mitigation** actions, e.g. to ration the availability of electricity or to evacuate the people.
   - ELECTRICITY RATIONING (Mitigation)
   - **EVACUATION** (Mitigation)
5. User decision: what are the resources used to mitigate the situation?
   - **ALLOCATED RESOURCES** (Mitigation)
6. Simulation: progress of the evacuation. Is the impact of the mitigation actions sufficient within the **scenario** timeframe?
   - **EVACUATION** (Impact)
7. Evaluation: The **economic impact** of the decisions and final outcome assessed with different indicators
   - COST (Outcome)
   - **INDICTORS** (Outcome)

The following data is assumed to be available or to be provided by the user:

1. Population **model** of the affected geographical area
2. Electric distribution network topology of the affected geographical area
3. Basic information about the residental buildings, i.e. type of house, means of heating, level of insulation, year of construction
4. Available resources in the initial state to mitigate the situation

Simulation Cases

The Reference Application for Nordic Winter Storm has the following simulation cases:

1. The meteorolgical authority provides the **Weather Model** representing the simulated weather conditions for the target area for whole duration of the scenario.

Weather Model is provided as part of the World State data contents.
2. Impact of the electricity availability problems is manipulated by the user, i.e. user in the role of game master decides which areas are left out of electricity in each world state
3. Impact of the cold weather and lack of heating combined is simulated with the **House Cooling Simulation Model**.
4. Impact of the mitigation actions (i.e. evacuation) is simulated with the **Evacuation Resource Simulation Model**.

Prepredness Planning

Preparedness Planning component of the reference application enables the **end user** to perform the following tasks:

1. User can create preparedness plans based on analysis of threats, vulnerabilities and risks that are relevant to the geographical area or domain of interest. The preparedness plans are created in the pre-**incident** phase.
2. User can excecute the created preparedness plans. As a results the plan proposes different alternatives for resources to be allocated to mitigate the situation.
3. As the outcome of the user's selection the preparedness plan allocates the resources to mitigate the situation.

Evaluation

The outcome of the decisions is analyzed using two tools:

1. **Economic Impact Model** provides the economic result of the decision, e.g. cost of resources used in evacuation.
2. **Indicators**: different indicatiors are used to illustrate the non-economic outcome of the decision, e.g. "how many people still needs to be evacuated"

Application usage in different phases

1. In the Pre-incident phase, user creates the initial world state, i.e. populates the World State Repository with information about population, **infrastructure**, resources and weather data, using the **OOI Management Views**. Based on **risk analysis**, user creates the preparedness plans using the **Preparedness Planning Views**.
2. During the incident, the user in the role of game master, decides which areas are having electricity distribution problems and manipulates the OOI information respectively. User running the simulation gets visualized information on which areas are impacted by the situation using the **Simulation Model View**. User makes decisions to mitigate the situation by either rationing the electicity (i.e. giving electricity to the currently impacted areas and taking the power off from some other areas). Alternatively, user can allocate evacuation resources using the **Preparedness Plan Excecution View**. The preparedness plan gets the information on available resources from the World State Repository and returns the information on allocated resources, respectively. The Economic Impact Model provides the cost of the decisions in the **Economic Impact Calculation View**. At any stage user can browse the world state contents using the **Worldstate View** and visualize the situation in the selected world state in the **Simulation Model View**.

3. In the Post-incident phase, the final outcome of the scenario is calculated as KPIs and illustrated in the **Scenario Analysis and Comparison View**.

Implementation:

The figure below illustrates the high level **architecture** of the **reference application** for the Nordic Winter Storm Domain. The user **interface** concists of several views for e,g, setting up the input data, creating and viewing the scenarios. creating and using the **preparedness** plans, interacting with the **simulation** models and visualizing the **simulation results**, calcuated costs and KPIs. The control paths between the BBs are not shown in the figure.

High Level Architecture (BB Level) of the reference application for the Nordic Winter Storm domain

Constituents:

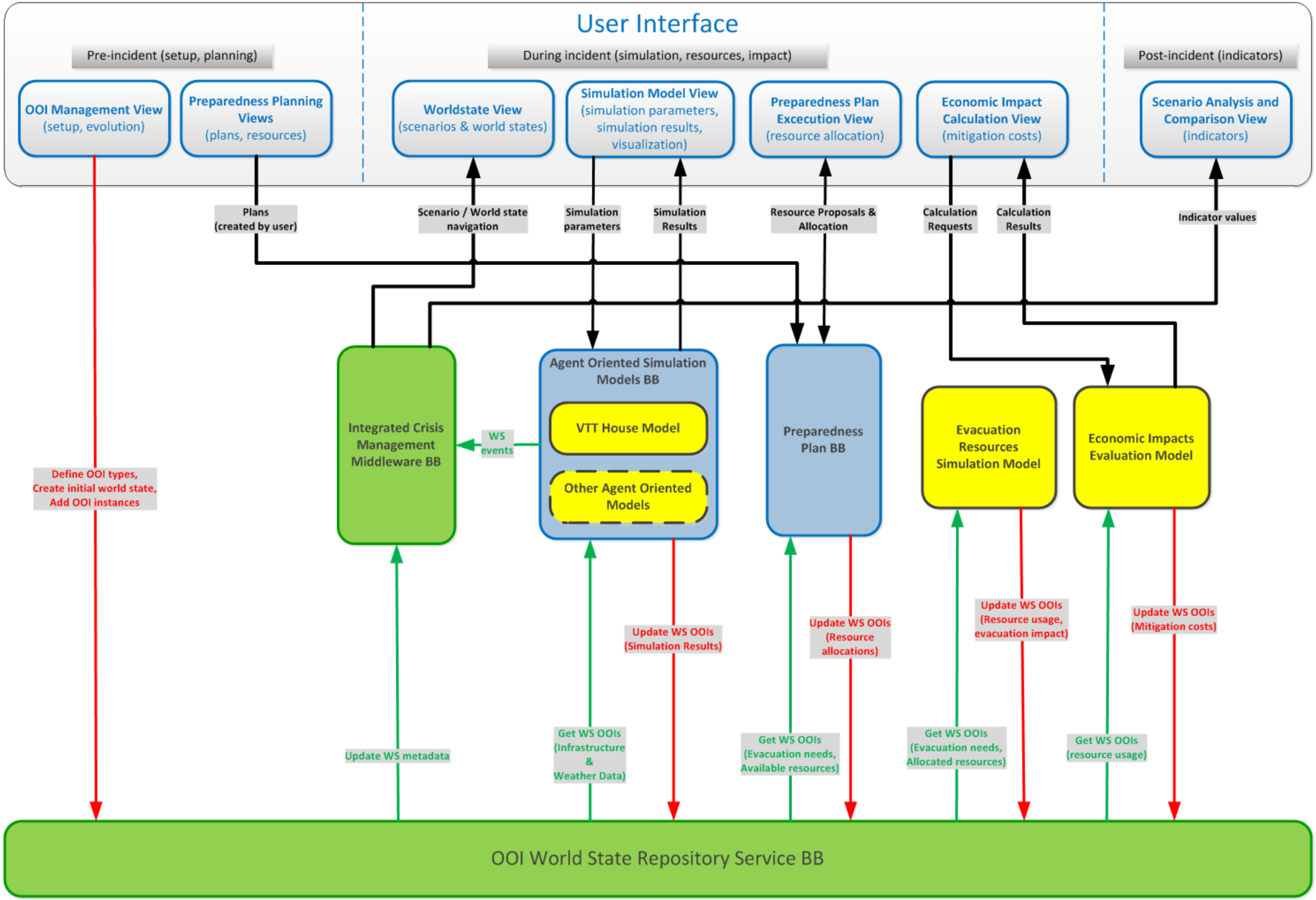
OOI World State Repository BB

Integration level: Integrated

OOI **World State** Repository acts as the centralized data repository in this **application**.

The following OOI types and their attibutes are used to represent the world in each state:

- Geocell (1 km x 1 km grid):
  - weather (temperature, wind)
  - electricity network status
  - living condition index
  - building condition index
  - population
  - **infrastructure** (residental buildings of different types)
  - needs (e.g. need for **evacuation**)
  - **emergency** level
- Buildings
  - location
  - buiding type


User Interface
Pre-incident (setup, planning)
During incident (simulation, resources, impact)
Post-incident (indicators)
OOI Management View (setup, evolution)
Preparedness Planning Views (plans, resources)
Worldstate View (scenarios & world states)
Simulation Model View (simulation parameters, simulation results, visualization)
Preparedness Plan Excecution View (resource allocation)
Economic Impact Calculation View (mitigation costs)
Scenario Analysis and Comparison View (indicators)
Plans (created by user)
Scenario / World state navigation
Simulation parameters
Simulation Results
Resource Proposals & Allocation
Calculation Requests
Calculation Results
Indicator values
Integrated Crisis Management Middleware BB
WS events
Agent Oriented Simulation Models BB
VTT House Model
Other Agent Oriented Models
Preparedness Plan BB
Evacuation Resources Simulation Model
Economic Impacts Evaluation Model
Define OOI types, Create initial world state, Add OOI instances
Update WS metadata
Get WS OOIs (Infrastructure & Weather Data)
Update WS OOIs (Simulation Results)
Get WS OOIs (Evacuation needs, Available resources)
Update WS OOIs (Resource allocations)
Get WS OOIs (Evacuation needs, Allocated resources)
Update WS OOIs (Resource usage, evacuation impact)
Get WS OOIs (resource usage)
Update WS OOIs (Mitigation costs)
OOI World State Repository Service BB

    - in-house temperature
    - eligibility status
- Resources
    - location
    - **resource** type (e.g. bus, ambulance, hotel)
    - capabilities (e.g. people transportation, accomodation)
    - availability state (e.g. available, incoming to target, outgoing from target)
    - usage (e.g. amount of kilometers driven in this world state)
    - travel time to target (in case of tranportation resource)
    - average speed (in case of transportation resource)
- Costs
    - transportation costs
    - accomodation costs
    - damage costs

The attributes represent the state of each OOI within the given world state.

Integrated Crisis Management Middleware BB

Integration level: Todo for V2

ICMM **BB** provides the platform for integration and management of the information from different sources, e.g. simulations, indicators and the **world state** **model**.

Agent Oriented Simulation Models

Integration level: Integrated

Agent Oriented **Simulation** Models **BB** provides an agent-based interaction and visualisation platform for the models used in the **Reference Application**.

Preparedness Plan BB

Integration level: Integrated

The **Preparedness** Plan **BB** enables the end-user to create Preparedness Plans based on the analysis of risks, threats and vulnerabilities. Preparedness Plans can be created and maintained continuously as part of the strategic planning phase but also during the operational use. In the **emergency** **situation** the Preparedness Plan BB acts as an decision support tool, proposing actions to be taken to mitigate the situation, as described by the end-user following the pre-defined narrative. Preparedness Plan BB utilizes the **resource** information stored into the **World State** Repository. Preparedness Planning Views and Preparedness Plan Excecution View are the web-based UIs of the Preparedness Plan BB

VTT House model

Integration level: Integrated

VTT House **Model** calculates the cooling gradient of the residental building according to given weather model (temperature, wind) and the building information (heating type, insulation, building type).

Economic impacts model

Integration level: Integrated

The Economic Impacts **Evaluation** **Model** is used in this **application** to provide economical factors to decisions between different actions to be taken. For example, an decision whether to evacuate the residents of the **affected area** or to repair the **infrastructure** (electricity, roads) to maintain the area habitable may be taken using an calculated economic impacts as an input.

Worldstate View

Integration level: Todo for V2

Worldstate View is used to visualise and to manipulate the information related to the worldstates and worldstate transitions

Economic impacts analysis view

Integration level: Integrated

The **Economic Impact** Calculation View is used to input the data used in calculations, to initiate the calculation processa and to present the calculated results to the **end user**, to be used to support decisions related to **response** tasks and their priorities.

Scenario Analysis and Comparison View

Integration level: Todo for V2

The **Scenario** Analyisis and Comparison Support View is used to visualise the **KPI** data to the **end user** in order to support the decisions related to **response** tasks and their priorities.

OOI Management View

Integration level: Integrated

This view is used by administrators to define the types of OOIs that are used within the **Reference Application**.

Evacuation Resources Simulation Model

Integration level: Todo for V2

This **model** is used to calculate the **impact** of the **evacuation** resources allocated to mitigate the **situation**.

It calculates the progress of the evacuation and **resource** usage between **world** states.

Documentation:

Specifications

- Pilot A Scenario **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP52/01%20work%20in%20progress/CRISMA%20Pilot%20A%20scenario_english.docx**

Downloads:

Team:

Ari Kosonen

Development roles: architecture, requirements, specifications

Maintaining the **high-level architecture** of the **reference application**.

Requirements and specifications related to **Preparedness** Planner.

Jussi Yliaho

Development roles: modelling

Markus Jähi

Development roles: modelling

Kalev Rannat

Development roles: modelling

Antti Järvinen

Development roles: development

Technical lead; development of the **Preparedness** Planner

Data:

Weather data

Weather data concisting of temperature and wind strength values. Temporal resolution of the data is typically few hours and geographical resolution is one geocell (e.g. 1 km x 1 km) or larger.

Population data

Population data concists of population categorization of the people residing within a given geocell in each **world state**.

Building data

Building data concists of categorization of residental buildings within a given geocell in each **world state**.

Electric grid topology data

Topology of the electric grid, in resolution of geocells (i.e. 1 km x 1 km squares). A disturbance in a given point of the electric grid is assumed to have an **impact** to the whole geocells which the failing point serves.

Resource data

Resource data concists of information about resources that potentially can be used to mitigate the **crisis** **situation**.
For each resource unit, following information, for example, is provided:
- **Capability**. which the resource unit can provide (e.g. transportation of 50 persons)
- Default location
- Status (e.g. availability and related constrains)
- Cost data

Resource data may originate from external resource databases or **it** can be added and edited by the user.

Reference Application for the Resource Management Training Support

Authors: **Denis Havlik**
Oren Deri
Chaim Rafalowski
Tomer Kaplan
Peter Kutschera
Manuel Warum
Kalev Rannat

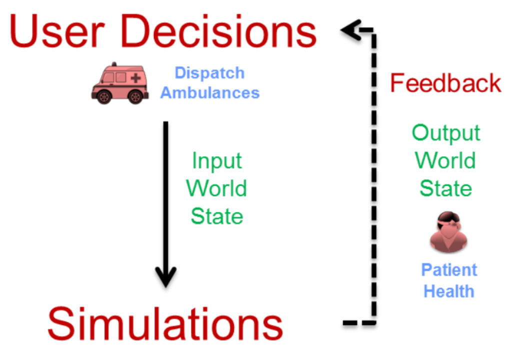
This **reference application** showcases the use of **Mashup platform** + widgets, Indicators and analysis services, Agents platform and **World State** repository in teaching of the **crisis** managers and assessing their knowledge. **It** demonstrates how interaction of users with a software **system** where various resources and population are represented by **agent** models can be used to assess and improve the reasoning of the decision makers in simulated (desktop training) **emergency** situations. The "educational" use of the application relies on the principle of trying out different strategies for resolving of a well-defined crisis and comparing the outcomes.

in **Crisis Management** (deskop) Training Scenarios and provides the technological platform for the Israeli **CRISMA** **validation** application (**Pilot** C).

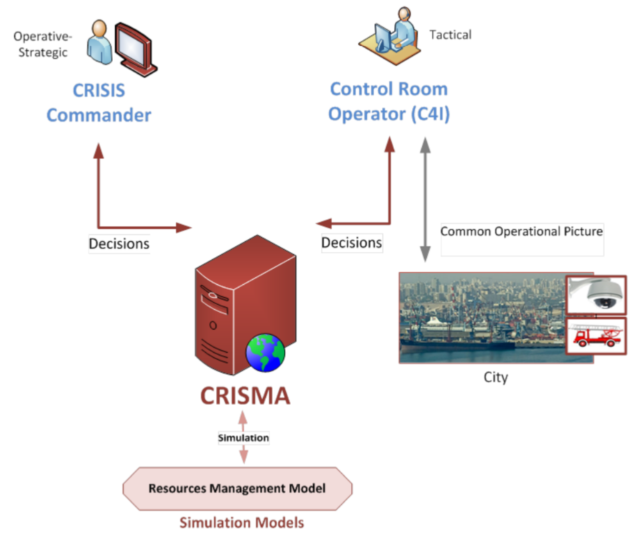
Goals: Using models and simulations to improve **decision making** **capability** of the crisis managers

Stakeholders : Crisis Managers at operative, strategic and tactical levels.

Storyboard



This application use Pilot C storyboard. Pilot C deals with the management of casualties following an accidental pollution in an urban setting. In order to create a "real" environment to play with, the terrain of the city of Ashdod will be used, though the demographic data to be used will be fictional. The pilot uses a case a chemical dispersal that causes casualties, as the story board, to be used to train **incident** commanders in decision making during a **disaster**.



As CRISMA tools calculate the evolution of the **situation** (in this case – medical condition of the patients) based on the environmental conditions (concentration of the contaminant), the time elapsed, the resources acting on the scene, and base line condition of the victims, it represents the commanders with a "result" of their previous decisions. The playback functionality of CRISMA allows the commanders to go back in time, take another course of action, and compare the results of the different decisions taken.

Implementation:

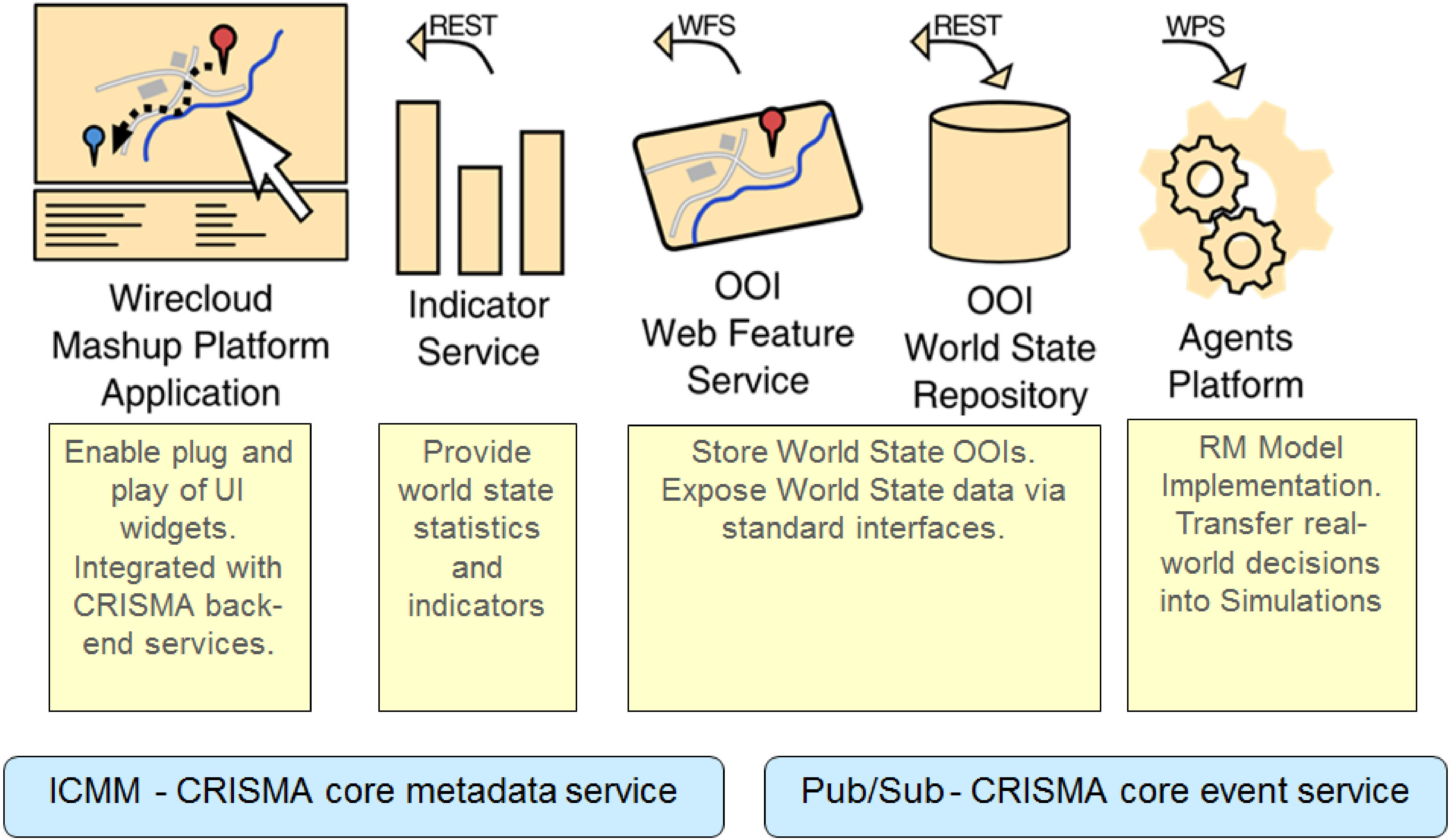
Basic Technical Concept

The main concept is to provide a unified **world state** storage that enables the UI create the initial world state and encode user commands. Then the RM **model** can decode this input world state and execute the commands while producing new time stamped world states. Consumers such as UI views, visualize the world stated to the user. Other analysis models (such as Indicators Model) can inspect the world state evolvements and provide additional insights.

The main model “World state” represents a time stamped snapshot and aggregated in a “world state” timeline that called “Simulation session” that represent a single training session.

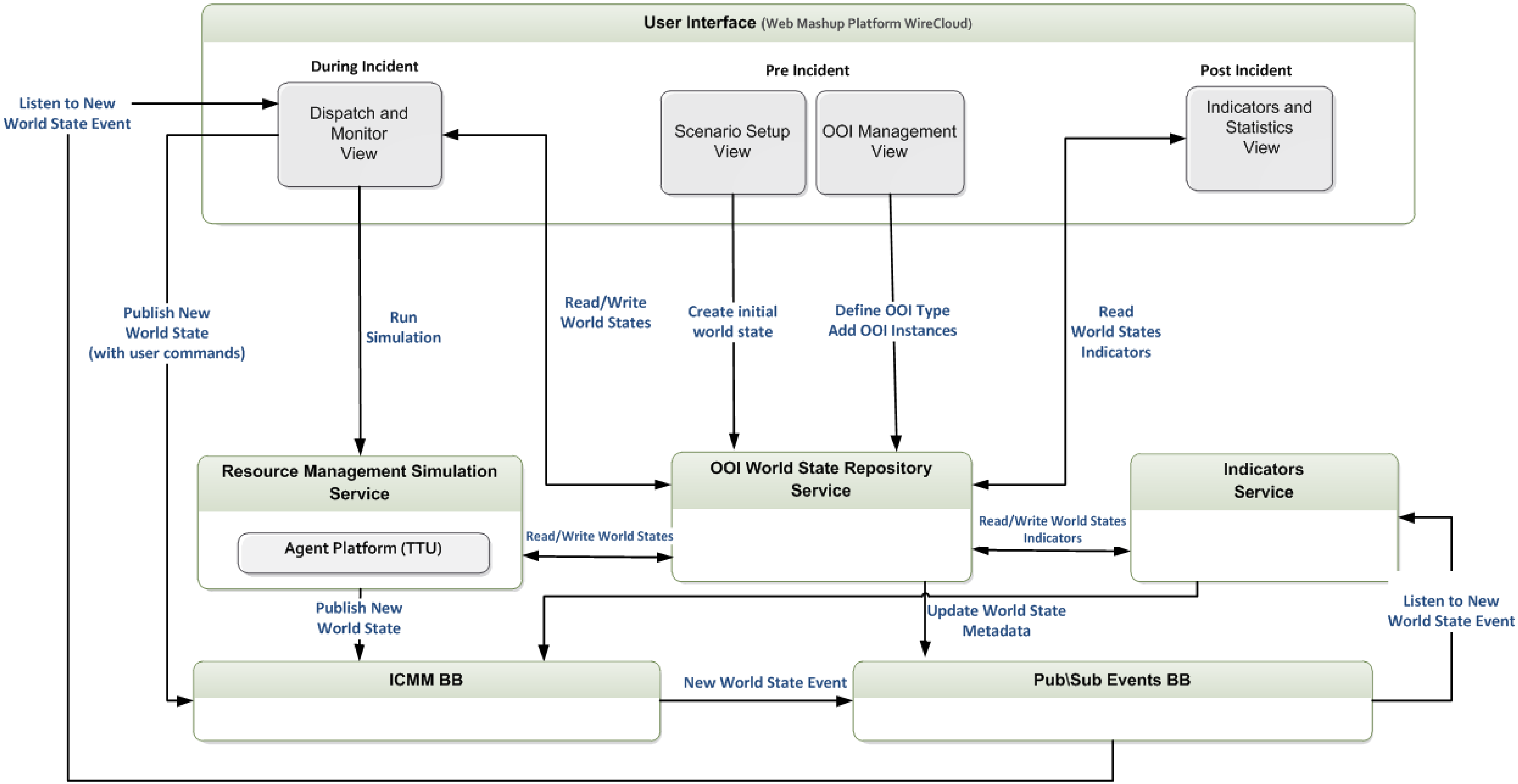
The RM **system** uses **CRISMA** **infrastructure** component, such as Pub/Sub **BB** for publishing events between components and **CCIM** in order to register the

Simulation world stated and enable external tools to access and investigate the **Simulation results**.



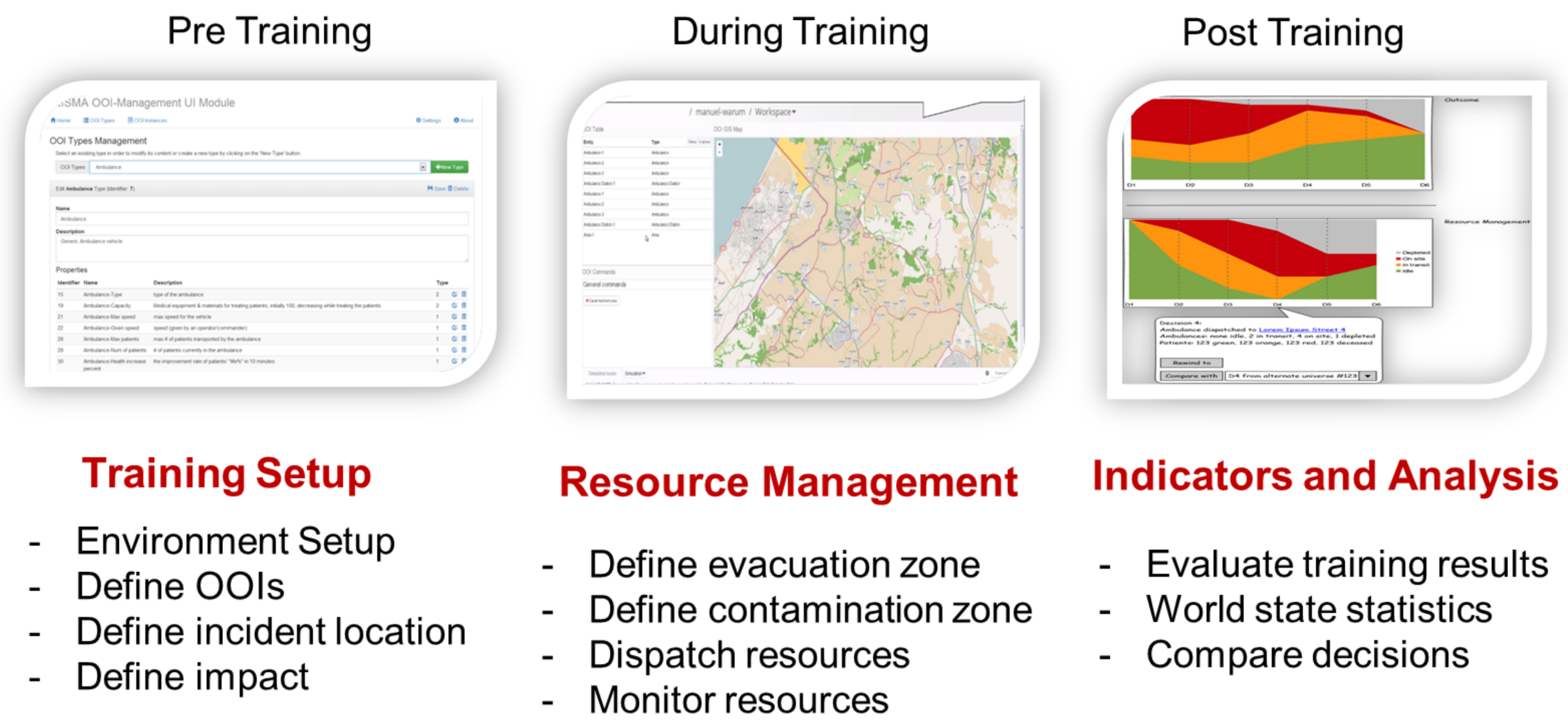
High level Architecture

The high level architecture of this **application** is shown below. Iti is important to keep in mind that the "Web nashup Platform" and the"Ressource Management Simulation **Service**" are both just containers capable of hosting various **GUI** elements (Web **Mashup Platform**) and Simulation models (Ressource Management Simulation Service). Different models used in this application are presented in the "Constituents" tab.



Main User Interface Views:

Constituents:



Agent Oriented Simulation Models

Integration level: Integrated

This **building block** is the platform on which all **OOI**-related models used in this applications are realized. **It** governs the development of all objects of interest (OOIs), as well as the interactions between them, taking into account the initial **world state** and the subsequent decisions.

UI Integration Platform BB

Integration level: Integrated

All user **interface** "views" defined in this **application** are realized as the prefab mashup applications running on top of the Wirecloud Application **Mashup Platform**, which is a reference implementation of the UI Integration Platform **BB**

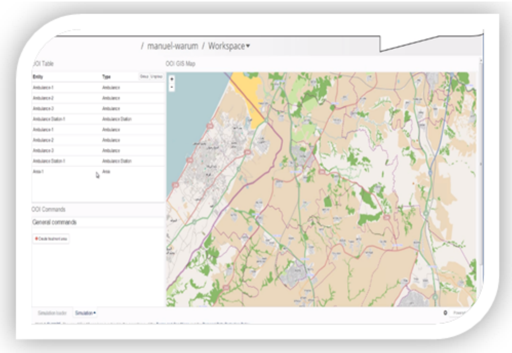
Resource Management Training Simulation Scenario Setup View

Integration level: Todo for V1

This View is used by the trainer and/or administrator to set up the training **scenario**, e.g. by choosing the training area indicating the whereabouts of the citizens, availability of the ambulances, state of the roads etc.

Resource Management Training Dispatch and Monitor View

Integration level: Todo for V1

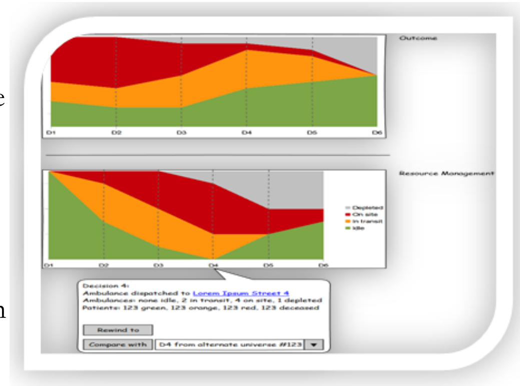


This is the main view used by the decision maker (trainee) to indicate his or her decisions during the **exercise**.

Resource Management Training Indicators and Statistics View

Integration level: Todo for V1

This view is used by the trainee, trainer and/or by the board of examiners to analyse the results of the training. **It** allows the users to visualize various indicators relevant for the **crisis management** (mainly the Key Performance Indicators, KPIs) and compare the results of different decisions.



OOI World State Repository BB

Integration level: Todo for V1

Central data repository for all OOIs used in this **application**. Currently: ambulances, citizens/patients and CDM areas. In order to simplify the **API** and decouple the **GUI** from the **model**(s), the **OOI** state is also used to encode the decisions. for instance, the "destination" **parameter** is interpreted by the model depending on the OOI type and context:

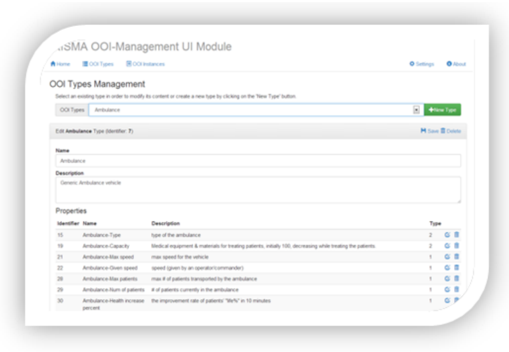
- empty ambulances drive to their destination
- ambulances with patients drive to patients destinations
- an area destination is assigned to patients after treatment.

OOI Management View

Integration level: Todo for V1

This view is used to define the types of OOIs that are used in the training **scenario**. **It** is de-facto the management **interface** for the **OOI** **World State** Repository **BB** and normally used only by programmers and/or technically savvy administrators as a pre-requisite to set-up the training. In addition, it allows this view can be used to clean up the database from old results which aren't needed any more, as well as to set up the World State details which can not be manipulated using the more user-friendly Scenario Setup View **GUI**.

Indicator Building Block



Integration level: Integrated

BasicIndicators implementation of the Indicators **Building Block** is used in this **demonstrator** to calculate some simple indicators such as the number of patients in different conditions. These results are attached to **World** states, stored in the **World State** repository **BB** as and visualized using the Indicators and Statistics View.

Integrated Crisis Management Middleware BB

Integration level: Todo for V1

ICMM provides the **CRISMA** applications with information on available (computational) resources, thus simplifying the **task** of designing other parts of the **system**.

Publish Subscribe Context Broker BB

Integration level: Todo for V1

The Pub/Sub context broker receives messages (events) from other constituents and dispatches **it** to all constituents which registered their interest. This is used to simplify the coordination of the work between the different BBs, in particular to send out the note that new **World** states are available. This type of events is for instance used by the **Indicator** **BB** to trigger the indicator calculation, and by the **simulation** models platform in order to trigger new world **transition** calculation.

Pub/sub events are also used by the **ICMM** to populate its **catalogue** of World States and simulation runs (training sessions).

Integration level: Todo for V1

This **simulation model** is realised using the TTUs agents platform and governs the movement of the OOIs of the "ambulance" type and their interaction with the patients. In this **application**, the ambulances can be ordered to head towards some destination, treat patients at the spot, and to transport then to another location. This can be subject to some conditions, e.g. "treat contaminated patients" or "transport decontaminated patients with lifeline below X% to the hospital.

Patients model

Integration level: Todo for V1

This **simulation model** is realised using the TTUs agents platform and governs the movement and lifelines of the OOIs of the "patient" (or citizen) type. In this **application**, the patients are negatively affected (poisoned) by the bromine, and positively affected by decontamination measures at the dedicated decontamination area and by other treatment measures which can be provided by the ambulances (in fact by the paramedics, but we do not explicitly model the paramedics in this demo).

In addition, the patients (citizens) can be also given the order to evacuate and then decide to do so (or not). Whether this behaviour is governed by the patients model or by a separate "**evacuation**" model is yet to be decided (TODO: update the description accordingly)

Resource Management Tactical Training BB

Integration level: Todo for V1
This **BB** represents a desktop **application** that ncorporate the **Resource Management** "views" realized as web mashups for improved user experience. **Simulation results** would be presented on the desktop application **GIS** map in order to support a realistic tactical training session.

The following integration tasks are planned:

- Develop **HTTP** **REST** **API** that support manipulating OOIs (Situator Assets).
- Develop **CRISMA** Situator Gateway that enable Real-Time Sync of **OOI** data with Situator's Assets.
- Display Models Results via **WFS**/**GML** using ESRI ArcGIS
- Develop CRISMA UI Widgets host to enable using CRISMA web application inside Situator desktop application.
- Develop **World State** Geographic viewer using the OOI-WSR WFS API that support display World State as Situator Map Layer and overlay of multiple World States.

Integrated Planning View

Integration level: Todo for V2
In the v2, this **reference application** will be extended to incorporate the planing aspect which is required by the **pilot** C. Most likely result of this action is that: (1) the German and Israeli pilots will end up sharing almost all of the code, with the exception of the actual ressoruce models; and (2) we should be able to **test** training and planing functionality on both of these pilot sites.

Documentation:

Specifications

The following document describes the software **architecture** and hardware deployment of the **Resource Management** **application**. The document is also a precursor for the **Pilot** C (Ressource Management Training in Israel) architecture description and will be extended into a fully fledged pilot **application architecture** description over time.

- Resource Management Architecture
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP54/01%20work%20in%20progress/Architecture/CRISMA_PilotC_Resource_Management_Architecture_V0.1.doc

Downloads:

TODO: add a couple of words & a link to Wirecloud mashup(s) relevant to this **application**.

Team:

Oren Deri

Development roles: architecture, coordinaton, development, integration, specifications

Coordinates the technical work of other developers involved in this **application**. Assures the specifications and developments evolve in accordance with the evolving **user requirements** and the lessons learned during development. Main **developer** of the **OOI**-RM platform and responsible for the integration of various constituents in a working application.

Chaim Rafalowski

Development roles: requirements, validation

Owner of the Israeli **pilot** and representative of the end users (**MDA** trainers and trainees).

Peter Kutschera

Development roles: configuration, development, installation, integration, support

Responsible for development and integration of **indicator** and Pub/Sub BBs

Denis Havlik

Development roles: architecture, coordinaton, mediation, specifications

Mediation between users and developers with various backgrounds. Helps the development team to find a right balance between exploring the scientifically interesting "universe of possibilities" and adopting the pragmatic solutions which can be realized within the project scope.

Kalev Rannat

Development roles: development, modelling

Responsible for development of the ressource/**OOI** models and their realization in the agents platform.

Manuel Warum

Development roles: development, integration

Responsible for development of the widgets and mashup applications.

Tomer Kaplan

Development roles: requirements, testing, validation

Second representative of the end users. Main **MDA** contributor in the second half of the project.

Data:

Background maps

In this **demonstrator**, the open streetmap maps are used only to put the training in a context familiar to trainees. Open streetmap could be easily replaced by a different map **service** and does not affect the ressource (**OOI**) models.

Patient

Patients are represented by Patient-type OOIs in frames of **Agent**-Oriented **Simulation Model** (AOSM) designed for this **reference application**. As all OOIs, the Patient-**OOI** **it** is described by its unique identifier, properties and the values assigned to these properties at a certain time instance (or **World State**).

The most important property is the Patient-life. The Patient-life (Life %) represents the Patient's life condition or "life-level". A healthly patient has a Patient-life=100% and the life-level changes as a function of time, patients category (children/adults/elderly), external conditions (level of bromine in the air), pre-conditions (previous bromine contamination) as well as on the help received from paramedics.

Identifier	Name	Description	Type

Table: OOI type Patient and description of the properties.

In our reference application, the patients are passive entities (though human beings with life characteristics, in the model they do not communicate, make decisions or move by their own will). Concrete figures governing the Patient-life change as a function of above mentined factors have been postulated by **MDA**, based on their experience with the real patients. They can be easily changed, e.g. to model patient-poisoning with different substances.

Once the patient is poisoned, ira life conditions will detorriate in time and without help some of the Patients will die. However, the rate of detoriation will slow down if the patient is evacuated from the danger zone. Patient's health can improve as a result of interaction between a medical team (a **capacity** given for special OOIs type Vehicle) and a Patient, and the lightly poisoned patients can even recover without help (in a healthly environment).

Ambulance

Ambulance is the most-important **OOI** type in our **simulation**. Ambulance is a special type of Vehicle which can move and interact with the patients. The actions an ambulance can perform are:

1. **Move**. The rules for moving are the same as for the other vehicles.

2. **Interract** with the patient. Ambulances can perform pre-triage, triage and treat the patients on the field
3. **Evacuate** the patients to the hospital

The "Ambulance" thus represents not only a vehicle but also the medical personell which has been assigned to this vehicle. Obvious shortcoming of this **model** is that ambulances can't simply bring some paramedics to the site of **accident** and then leave without them, but according to **DRK** and **MDA**, this is not a common practice.

All properties specific to an an ambulance are summarized in a table below. Remaining Vehicle OOIs

Identifier	**Name**	**Description**	**Type**
TODO	dd	dd	dd
		dd	dd
dd	dd		

Table: Ambulance OOI properties

The Ambulance-life property has a different meaning than the patient-life. **It** represents the ability of the Ambulance to treat patients. The Ambulance-life diminishes with every action and eventually the Ambulance has to return to the station for re-filling.

Further ambulance-specific properties govern the ability of the ambulance to treat patients (treatment type and time required for treatment), as well as the transport **capacity** (number of patients that can be simultaneously treated.

Strategic Areas

Strategic Areas are special types of OOIs which are characterised by their type and position.

Area types supported in this **application** are: **Evacuation**, Triage, Treatment, Ambulance Station and Hospital. These areas (or zones) do not have an active **role**. They are simply used as a way to simplify the way to issue the commands and facilitate setting up of the **crisis management** workflows.

Areas of the type "Ambulance station" and "Hospital" have a special role in the application. First, they are set-up by the trainer and can't be moved by the trainees.

Second the ambulance station can be used to re-fill the Ambulance-life, and the patients which arive in a Hospital (alive) are considered "safe" independently on their previous condition and effectively removed from further **simulation**.

Obvious shortcoming of this **model** is that **it** assumes the hospital is capable of instanteneous healing and can't be overwhelmed by a large number of new patients. For use cases where hospital-congestion is likely, hospitals would have to be modelled similarly to a combination of a pasive strategic area and a very potent but unmovable Ambulance.

Toxic plume

Toxic Plume is a special **OOI** type representing the danger area. **It** is characterised by toxic substance type and concentration. Concentration is a function of space and time, since the plume moves and diffuses over time.

Traffic congestion area

Not sure if we have this... TODO

4. Building Blocks and Models

CRISMA Building Blocks are abstract specifications of the functional elements of the **CRISMA Framework**. Actual implementations of the Building Blocks are presented under "Components"

Building Blocks are generic, composable, adaptable as well as domain- and location-independent and thus transferable to different **crisis management** domains. Building Blocks come in three flavours: **Infrastructure**, Integration and User Interaction Building Blocks.

Since this book primarily targets the software developers and integrators, CRISMA "Models" are presented as a special type of Building Blocks in section 4.4. Like (other) Building Blocks, the Models are also abstract specifications which need to be realized by one or more software Components and integrated in CRISMA applications.

It is however important to keep in mind that Models are inherently more **application**-specific and in many cases also far more complex than other Building Blocks. Accurate models of e.g. earthquake, weather or inundation are complex and seldom developed from scratch. In fact, the modellers often refer to a time-consuming **task** of correctly parametrizing an existing piece of software, assuring it is fed with appropriate data and testing the validity of the results with **test** data as "**model** development".

From the software **developer** or integrator point of view, the challenge therefore often lies in the need to assure that some ancient piece of software can be parametrized, fed with data and the results accessed within the new applications. In CRISMA Framework, the task of integrating such software is supported by the Building Blocks of the "Integration" type.

4.1 Infrastructure

CRISMA building blocks can be roughly separated into BBs providing the basic **framework** functionality ("**infrastructure**"), the BBs simplifying the integration of the legacy data and software ("Integration") and the BBs providing the **GUI** elements for the CRISMA applications. This section groups all CRISMA BBs of the "infrastructure" type.

Integrated Crisis Management Middleware BB

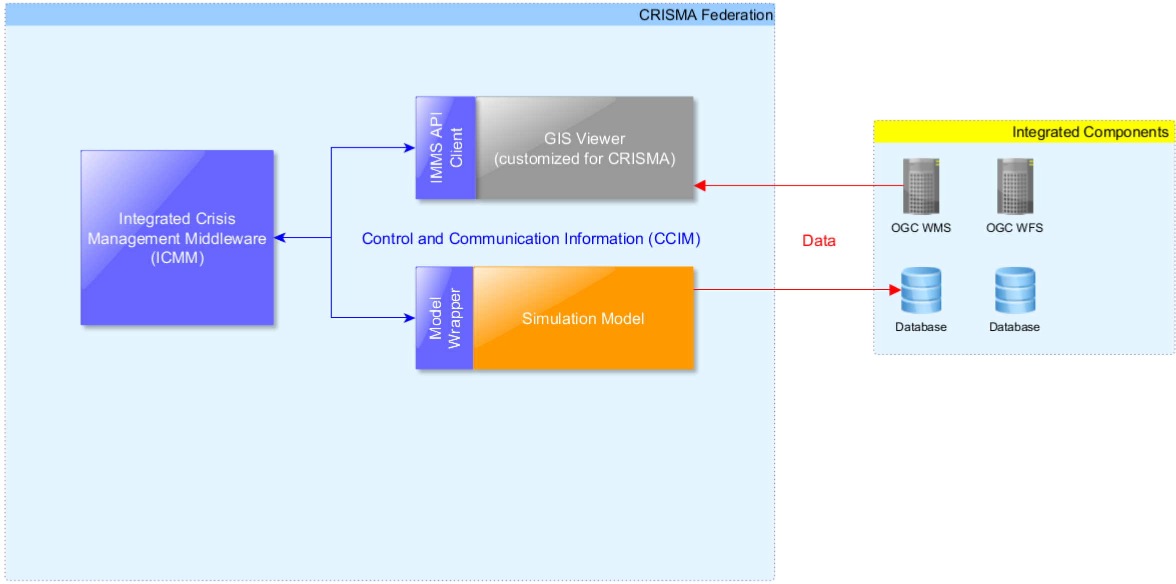
BB Description:

The **ICMM** is a central **Building Block** in every **CRISMA Application**. **It** connects **Crisis Management** Simulations with the Analysis and Decision Support functionality of CRISMA by providing a central repository for harmonized **world state** and **indicator** information. The ICMM is fed by simulations providing the basic information to be used for world state analysis and decision support Building Blocks. On an ICT conceptual level the ICMM is a generic distributed **resource**-oriented Control and Communication Information Management **System**. Thereby it is important to note, that the term 'resource-oriented' in the ICMM refers to the **concept** of generic resources as used in the context of the Resource Oriented **Architecture** (ROA).

An ICMM 'resource' may represent Control and Communication Information on simulations, world states, simulations cases, etc. Therefore a clear distinction between the management of Objects of Interest which represent abstractions of real world phenomena (e.g. **capacity** resources) and the management of 'virtual' Control and Communication Information entities has to be made. Of course, the ICMM may also provide and manage Control and Communication Information about objects of interest (world state data) which are in turn stored and managed by the dedicated **Object of Interest** Worldstate Repository (**OOI**-WSR) Building Block. To avoid ambiguities with the term 'resource' we call 'resources' managed by the ICMM 'Control and Communication Information entities' or just 'entities'.

Separation of Control and Communication Information and Data Flows

The ICMM provides a real time publish-subscribe Control and Communication Information bus that connects CRISMA Federates in a uniform simple and transparent way. It exposes several **RPC** based **HTTP** APIs that follow basic **REST** principles. In a **CRISMA Federation**, data may be exchanged directly between federates (peer to peer) while the exchange of Control and Communication Information (information about **simulation** cases, control flows, events, etc.) has to be performed by using the APIs provided by the ICMM.



Interaction between the ICMM and other Building Bocks and Integrated Components

To facilitate **interoperability** between different Federates and even Federations, the ICMM has to define a lightweight, common, minimal and generic **CCIM** while each CRISMA Federation can define its individual extension. The common CRISMA Core CCIM conceptualises the shared commonalities of all CRISMA Reference Applications that can be described by Control and Communication Information. The Core CCIM defined by the ICMM and exposed through the ICCM APIs is based on the **Conceptual Business Logic** of CRISMA and thus describes the world in a crisis management simulation in terms of Worldstates and Worldstate Transitions

Core CCIM of the CRISMA Framework

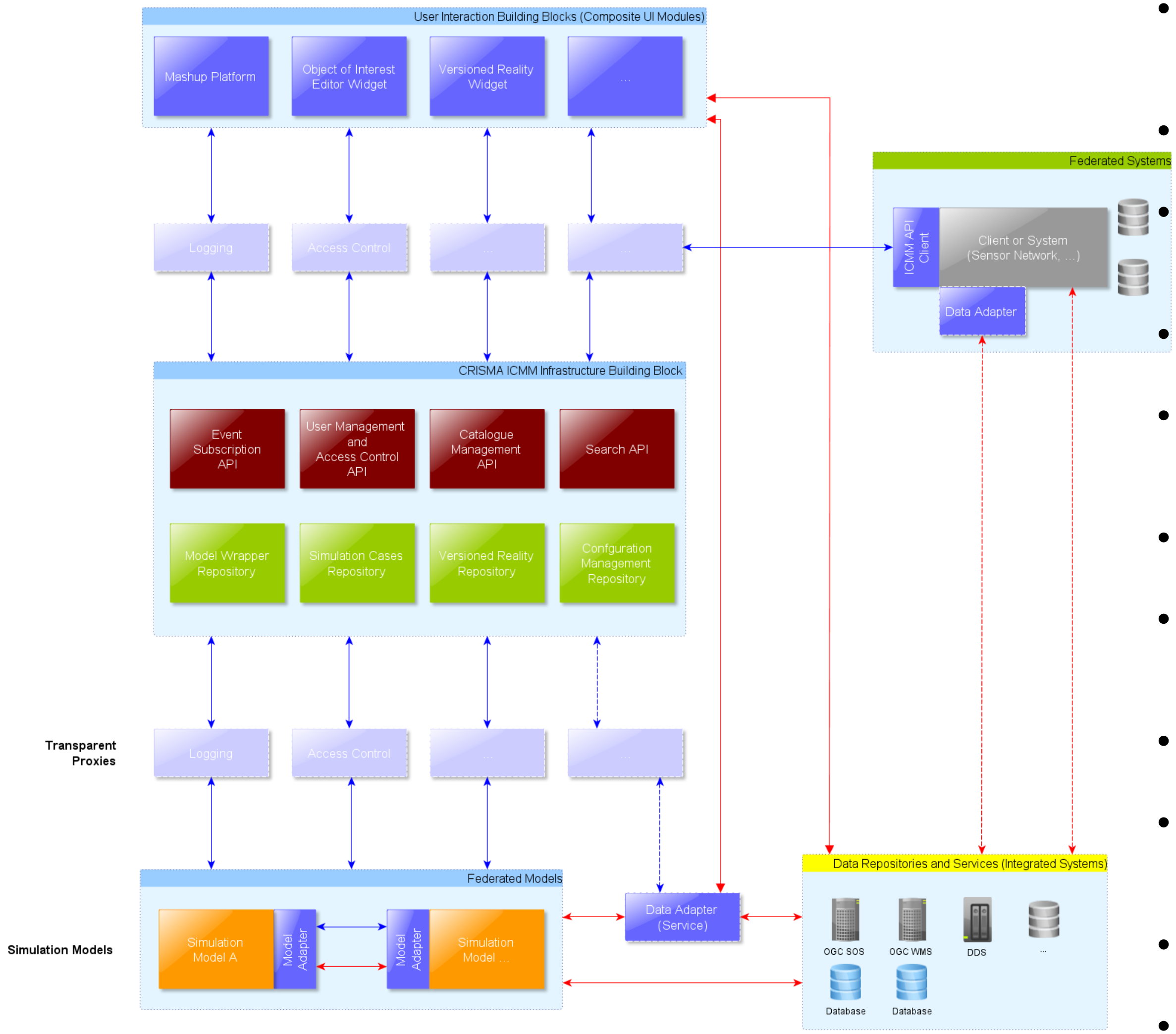
The core CCIM is the basis for all application specific information models (Application CCIMs). The Core CCIM defines the information classes of the conceptual business logic of CRISMA on abstract and generic level. Well defined extension points for Application CCIMs have been included.. Interoperability within Federations is realised with help of the federation-specific extensions, interoperability between Federations with help of the common Core CCIM. The agreement on this minimal yet exhaustive Core CCIM enables the development of generic Software Components that can be reused in any CRISMA Application.

Rationale

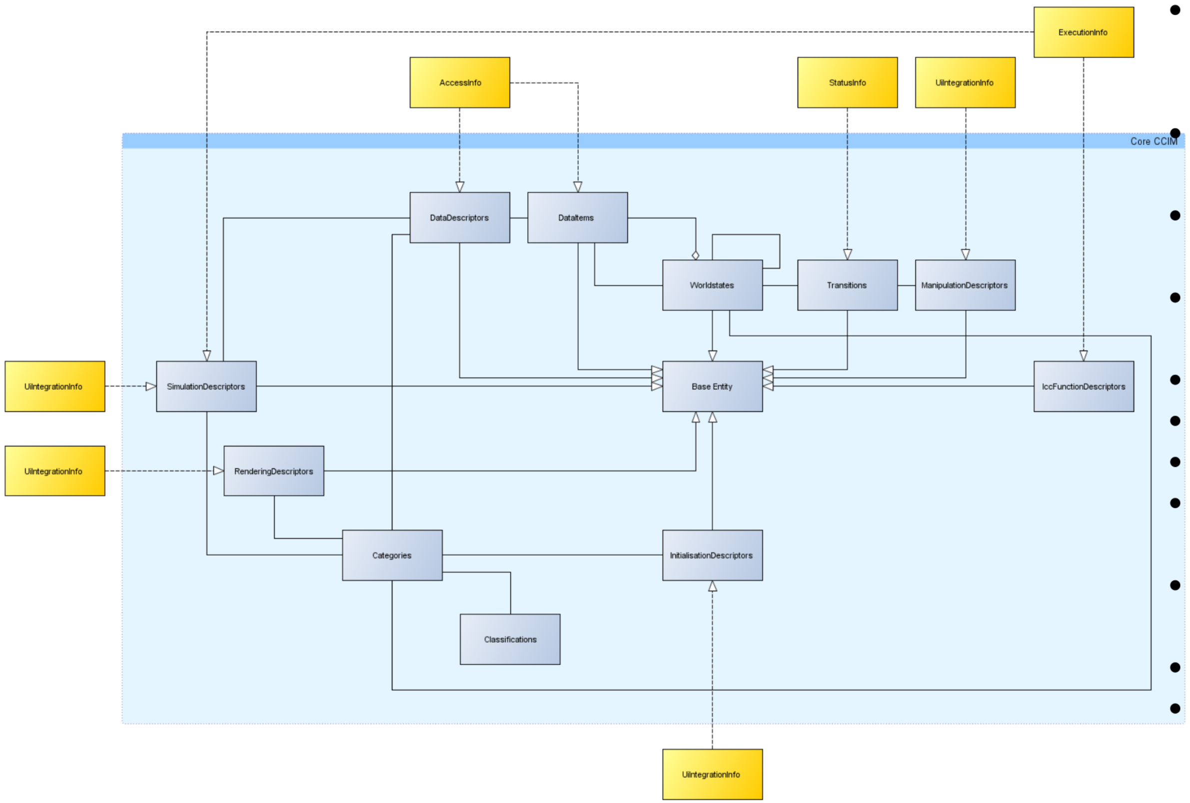
On the one hand, Federates must be loosely coupled to make integration of legacy systems (e.g. training platform, **pilot** legacy applications, …) as easy as possible and other hand, the **ICMS** must provide sufficient means to keep the effort for building new CRISMA Applications (Pilot Applications) as low as possible. **Meta-information** (Control and Communication Information) based approaches have proven to be suitable for non-intrusive integration of heterogeneous software systems.

Key Functionality

- The ICMM shall support any kind of CCIM entities, including for example worldstates, simulations, worldstate data, etc.
- The ICMM REST **API** shall allow the management of CCIM entities in a uniform, transparent and user-friendly way regardless of the concrete **entity** type (e.g. the same basic CRUD operations for worldstate entities, simulation entities, ...).
- The ICMM shall provide an easy to use and easy to understand web-based HTTP RESTful API.



- The ICMM shall support mechanisms to subscribe to CCIM entities (e.g. change of entity state (simulation started, **model** result available, ...) managed by the ICMM.
- The ICMM shall specify a client-side API to propagate the events to subscribed clients (push notifications or active **event** propagation).
- The ICMM shall specific and server-side API and/or mechanisms to let clients request status changes (pull notifications or passive event propagation).
- The ICMM shall be able to save and restore the actual state of a CCIM entity or a collection of meta-information entities (versioning).
- The ICMM shall allow to build modularised and distributed CCIMs (e.g. one general base model and several Federation (pilot) specific specialized information models).
- The ICMM shall support links between CCIM entities and aggregations of CCIM entities.
- The ICMM shall support generalisation and specialisation relationships (**interface** inheritance) of CCIMs including cross domain relationships.
- The ICMM shall support a (lifetime) unique reference for identifying (URN) and accessing (URL) a CCIM entity
- The ICMM shall support complex CCIM entities that are made of references to other entities (thus, references can be resolved on demand).
- The ICMM shall support discovery of CCIM entities (residing in distributed repositories), including spatial discovery.
- The ICMM shall support the creation of catalogues, possibly spanning distributes CCIM repositories.
- The ICMM shall provide secured and controlled access to CCIM entities, including support for **authentication**, **authorisation**, encryption, etc.
- The ICMM shall support mechanisms to deal with the concurrent modification of CCIM entities.
- The ICMM shall allow decoupling of **CCI** from actual data (e.g. objects of interest, model i/o).
- The ICMM shall leverage transparent access to decentralised stored data (**DDS**, **WMS**, **RDMBS**, document oriented) by providing CCI for data access.
- The ICMM shall be realised as distributed system (no single server) where various distributed CCIM repositories (based on different modularised CCIMs) can be created.



- The ICMM should support the creation/import of CCI from different data sources to establish a unified view and on heterogeneous data.
- The ICMM shall provide a CCIM for the management of world states.
- The ICMM shall send a mail to the system administrator to inform of a new registration to approve.
- The ICMM shall allow the administrator to visualize the registered users and their associated rights.
- The ICMM shall allow the administrator to edit user groups.
- The ICMM shall allow the administrator to register new user rights.
- The ICMM shall display a registration form for new users.
- The ICMM shall allow the already registered users to log in into the system
- The ICMM shall allow the user to log out and remove the user account.
- The ICMM shall display to an user its profile.
- The ICMM shall display a page that allows the user to register or modify its personal information.

Realisation

The **cismet** cids System implements some of the features of the ICMM, especially the modular **Control and Communication Information model**. It is **open source** and can be extended as required. Additionally, it provides powerful administration tools for Control and Communication Information management and import (Visual Information Modelling and **ETL** Tools). The screen shot below shows the server management console of the ICMM that is implemented on basis of the cids server software.

cids Server Console

Documentation:

Functional Description

Functional Description of the **ICMM**. Located on the **VTT** WS.

- ICMM Functional Description

https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20Functional%20Building%20Blocks/v2.0/Integrated%20Crisis%20Management%20Middleware.doc

User Guide

Documentation of the Core **Control and Communication Information Model** of the **CRISMA Framework**. Located on the **VTT** WS.

- Core CCIM Documentation
  http://goo.gl/ruXmyv (http://goo.gl/ruXmyv)

User Guide

UML Sources of the Core **Control and Communication Information Model** of the **CRISMA Framework** in XMI format. Located on the **VTT** WS.

- Core CCIM UML (XMI)

https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20CCIMs/Core%20CCIM.xmi.zip

Specifications

- D32.2 ICMS Architecture Document V2 **http://www.crismaproject.eu/deliverables/CRISMA_D322_public.pdf (http://www.crismaproject.eu/deliverables/CRISMA_D322_public.pdf)**

Realization:

cids System

The **cismet** cids **System** implements some of the features of the **Integrated Crisis Management Middleware** **Infrastructure** **Building Block**, especially the modular **Control and Communication Information model**. **It** is **open source** and can be extended as required. Additionally, it provides powerful administration tools for Control and Communication Information management and import (Visual Information Modelling and **ETL** Tools).

Related Models and BBs:

Cascade Events Configuration and Interaction View

The **Cascade Effects** View is a User interaction Building that allows a user to configure and run a Cascade Effects **Scenario**. The user can select a triggering **event** (for example, an earthquake) and provide may either specify the characterisation of the event (**Simulation Control Parameter**) and thus initiate a new **Simulation Model** Run for this particular event, or select (if available) the output of a past event or an event already simulated.

Data Integration

The Data Integration **Building Block** provides components that can be used to easily serve data in a **CRISMA**-compliant (**OGC** open **standard** compatible) way so that other Building Blocks may use them for further processing like viewing or editing. That way data can be made accessible for CRISMA components.

Indicator Building Block

The Indicators **Building Block** (formerly Algebraic **Evaluation**) is a component that allows definition, storage and evaluation of “simple” algebraic models in order to evaluate consequences of decisions made by **CRISMA** users during a training session. In this context “simple” means that there is no large number crunching needed. Usage is for calculation of key performance indicators, heuristic models and to some extend the implementation of models actually running in interactive **GIS** environments in order to make them usable as services in the **CRISMA framework**.

Multi Criteria Analysis and Decision Support View

The Multi **Criteria** Analysis View and Decision Support View is a User Interaction **Building Block** that allows performing a ranking of different **Crisis Management** Scenarios with respect to specific Criteria.

Publish Subscribe Context Broker BB

The Publish Subscribe Context Broker **Building Block** is a cross-over between an **event** broker which accepts events and dispatches them to subscribers and an access **service** providing the information on current state of the "world". Thereby, the Publish Subscribe Context Broker Building Block can be used to realize the event subscription and event delegation functionality of the **ICMM**. Thus, the Building Block is both suitable for dispatching events related to Control and Communication Information (**CCI**) managed by the ICMM and events related to (Worldstate) Data managed by arbitrary data access services, e.g. the **OOI** **World State** Repository **BB** or implementations of the Data Integration BB (**WMS**, **WFS**, …).

Scenario Analysis and Comparison View

The **Scenario** Analysis and Comparison View is able to visualise **Indicator** and **Criteria** data in a way that users are able to analyse and compare different Simulated **Crisis Management** Scenarios and ultimately come to a decision which fits the **simulation objective** best for a specific **Simulation Case**.

Simulation Model Integration BB

The **Simulation Model** Integration **Building Block** provides components that can be used to easily enable simulation models to participate in a **CRISMA Application**. Due to the heterogeneity of existing simulation models CRISMA has to ensure that they can be integrated in a standardized way and ultimately become a **CRISMA Federate**.

That way they are made CRISMA-aware. The envisioned technique to be used is that of so called wrapping so that this Building Block provides a piece of software that can be used to make (existing) simulation models CRISMA-aware with as little effort as possible with respect to the individualities of the different simulation models.

Worldstate View

The Worldstate View **Building Block** is a set of generic Widgets (Mashable Composite UI Modules) that allow to visualise Control and Communication Information

(**CCIM**) related to Worldstates and Worldstate Transitions. Thus, **it** operates mainly on common **meta-information** about the **world** rather than the real data of the world (**World State** Data Slots).

OOI World State Repository BB

Authors: **Oren Deri**
Denis Havlik
Peter Kutschera

BB Description:

OOI-WSR is a **Resource Management** related **Building Block** that enables archiving, querying and manipulation of Objects of Interest (OOI) **world state** data. This **BB** serves as a Repository **service** for OOI data that can be consumed or manipulated by other interaction or functional building blocks and resource management models.

OOI-WSR handels the following data models:

- OOI – **Entity** (**object**) that manipulated by the user or automatically by some resource management mathematical **model**. Contains a flat list of properties, that each construct from one of the following types: Number, Text, Date and Time, Geography and Geometry.
- **Simulation** Session – Represent a training or planning simulation session that initiated manually by the user. **It** usually involved defining the initial world state and then applies user decisions, via running resource management models or manually updates the OOI state. Each OOI state must be associated to specific world state instance.
- World State – Represent a meaningful OOI data snapshot at specific time. The world state holds list of OOI instances with state that relevant to specific point in time. It usually represent the following:
    - Initial state of simulation Session
    - Output result of model execution at specific point in time.
    - Manual manipulation of user at specific point in time.
    - Final state of Simulation session

In a **CRISMA Federation**, data may be exchanged directly between federates (peer to peer) while the exchange of Control and Communication Information (information about simulation cases, control flows, events, etc.) must be performed using the APIs provided by the **ICMM**. Therefore OOI-WSR publishes all world state changes to the ICMM BB.

Rationale

The OOI-WSR supports the basic functionality of store query and manipulates OOI data that requested by Resource Management related CRISMA related BBs and federations. User Interaction BBs requires quick and simplified way to access world state information in order to visualize it via Plain text, **GIS** display Tabular display and graphs. Functional BB that Analyse **simulation results** require directs access to the data in order to generate KPIs and identify trends. CRISMA Models requires access to the initial OOI world state and update the new world state based on simulation results.

Therefore we need a standalone, loosely-coupled, OOI storage repository that enables sharing and reusing OOI archiving and query services.

Key Functionality

The need for this Building Block can be derived from the following **user requirements**:

- Provide unified and flexible way to model OOI data in a human readable format.
- Enable easy query of world state OOI versions, and serves as the default integration method between the CRISMA **CCIM** and resource management models.
- Support interaction with existing decision support and **crisis management** systems by providing a **standard** unified way to access simulation related world state OOI data.
- Provide a storage service **infrastructure** as a basis for the harmonized, standardized and user friendly integration of new and existing simulation models.
- Enable concurrent access of UI and backend clients and allows OOI world state data sharing by all parties involves in crisis management simulation.
- Support the integration of OOI data originated from external sources and construct of various types and formats.
- support the communication with existing IT tools for crisis management to address **decision making** alternatives, perform what-if scenarios, update **system** information with real time data, derive quantitative and qualitative data from existing models and introduce new features to existing simulation tools if required.
- Support the generic approach to be able to support the very different **pilot** applications as well as applications beyond in the crisis management context.

Documentation:

Functional Description

TO do

- Building Block Description Document **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20Functional%20Building%20Blocks/v2.0/OOI%20World%20State%20Repository%20Building%20Block%20Description%20V1.doc**

Demo

This video shows how to use the **OOI** **World State** Repository

Realization:

OOI World State Repository

OOI-WSR is a **Resource Management** related module that enables archiving, querying and manipulation of OOI **world state** data. This module serves as a Repository **service** for OOI data that can be consumed or manipulated by other interaction or functional building blocks and resource management models.

Related Models and BBs:

Indicator Building Block

The Indicators **Building Block** (formerly Algebraic **Evaluation**) is a component that allows definition, storage and evaluation of "simple" algebraic models in order to evaluate consequences of decisions made by **CRISMA** users during a training session. In this context "simple" means that there is no large number crunching needed. Usage is for calculation of key performance indicators, heuristic models and to some extend the implementation of models actually running in interactive **GIS** environments in order to make them usable as services in the **CRISMA framework**.

OOI Management View

This **Building Block** enables to view and edit the actual data available for specific **scenario** or **simulation**. **It** enables the **system** administrator to detect the **OOI** data type's properties and based on the OOI Information Models definitions. It also enables to monitor and edit a specific snapshot context (time, user, and workflow).

Publish Subscribe Context Broker BB

The Publish Subscribe Context Broker **Building Block** is a cross-over between an **event** broker which accepts events and dispatches them to subscribers and an access **service** providing the information on current state of the "world". Thereby, the Publish Subscribe Context Broker Building Block can be used to realize the event subscription and event delegation functionality of the **ICMM**. Thus, the Building Block is both suitable for dispatching events related to Control and Communication Information (**CCI**) managed by the ICMM and events related to (Worldstate) Data managed by arbitrary data access services, e.g. the **OOI** **World State** Repository **BB** or implementations of the Data Integration BB (**WMS**, **WFS**, …).

Resource Management Tactical Training BB

Resource Management Tactical Training **Building Block** (RMTT **BB**) simplifies the **task** of designing the Tactical Training applications for control room operator and on-scene commanders. RMTT enables a Trainee to learn **emergency management** by assigning tasks to various resources and analysing the results in a virtual environment.

Cascading Effects Model

The cascading effect **model** for dynamic **scenario** assessment calculates the probability of attainment of cascading events scenarios, given an initial triggering **event**, or estimate consequence paths given the occurrence of selected scenarios.

Publish Subscribe Context Broker BB

Authors: Hermann Huber
Denis Havlik

BB Description:

The Publish Subscribe Context Broker **Building Block** is a cross-over between an **event** broker which accepts events and dispatches them to subscribers and an access **service** providing the information on current state of the "world". Thereby, the Publish Subscribe Context Broker Building Block can be used to realize the event subscription and event delegation functionality of the **ICMM**. Thus, the Building Block is both suitable for dispatching events related to Control and Communication Information (**CCI**) managed by the ICMM and events related to (Worldstate) Data managed by arbitrary data access services, e.g. the **OOI** **World State** Repository **BB** or implementations of the Data Integration BB (**WMS**, **WFS**, …).

The Building Block works with context elements, that is structured pieces of data describing the state of a certain **entity** (in **CRISMA** case this could be e.g. objects of interest or arbitrary **CCIM** Entities like Worldstates). Each Context Element has multiple attributes, which can be set independently by incoming events. A result is an updated context (in CRISMA case this could be e.g. an ambulance changing its status from "idle" to "moving" or a new worldstate entity is added to a progressing **scenario**).

Context updates can be dispatched to "context subscribers", but a latest version of the Context element can also be accessed online using the Pub/Sub service **interface** (**REST** **API**). In order to avoid the use of obsolete context elements, each context element can be assigned with an expire-date and automatically purged from the Pub/Sub after this date.

Rationale

An event handler allows us to simplify the **application** design by circumventing (or at least minimizing) the need for direct interaction between BBs and for hard—coding such interactions. Event handling is also a central **concept** of the **Integrated Crisis Management Middleware** (ICMM). The Publish/Subscribe Context Broker BB can be used to transparently extend the ICMM with event subscription and event delegation functionality.

A Context Broker exposes the (**standard**) interfaces for retrieval of the context information, events and other data from the Context or Data/Event Producers to the Context or Data/Event Consumers. The consumer doesn't need to know where the data are located and what is the native protocol for their retrieval. **It** will just communicate to the Context Broker through a well-defined interface specifying the data it needed in a defined way: on request or on subscription basis. The Context Broker will provide the data back to the consumer when queried, in case of "on-request", or when available, in case of "on-subscription" communication mode.

Key Functionality

The expected key functionalities of the Pub/Sub can be summarised as:

- support a wide range of data and **meta-information** (see data **model** above).
- allow the management of context elements/events in a uniform, transparent and user-friendly way regardless of the concrete entity type
- provide an easy to use and easy to understand web-based API (**HTTP** REST)
- be usable by a wide variety of different clients (**resource management** **GUI**, Model Wrappers, serious gaming engine, mobile app, web browser, ...)
- support mechanisms to subscribe to events/context elements (e.g. change of entity state (**simulation** started, model result available, ...)
- allow access to valid context elements (latest version unless expired)

Related Building Blocks and Interactions

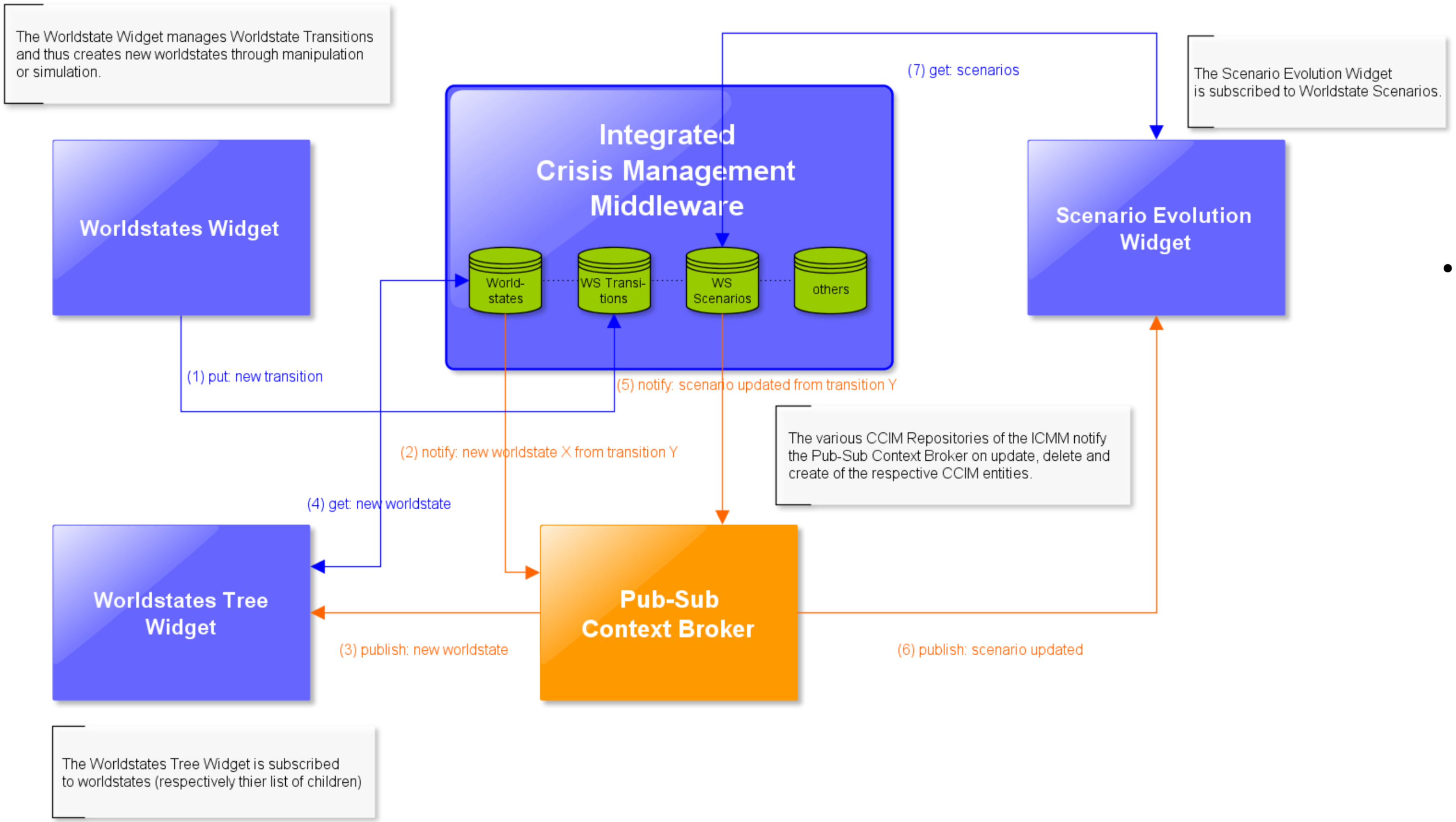
The ICMM and other Building Blocks with RESTful service interfaces that want to benefit from publish subscribe functionality interact with the Pub/Sub Context Broker; There are basically two general interaction scenarios for the usage of the Publish/Subscribe Context Broker.
The first is related to the subscription to Control and Communication Information (CCIM) and the publishing of the related events. As shown in the Figure below, the Publish/Subscribe Context Broker is used by the ICMM to delegate events to CRISMA Federates. In the example below both the **Scenario Evolution** **Widget** (a User Interaction Building Block and Composite UI Module, respectively) are subscribed to certain CCIM Entities (worldstate, scenario, …) and want to get notified once a new CCIM entity of a specific type is created, deleted or updated in the respective ICMM repository. Once a CCIM Entity is updated, the ICMM will notify the Publish/Subscribe Context Broker which in turn publishes the notification to the subscribed clients.

ICMM Event Delegation with help of the Publish/Subscribe Context Broker

The second interaction scenario which is very similar to the first one is related to worldstate data, e.g. Objects of Interests. A Data Integration Building Block serving a specific type of data, e.g. an **OGC** WFS-T providing OOI Features, notifies the Publish/Subscribe Context Broker once an OOI has been created, updated or deleted. For example, if the location property of an OOI is changed (a moving ambulance) all interested clients are notified of the new position of the OOI.

Documentation:

Specifications



Publish Subscribe Context Broker is a identical to FI-Ware ContextBroker. Full specifications are available on FI-Ware wiki

- FI-Ware ContextBroker Specifications **http://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.Data.ContextBroker (http://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.Data.ContextBroker)**

Realization:

Orion Context Broker

The Orion Broker is a reference implementation of the FI-Ware Publish/Subscribe Context Broker GE, providing the NGSI9 and NGSI10 interfaces. In **CRISMA**, **it** is used as a reference implementation of the CRISMA Publish Subscribe Context Broker **BB**. The Context Broker is used by other building blocks to exchange events and process updates.

Related Models and BBs:

Indicator Building Block

The Indicators **Building Block** (formerly Algebraic **Evaluation**) is a component that allows definition, storage and evaluation of "simple" algebraic models in order to evaluate consequences of decisions made by **CRISMA** users during a training session. In this context "simple" means that there is no large number crunching needed. Usage is for calculation of key performance indicators, heuristic models and to some extend the implementation of models actually running in interactive **GIS** environments in order to make them usable as services in the **CRISMA framework**.

Integrated Crisis Management Middleware BB

The **ICMM** is a central **Building Block** in every **CRISMA Application**. **It** connects **Crisis Management** Simulations with the Analysis and Decision Support functionality of CRISMA by providing a central repository for harmonized **world state** and **indicator** information. The ICMM is fed by simulations providing the basic information to be used for world state analysis and decision support Building Blocks. On an ICT conceptual level the ICMM is a generic distributed **resource**-oriented Control and Communication Information Management **System**. Thereby it is important to note, that the term 'resource-oriented' in the ICMM refers to the **concept** of generic resources as used in the context of the Resource Oriented **Architecture** (ROA).

OOI Management View

This **Building Block** enables to view and edit the actual data available for specific **scenario** or **simulation**. **It** enables the **system** administrator to detect the **OOI** data type's properties and based on the OOI Information Models definitions. It also enables to monitor and edit a specific snapshot context (time, user, and workflow).

OOI World State Repository BB

OOI-WSR is a **Resource Management** related **Building Block** that enables archiving, querying and manipulation of Objects of Interest (OOI) **world state** data. This **BB** serves as a Repository **service** for OOI data that can be consumed or manipulated by other interaction or functional building blocks and resource management models.

Worldstate View

The Worldstate View **Building Block** is a set of generic Widgets (Mashable Composite UI Modules) that allow to visualise Control and Communication Information (**CCIM**) related to Worldstates and Worldstate Transitions. Thus, **it** operates mainly on common **meta-information** about the **world** rather than the real data of the world (**World State** Data Slots).

4.2 Integration

Integration BBs simplify the **task** of integrating other BBs, models and legacy applications in **CRISMA** applications.

Agent Oriented Simulation Models

BB Description:

The **Agent**-Oriented **Simulation** Models **Building Block** serves for the development of dynamic maps – specific (individual-based) simulation models composed of interacting software agents situated in some environment. This Building Block comprises a collection of generic agents and interaction templates for dynamic map construction, for describing, defining and specifying points, areas and layers of interest. **It** provides furthermore a dynamic-map-based user **interface** for interaction and visualization. Thus, it can be considered both an Integration and User Interaction Building Block.

The Agent-Oriented Simulation Models Building Block extends the **OGC** **WPS** based **Model** Integration **Concept** introduced in the **CRISMA Framework Architecture** V1 (D32.1, 2013) by real-time and agent-oriented simulations. In contrast to the loosely coupled WPS based approach, the tight coupling between user interface and dynamic agents (constitution the agent-oriented **simulation model**) in this Building Block makes it possible to follow a simulation in real time and to influence a running simulation.

For decision support in **crisis management** is essential to provide good **situational awareness**, including data fusion from and dissemination to different sources. Agent-oriented simulations, e.g. agent-based simulations of the crisis evolving in some environment, include simulation model for the environment and agents' behaviours and, interaction patterns. Simulations serve for developing better understanding of situations and relationships between processes in the crisis area/aspect of interest.

The Agent-Oriented Simulation Models Building Block comprises of a set of simulation/monitoring tools and services for decision support. It consists of a collection of generic agents and interaction templates for (dynamic) map construction, for describing, defining and specifying points, areas and layers of interest. Agent-Oriented Simulation Models Building Block offers tools and generic components to describe/simulate evolution of the crisis and Points of Interest (POI) important for Crisis Management in spatiotemporal context (on a dynamic self-organizing map as an operative model of crisis or a crisis **situation**).

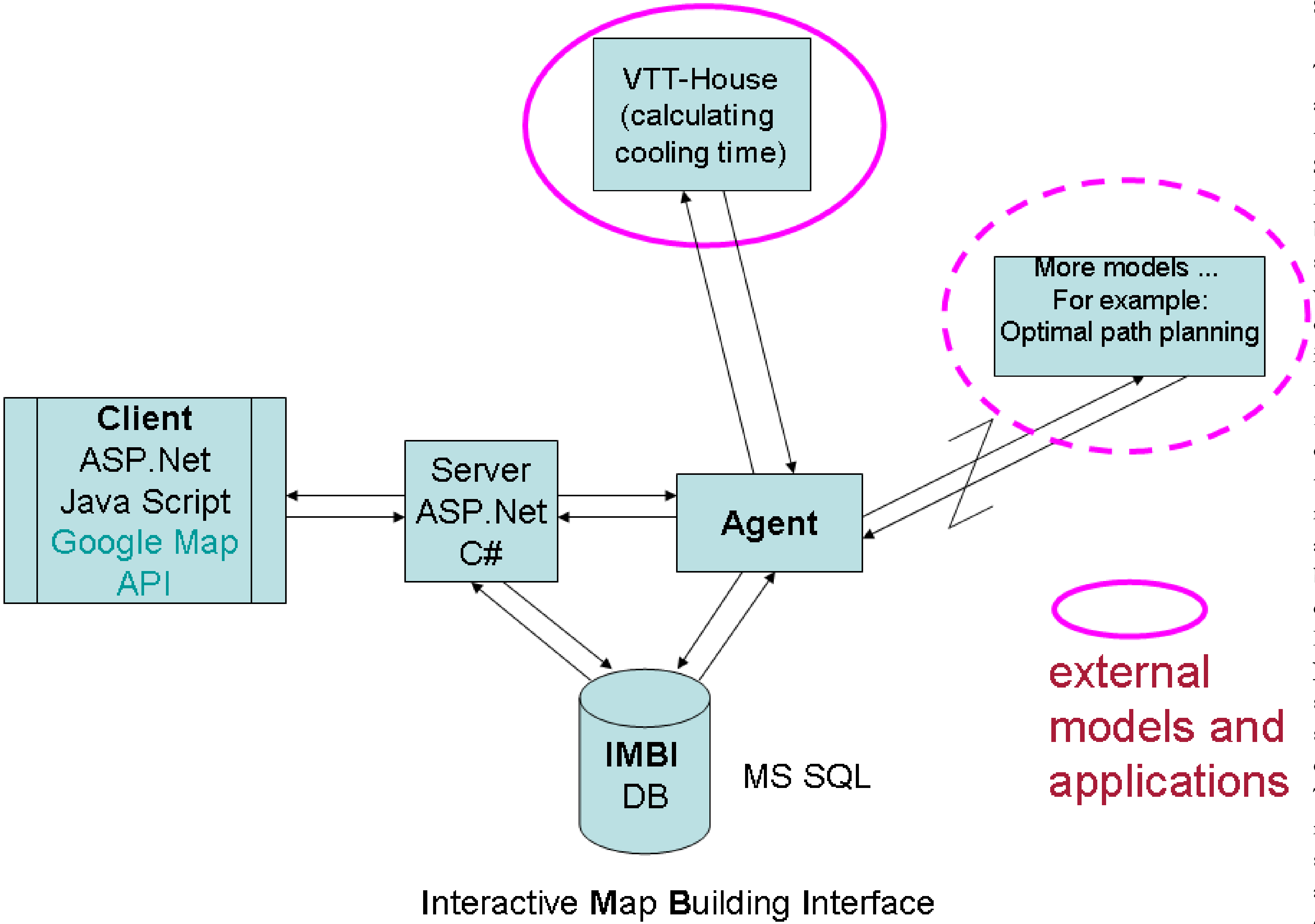


Figure 1: Example of integration of Simulation Models

The Figure 1 shows an example on how the Agent-Oriented Simulation Models Building Block can be used to integrate simulation models with other components. The illustration explains the components and functional connections between the simulation model (Agents) and an in-memory business intelligence data base (IMBI DB) at one side and **VTT**-House simulation model, some route planner, etc, at another side. The Agent acts as a mediator between server-**application** and other components. The Client is connected to the server-application and the last, in turn, with a DB and the Agent. The Agent mediates all connections, forming/constituting the IMBI DB objects with corresponding properties.

Rationale:

Applying this Building Block offers a possibility to create simulation environments. This Building Block is a convenient way to obtain better understanding of situations and complex relationships between different actors and processes in the crisis area. Additionally, the active components (software agents) in these virtual environments can monitor the sensors (or other data sources) in real crisis, analysing the data helping to evaluate continuously evolving crisis situations.

Also, the simulation environment can interact with external Crisis Management models, to capture information provided by them. For example, the user can follow, how the cooling or damage of buildings could affect the needs for **evacuation** or for supplying of material resources, what could be an optimal way to share or re-share the resources.

Offering general overview about processes and actors in the crisis, it can also serve as a convenient tool supporting resources management, especially for situations where the user can get additional and valuable information from agent's interactions and interactions with the simulation environment, analyzing situations that don't have clear analytic solution.

This Building Block is intended to be used together with **Resource Management** Building Blocks in CRISMA, to offer a platform for Agent-based Modeling and Simulations for Objects of Interest.

Key Functionality:

- Run simulations with varying data.
- Showing information resulting from the simulation **scenario** both graphically (map-based) and/or in a table format.
- Capabilities to edit/modify some parameters of the simulation.
- **Capability** to edit (create, modify and delete) OOIs.
- Capability to edit **OOI** properties' values.
- Capability to start, pause/continue and stop the simulation.
- Capability to communicate with CRISMA federates (over OOI-WSR and **ICMM**).
- Capability to communicate with CRISMA federates (over OOI-WSR and ICMM).
- Capability to communicate with CRISMA federates (over OOI-WSR and ICMM).
- Capability to communicate with CRISMA federates (over OOI-WSR and ICMM).
- Capability to model interactions between different Agents.
- Capability to model interactions between different Agents.
- Capability to model situation awareness.

- Capability to perform a **world**-state **transition** (simulation) and to configure the interval for creating world-state snapshots.
- Capability to follow a world-state **transition** in real time and to show the progressing status of an intermediate world-state.

REALISATION:

The Dynamic Map Agents Software owned by CRISMA partner **TTU** is a suitable candidate for implementing this Building Block.

Dynamic Map Agents has been used in different user-specific applications. It needs certain work before getting adapted to CRISMA-specific needs. The (generic) map agents must be constructed/developed according to certain crisis types and scenarios (either for real crisis management or training), (sub)models and auxiliary tools chosen.

Documentation:

Functional Description

Functional requirements can be found in:

- CRISMA Technology, Concepts & Technical Requirements Report V2
  http://www.crismaproject.eu/deliverables/CRISMA_D312_public.pdf (http://www.crismaproject.eu/deliverables/CRISMA_D312_public.pdf)

Realization:

Dynamic Map Agents

This is a description of a software candidate that offers tools for **agent**-based simulations for the implementation of a functional **Building Block** (Agent-Oriented **Simulation** Models Building Block), together with **OOI** **World State** Storage **Service** (**WFS**) **BB** (**NICE**), **Resource management** integration and deployment (**WPS**) BB (CASS), Indicators BB (**AIT**), Tactical RM Training BB (Situator integration) (**NICE**) and RM Training Dispatch and Monitor View (**AIT**), RM Training Simulation **Scenario** Setup View (AIT), RM Training Indicators and Statistic View (AIT).

Related Models and BBs:

Patients model

This **model** governs the behaviour of the patients in Israeli and German pilots. Full description shall be provided by **TTU**.

VTT House model

Model for estimating the extreme cold weather related **vulnerability** curves for buildings.

Resource Allocation Tactic Model

This **model** replicates the decision making process of **crisis** managers in **resource management** planning applications. **It** allows us to **test** the planning scenarios with different management strategies, e.g. "prioritize transport to hospital" or "prioritize treatment on the field". It realizes a second layer above the Ambulance Model in order to being capable of running simulations with predefined strategy settings. These different strategy settings for resource planning allow users determining the best strategy in a specific **scenario**. Depending on the strategy ambulances in an idle state are assigned to certain commands before rerunning the ambulance model **simulation**.

Data Integration

BB Description:

The Data Integration **Building Block** provides components that can be used to easily serve data in a **CRISMA**-compliant (**OGC** open **standard** compatible) way so that other Building Blocks may use them for further processing like viewing or editing. That way data can be made accessible for CRISMA components.

The envisioned technique to make data available is to use standardized OGC services:

- **WMS** - delivers raster maps
- **WFS** and WFS-T - mainly vector data
- **WCS** - raster data
- WCPS - raster data with processing functionality (optional)
- **SOS** and SOS-T - time series data (optional)

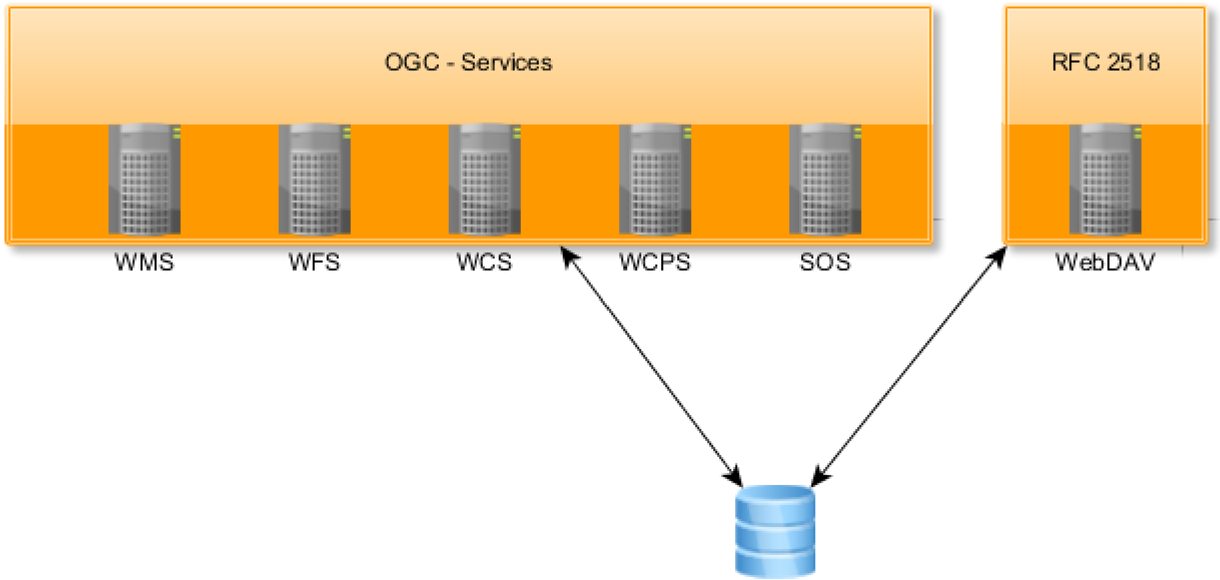
There might the necessarily to access data that can’t be mapped to the OGC services. In this case **WebDAV** shall be used.

Rationale

Data used in CRISMA has to be made available in a uniform way so that CRISMA components may access **it** standardized. Also CRISMA produces data that has to be stored.

Key Functionality

The key functionalities of the Data Integration Building Block are defined by the following functional requirements.



- The Data Integration Building Block shall provide **API** functions that deliver descriptions of the respective data it can provide.
- The description of the data provided by the Data Integration Building Block shall contain machine-processable **meta-information** about result data (e.g. OGC Capabilities Document). This includes information on data formats, possible query mechanisms, etc.
- The Data Integration Building Block shall be able to make data available in a CRISMA-compliant way using standardized OGC WMS interfaces
- The Data Integration Building Block shall be able to make data available in a CRISMA-compliant way using standardized OGC WFS interfaces
- The Data Integration Building Block shall be able to make data available in a CRISMA-compliant way using standardized OGC SOS interfaces
- The Data Integration Building Block shall provide a way to add new datasets to the already available OGC WMS.
- The Data Integration Building Block shall provide a way to add new datasets to the already available OGC WFS using standardized OGC WFS-T interfaces
- The Data Integration Building Block shall provide a way to add new datasets to the already available OGC SOS using standardized OGC SOS-T interfaces
- The Data Integration Building Block shall be able to make data available in other **application** specific formats.
- The Data Integration Building Block shall provide convenient configuration and extension possibilities that leverage easy integration of many different types of data sources.

The WFS-Transactional standard defines operations to create new features and the SOS-Transactional standard defines operations to create new "observations".

In CRISMA data is immutable thus implementers of the Data Integration Building Block have to take care that an already published **resource** will always remain accessible using the identifier it has been published with.

Documentation:

Specifications

The **OGC** standards are available at the OCG web site (**http**://**www.opengeospatial.org/standards (http://www.opengeospatial.org/standards)**):

- **WMS (http://www.opengeospatial.org/standards/wms)**
- **WFS and WFS-T (http://www.opengeospatial.org/standards/wfs)**
- **WCS (http://www.opengeospatial.org/standards/wcs)**
- **WCPS (http://www.opengeospatial.org/standards/wcps)**
- **SOS and SOS-T (http://www.opengeospatial.org/standards/sos)**

RFC's (Request for comments) are available from many sources.

- HTTP Extensions for Distributed Authoring — **WebDAV**, **RFC 2518**
  - The original specification
- HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV), **RFC 4918**
  - Actualisation

- OGC standards
  http://www.opengeospatial.org/standards (http://www.opengeospatial.org/standards)

Realization:

GeoServer

GeoServer is an **open source** software server written in Java that allows users to share and edit geospatial data. Designed for **interoperability**, **it** publishes data from any major spatial data source using open standards.

MapServer

MapServer is an **Open Source** geographic data rendering engine written in C. Beyond browsing **GIS** data, MapServer allows you create “geographic image maps”, that is, maps that can direct users to content. For example, the Minnesota DNR Recreation Compass provides users with more than 10,000 web pages, reports and maps via a single **application**.

Related Models and BBs:

GIS View Building Block

The **GIS** View is a User Interaction **Building Block** that enables the visualisation and manipulation of geospatial data. Geospatial data plays a predominant **role** in all **crisis management** related applications, because most if not all information playing a role in crisis management has a geospatial component.

Integrated Crisis Management Middleware BB

The **ICMM** is a central **Building Block** in every **CRISMA Application**. **It** connects **Crisis Management** Simulations with the Analysis and Decision Support functionality of CRISMA by providing a central repository for harmonized **world state** and **indicator** information. The ICMM is fed by simulations providing the basic information to be used for world state analysis and decision support Building Blocks. On an ICT conceptual level the ICMM is a generic distributed **resource**-oriented Control and Communication Information Management **System**. Thereby it is important to note, that the term 'resource-oriented' in the ICMM refers to the **concept** of generic resources as used in the context of the Resource Oriented **Architecture** (ROA).

Simulation Model Integration BB

The **Simulation Model** Integration **Building Block** provides components that can be used to easily enable simulation models to participate in a **CRISMA Application**. Due to the heterogeneity of existing simulation models CRISMA has to ensure that they can be integrated in a standardized way and ultimately become a **CRISMA Federate**.

That way they are made CRISMA-aware. The envisioned technique to be used is that of so called wrapping so that this Building Block provides a piece of software that can be used to make (existing) simulation models CRISMA-aware with as little effort as possible with respect to the individualities of the different simulation models.

Simulation Model Interaction View

The **Simulation Model** Interaction **Widget** **Building Block** is a Composite UI Module that lets end users interact with the various simulation models exposed by a Simulation Model Integration Building Block. **It** is responsible for collecting simulation model parameters, launching the respective simulation model, showing status information and retrieving simulation model results. It also constitutes the **GUI** to the Simulation Model Integration Building Block.

Evacuation model for coastal submersion

The **evacuation** **model** developed for coastal submersion in Charente-Maritime (France) used the software LSM2D. This model cover the Rivedoux-Plage area in Ré Island as well as the area on the coast from Yves to Chatellaion. Different scenarios of evacuation could be simulated. The main results of this model are:

- the estimated time to evacuate the population
- the estimated closed roads

- the estimated casualties on population
- the estimated impacts on buildings due to the flood

Cascading Effects Model

The cascading effect **model** for dynamic **scenario** assessment calculates the probability of attainment of cascading events scenarios, given an initial triggering **event**, or estimate consequence paths given the occurrence of selected scenarios.

Indicator Building Block

Authors: Hermann Huber
Denis Havlik

BB Description:

The Indicators **Building Block** (formerly Algebraic **Evaluation**) is a component that allows definition, storage and evaluation of “simple” algebraic models in order to evaluate consequences of decisions made by **CRISMA** users during a training session. In this context “simple” means that there is no large number crunching needed. Usage is for calculation of key performance indicators, heuristic models and to some extend the implementation of models actually running in interactive **GIS** environments in order to make them usable as services in the **CRISMA framework**.

Rationale

Business rationale: The **CRISMA application** domains are diverse and so are the decisions made by CRISMA users. Key performance indicators need to be calculated in order to provide feedback regarding the **impact** of a specific decision and to allow comparison between different decisions. Impact in this sense refers to the question of how decisions affect the current **scenario evolution** (i.e. material costs, lost productivity, number of death, time needed for normalization).

Technical rationale: Deliverable D32.1 describes the process of creating new **world** states by executing a **world state** **transition**. New world states resulting from a **transition** are said to be complete only after a successful calculation of indicators, **criteria** and cost, i.e. executing the **ICC** Functions. The ICC Functions don’t change within a single **simulation case** and are identical for all transitions. This building block provides functionality required to create, manage and execute the ICC Functions.

Key Functionality

- The Indicators **BB** shall support **OGC** **WCS** **service**
- The Indicators BB shall support OGC **WPS** service
- The Indicators BB shall support OGC **WFS** service

- The indicators BB enables to specify and manage custom **indicator** models in order to calculate application specific indicators
- The calculation process of the Indicators BB can be triggered manually (on demand)
- The calculation process of the Indicators BB can be triggered automatically (**event** based)
- The Indicators BB supports version management of both, models and indicator results
- The Indicators BB shall provide means to maintain multiple versions of calculated indicators
- The Indicators BB provides a UI component (client software) that allows scenario managers to modify indicator models
- The Indicators BB shall support calculations with any kind of geometries like points, lines, grids, coverages, political areas, etc.
- The Indicators BB shall support means to deal with multiple CRISMA use cases in parallel
- The Indicators BB shall provide means to access and retrieve data from multiple World States
- The Indicators BB provides calculated results in various formats (i.e. table format, graphical format)
- The Indicators BB shall provide means to export calculated results (CSV, Excel)
- The Indicators BB shall support CVS and EXCEL data import
- The Indicators BB shall inform the **ICMM** or Pub/Sub Broker BB when new data is available.
- The Indicators BB shall support triggering by appropriate events in additional to WPS

Realisation

Two realizations of this Building block exist in CRISMA: **BasicIndicators and the Emikat**.

- BasicIndicators is basically a Python script with WPS **interface**, which can be easily customized by a programmer. **It** is based on **PyWPS** and available as **Open Source**.
- Emikat is a business-intelligence and decision-support GIS-like tool suitable for building of complex models and spatial indicators.
    - Emikat is a mature commercial product which has been primarily developed for the austrian governmental agencies to simplify emission monitoring and simulatiing the impact of future developments on emissions and imissions in respective areas.
    - Unlike BasicIndicators, Emikat also offers a dedicated **GUI** client component which allows professionals to develop the models and indicators without the help of a professional programmer.

Although both of these reference implementations can be used to calculate some indicators, their raison d'être is very different. BasicIndicators should be primarily used for indicators which can be easily encoded in a couple of lines of Python code, such as "total number of deaths in an **accident**".

On the other hand, Emikat primarily targets the use cases where **Indicator function** is rather complex and the result is more than just a set of one or few numbers. For example, Emikat could be used to calculate the geographic distribution of a **hazard** based on the known distribution of **risk** sources and elements of risk. Depending on the way such calculation is positioned in a **CRISMA system**, such functionality can be either called an indicator (result is not considered part of the World State and not used in WS transitions) or a **model** (result is part of the WS and used by other models in WS transitions).

Documentation:

Functional Description

Main functions supported by this Building Blocks are:

- Completing the **World** States by creating, managing and executing **ICC** Functions
- Calculating key performance indicators to quantify decision impacts and provide feedback to **CRISMA** users
- Publishing the models which were developed within **GIS** systems as web services
- Analysing and visualizing the results in many different ways (i.e. **Web Coverage Service**)

Realization:

Emikat

Emikat is a **data management** and modelling **system** for spatially resolved data. The system allows the integration of emission-relevant activity data (e.g. from surveys, traffic models or statistics) as well as the definition and analysis of **model** scenarios. Scenarios allow a comparative examination of model results corresponding to different versions of data inventories – for example the overall effects of trends in emission generating activities and pollution control measures or the influence of different calculation models on estimated results.

BasicIndicators

BasicIndicators is a reference implementation of the **Indicator** **BB**. **It**'s **role** is to calculate the some of the indicators needed in **CRISMA** applications and store the results in the **World State**.

Related Models and BBs:

Integrated Crisis Management Middleware BB

The **ICMM** is a central **Building Block** in every **CRISMA Application**. **It** connects **Crisis Management** Simulations with the Analysis and Decision Support functionality of CRISMA by providing a central repository for harmonized **world state** and **indicator** information. The ICMM is fed by simulations providing the basic information to be used for world state analysis and decision support Building Blocks. On an ICT conceptual level the ICMM is a generic distributed **resource**-oriented Control and Communication Information Management **System**. Thereby it is important to note, that the term ‘resource-oriented’ in the ICMM refers to the **concept** of generic resources as used in the context of the Resource Oriented **Architecture** (ROA).

OOI World State Repository BB

OOI-WSR is a **Resource Management** related **Building Block** that enables archiving, querying and manipulation of Objects of Interest (OOI) **world state** data. This **BB** serves as a Repository **service** for OOI data that can be consumed or manipulated by other interaction or functional building blocks and resource management models.

Publish Subscribe Context Broker BB

The Publish Subscribe Context Broker **Building Block** is a cross-over between an **event** broker which accepts events and dispatches them to subscribers and an access **service** providing the information on current state of the “world”. Thereby, the Publish Subscribe Context Broker Building Block can be used to realize the event subscription and event delegation functionality of the **ICMM**. Thus, the Building Block is both suitable for dispatching events related to Control and Communication Information (**CCI**) managed by the ICMM and events related to (Worldstate) Data managed by arbitrary data access services, e.g. the **OOI** **World State** Repository **BB** or implementations of the Data Integration BB (**WMS**, **WFS**, …).

Resource Management Training Indicators and Statistics View

The **Resource Management** Training Indicators and Statistics View **Building Block** is a User Interaction Building Block that will be realised as Mashable Composite UI Module. **It** focuses on the visual presentation of statistics and key indicators of a given Worldstate in order to provide a quick overview of the **situation** and to allow for comparison between any two given Worldstates.

Scenario Analysis and Comparison View

The **Scenario** Analysis and Comparison View is able to visualise **Indicator** and **Criteria** data in a way that users are able to analyse and compare different Simulated **Crisis Management** Scenarios and ultimately come to a decision which fits the **simulation objective** best for a specific **Simulation Case**.

Population exposure model

Model for distributing population in spatial and temporal dimensions.

This model uses temporal and spatial proxies in order to disaggregate the population from administrative units to spatio-temporal grids. The outcome is used in **CRISMA** as basis for time-dependent **exposure** assessment and in further steps as a base data for **evacuation** and casualty modeling (Aubrecht et al., 2014ab;

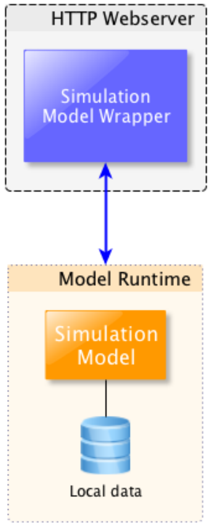
Steinnocher et al., 2014).

Simulation Model Integration BB

BB Description:

The **Simulation Model** Integration **Building Block** provides components that can be used to easily enable simulation models to participate in a **CRISMA Application**. Due to the heterogeneity of existing simulation models CRISMA has to ensure that they can be integrated in a standardized way and ultimately become a **CRISMA Federate**.

That way they are made CRISMA-aware. The envisioned technique to be used is that of so called wrapping so that this Building Block provides a piece of software that can be used to make (existing) simulation models CRISMA-aware with as little effort as possible with respect to the individualities of the different simulation models.



The current model integration approach described in the **CRISMA Framework** **Architecture** is based on **OGC** **Web Processing Service**. The **WPS** provides rules for standardizing the way how the processes and their inputs and outputs are described, how a client can request the execution of the processes and how the outputs of the process are delivered. Thereby, also several simulation models may be exposed as one WPS **interface** thus supporting model changing, **cascading effects**, etc.). The CRISMA model chaining **concept** enables users to access a specific WPS that takes care of the correct orchestration of several single WPS simulation models.

It is important to note that the Simulation Model Integration Building Block does not have to expose an **API** other than that supported by the OGC WPS **standard**. CRISMA clients of the Simulation Model Integration Building Block (e.g. The Simulation Model Interaction **Widget** Building Block)

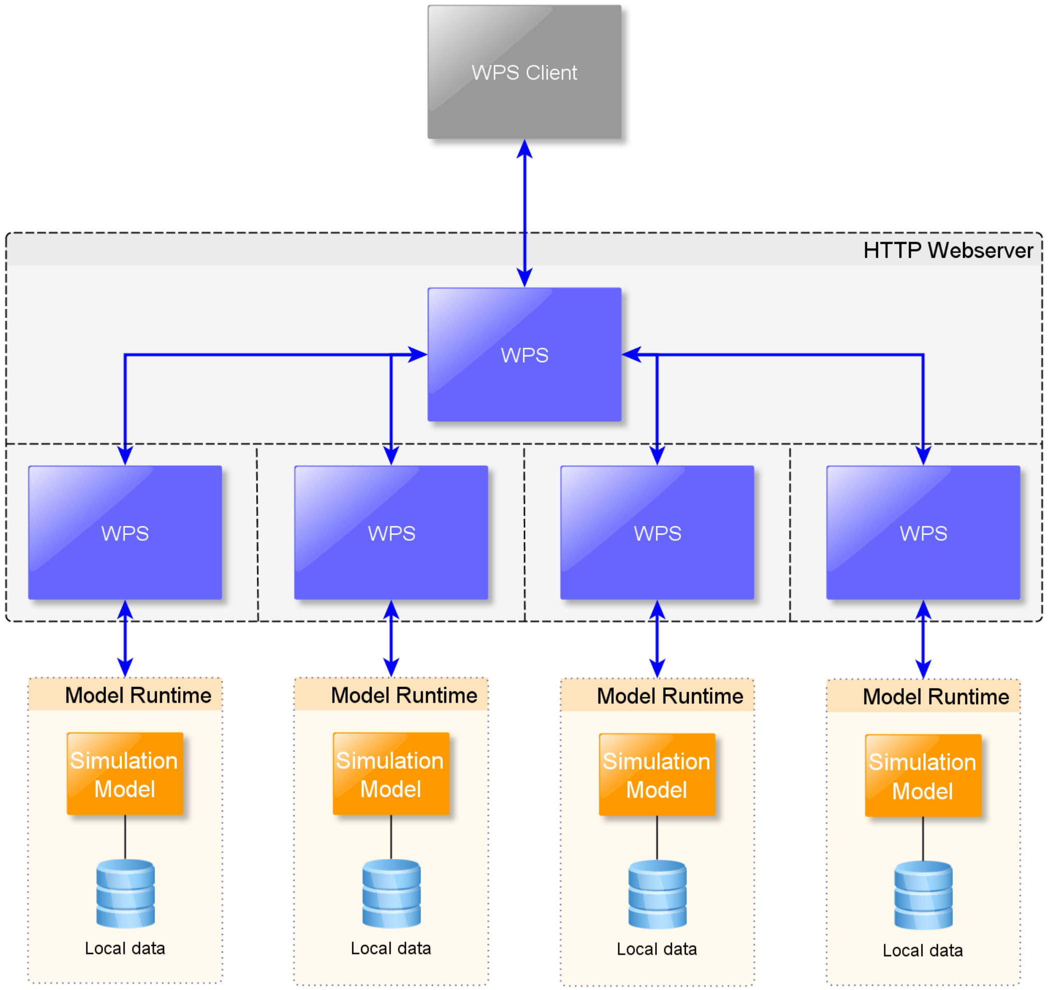
Rationale:

CRISMA heavily relies on the execution of models. Existing models are rarely accessible in a uniform way and newly developed ones should also be based on a certain standard this Building Block aims at providing these features.

Key Functionality:

The key functionalities of the Simulation Model Integration Building Block are defined by the following functional requirements.

- The Simulation Model Integration Building Block shall provide a wrapper for simulation models to make them compatible to the CRISMA model integration approach (based on OGC WPS) regardless of the model runtime.



- The Simulation Model Integration Building Block shall be able to expose several different simulation models as one Federated Simulation. The Federated Simulation must appear as one single simulation model.
- The Simulation Model Integration Building Block shall provide API functions that deliver descriptions of the respective model in a CRISMA-compliant way (WPS capabilities document).
- The description of the simulation model wrapped by the Simulation Model Integration Building Block shall contain a human readable description of the purpose of the simulation model.
- The description of the simulation model wrapped by the Simulation Model Integration Building Block shall contain machine-processable **meta-information** about the supported, required and optional parameters that have to be available in order to perform a new model run. This includes information how this data can be provided, e.g. if it can be transmitted prior to the actual execution request and where it shall reside, if it shall be part of the actual execution request, if the actual execution request may contain links to the actual data. Additionally it must state what formats and encodings are supported for each **parameter** as well as being able to provide validity ranges if applicable.
- The description of the simulation model wrapped by the Simulation Model Integration Building Block shall contain machine-processable information about its result data. This includes information on supported data formats and means for retrieving the data (Example: the result of a run is map raster data and can be fetched at a specific **WMS** using a certain URL).
- The Simulation Model Integration Building Block shall provide machine-processable information about the status of a specific execution (**simulation run**). Requesters may be informed at any time whether an execution is Scheduled, Running, Paused, Failed, or Finished.
- The Simulation Model Integration Building Block should provide machine-processable information about the expected execution time for a certain run and the remaining execution time or at least an indication of the actual status.
- The Simulation Model Integration Building Block should provide detailed error messages e.g. in case of a failed execution.
- The Simulation Model Integration Building Block shall provide API functions to launch the simulation model using parameters and model initialisation data provided by the user

- The Simulation Model Integration Building Block shall be able to resolve and retrieve input data from CRISMA Federates (Data Integration Building Block) and Integrated Components (e.g. **FTP** or **WebDAV** Server).
- The Simulation Model Integration Building Block shall be able to retrieve model output data and publish them via CRISMA Federates (Data Integration Building Block) or Integrated Components (e.g. FTP or WebDAV Server).
- The Simulation Model Integration Building Block shall provide API functions to access model output data via a Data Integration Building Block component interface.
- The Simulation Model Integration Building Block shall provide convenient configuration and extension possibilities that leverage easy integration of many different types of simulation models.
- The Simulation Model Integration Building Block should be able to interact with the CRISMA Middleware (**ICMM**) to obtain information about data adapters, to attach model run status information to a worldstate **transition**, etc.
- The Simulation Model Integration Building Block shall provide convenient configuration and extension possibilities that leverage easy integration of many different types of simulation models.

Documentation:

Functional Description

TODO: link to functional description of this **BB**

Specifications

Model Integration **BB** **service** **interface** is the "**Web Processing Service**" (**WPS**). WPS is a well-established **standard** of the **Open Geospatial Consortium** (**OGC**). The specifications of the service interfaces and data models are available **free of charge** from the OGC web site.

- OGC WPS specifications
  http://www.opengeospatial.org/standards/wps (http://www.opengeospatial.org/standards/wps)

Realization:

PyWPS

PyWPS (Python **Web Processing Service**) is an implementation of the **Web processing Service (http://www.opengeospatial.org/standards/wps) standard** from **Open Geospatial Consortium (http://opengeospatial.org)** . Within **CRISMA**, the PyWPS is (can be) used to make legacy models and other software available within the **CRISMA Framework**. In order to do so, one has to extend the basic PyWPS classes and introduce invocations of the own software. By nature of GPL, this extension is automatically licensed under GPL as well, but the invoked software (e.g. models) is not linked to PyWPS and can be published under a different license.

Related Models and BBs:

Cascade Events Configuration and Interaction View

The **Cascade Effects** View is a User interaction Building that allows a user to configure and run a Cascade Effects **Scenario**. The user can select a triggering **event** (for example, an earthquake) and provide may either specify the characterisation of the event (**Simulation Control Parameter**) and thus initiate a new **Simulation Model** Run for this particular event, or select (if available) the output of a past event or an event already simulated.

Data Integration

The Data Integration **Building Block** provides components that can be used to easily serve data in a **CRISMA**-compliant (**OGC** open **standard** compatible) way so that other Building Blocks may use them for further processing like viewing or editing. That way data can be made accessible for CRISMA components.

Integrated Crisis Management Middleware BB

The **ICMM** is a central **Building Block** in every **CRISMA Application**. **It** connects **Crisis Management** Simulations with the Analysis and Decision Support functionality of CRISMA by providing a central repository for harmonized **world state** and **indicator** information. The ICMM is fed by simulations providing the basic information to be used for world state analysis and decision support Building Blocks. On an ICT conceptual level the ICMM is a generic distributed **resource**-oriented Control and Communication Information Management **System**. Thereby it is important to note, that the term 'resource-oriented' in the ICMM refers to the **concept** of generic resources as used in the context of the Resource Oriented **Architecture** (ROA).

Preparedness Plan BB

The **Preparedness** Plan **Building Block** is a decision support mechanism, which helps the decision maker to take the needed actions in case of **emergency** according to plans based on analysis of threats, vulnerabilities and possible emergency scenarios.

Simulation Model Interaction View

The **Simulation Model** Interaction **Widget** **Building Block** is a Composite UI Module that lets end users interact with the various simulation models exposed by a Simulation Model Integration Building Block. **It** is responsible for collecting simulation model parameters, launching the respective simulation model, showing status information and retrieving simulation model results. It also constitutes the **GUI** to the Simulation Model Integration Building Block.

Coastal Submersion Model

The Costal Submersion **Model** is a 2D-hydrodynamic model based on the **open source** TELEMAC-MASCARET **system**. In TELEMAC-MASCARET system, we use mainly TELEMAC2D to calculate the time and space dependent hydrodynamic characteristics such as water levels, velocities, discharges.

Evacuation model for coastal submersion

The **evacuation** **model** developed for coastal submersion in Charente-Maritime (France) used the software LSM2D. This model cover the Rivedoux-Plage area in Ré Island as well as the area on the coast from Yves to Chatellaion. Different scenarios of evacuation could be simulated. The main results of this model are:

- the estimated time to evacuate the population
- the estimated closed roads
- the estimated casualties on population
- the estimated impacts on buildings due to the flood

Resource Management Model

The **Resource Management** Models developed in the **CRISMA** project are built upon the **OOI** **concept** with different context dependent behavioral patterns for different **crisis** domains. Thus, there is no overall generic and all-purpose Resource (OOI) Management **Model**, but a set distinct models for different types of resources (e.g. ambulances, patients) and different situations. However, such domain and crisis specific Resource Management Models can be implemented on basis of the general **Agent**-Oriented **Simulation** Models **Building Block**, with its functionalities described in D312 .

Road network vulnerability model (RNV)

Model for the assessment of probability of road link interruption due to earthquakes.

Dikes vulnerability model

The dikes **vulnerability model** is a model programmed in python which allow to calculate the potential statistical **impact** on dikes depending on their status.The model is based on the damage probability matrix.

In view of results of dike vulnerability model, the user could make an informed choice on break or failure dikes for local **simulation**.

Building impact model

Model for the assessment of expected damage on building classes due to earthquakes

Time Dependent Vulnerability model (TDV)

Model for the assessment of time-dependent damage on **elements at risk**.

UI Integration Platform BB

BB Description:

The UI Integration Platform **Building Block** is a component that is able to host Composite UI Modules. Composite UI Modules are User Interaction Building Blocks that are realised as HTML5 and JavaScript widgets. **It** constitues the Runtime Environment of the Composite UI Modules as they - by their nature - cannot be used as stand-alone applications.

UI Integration Platform

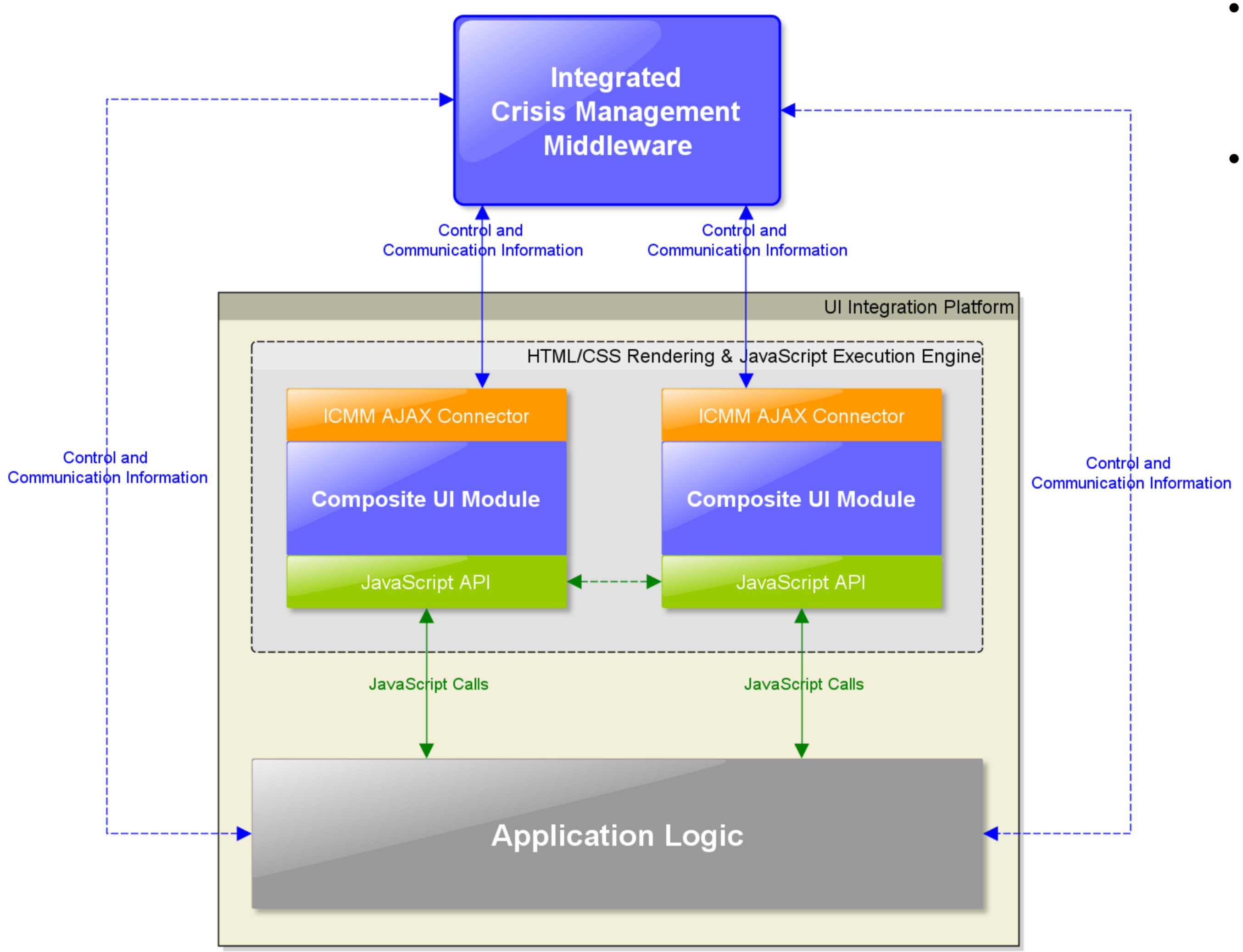
In general, the UI Integration Platform will not be used to render a single Composite UI Module but a set of Mashable Composite UI Modules (e.g. Wirecloud Widgets) which have been combined into a Mashup **Application** with the help of the UI **Mashup Platform** (e.g. Wirecloud).

Rationale

In order to be able to actually run a **CRISMA Application** a Runtime Environment for the Composite UI Modules is needed as they - by their nature - cannot be used as stand-alone applications.

Key Functionality

- The UI Integration Platform must be able to render Composite UI Modules.
- The UI Integration Platform must support HTML5.
- The UI Integration Platform must support **CSS3**.
- The UI Integration Platform must support JavaScript.



- The UI Integration Platform must not impose any restrictions or requirements on Composite UI Modules that are in conflict with technical and functional requirements on Composite UI Modules.
- The UI Integration Platform should be able to communicate with Composite UI Modules via a JavaScript **API** exposed by the Composite UI Module

Realisation and Software Components

There exist several pieces of software that are able to provide the key functionality such as any modern browser (Firefox, Opera, etc.) or platforms like JavaFX. Hence there is virtually no development effort needed. When the UI Integration Platform is part of an existing application and the Composite UI Module shall be integrated into this application, the UI Integration Platform should be able to support interaction and communication between the existing application and Composite UI Module via the Composite UI Module's JavaScript API.

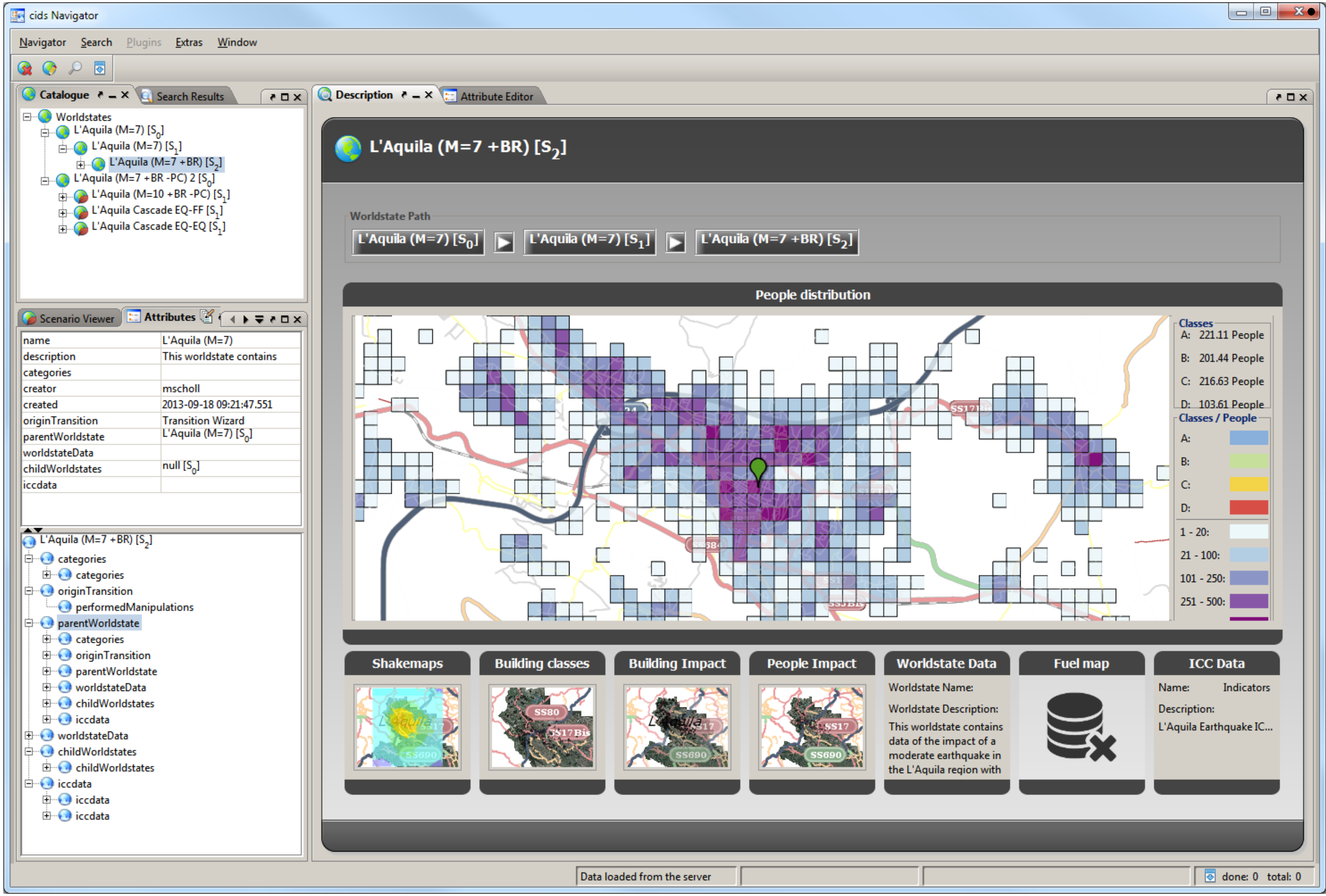
UI Integration Platform of the CRISMA Technology Demonstrator

Documentation:
Functional Description

Functional Description of the "UI Integration Platform" **Infrastructure** / User Interaction **Building Block**

- UI Integration Platform BB Functional Description **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20Functional%20Building%20Blocks/v2.0/UI%20Integration%20Platform.doc**

Specifications



D32.2 ICMS Architecture Document V2

http://www.crismaproject.eu/deliverables/CRISMA_D322_public.pdf (http://www.crismaproject.eu/deliverables/CRISMA_D322_public.pdf)

Realization:

Situator Training System

Situation Management is a solution approach that comprises tools and methods for coordinating the interaction between people, technologies, and responses. **NICE** Systems is a major player in the Situation Management market with over 20 years in experience in the industry and expertise in a wide variety of industries such as seaports, airports, railways, banking, government, telecom, utilities, military, manufacturing, etc.

---

cids Navigator

Cids Navigator is one of the software components that implement the Functional **Building Block** UI Integration Platform of the **CRISMA Framework**.

Related Models and BBs:

Integrated Planning View

The Integrated Planning View **BB** is a generic integrated view for the configuration and inspection of arbitrary **crisis management** scenarios in planning situations. The **CRISMA** planning use cases (most use cases of CRISMA) intend to realize functionalities that are related to opening **world** states, changing them and storing them. Further running simulations, inspecting them and comparing results is addressed in all of them and the user work for planning purposes in a calm office environment.

UI Mashup Platform

The **Mashup Platform** **Building Block** acts as a runtime environment for Mashable Composite UI Modules (widgets) and provides inter-**widget** communication, persistent per-widget configuration as well as proxy capabilities. The platform allows the user (in this case the person configuring a **CRISMA application**) to create a HTML5-based front-end by selecting certain user components (widgets and operations) that can be combined into a mashup. These user components are the **interface** for users to underlying back-end services. The user components themselves are provided by a 3rd party platform through a **catalogue**, from which a range of components can be installed into a mashup and connected ("wired") using a graphical editor.

Worldstate View

The Worldstate View **Building Block** is a set of generic Widgets (Mashable Composite UI Modules) that allow to visualise Control and Communication Information (**CCIM**) related to Worldstates and Worldstate Transitions. Thus, **it** operates mainly on common **meta-information** about the **world** rather than the real data of the world (**World State** Data Slots).

UI Mashup Platform

BB Description:

The **Mashup Platform** **Building Block** acts as a runtime environment for Mashable Composite UI Modules (widgets) and provides inter-**widget** communication, persistent per-widget configuration as well as proxy capabilities. The platform allows the user (in this case the person configuring a **CRISMA application**) to create a HTML5-based front-end by selecting certain user components (widgets and operations) that can be combined into a mashup. These user components are the **interface** for users to underlying back-end services. The user components themselves are provided by a 3rd party platform through a **catalogue**, from which a range of components can be installed into a mashup and connected (“wired”) using a graphical editor.

A mashup platform simplifies the **task** of creating graphical user interfaces using pre-defined items (widgets). The widgets can be provided by any 3rd party platform through a catalogue, from which a range of components can be installed into a mashup and connected (“wired”) using a graphical editor. Unless the custom widgets are required to fulfil any requirements, this approach requires next to no knowledge about software development or the inner workings of the widgets or of the mashup platform.

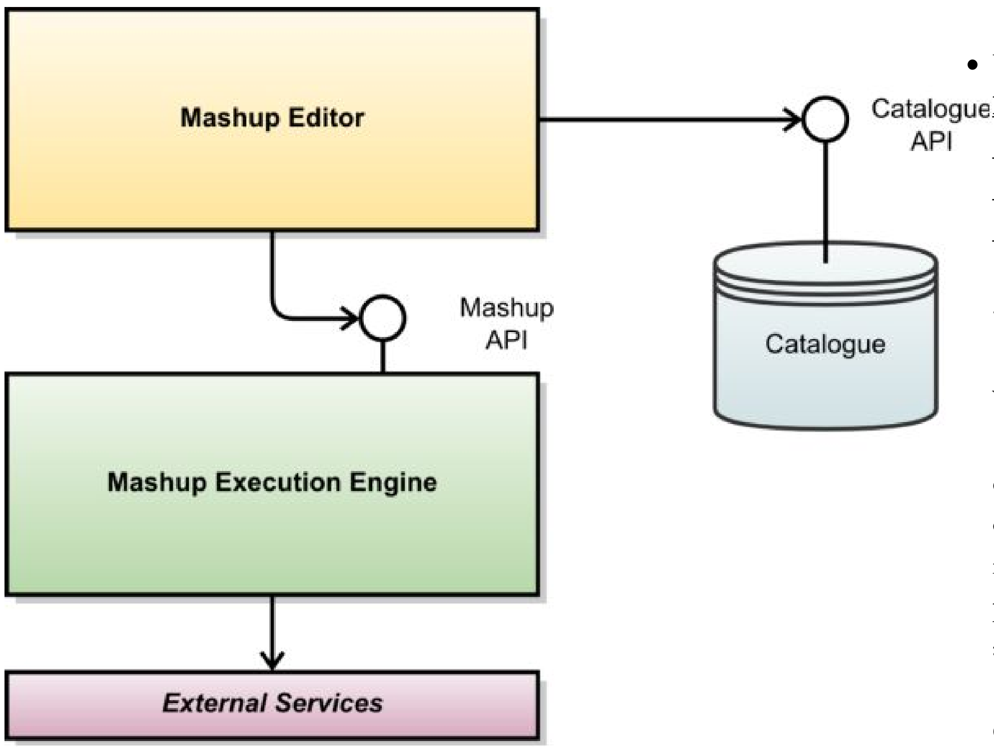
In a CRISMA context, the mashup platform is primarily used by administrators and developers to create the the application-specific mashup applications ("Views") from generic widgets. Such mashups provide a convenient way to quickly develop the functional prototypes during the development phase of individual CRISMA applications. In a subsequent development, the mashups can be either used as stand-alone web applications, embedded in legacy web- and desktop applications, or even re-implemented as additional functionality of the legacy applications.

The figure below illustrates the overall **architecture** of the UI Mashup Platform consisting of:

- a catalogue to store, publish and manage available widgets and mashups,
- a mashup editor that allows the creation of new mashup applications by combining and configuring existing widgets,
- a mashup execution engine responsible for providing any functionality required to execute a mashup and provide additional means of logging and debugging,
- the possibility to attach to external services though the mashup execution engine to make use of components not part of the mashup application or any widget itself.

Second figure (below) illustrates a possible configuration of a mashup application from an abstracted point of view. Any number of widgets and operators (with any number of input and output endpoints themselves) are connected and listen to and/or publish **event** data. In addition, services such as externally managed Pub/Sub brokers or external **JSON**/**XML** web services can be connected to in order to facilitate data from externally managed services and data sources.

Documentation:



Functional Description

- UI Mashup Platform - Functional Description **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20Functional%20Building%20Blocks/v2.0/UI%20Mashup%20Platform%20-%20Functional%20Description.doc**

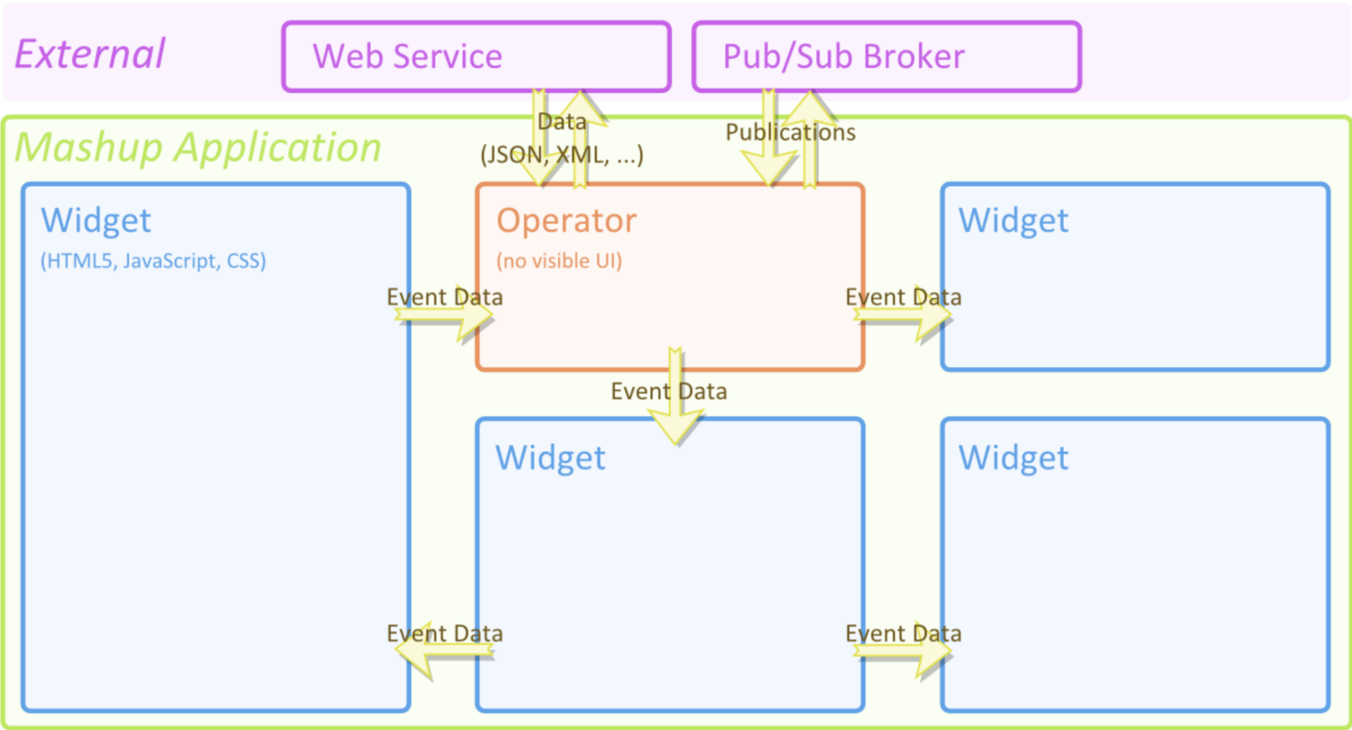
Realization:

Wirecloud Application Mashup Platform

Wirecloud is a reference implementation of the **FI-Ware (http://www.fi-ware.eu/)** (Web) **Application** Mashup Generic Enabler. **It** allows users with no programmers experience to rapidly build web applications by "wiring" the available widgets and operators and positioning them on a screen. In **CRISMA**, the Wirecloud is used as a reference implementation of the **UI Mashup Platform** **Building Block** and as a platform for development of the CRISMA web widgets and Views (Mashup applicaitons) based on these widgets.

Currently available CRISMA-specific widgets and operators for this platform:

- **Charts**
- **OOI** Commands
- **OOI GIS Map**
- **OOI Table**
- **Simulation** Picker
- **Worldstate Loader** (to be replaced by an interactive **Worldstate Picker** in future iterations)
- Worldstate Saver



Related Models and BBs:

Integrated Planning View

The Integrated Planning View **BB** is a generic integrated view for the configuration and inspection of arbitrary **crisis management** scenarios in planning situations. The **CRISMA** planning use cases (most use cases of CRISMA) intend to realize functionalities that are related to opening **world** states, changing them and storing them. Further running simulations, inspecting them and comparing results is addressed in all of them and the user work for planning purposes in a calm office environment.

Preparedness Plan BB

The **Preparedness** Plan **Building Block** is a decision support mechanism, which helps the decision maker to take the needed actions in case of **emergency** according to plans based on analysis of threats, vulnerabilities and possible emergency scenarios.

Resource Management Training Dispatch and Monitor View

The Dispatch and Monitor view provides a high-level overview over the **resource management** **simulation**'s **world state**. Its purpose is to display one world state at a time and allow the user to distribute resources (ambulances, etc.) among different areas where the **crisis** plays out.

Resource Management Training Indicators and Statistics View

The **Resource Management** Training Indicators and Statistics View **Building Block** is a User Interaction Building Block that will be realised as Mashable Composite UI Module. **It** focuses on the visual presentation of statistics and key indicators of a given Worldstate in order to provide a quick overview of the **situation** and to allow for comparison between any two given Worldstates.

Resource Management Training Simulation Scenario Setup View

The **Resource Management** Training **Simulation** **Scenario** Setup View is a User Interaction **Building Block** that will be realised as Mashable Composite UI Module. **It** allows the creation of new resource management simulations or modification of existing ones. It allows the user - most likely a trainer - to create incidents and scenes as well as the creation and management of objects of interest (**OOI**) instances as shown in the figure below.

Simulation Model Interaction View

The **Simulation Model** Interaction **Widget** **Building Block** is a Composite UI Module that lets end users interact with the various simulation models exposed by a Simulation Model Integration Building Block. **It** is responsible for collecting simulation model parameters, launching the respective simulation model, showing status information and retrieving simulation model results. It also constitutes the **GUI** to the Simulation Model Integration Building Block.

UI Integration Platform BB

The UI Integration Platform **Building Block** is a component that is able to host Composite UI Modules. Composite UI Modules are User Interaction Building Blocks that are realised as HTML5 and JavaScript widgets. **It** constitues the Runtime Environment of the Composite UI Modules as they - by their nature - cannot be used as stand-alone applications.

Worldstate View

The Worldstate View **Building Block** is a set of generic Widgets (Mashable Composite UI Modules) that allow to visualise Control and Communication Information (**CCIM**) related to Worldstates and Worldstate Transitions. Thus, **it** operates mainly on common **meta-information** about the **world** rather than the real data of the world (**World State** Data Slots).

4.3 User Interaction

Many of the **CRISMA** Building Blocks provide specifications of the reusable **graphical user interface** elements. At the project start, our intention was to describe each of the CRISMA widgets as a **building block**, but this turned out to be impractical due to a large number of widgets. Therefore we eventually decided to present the widgets only at the component level (==implementation/software) and to keep the **BB**-level descriptions only for the very complex widgets and for the "Views".

Views are defined as functional **GUI** elements which provide some functionality required by CRISMA applications AND which can be realized by combining several of the widgets. At a realization level, most of the CRISMA widgets are implemented in HTML5/Javascript and built in such a way that they can be used either both stand-alone and within the Wirecloud **mashup platform**. This solution speeds up the **application** prototyping, while at the same time allowing the re-use of the same GUI elements without the mashup platform.

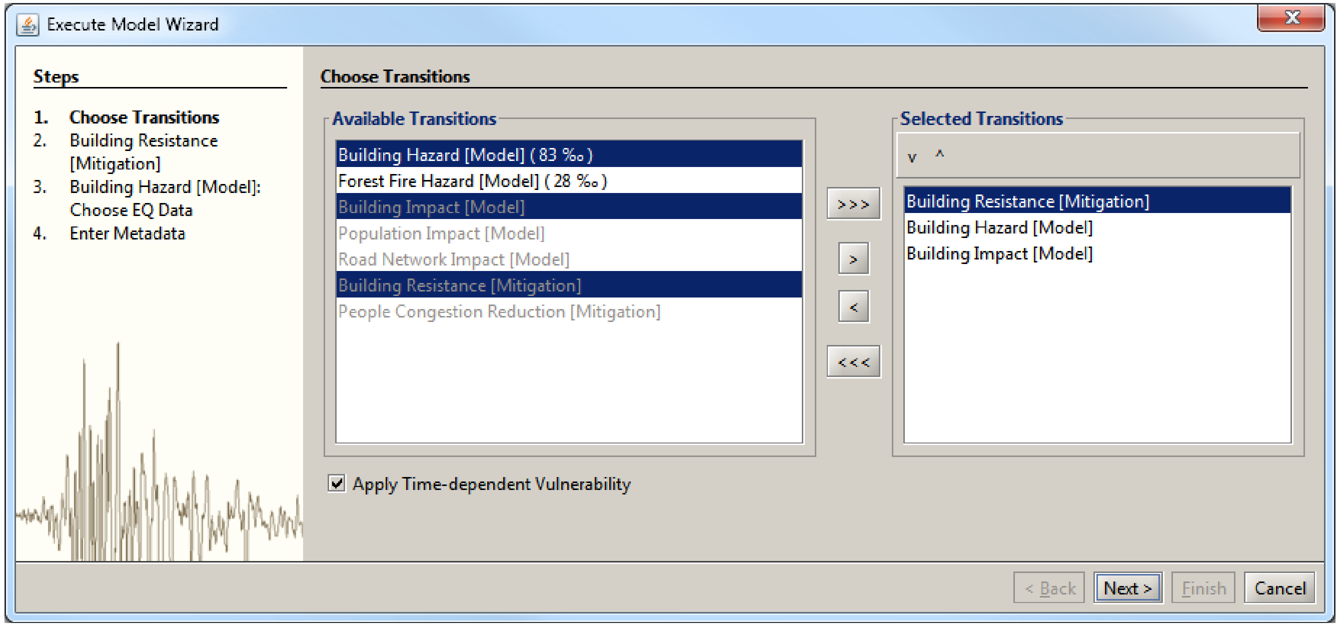
Cascade Events Configuration Widget

Version: 1.0
This is a java-based prototype of the **Cascade Events** Configuration and Interaction View. **It** is mainly used in the **CRISMA** **Technology Demonstrator**.

The Cascade Events Configuration **Widget** a new software component that is developed from scratch in phase two of WP34. In contrast to the technological requirements for new software developments that demands the implementation of new User Interaction Building Blocks as Composite UI Module (HTML and JavaScript), the Cascade Events Configuration Widget is developed in Java in the context of ongoing prototyping activities to be able to demonstrate the capabilities of the **CRISMA Framework** (Technology Demonstrator).

Configuration of the Transition Matrix

Cascade Events Configuration for Earthquake



Cascade Events Configuration for Forest Fire

The Cascade Events Configuration Widget is implemented as cids navigator (UI Integration Platform) Wizard.

Documentation:

Functional Description

- Building Block Description
  https://crisma-cat.ait.ac.at/bbs/cascade-events-configuration-and-interaction-view

Downloads:

Sourcecode on GitHub

- Sourcecode on GitHub
  https://github.com/cismet/cids-custom-crisma

Binaries in the **cismet** Software Repository

- Software Repository
  https://repo.cismet.de/

Team:

Martin Scholl

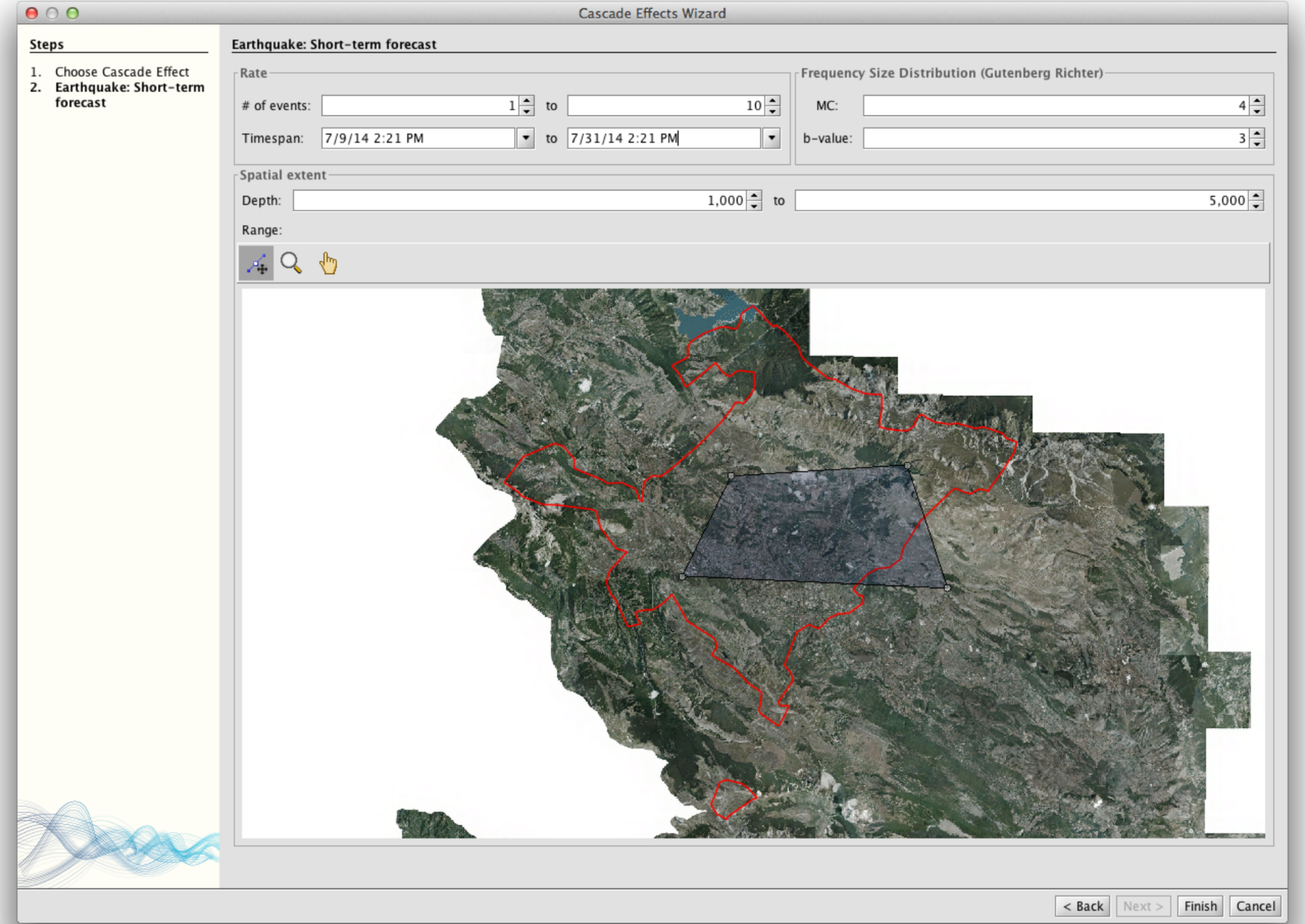
Development roles: development, installation, support

Main **developer** of the **Widget** and point of contact for installation and configuration support.

Cascade Events Configuration and Interaction View

BB Description:

The **Cascade Effects** View is a User interaction Building that allows a user to configure and run a Cascade Effects **Scenario**. The user can select a triggering **event** (for example, an earthquake) and provide may either specify the characterisation of the event (**Simulation Control Parameter**) and thus initiate a new **Simulation Model** Run for this particular event, or select (if available) the output of a past event or an event already simulated.



When the triggering event has been selected and characterized (e.g. a earthquake as shown in the screen shot above), the View shows the possible paths of event chains that are available according to the respective **Transition Matrix**. The user may select one of the paths and the Cascade Effects View will highlight eventual secondary events triggered after the previous one. In this way, the user may select a specific chain of events that are interested to assess or to interrupt a chain if he decides to stop the analysis in an intermediate point. For each of the events, the user may either characterise the event by providing the respective simulation control parameters or select the output of a past event.

Documentation:

Specifications

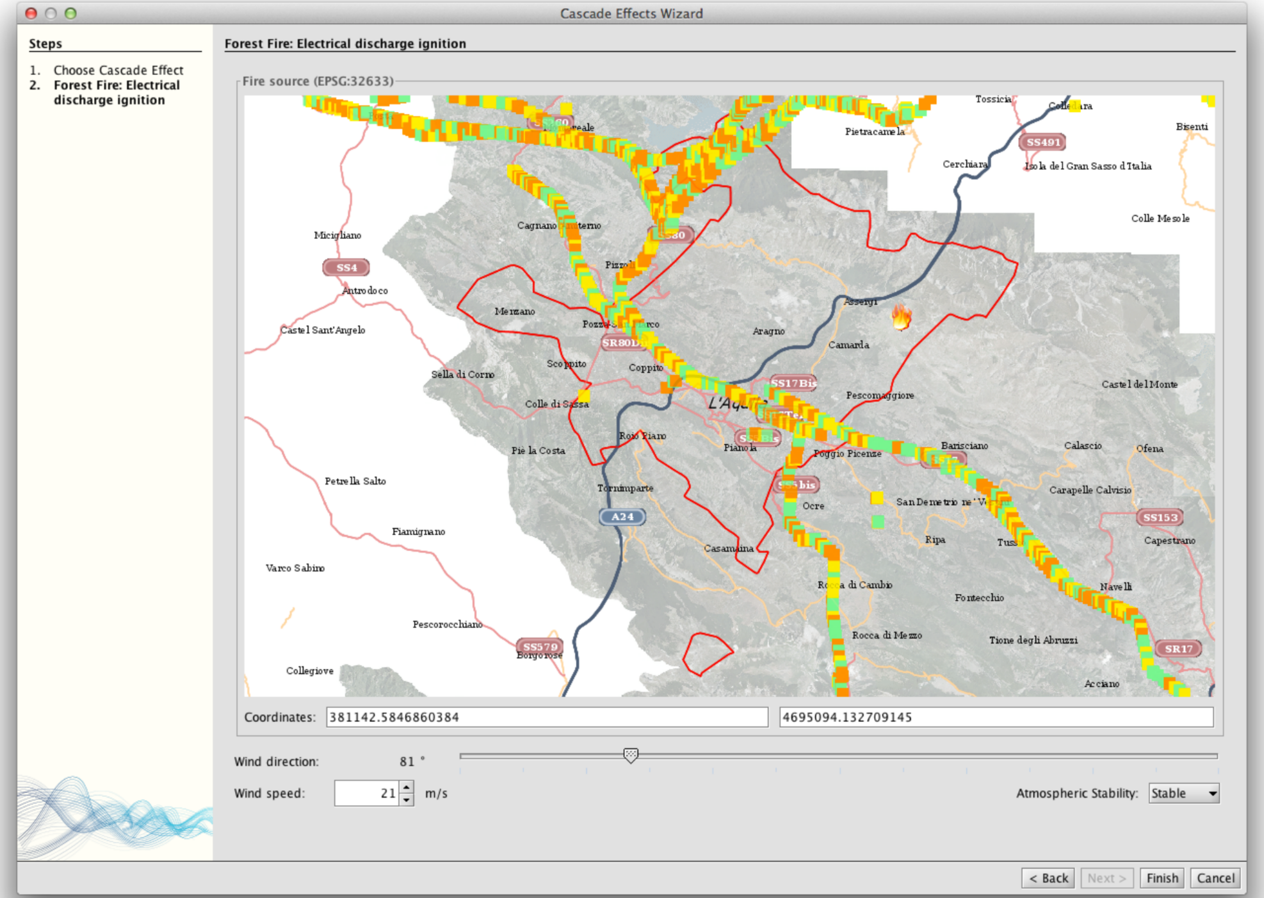
Reports describing the **concept** **model** for dynamic **scenario** assessment due to **cascade events** and the prototype **application** for time-dependent **vulnerability** for systems at risks.

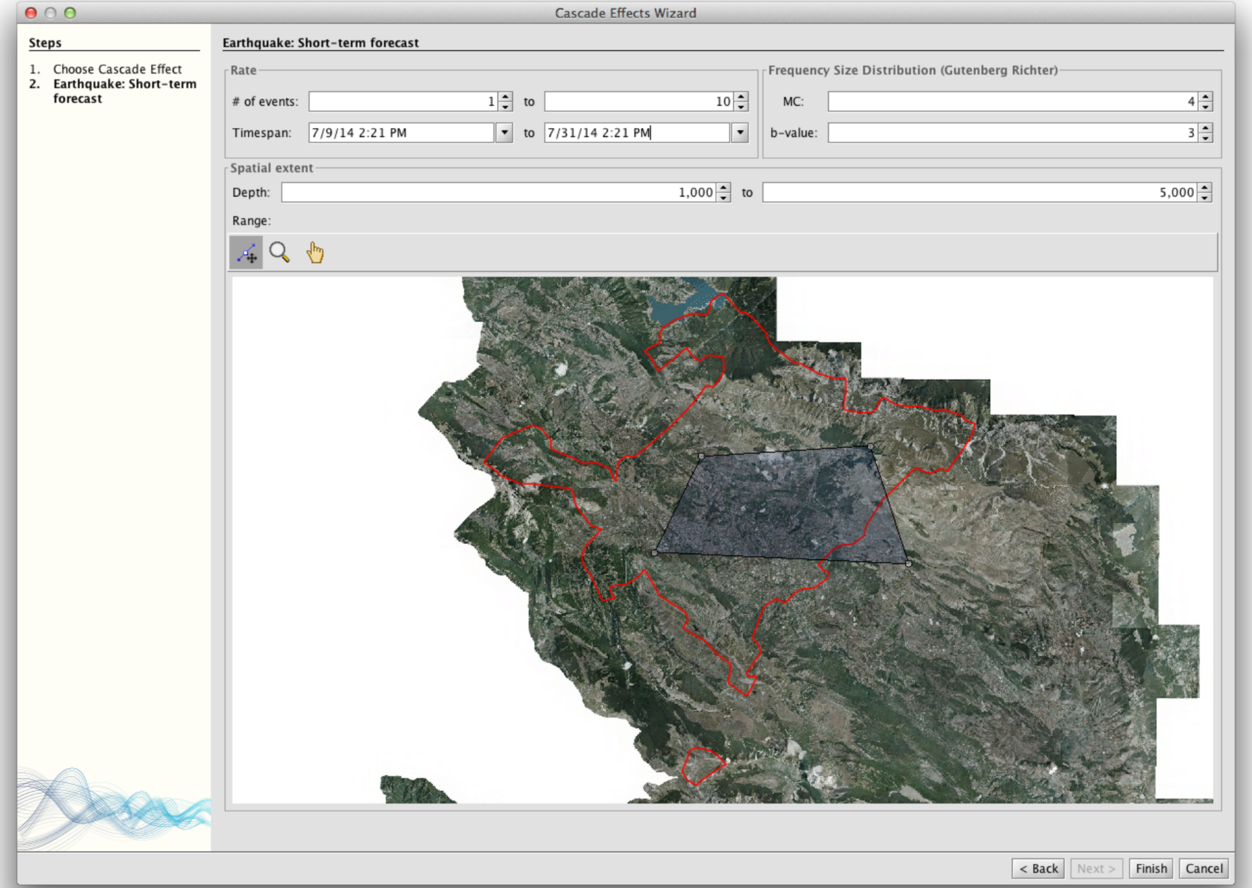
- D42.1 Dynamic scenario concept models **https://workspace.vtt.fi/sites/eu_crisma/Deliverables/Accepted%20by%20the%20Commission/CRISMA_D421_final.pdf**
- D43.1 Version 1 of Dynamic vulnerability functions, Systemic vulnerability, and Social vulnerability **https://workspace.vtt.fi/sites/eu_crisma/Deliverables/Sent%20to%20the%20Commission/CRISMA_D431_final.pdf**

Realization:

Cascade Events Configuration Widget

This is a java-based prototype of the **Cascade Events** Configuration and Interaction View. **It** is mainly used in the **CRISMA** **Technology Demonstrator**.


Cascade Effects Wizard
Steps
1. Choose Cascade Effect
2. Forest Fire: Electrical discharge ignition
Forest Fire: Electrical discharge ignition
Fire source (EPSG:32633)
L'Aquila
Coordinates:
381142.5846860384
4695094.132709145
Wind direction:
81 °
Wind speed:
21
m/s
Atmospheric Stability:
Stable
< Back
Next >
Finish
Cancel



Related Models and BBs:

Integrated Crisis Management Middleware BB

The **ICMM** is a central **Building Block** in every **CRISMA Application**. **It** connects **Crisis Management** Simulations with the Analysis and Decision Support functionality of CRISMA by providing a central repository for harmonized **world state** and **indicator** information. The ICMM is fed by simulations providing the basic information to be used for world state analysis and decision support Building Blocks. On an ICT conceptual level the ICMM is a generic distributed **resource**-oriented Control and Communication Information Management **System**. Thereby it is important to note, that the term 'resource-oriented' in the ICMM refers to the **concept** of generic resources as used in the context of the Resource Oriented **Architecture** (ROA).

Simulation Model Integration BB

The **Simulation Model** Integration **Building Block** provides components that can be used to easily enable simulation models to participate in a **CRISMA Application**. Due to the heterogeneity of existing simulation models CRISMA has to ensure that they can be integrated in a standardized way and ultimately become a **CRISMA Federate**.

That way they are made CRISMA-aware. The envisioned technique to be used is that of so called wrapping so that this Building Block provides a piece of software that can be used to make (existing) simulation models CRISMA-aware with as little effort as possible with respect to the individualities of the different simulation models.

Simulation Model Interaction View

The **Simulation Model** Interaction **Widget** **Building Block** is a Composite UI Module that lets end users interact with the various simulation models exposed by a Simulation Model Integration Building Block. **It** is responsible for collecting simulation model parameters, launching the respective simulation model, showing status information and retrieving simulation model results. It also constitutes the **GUI** to the Simulation Model Integration Building Block.

Cascading Effects Model

The cascading effect **model** for dynamic **scenario** assessment calculates the probability of attainment of cascading events scenarios, given an initial triggering **event**, or estimate consequence paths given the occurrence of selected scenarios.

Time Dependent Vulnerability model (TDV)

Model for the assessment of time-dependent damage on **elements at risk**.

Economic impacts analysis view

Authors: **Jussi Yliaho**
Markus Jähi

BB Description:

This **Building Block** is an **economic evaluation** tool to support **crisis management** and to be used in the **preparedness** phase for planning and training purposes. The main objective of an economic evaluation in **CRISMA** is:
- to present the economic impacts arising from crises (ex post performance) and
- to assess different **mitigation** proposals and their costs/benefits (ex ante planning).

This Building Block is a combination of the two V1 **economic impact** related components (see figure below) and therefore its purpose is two-fold. Firstly, its purpose is to collect cost related data (economic parameters) which is not readily available from the Worldstate description and is needed in economic impact calculation from a user. Secondly, its purpose is to show results related to economical evaluation.

This Building Block is a **GUI** client invoking the Economic Impact Calculation **Service** which is provided by SP4 - Models for Multi-Sectoral Consequences (**Task** 4.4.2 - **Simulation** of economic impacts and consequences).

The main functionalities to be provided by this building block are:
- The user should be able to define the desired type of economic impact analysis.
- The user should be able to see Worldstate values related to economical calculations (e.g. number of destroyed buildings, used working hours for **emergency** personnel).
- The user should be able to fill in on a form cost data related to **response**, damages and mitigation measures.
- From the building block user input values need to be transferred to the Economic Impact Calculation Service.
- To present the economic impacts arising from crises (ex post performance).
- To assess different mitigation proposals and their costs/benefits (ex ante planning).

CRISECON components:

Documentation:

Functional Description

Downloads:

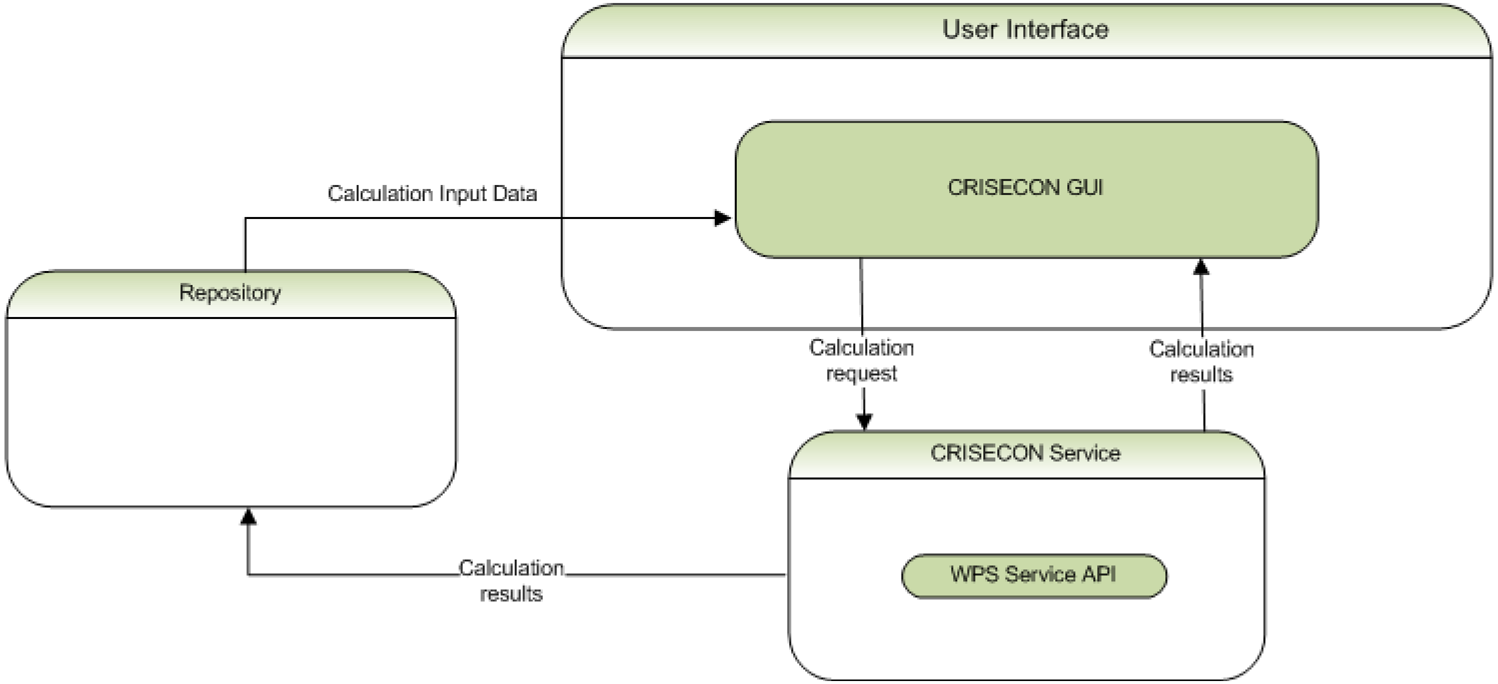
- https://crisma-cat.ait.ac.at/system/files/uid_67/CRISECON_model_and_software_D442_2.doc

Realization:

CRISECON Service

CRISECON **Service** implements, together with the related UI component CRISECON **GUI**, the CRISECON **model** developed in SP4.

The key functionalities CRISECON Service are:



CRISECON GUI

CRISECON **GUI** (see figures below) implements, together with the related web-**service** component CRISECON Service, the CRISECON **model** developed in SP4.

Related Models and BBs:

Economic impacts model

Model for:

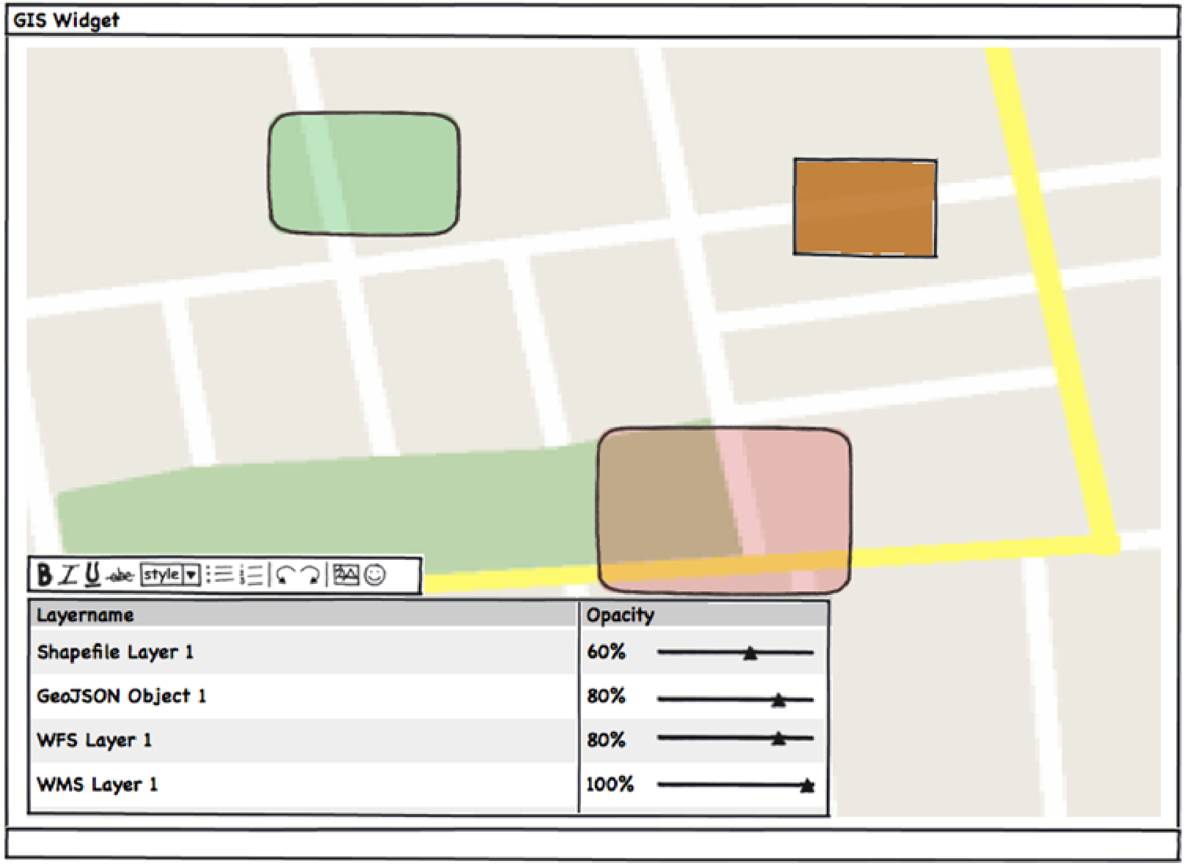
- presenting economic impacts arising from crises (ex post performance) and
- assessing different **mitigation** proposals and their costs/benefits (ex ante planning).

GIS View Building Block

BB Description:

The **GIS** View is a User Interaction **Building Block** that enables the visualisation and manipulation of geospatial data. Geospatial data plays a predominant **role** in all **crisis management** related applications, because most if not all information playing a role in crisis management has a geospatial component.

The GIS View must be able to handle data delivered by the Data Integration Building Block but also other well-known GIS formats. Furthermore, **it** must be possible to visualise geospatial data from multiple sources simultaneously to allow a comparison of different scenarios (GIS layers). Moreover, the GIS View shall not only be able to show static information like background maps (e.g. road transport **infrastructure**, topographic maps) but also dynamic content which is updated according to a progressing crisis management **simulation**. For example, the current location of resources and actors, as well as the status of other objects of interest should be shown in their spatial context. Additional user interaction features of the GIS View include, for example, the ability to interactively manipulate objects on the map.



GIS View GUI Example

Rationale

CRISMA Applications by nature have to deal with geospatial data.

Key Functionality

- The GIS View shall be able to visualise geo-spatial data from multiple sources simultaneously.
- The GIS View shall provide **API** functions to get and set data sources for visualisation and manipulation programmatically.
- The GIS View shall provide means to order multiple data sources via its user **interface** to effectively influence the visualisation (layering).
- The GIS View shall provide API functions to order multiple data sources.
- The GIS View shall be able to handle every data source as separate unit (layer sandboxing): asynchronous retrieval, individual error handling and visualisation.
- The GIS View shall provide means to manipulate visualisation properties per data source via its user interface, e.g. layer transparency.
- The GIS View shall provide API functions to manipulate visualisation properties per data source programmatically.
- The GIS View shall support for an edit mode to edit geo-spatial data.
- The GIS View shall support API functions to programmatically activate the edit mode.
- The GIS View shall be able to assign an edit action performed by the user to a worldstate **transition**.
- The GIS View shall provide API functions to propagate edit events to related Building Blocks, e.g. the Worldstates View.
- The GIS View shall provide means to edit geographical **feature** data including feature properties and feature geometries thus effectively editing worldstate data

directly.

- The GIS View shall provide API functions to edit geographical feature data programmatically.
- The GIS View shall provide means to create new geographic features via its user interface.
- The GIS View shall provide means to create new geographic features programmatically via respective API functions.
- The GIS View shall be able to edit **WMS** request parameters thus effectively creating a new worldstate data item.
- The GIS View shall provide an API functions to edit WMS request parameters programmatically.
- The GIS View shall be able to edit **WFS** request parameters thus effectively creating a new worldstate data item.
- The GIS View shall provide an API functions to edit WFS request parameters programmatically.
- The GIS View shall provide means to create new point geometries via its user interface.
- The GIS View shall provide means to create new line geometries via its user interface.
- The GIS View shall provide means to create new polygon geometries via its user interface.
- The GIS View shall provide means to create new geometries from manual coordinate input via its user interface.
- The GIS View shall provide means to create new geometries from Extended Well-Known Text/Binary (EWKT/B) input via its user interface.
- The GIS View shall provide the general navigation functionalities Zoom, Pan, MoveTo and Home.
- The GIS View shall provide general location history navigation functionalities.
- The GIS View shall provide undo and redo (edit history) functionalities.
- The GIS View shall support **OGC** WFS data sources.
- The GIS View shall support OGC WFS-T data sources.
- The GIS View shall support OGC WMS data sources.
- The GIS View shall support OGC WMS-T data sources.
- The GIS View shall support GeoJSON data sources.
- The GIS View shall support ESRI Shapefile data sources.
- The GIS View shall support different EPSGs.
- The GIS View shall provide convenient configuration mechanisms that allow configuring default data sources (e.g. background layers) and other configurable properties.
- The GIS View shall come with a set of default backgrounds maps (e.g. OpenStreetMap, Google Maps, etc.) that can be used in cases when no detailed maps of a specific are of interest are available.
- The GIS View shall support feature-type specific custom rendering and editing functionalities, e.g. custom visualisations for different types of OOIs.

Documentation:

Functional Description

Funtional Description of the **GIS** **Widget** **BB**.

- Functional Description **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20Functional%20Building%20Blocks/v2.0/GIS%20Widget.doc**
- D32.2 ICMS Architecture Document V2 **http://www.crismaproject.eu/deliverables/CRISMA_D322_public.pdf (http://www.crismaproject.eu/deliverables/CRISMA_D322_public.pdf)**

Realization:

Wirecloud OOI GIS Map

A map **widget** showing **Object Of Interest** (**OOI**) on a background map.

---

cismap

Cismap is one of the software components that implement the Functional **Building Block** **GIS** **Widget** of the **CRISMA Framework**.

Leaflet.js

Leaflet.**js** is one of the software components that implement the Functional **Building Block** **GIS** **Widget** of the **CRISMA Framework**. **It** is used "as is" and thus not extended or modified for CRISMA.

OpenLayers

OpenLayers is one of the software components that implement the Functional **Building Block** **GIS** **Widget** of the **CRISMA Framework**. **It** is used "as is" and thus not extended or modified for CRISMA.

Related Models and BBs:

Data Integration

The Data Integration **Building Block** provides components that can be used to easily serve data in a **CRISMA**-compliant (**OGC** open **standard** compatible) way so that other Building Blocks may use them for further processing like viewing or editing. That way data can be made accessible for CRISMA components.

Preparedness Plan BB

The **Preparedness** Plan **Building Block** is a decision support mechanism, which helps the decision maker to take the needed actions in case of **emergency** according to plans based on analysis of threats, vulnerabilities and possible emergency scenarios.

Worldstate View

The Worldstate View **Building Block** is a set of generic Widgets (Mashable Composite UI Modules) that allow to visualise Control and Communication Information (**CCIM**) related to Worldstates and Worldstate Transitions. Thus, **it** operates mainly on common **meta-information** about the **world** rather than the real data of the world (**World State** Data Slots).

Cascading Effects Model

The cascading effect **model** for dynamic **scenario** assessment calculates the probability of attainment of cascading events scenarios, given an initial triggering **event**, or estimate consequence paths given the occurrence of selected scenarios.

Integrated Planning View

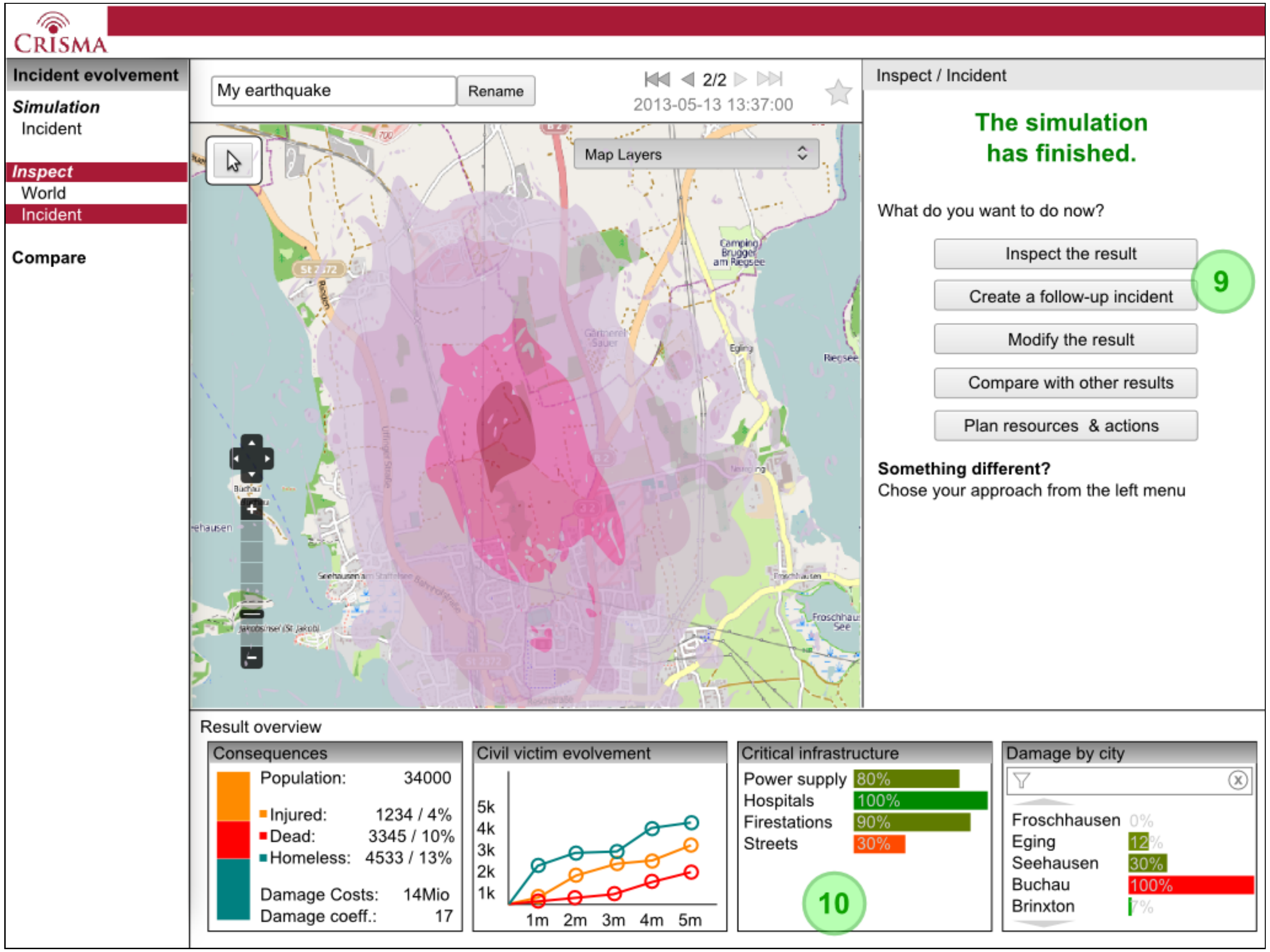
BB Description:

The Integrated Planning View **BB** is a generic integrated view for the configuration and inspection of arbitrary **crisis management** scenarios in planning situations. The **CRISMA** planning use cases (most use cases of CRISMA) intend to realize functionalities that are related to opening **world** states, changing them and storing them. Further running simulations, inspecting them and comparing results is addressed in all of them and the user work for planning purposes in a calm office environment.

Figure 1 shows an exemplary main screen for the **use case** **incident** evolvement (**pilot** D - Italy).

Figure 1: Dialogue **situation** during interacting with the Integrated Planning View

Rationale



This BB conceptually integrates the following existing BBs as part of a holistic user **interface** **framework** for planning use cases:

- **World State** View (Display and visualize World State Metadata and Data)
- **Resource Management** Training **Scenario** Setup View (Creating/Editing Incident Scenarios for Training Sessions)
- World State Analysis (Displaying of World States, Indicators, **Criteria**)
- Decision Support View (Showing World States, Indicators, Criteria, Multi-Criteria Analysis)
- Resource Management Training Indicators and Statistics View (Display World States and Indicators for Training Session Statistics)

Under planning use cases we understand those use cases that use CRISMA applications in an office environment during the crisis management **preparedness** phase. Regularly those users are not interrupted regularly and a single user uses the respective **CRISMA application** for short-term or long-term planning of crisis management **risk** assessment, **infrastructure** or strategy development.

These use cases are:

- Long term planning view (pilot B) (**Reference application** for Coastal Submersion)
- Incident evolvement view (pilot D) (Reference appliocation for Earthquake and Forest Fire)
- Incident evolvement and resource view (pilot A) (Reference Application for Winter Storm)
- Resource planning view (pilot E) (Reference Application for Resource Planning)

The different roles in the planning use cases work in different environments e.g. office, command vehicle or command centres. Table 1 shows the average expertise and minimum and maximum expertise of those roles working in office environments.

Table 1: Average and Range of Expertise for all Office Users in the Planning Use Cases

	Expertise/Usage Frequency		
	IT	**Crisis Management**	**Usage of CRISMA Application**
Average	middle-high	low-middle	middle-high
Minimum and Maximum	low – high	low – high	low-middle – high

The average expertise in information technology of office users is about 3 of 4 points. They are moderate in crisis management but will use CRISMA applications quite often.

For the office users IT skills reaches from low to high expertise. The same applies for the expertise in crisis management. This is a main finding for the design of an interaction **concept** for these applications, as ease-of-learning usability goals therefore may also apply beside pure interacting effectiveness and ease of use goals.

In the following selected parts of the concepts are briefly described.

Menu Bar

In the following a menu structure tailorable to all CRISMA planning use cases is described. Further an item leading to the comparison of several scenarios including a simple world state analysis is included.

For the interaction concept this capabilities result in the main menu structure: simulate, modify and inspect. The user may simply open a world state and inspect it, run a **simulation** on this world state or modify the world state manually. Additionally an important **feature** is the comparison of several world states, which results in another menu item. All these items are need as the user interactions needed for these group actions are substantially different.

The Simulation mode is used to parameterise and run simulations. The inspect mode allows viewing the the results subsequently. The modify mode is used to modify a world state. In all these modes, the user is allowed to interact with every building or other objects on a map. The interaction with that objects (either parameterising, inspecting or viewing) can for instance in the modify mode be used to optimise buildings for better resistance on a flood or an earthquake. Each modification or series of modifications creates a new world state.

The CRISMA **conceptual business logic** distinguishes between world, incident, **response** and restore regarding the scenarios (world states) and the models. From a technical point of view this is not necessary and for some use cases this may also be not necessary as only one of the dimensions are relevant. Nevertheless there are use cases necessitating substantial different user interfaces for two kinds of scenarios. E.g. for the long term planning use cases the user interaction sequence is to first open a scenario and perform a **mitigation** option or explicitly not perform one. Mitigation options in the interaction concept means to modify a world state manually before an incident **simulation run** is started. So the user in this case first navigates to the modification of the world e.g. places a dike at a certain position and afterward switches to simulate and runs his **model**. Another example is the response scenario that shall be changeable regarding rescue infrastructure like hospitals etc., or the same hospital that is relevant as a building with damage degree for the incident scenario.

Figure 2: Menu showing all Possible Entries

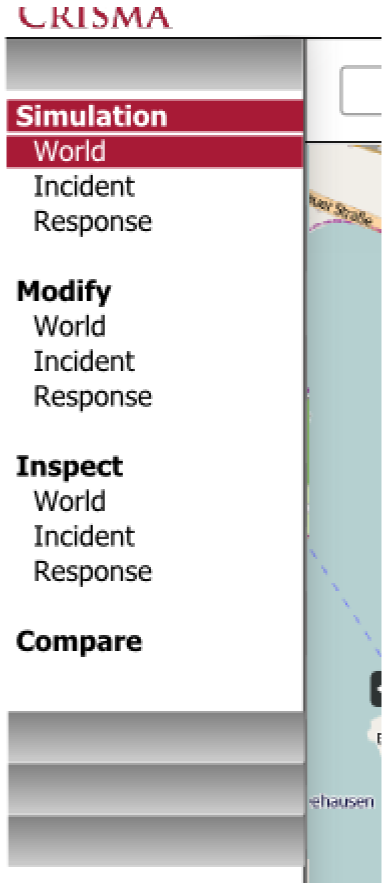
Figure 2 shows all possible entries of the menu. The menu will not look like this in any CRISMA application (except possibly a demo application) as each use case focuses on some of the aspects. Also it may be not necessary to have e.g. the sub-item “incident” if there is no other kind of simulation foreseen in a respective use case. The hierarchy of the menu items was elaborated iteratively. The most important entry point for the user working with planning applications is “what he is capable of” with the system. This is reflected by Simulate, Modify, Inspect and Compare. For some use cases these capabilities may differ per user roles working with the system. In this case menu bar just contains some of these capabilities. The second important hint for the user is “what am I dealing with”. For some use cases there is just one kind of scenario relevant. For some others there are two kinds of scenarios relevant as for instance resource planning where the initial incident scenario as well as the generated response scenario matters.

Scenario Navigation Bar

Crisis management scenarios are stored and handled using the world state concept. However the term world state is not used on the user interface. Changing world states through manual changes or simulations always results in new world states. These new world states then link to the previous ones for traceability reasons. From a user’s point of view these “intermediate” world states that are not “leaves” in the world state tree usually are not interesting any more. As default behaviour the planning interaction concept therefore foresees that they are omitted and just the leaf world states are visible with a defined name. These leaf world states represent the whole scenario, which is defined as a world state branch in the user interface concept.

Figure 3 shows the navigation bar that is always visible when the user is working with one scenario.

Figure 3: Scenario Navigation Widgets allow Navigating to previous States of a Scenario



The showed scenario has two states and shows the last state as default. The user can navigate back to intermediate states using the depicted navigation buttons.

The starring concept allows a user defining his or her favourite scenarios by simply clicking on the star behind the representation of a scenario on the user interface

Predefined Values for Input Fields

It is generally considered as good practice, to prefill input fields, if possible, so that they can remain untouched as a rule. The goal is to reduce input errors and to ensure a speedy workflow (cf. "Good defaults" Tidwell 2011, 385). Nevertheless, incorrect entries have to be marked ((d), see "Same Page Error Messages" Tidwell 2011, 388). Because of hierarchically related objects in the configuration of the world, it is necessary to also prefill including child objects. This can be achieved with templates.

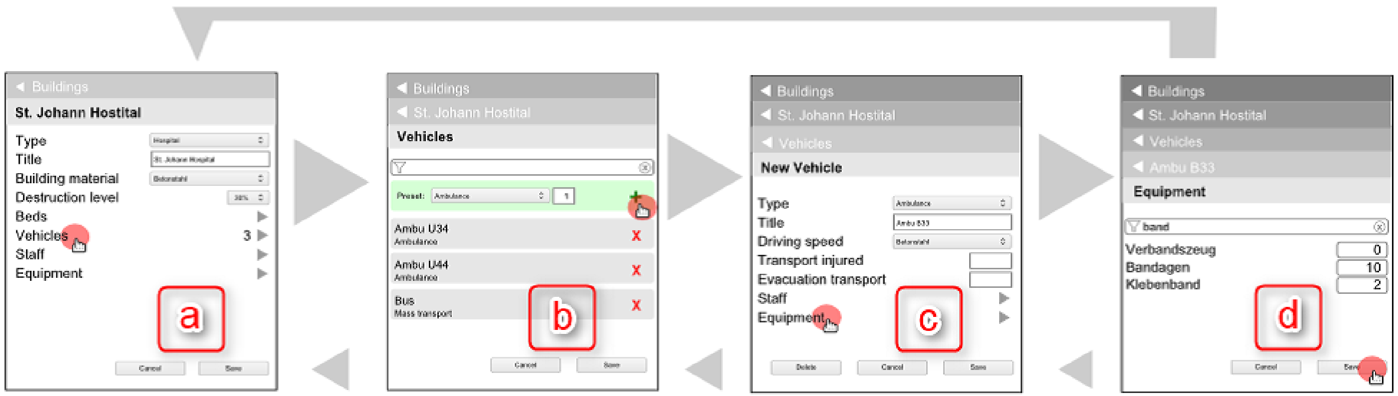


Figure 4: Navigation in the Breadcrumb Stack

Filtering Attributes

If an **object** has many properties that are not easy to perceive for the user, a filtering mechanism supports him. If for example an ambulance has 100 potential quantifiable materials the user can filter for specific properties (see Figure 4 (d) and Figure 5). Figure 5 illustrates that the breadcrumb stack may Increase with a Level in the Hierarchy Down (a), can any Time a Step Back but can also Skip Many Levels as required by the "Breadcrumb Stack". b) Shows an Object List, c) the Properties of one of the Objects (B), and d) the Filtering Means on the Example of Quantitative Properties.

Figure 5: Filter Component

The filter option is placed directly on the form as the first item under the "breadcrumb stack". Even for forms with few items the general availability of filtering may be useful. Showing or hiding the function could affect the user's conformity with expectation and have negative effects on the user's spatial memory - ("Spatial Memory" Tidwell 2011, 17).

Working with Templates

Templates are different configurations of an object type with different but required configuration. Figure 6 shows in (c) and (e) variants of the original selection, to add new objects. The individual designs illustrate: a) Configuration on Top-Level, b) Configuration with Direct and Lower-Level Attributes, c) Adding of Objects Using Templates, d) Marking of Errors and e) A Presentation of Direct and Visible Attributes in a Tabular Form

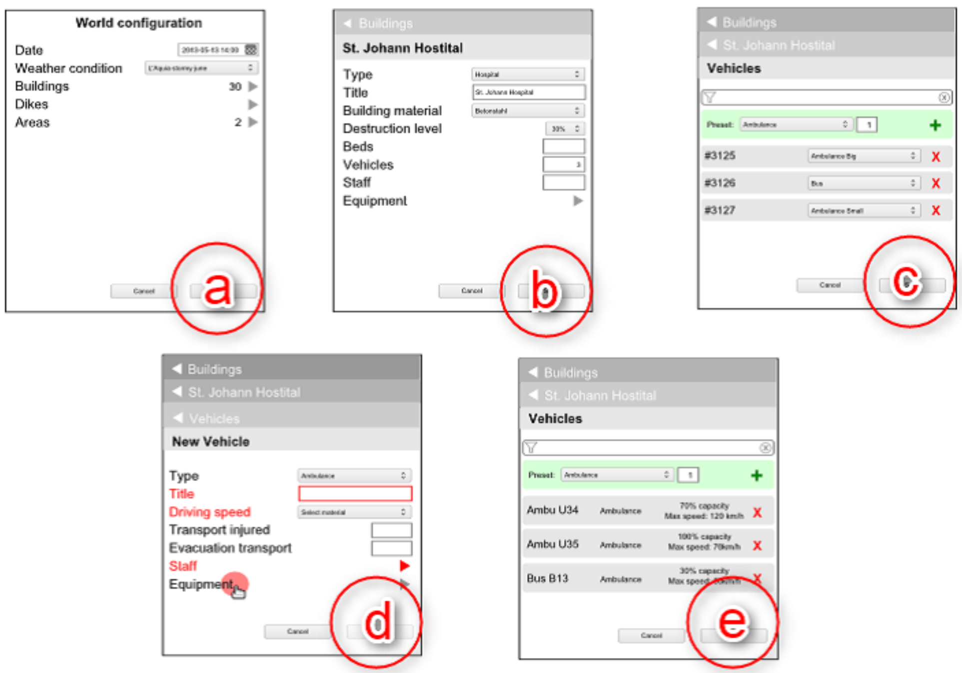


Figure 6: Exemplary Views of the Configuration Component

The following two approaches for working with hierarchical templates are possible:

- The object hierarchy for a type is fixed, the user only decides to instantiate it.
- The user chooses an initial object configuration and may change the values subsequently.

Template objects are always introduced as copies in order to being able to change each property individually. Copying instead of referencing further avoids the same simulation run under the same conditions giving different results when changing the template.

When using templates it may be useful to foresee attributes which allow an individualisation of the concrete object e.g. the percentage of the available equipment of an ambulance or the degree of damage of a building caused by an earthquake.

Shall the user be able to change an object's configuration it may be useful to save the information on which template the object is based on in order to offer a return path. Changing a template may influence existing derived objects and their properties; it should not change the properties' values.

The "numbers only mode" is a variant of the templates concept. It can be applied at any level of the hierarchy. For the user he or she can spin-down the hierarchy ends at this point, even though there are more hierarchy levels. Just a counting number for each sub-object can be changed. The objects itself are not changeable and may correspond to a template, if selected. The "numbers only mode" is a variant of the template concept. Again copies of a fully preconfigured template object are generated when a user increases such a number.

Compare Interaction Concept

A big benefit of the CRISMA **architecture** is the **capability** of comparing different scenarios of one **simulation case** with each other. This allows the user to choose suitable strategies out of several ones or to decide e.g. for improving buildings to get more resistant for earthquakes or floods.

Figure 35 shows the selection list of several scenarios. In the text field there is a possibility to filter shown results by typing some characters into it. To only show self-created scenarios a checkbox named "only-mine" has to be activated. If a user wants to select scenarios from different users, he or she is able to disable that checkbox, which in default will be selected. Above the "only-mine" checkbox, there are three possible attributes for ordering scenarios (date, title or username).

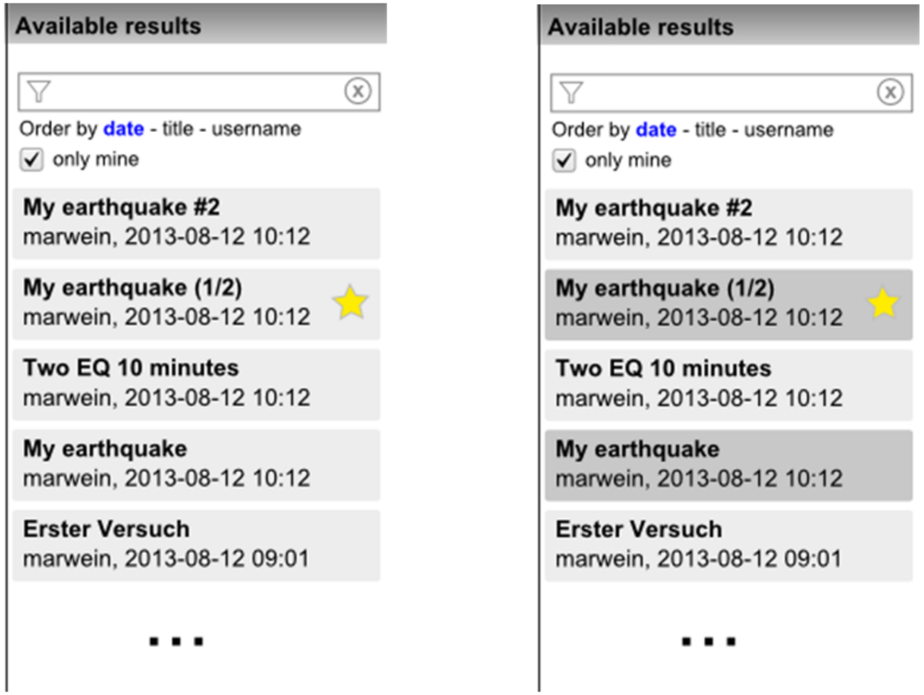


Figure 7: Configuration of Compare Mode with no States (a) and Two States (b) selected

After this initial filtering of scenarios, they can be selected for comparison by clicking. In Figure 7 (b) the scenarios "My earthquake (1/2)" and "My earthquake" are selected. The first one is an intermediate scenario state as it has the extension "(1/2)". The selected items are marked in a darker grey as the unselected ones. Additionally there is a list **widget** showing actual selected scenarios where the user is able to reorder selected scenarios for such comparisons where ordering is relevant (see Figure 8).

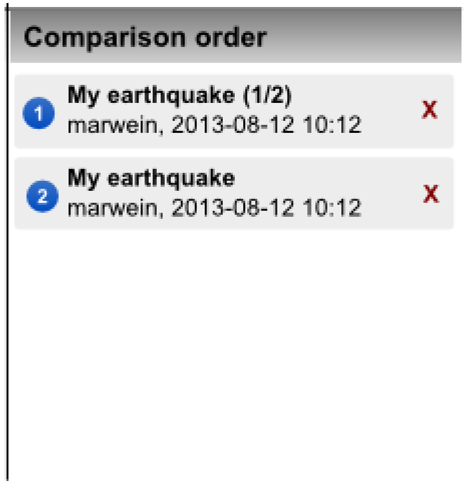


Figure 8: A Widget Visualises the Selected Scenarios

Regularly only the leaf world state (representing the scenario of the whole branch) is shown in list. However, if a user "stars" an intermediate state of a scenario, these intermediate states are shown, too.

After selecting two or more scenarios users are able to select indicators for visualising the results as shown in the left part of Figure 9.

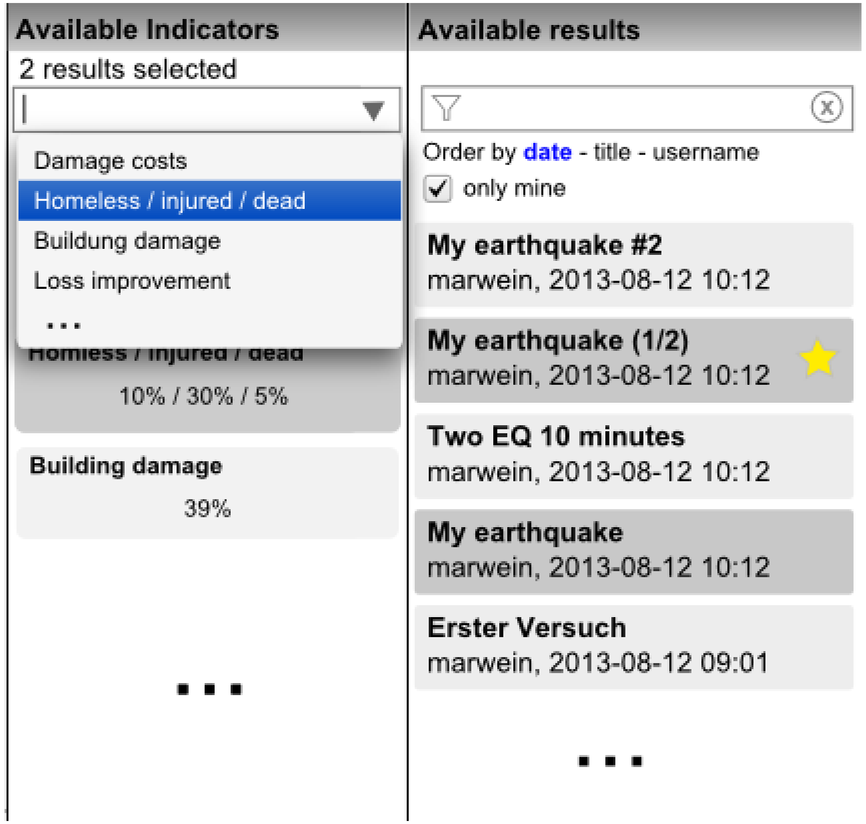


Figure 9: Compare View During the Selection of a Result Set to Display.

If the user selected one or more indicators, they are shown in the list view (see Figure 10 (a)). By selecting one of them, it can be configured as shown in Figure 10 (b) to set the visualisation form which will appear or refresh right after clicking “Apply”.

Figure 10: Selected Indicators (a) and their Options (b) in Compare View

That selection will be visualised in a panel on the left side of CRISMA application. As shown in Figure 11 a diagram appears to see differences between the selected scenarios. To differentiate the scenarios, they marked with a number each as defined in Figure 11.

Figure 11: Compare View Diagram to Visualise Comparison of Two Scenarios

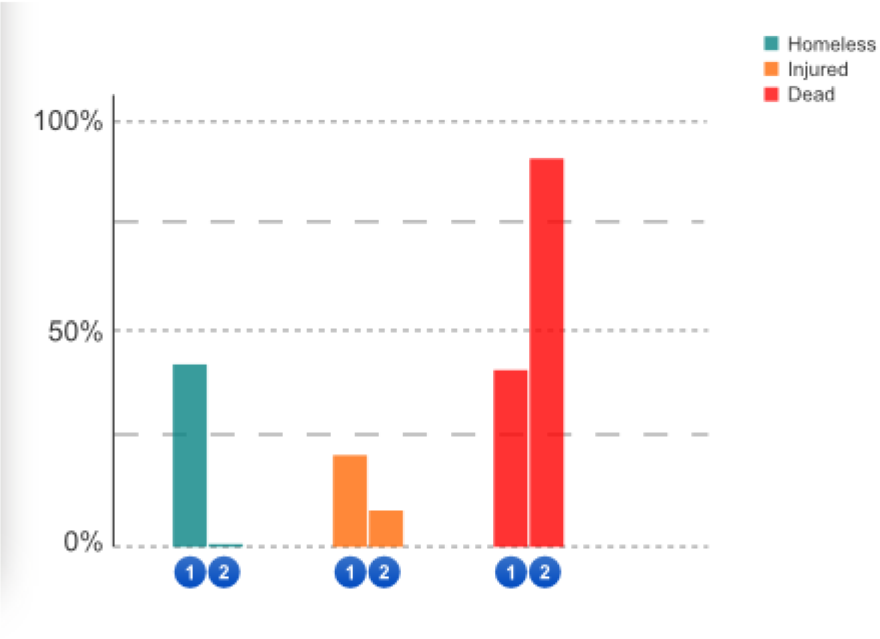
References

- Tidwell, Jenifer. Designing interfaces. 2nd ed. Sebastopol, CA: O'Reilly, 2011.

Documentation:

Specifications

The **concept** has the following main components:



- menu main views: inspect, simulate, Modify, Compare

For all mentioned views besides compare:

- **Scenario** navigation bar (top of screen)
- Map or other visualisation (centre of screen)
- configuration component (including "breadcrumb stack" menu) (right of screen)
- results /charts view (bottom of screen)

The Compare View:

- Map or other visualisation (top)
- charts (below)
- scenario selection (right)
- scenario selected (right bottom)
- **indicator** selection (centre-right)
- indicators selected (centre-right)
- indicator configuration(centre bottom)

- Detailed generic description and an exemplary realisation for earthquake and incident-evolvement planning **https://workspace.vtt.fi/sites/eu_crisma/Deliverables/Sent%20to%20the%20Commission/CRISMA_D251_final.pdf**

Realization:

Wirecloud Application Mashup Platform

Wirecloud is a reference implementation of the **FI-Ware (http://www.fi-ware.eu/)** (Web) **Application** Mashup Generic Enabler. **It** allows users with no programmers experience to rapidly build web applications by "wiring" the available widgets and operators and positioning them on a screen. In **CRISMA**, the Wirecloud is used as a reference implementation of the **UI Mashup Platform** **Building Block** and as a platform for development of the CRISMA web widgets and Views (Mashup applicaitons) based on these widgets.

Currently available CRISMA-specific widgets and operators for this platform:

- **Charts**
- **OOI** Commands

- **OOI GIS Map**
- **OOI Table**
- **Simulation** Picker
- **Worldstate Loader** (to be replaced by an interactive **Worldstate Picker** in future iterations)
- Worldstate Saver

Wirecloud Charts

This **widget** displays stacked bar charts for **indicator** values for a given worldstate.

Wirecloud OOI GIS Map

A map **widget** showing **Object Of Interest** (**OOI**) on a background map.

Exercise Worldstate Data Chart Widgets (Wirecloud)

Several widgets that display worldstate data from **ICMM** in graphical form (as pie charts and stacked bar charts) and in table form. Developed with AngularJS.

Configuration Component

The configuration component is the core component of the Integrated Planning View. **It** has been exemplary implemented visualizing the Charts in the Debriefing View of the **Reference Application** for **Exercise**-Support in **Pilot** E.

Related Models and BBs:

Multi Criteria Analysis and Decision Support View

The Multi **Criteria** Analysis View and Decision Support View is a User Interaction **Building Block** that allows performing a ranking of different **Crisis Management** Scenarios with respect to specific Criteria.

Resource Management Training Indicators and Statistics View

The **Resource Management** Training Indicators and Statistics View **Building Block** is a User Interaction Building Block that will be realised as Mashable Composite UI Module. **It** focuses on the visual presentation of statistics and key indicators of a given Worldstate in order to provide a quick overview of the **situation** and to allow for comparison between any two given Worldstates.

Resource Management Training Simulation Scenario Setup View

The **Resource Management** Training **Simulation** **Scenario** Setup View is a User Interaction **Building Block** that will be realised as Mashable Composite UI Module. **It** allows the creation of new resource management simulations or modification of existing ones. It allows the user - most likely a trainer - to create incidents and scenes as well as the creation and management of objects of interest (**OOI**) instances as shown in the figure below.

Scenario Analysis and Comparison View

The **Scenario** Analysis and Comparison View is able to visualise **Indicator** and **Criteria** data in a way that users are able to analyse and compare different Simulated **Crisis Management** Scenarios and ultimately come to a decision which fits the **simulation objective** best for a specific **Simulation Case**.

Simulation Model Interaction View

The **Simulation Model** Interaction **Widget** **Building Block** is a Composite UI Module that lets end users interact with the various simulation models exposed by a Simulation Model Integration Building Block. **It** is responsible for collecting simulation model parameters, launching the respective simulation model, showing status information and retrieving simulation model results. It also constitutes the **GUI** to the Simulation Model Integration Building Block.

UI Integration Platform BB

The UI Integration Platform **Building Block** is a component that is able to host Composite UI Modules. Composite UI Modules are User Interaction Building Blocks that are realised as HTML5 and JavaScript widgets. **It** constitues the Runtime Environment of the Composite UI Modules as they - by their nature - cannot be used as stand-alone applications.

UI Mashup Platform

The **Mashup Platform** **Building Block** acts as a runtime environment for Mashable Composite UI Modules (widgets) and provides inter-**widget** communication, persistent per-widget configuration as well as proxy capabilities. The platform allows the user (in this case the person configuring a **CRISMA application**) to create a HTML5-based front-end by selecting certain user components (widgets and operations) that can be combined into a mashup. These user components are the **interface** for users to underlying back-end services. The user components themselves are provided by a 3rd party platform through a **catalogue**, from which a range of components can be installed into a mashup and connected ("wired") using a graphical editor.

Worldstate View

The Worldstate View **Building Block** is a set of generic Widgets (Mashable Composite UI Modules) that allow to visualise Control and Communication Information (**CCIM**) related to Worldstates and Worldstate Transitions. Thus, **it** operates mainly on common **meta-information** about the **world** rather than the real data of the world (**World State** Data Slots).

Multi Criteria Analysis and Decision Support View

BB Description:

The Multi **Criteria** Analysis View and Decision Support View is a User Interaction **Building Block** that allows performing a ranking of different **Crisis Management** Scenarios with respect to specific Criteria.

Together with the **Scenario** Analysis and Comparison View **it** realises the concepts for **Crisis Management Simulation Analysis** as defined by the **CRISMA Framework** **Architecture** and the **Model** for Decision-Making Assessment (D44.1, 2013). While the Scenario Analysis and Comparison View allows only a comparison of Indicators and Criteria for different scenarios, the Multi Criteria Analysis View and Decision Support View adds supplemental decision support functionalities. The Multi Criteria Analysis View and Decision Support View is composed of two different Widgets: The Decision Strategy **Widget** and the Decision Ranking Widget.

Decision Strategy Widget

The Decision Strategy Widget allows defining a weighting strategy for different Criteria to ultimately allow ranking of Crisis Management Scenarios in Multi Criteria Analysis. Thereby, a static weighting factor can be assigned to each **Indicator**. This factor specifies the contribution of the particular Criteria to the overall level of satisfaction of a particular Crisis Management Scenario. Additionally, a dynamic weighting factor can be specified which emphasises specific Criteria in relation to achieved level of satisfaction according to the **OWA** (Ordered Weighted Averages) method (refer to D44.1 for more details).

Decision Ranking Widget

The Decision Ranking Widget allows selecting a previously defined Decision Strategy and to apply it to a set of Crisis Management Scenarios. Crisis Management Scenarios are represented by **World** States (leaf nodes in a **World State** Tree). The Decision Support View can therefore be invoked from the Worldstate View (section 4.3.12) by selecting different World States for the comparison. After the Decision Strategy has been applied, the widget will present a list of World States ordered according to their level of satisfaction. Further visualisation and comparison of Indicators and Criteria can then be performed with help of the Scenario Analysis and Comparison View.

Documentation:

Specifications

Specification of the **concept** of multi **criteria** analysis.

- D32.2 ICMS Architecture Document V2
  http://www.crismaproject.eu/deliverables/CRISMA_D322_public.pdf (http://www.crismaproject.eu/deliverables/CRISMA_D322_public.pdf)

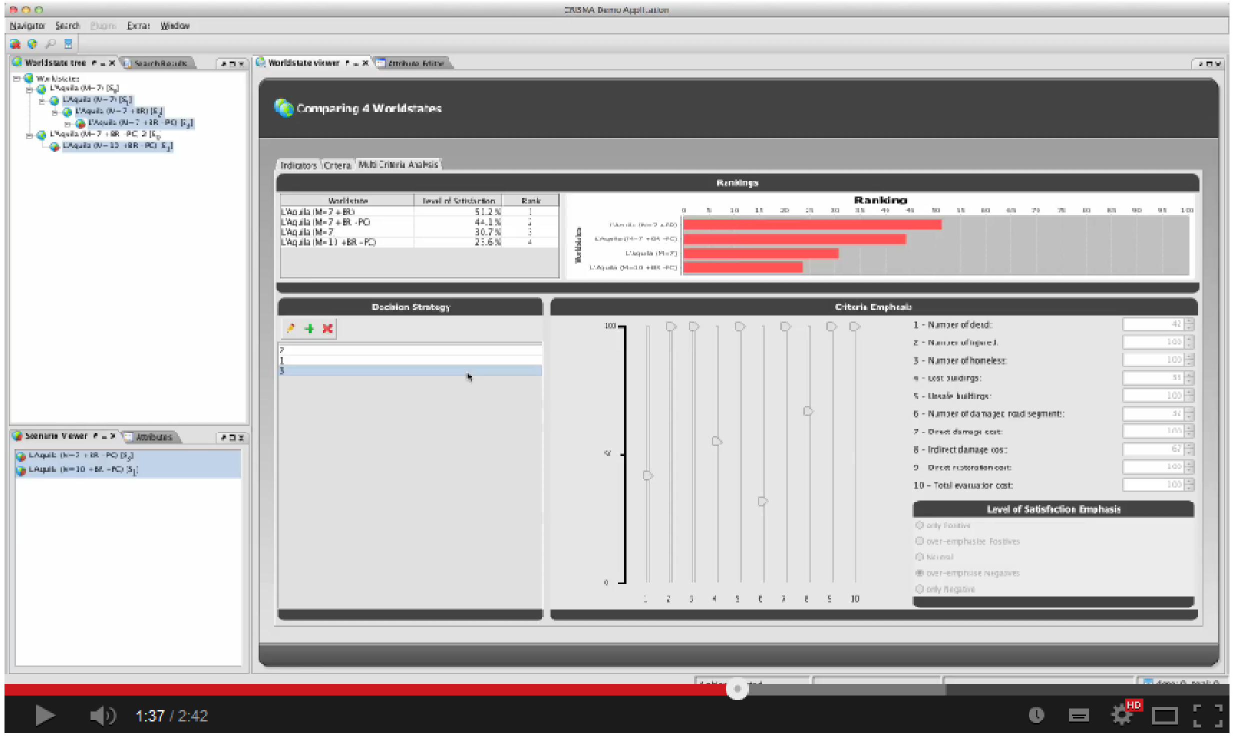
Demo

- Demonstrator Video for Scenario Analysis and DSS
  https://www.youtube.com/watch

Realization:

Multi Criteria Analysis and Decision Support Widget (Java)

This is the Java implementation of the Multi **Criteria** Analysis and Decision Support View.



Multi Criteria Analysis and Decision Support Widgets (JavaScript)

This is the JavaScript implementation of the Multi **Criteria** Analysis and Decision Support View **Building Block**.

Related Models and BBs:

Integrated Crisis Management Middleware BB

The **ICMM** is a central **Building Block** in every **CRISMA Application**. **It** connects **Crisis Management** Simulations with the Analysis and Decision Support functionality of CRISMA by providing a central repository for harmonized **world state** and **indicator** information. The ICMM is fed by simulations providing the basic information to be used for world state analysis and decision support Building Blocks. On an ICT conceptual level the ICMM is a generic distributed **resource**-oriented Control and Communication Information Management **System**. Thereby it is important to note, that the term ‘resource-oriented’ in the ICMM refers to the **concept** of generic resources as used in the context of the Resource Oriented **Architecture** (ROA).

Integrated Planning View

The Integrated Planning View **BB** is a generic integrated view for the configuration and inspection of arbitrary **crisis management** scenarios in planning situations. The **CRISMA** planning use cases (most use cases of CRISMA) intend to realize functionalities that are related to opening **world** states, changing them and storing them. Further running simulations, inspecting them and comparing results is addressed in all of them and the user work for planning purposes in a calm office environment.

Worldstate View

The Worldstate View **Building Block** is a set of generic Widgets (Mashable Composite UI Modules) that allow to visualise Control and Communication Information (**CCIM**) related to Worldstates and Worldstate Transitions. Thus, **it** operates mainly on common **meta-information** about the **world** rather than the real data of the world (**World State** Data Slots).

OOI Management UI Component

Version: 1

This **Building Block** enables to view and edit the actual data available for specific **scenario** or **simulation**. **It** enables the **system** administrator to detect the **OOI** data type's properties and based on the OOI Information Models definitions. It also enables to monitor and edit a specific snapshot context (time, user, and workflow).

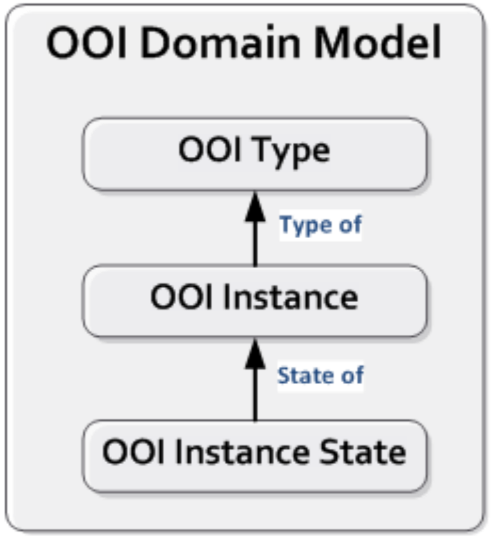
The OOI Editor Building Block has a generic UI for creating and instantiating different Data Types. It enables creating new instance from an existing data type or editing an existing **object** instance. Support for new data types can be done by creating a dedicated plugin that contains data adapter and an optional UI data visualization.

The relation between the different entities described in the following diagram:

The user **interface** will be constructed from two sub views:

OOI Types Editor – Create new OOI types and it associate properties.

OOI Types holds the following data:



- Unique identifier
- Textual description.

Each OOI Type property holds the following data:

- Unique identifier
- Textual description.
- Data type: Number, String, DateTime, Boolean, Geometry or Geography
- **Validation** rules (required/optional, min/max occurrences, length, etc.)

Screenshot:

OOI Instance Editor - Create new OOI instances that later can be used as simulation initial **world state**.

Screenshot: Noy yet avilable...

Documentation:

Implementation Plan

- Implementation Plan **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP34/01%20Work%20in%20Progress/Software%20Components/OOI-Editor%20View%20BB/OOI%20Editor%20-%20BB%20Implementation%20Plan.doc**

Demo

A guided tour that ilustrate the UI usage

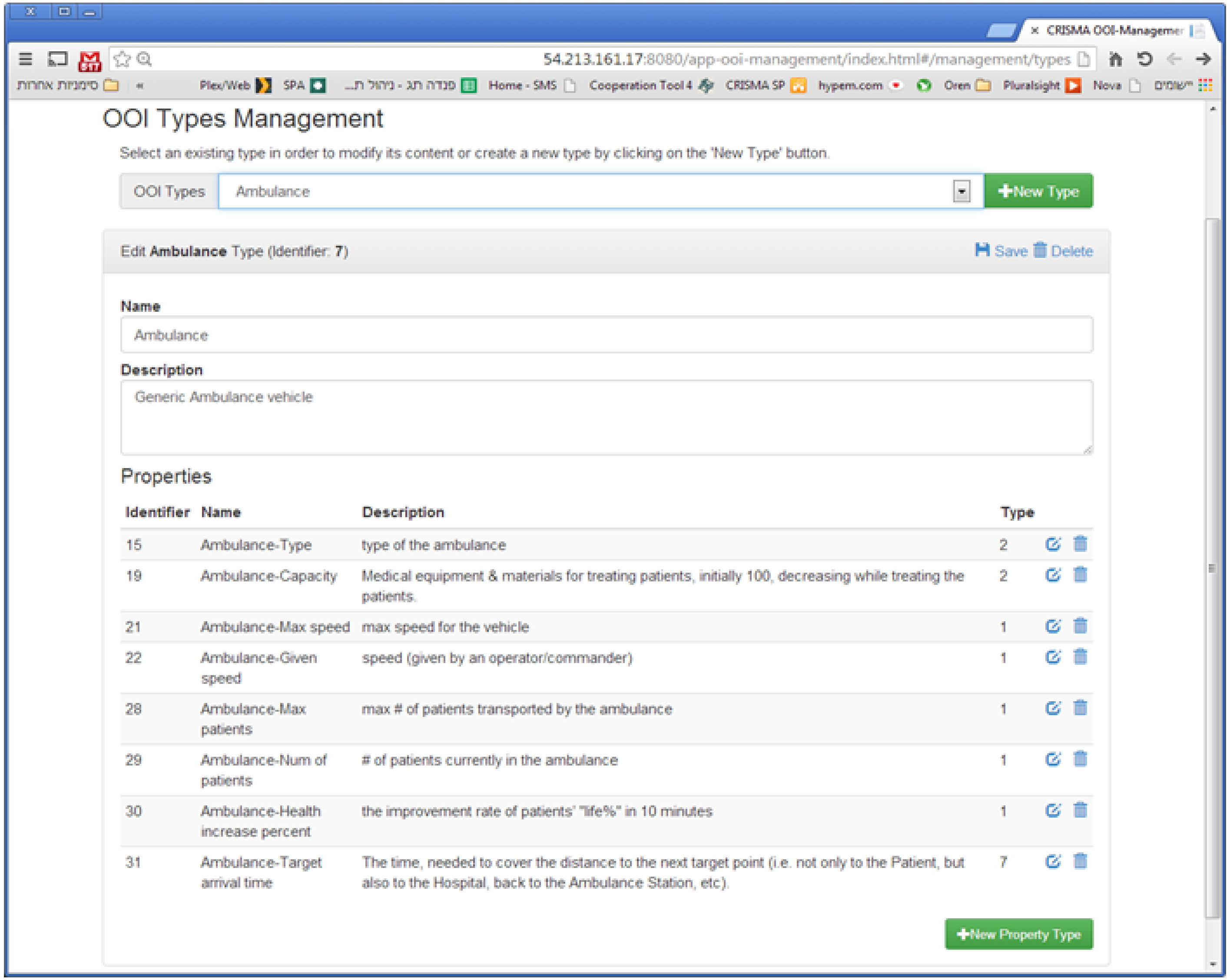
Downloads:

- Online Application **http://54.213.161.17:3030/app-ooi-management/index.html (http://54.213.161.17:3030/app-ooi-management/index.html)**

Team:

Oren Deri

OOI Management View



BB Description:

This **Building Block** enables to view and edit the actual data available for specific **scenario** or **simulation**. **It** enables the **system** administrator to detect the **OOI** data type's properties and based on the OOI Information Models definitions. It also enables to monitor and edit a specific snapshot context (time, user, and workflow).

The user interface will be constructed from two sub views:

OOI Types Editor – Create new OOI types and associate properties.

Suggested Mockup:

OOI Instance Editor - Create new OOI instances that later can be used as simulation initial **world state**.

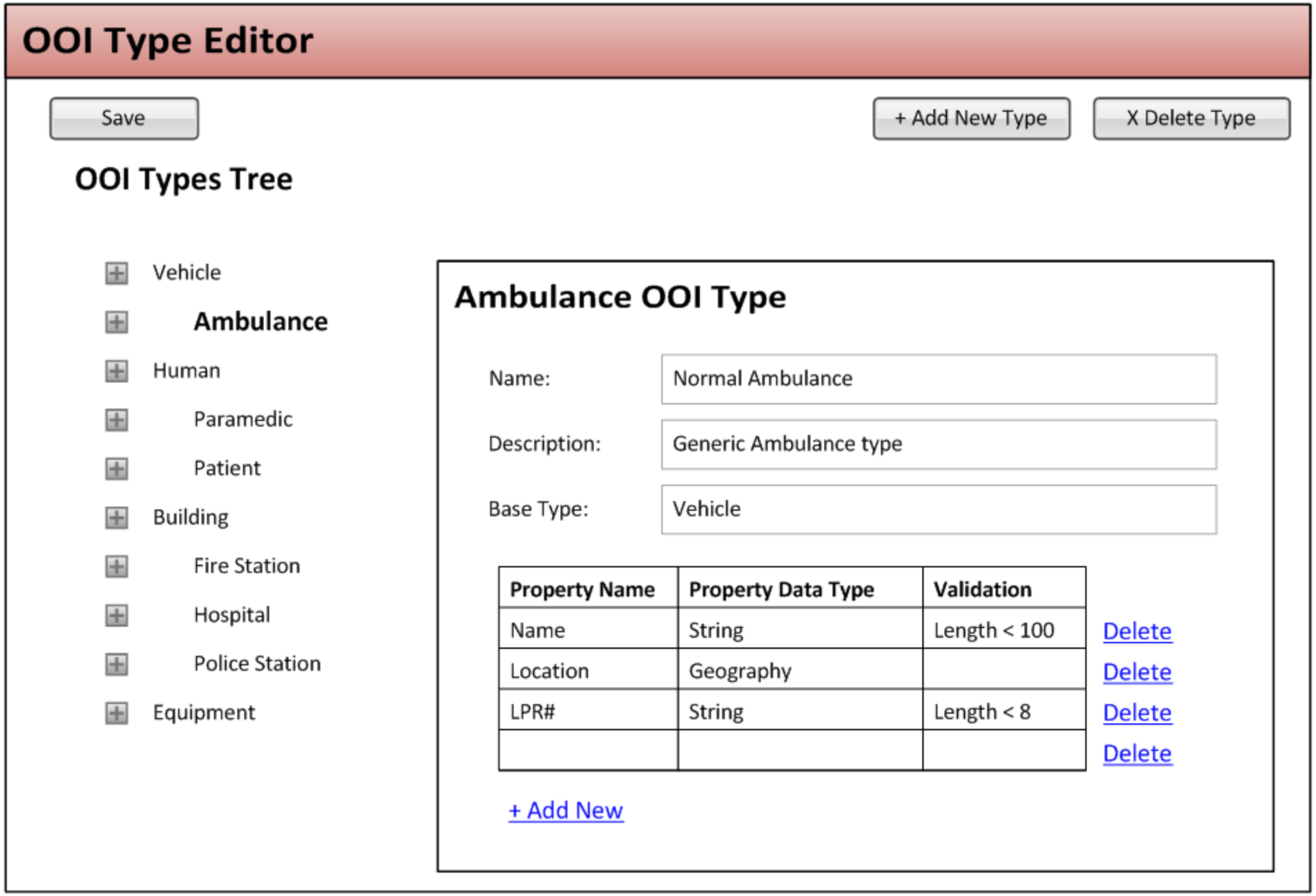
Suggested Mockup:

Rationale

For systems that support plug and play of models an OOI editor is a must have to define the data required by the **Model** and specify the instances for specific simulation execution. In opposite to a normal fixed Models simulation CRISMA needs an OOI Editor that can easily adapt new data types according to new OOI types exposed by the **ICMM**.

Key Functionality

The need for this Building Block can be derived from the following **user requirements**:



- Provide unified and flexible way to model OOI data in a human readable format.
- Support the generic approach to be able to support the very different **pilot** applications as well as applications beyond in the **crisis management** context.
- Needed in order to run any simulation that is OOI aware. Specifically for training system that involves **resource management** capabilities.

Documentation:

Functional Description

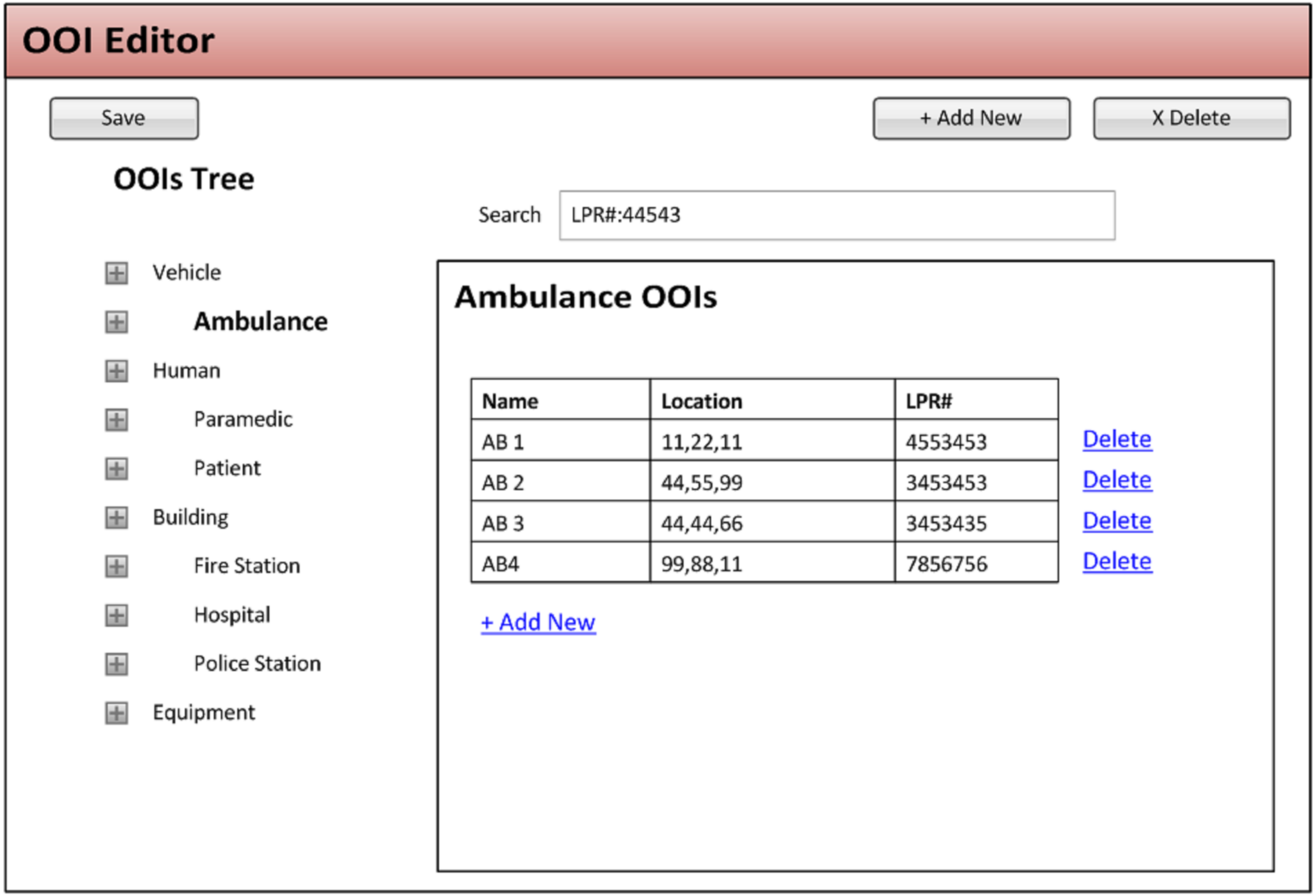
Functional description can be found as part of the formal **Building Block** description.

- Building Block Description Document **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20Functional%20Building%20Blocks/v2.0/OOI%20Editor%20Building%20Block%20Description%20V3.doc**

Realization:

OOI Management UI Component

This **Building Block** enables to view and edit the actual data available for specific **scenario** or **simulation**. **It** enables the **system** administrator to detect the **OOI** data type's properties and based on the OOI Information Models definitions. It also enables to monitor and edit a specific snapshot context (time, user, and



workflow).

Related Models and BBs:

OOI World State Repository BB

OOI-WSR is a **Resource Management** related **Building Block** that enables archiving, querying and manipulation of Objects of Interest (OOI) **world state** data. This **BB** serves as a Repository **service** for OOI data that can be consumed or manipulated by other interaction or functional building blocks and resource management models.

Publish Subscribe Context Broker BB

The Publish Subscribe Context Broker **Building Block** is a cross-over between an **event** broker which accepts events and dispatches them to subscribers and an access **service** providing the information on current state of the "world". Thereby, the Publish Subscribe Context Broker Building Block can be used to realize the event subscription and event delegation functionality of the **ICMM**. Thus, the Building Block is both suitable for dispatching events related to Control and Communication Information (**CCI**) managed by the ICMM and events related to (Worldstate) Data managed by arbitrary data access services, e.g. the **OOI** **World State** Repository **BB** or implementations of the Data Integration BB (**WMS**, **WFS**, …).

Resource Management Tactical Training BB

Resource Management Tactical Training **Building Block** (RMTT **BB**) simplifies the **task** of designing the Tactical Training applications for control room operator and on-scene commanders. RMTT enables a Trainee to learn **emergency management** by assigning tasks to various resources and analysing the results in a virtual environment.

Resource Management Model

The **Resource Management** Models developed in the **CRISMA** project are built upon the **OOI** **concept** with different context dependent behavioral patterns for different **crisis** domains. Thus, there is no overall generic and all-purpose Resource (OOI) Management **Model**, but a set distinct models for different types of resources (e.g. ambulances, patients) and different situations. However, such domain and crisis specific Resource Management Models can be implemented on basis of the general **Agent**-Oriented **Simulation** Models **Building Block**, with its functionalities described in D312 .

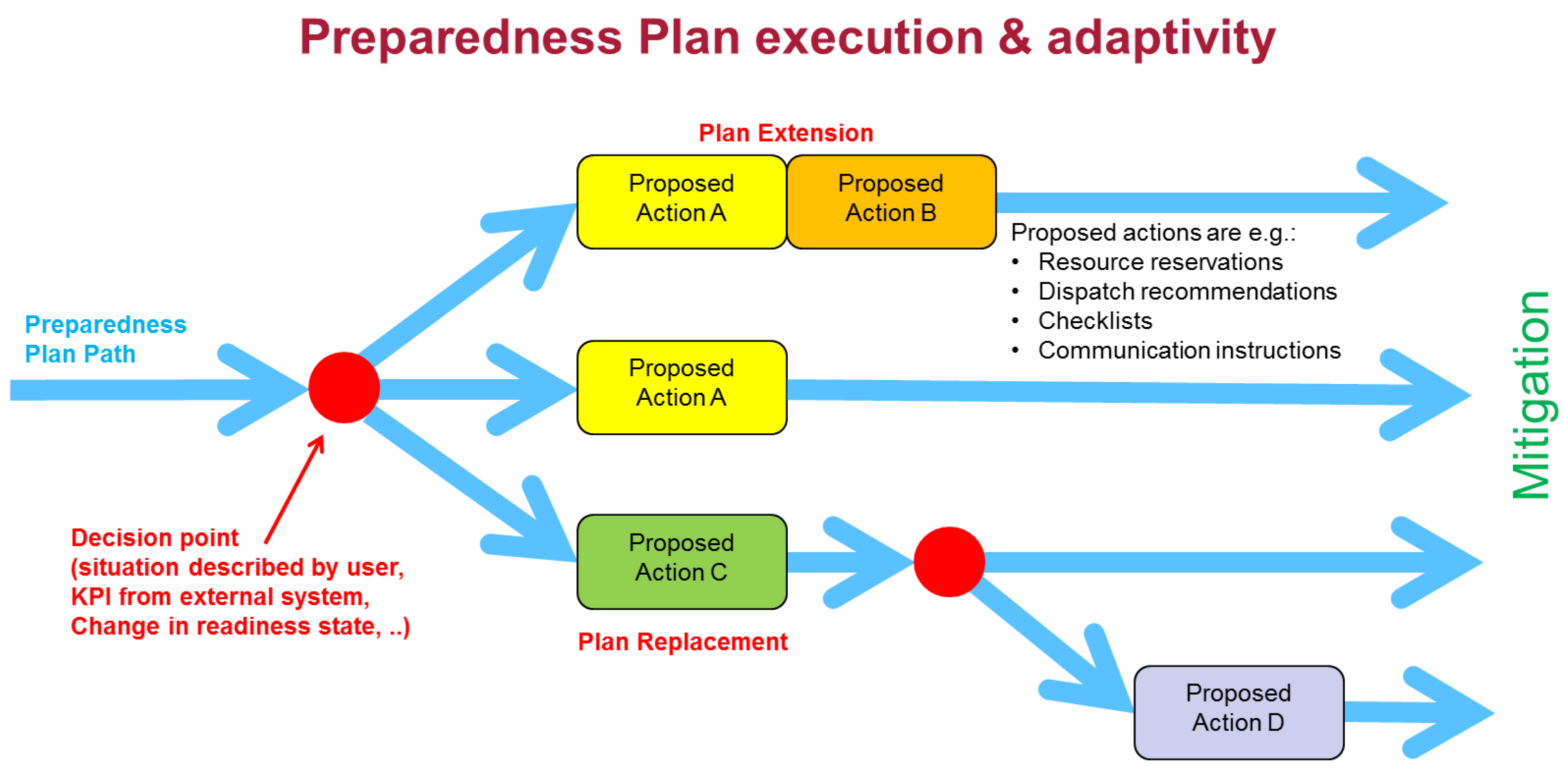
Preparedness Plan BB

BB Description:

The **Preparedness** Plan **Building Block** is a decision support mechanism, which helps the decision maker to take the needed actions in case of **emergency** according to plans based on analysis of threats, vulnerabilities and possible emergency scenarios.

Different phases of emergency require different kinds of actions. For example, in the early **warning** phase the actions can be reservations of resources according to existing contracts. If the **situation** further evolves to an actual emergency, the proposed actions may be e.g. dispatching of the **response** units or taking actions needed for **evacuation** of the affected people. Proposed actions for a certain situation may be provided, for example, as check lists or communication plans.

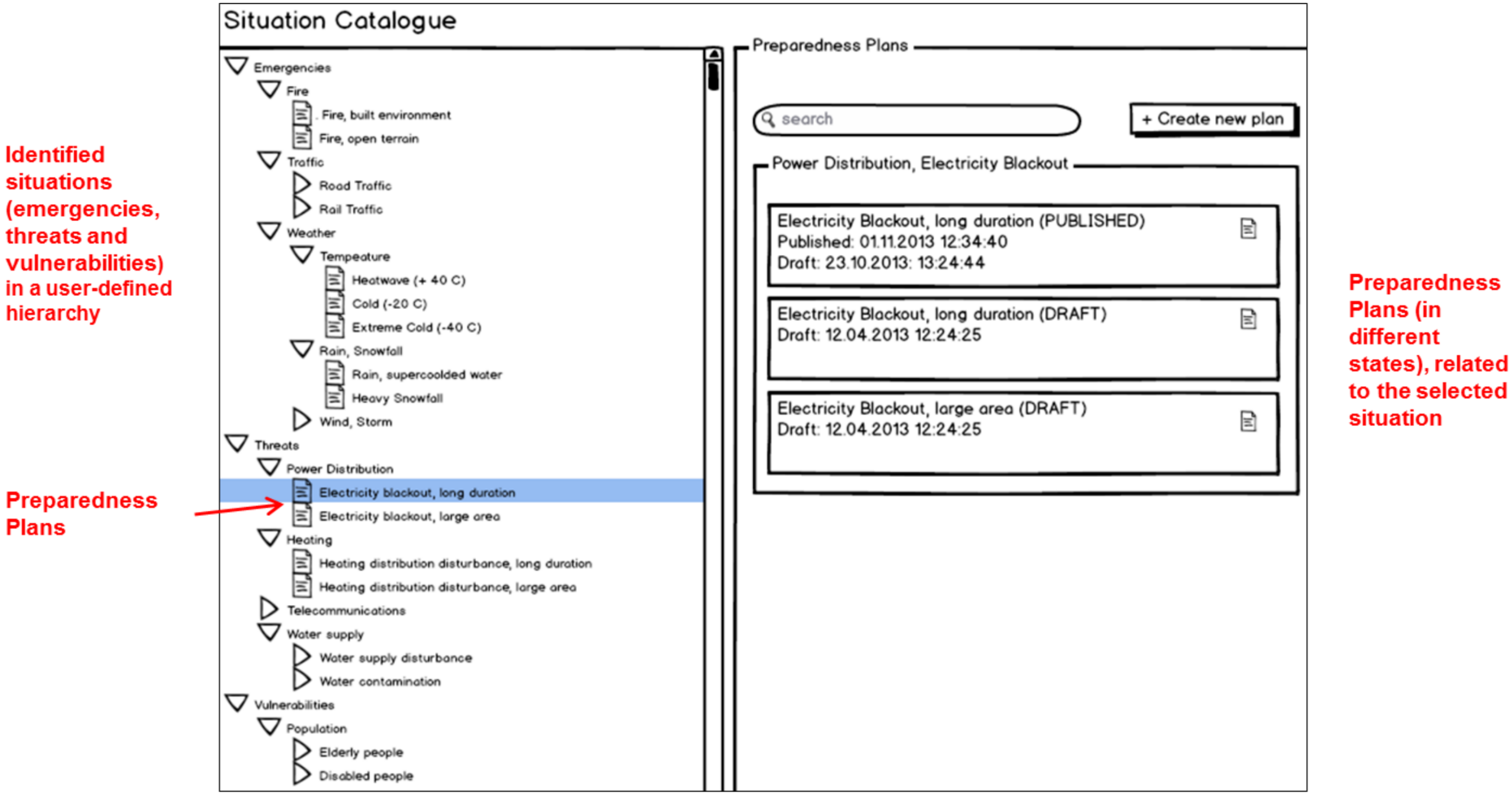
The planned scenarios are maintained as a Situation **Catalogue**, which consists of identified threats, vulnerabilities or possible emergencies. Each of the Situation Catalogue item describes one lowest level situation or phenomenon which requires some specific action. The items may occur simultaneously or can be alternative to each other. The actual situation is thus a set of individual situation catalogue items. Each item may add, modify or remove some proposed action from the instructions given to the user.



Preparedness Plan adaptivity during execution according to decision points.

Maintaining the scenarios in the Situation Catalogue (UI mockup).

The proposed actions can be described as instructions, checklists or **resource** allocations with the required capabilities based on earlier made resource reservations or contracts. The resources for response and **recovery** tasks in different scenarios are allocated according to pre-planned patterns and up-to-date resource and situation data. In the preparedness planning phase the possible response patterns are created for different emergency scenarios. During the actual emergency situation, the **BB** suggests the actions to be taken to the decision maker. These decision points can be either multiple-choice questions to which the user answers, free-text forms where the user describes the situation or analysis points where the proposed action is based on data provided by an external analysis tool or **model** (e.g. population data, weather information).



The proposed actions evolve according to user choices during the execution of the Preparedness Plan (UI mockup).

The Preparedness Plan building block consists of the Preparedness Plan **Service**, which implements the storage of the preparedness plans as well as the **business logic** to calculate the required response and proposed actions. User can access and use the Preparedness Plan Service using a web-based user **interface**. The Preparedness Plan Service connects to **OOI**-WSR BB to access situation and resource data and **it** may utilize results of the **simulation** models, also stored in to the OOI-WSR.

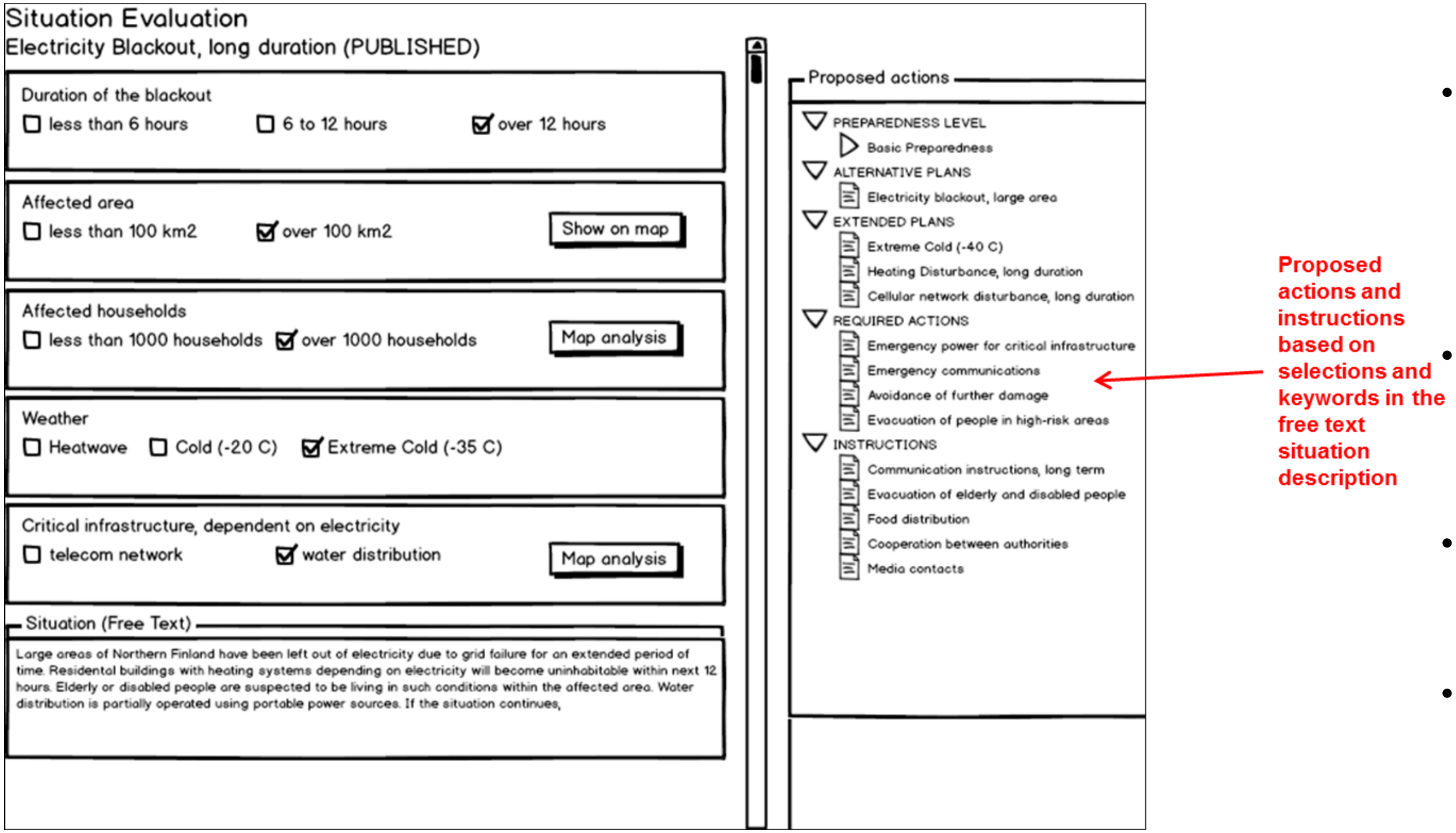
Rationale

The Preparedness Plan building block is needed to provide means to plan response tasks and actions and to provide decision support according to pre-defined response patterns. Preparedness Plan BB provides resource allocation alternatives to be choosen by the user.

Documentation:

Specifications

The **Preparedness** Plan **BB** is a generic decision support tool, which helps the decision maker to allocate resources for **response** and **recovery** tasks, according to pre-planned patterns and up-to-date **resource** and **situation** data.



Key functionality:

- The Preparedness Plan BB shall provide decision support to the user on taking actions (e.g. allocating response resources, informing other relevant parties) related to an regional **emergency** situation. (Mandatory)
- The Preparedness Plan BB shall enable the creation of a pre-defined narrative (i.e Preparedness Plan) as a response to different emergency situations. (Mandatory)
- The Preparedness Plan BB shall provide to the user information on available response resources and other operative bodies
- The Preparedness Plan BB shall enable building of Emergency Plans using real data on **exposure**, **hazard**, **capacity** resources etc.
- The Preparedness Plan BB shall enable ranking of different risks in the same area to identify the critical points to support planning
- The Preparedness Plan BB shall provide a map-based **capability** to plan deployment of the response bodies involved in emergency operations
- The Preparedness Plan BB shall enable a map-based, real-time visualization of the deployment of the search and rescue squads
- The Preparedness Plan BB shall enable a map-based, real-time visualization of the health care facilities in the surroundings of the affected areas
- The Preparedness Plan BB shall provide means to determine which resources are needed to handle incidents with respect to **impact** / damage scenarios
- The Preparedness Plan BB shall take into account the factors influencing response capacity (e.g. weather, traffic, driving speed, availability of equipment)
- The Preparedness Plan BB shall take into account the population data (demographics, **vulnerability**)
- The Preparedness Plan BB shall take into account the regional characteristics of the territory and the presence of the special infrastructures

Realisation:

The **Insta Preparedness Planner** implements the functionalities related for creating and maintaining the Situation **Catalogue** consisting of a hierarchy of vulnerabilities, threats and possible emergency situations which are relevant to the area of operation. Each item in the Situation Catalogue consists of several elements (decision points), each of which changes the combination of the proposed actions to mitigate the situation. The proposed action can be e.g. resource reservations with the required capabilities, communication plans, check lists or instructions to handle a specific situation.

Realization:

Insta Response Preparedness Planner

Insta **Response** **Preparedness** Planner is a decision support tool, based on existing **situation** assessment functionalities of the Insta Response product family. Preparedness planning is based on creation of a pre-defined narrative (i.e. the Preparedness Plan) as a response to different kinds of **emergency** or otherwise exeptional situations.

Related Models and BBs:

GIS View Building Block

The **GIS** View is a User Interaction **Building Block** that enables the visualisation and manipulation of geospatial data. Geospatial data plays a predominant **role** in all **crisis management** related applications, because most if not all information playing a role in crisis management has a geospatial component.

Simulation Model Integration BB

The **Simulation Model** Integration **Building Block** provides components that can be used to easily enable simulation models to participate in a **CRISMA Application**. Due to the heterogeneity of existing simulation models CRISMA has to ensure that they can be integrated in a standardized way and ultimately become a **CRISMA Federate**.

That way they are made CRISMA-aware. The envisioned technique to be used is that of so called wrapping so that this Building Block provides a piece of software that can be used to make (existing) simulation models CRISMA-aware with as little effort as possible with respect to the individualities of the different simulation models.

UI Mashup Platform

The **Mashup Platform** **Building Block** acts as a runtime environment for Mashable Composite UI Modules (widgets) and provides inter-**widget** communication, persistent per-widget configuration as well as proxy capabilities. The platform allows the user (in this case the person configuring a **CRISMA application**) to create a

HTML5-based front-end by selecting certain user components (widgets and operations) that can be combined into a mashup. These user components are the **interface** for users to underlying back-end services. The user components themselves are provided by a 3rd party platform through a **catalogue**, from which a range of components can be installed into a mashup and connected (“wired”) using a graphical editor.

Evacuation Resources Simulation Model

Evacuation Resources **Simulation Model** calcuates the **impact** of **resource** allocations chosen by the user (from the proposals based on the **preparedness** plan) to mitigate the **situation**.

Resource Management Tactical Training BB

BB Description:

Resource Management Tactical Training **Building Block** (RMTT **BB**) simplifies the **task** of designing the Tactical Training applications for control room operator and on-scene commanders. RMTT enables a Trainee to learn **emergency management** by assigning tasks to various resources and analysing the results in a virtual environment.

Operational Concept

The following figure illustrates a high level description of user interaction with the RMTT BB:

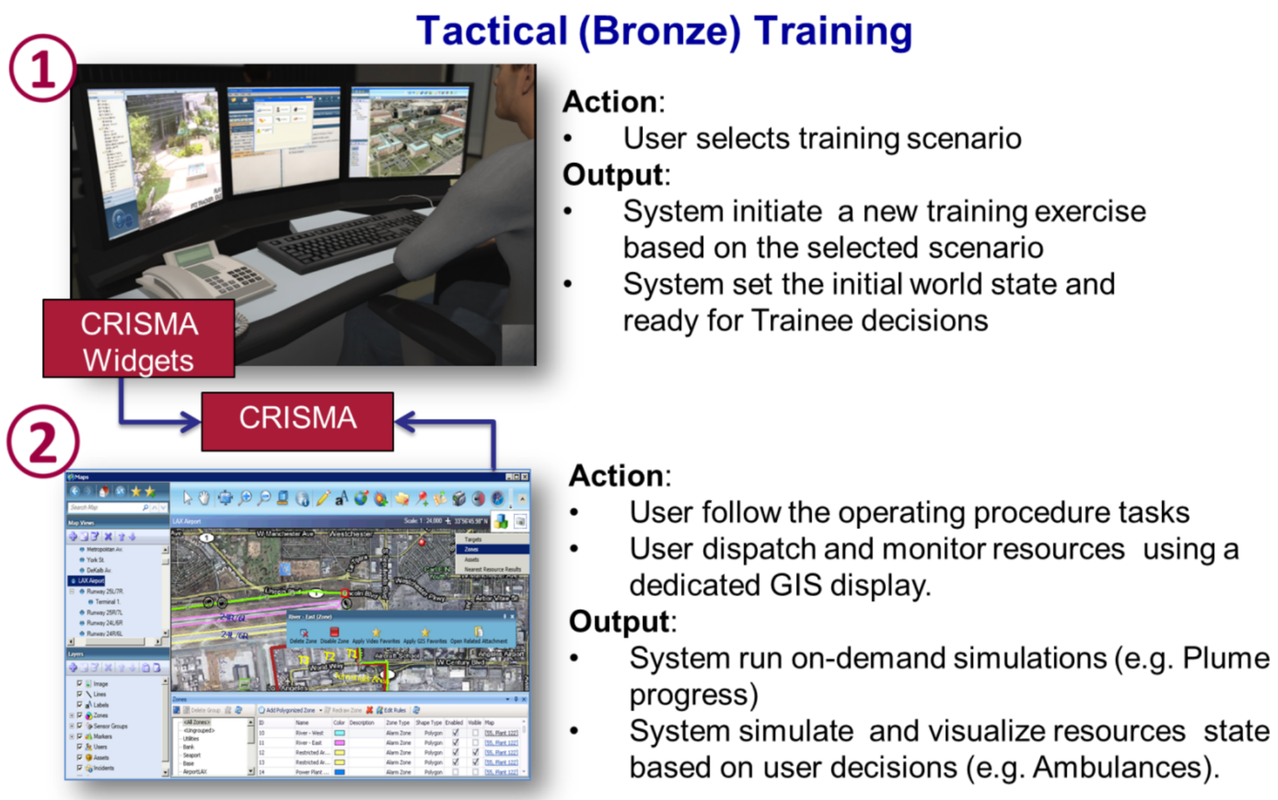
Implementing the RMTT involves the integration of UI views from both CRISMA UI views and **NICE** Situator UI views. The following views will be used:

Situator Assets Real-Time Management View

A highly customizable built-in console for tracking the location as well as monitoring the condition and environment surrounding an organization’s assets or resources. Asset types include human (employees, visitors, outside contractors), vehicles, equipment (such as GPS tracking devices), and any variety of custom asset types, such as buildings.

Situator Resource Real-Time Management View

A decision support and resource dispatching solution. Using the built-in resource management capabilities, dispatchers can view the current state of all organizational resources (sensors/responders/vehicle and human assets) in order to dispatch responders effectively when managing an **incident**. By analysing the activity and availability of responders in the area, the dispatcher can determine which responder is available to handle the **situation**, assign the responder to the incident and dispatch accordingly. Resources (vehicle and human assets) are assigned dynamic parameters, "availability" and "activity". These parameters help dispatchers better manage an incident by viewing if associated assets are available to be deployed. In Control Room, dispatchers can locate and associate nearest resources with incidents



CRSIMA OOI Monitoring and Dispatching View
See **BB description**.

CRISMA OOI Management View
See **BB description**.

Documentation:

Functional Description

Functional description can be found as part of the formal **Building Block** description.

directly from the Incidents Log, the relevant map or from the Nearest Resource Results map **GIS** module.

Situator Asset Editor
Enable to define new asset types. Only asset types defined in the Planning Tool are available in the Control Room. **NICE** Situator includes two built-in asset types – Vehicles and Human Assets. Administrators can dynamically add new asset types customizing the module to fit the specific needs of the organization. In addition, asset Subtypes can be added to Asset Types in Planning Tool. For example, under the Asset Type of Vehicles, subtypes Trucks, Patrol Cars, and Ambulances can be added. In Planning Tool, users can choose to show or hide any of the built-in Asset Type parameters. The hidden parameters do not display in Control Room. For example, if you hide the **parameter** 'Job Title' in Planning Tool, when creating a new human asset in Control Room the option to define that asset's Job Title will not be available. In addition, this parameter will not show when viewing the properties of a human asset in both the Map view and the Asset Management screen.

- Building Block Description Document **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20Functional%20Building%20Blocks/v2.0/Resource%20Management%20Tactical%20Training.doc**

Realization:

Situator Training System

Situation Management is a solution approach that comprises tools and methods for coordinating the interaction between people, technologies, and responses. **NICE** Systems is a major player in the Situation Management market with over 20 years in experience in the industry and expertise in a wide variety of industries such as seaports, airports, railways, banking, government, telecom, utilities, military, manufacturing, etc.

Related Models and BBs:

OOI Management View

This **Building Block** enables to view and edit the actual data available for specific **scenario** or **simulation**. **It** enables the **system** administrator to detect the **OOI** data type's properties and based on the OOI Information Models definitions. It also enables to monitor and edit a specific snapshot context (time, user, and workflow).

OOI World State Repository BB

OOI-WSR is a **Resource Management** related **Building Block** that enables archiving, querying and manipulation of Objects of Interest (OOI) **world state** data. This **BB** serves as a Repository **service** for OOI data that can be consumed or manipulated by other interaction or functional building blocks and resource management models.

Resource Management Model

The **Resource Management** Models developed in the **CRISMA** project are built upon the **OOI** **concept** with different context dependent behavioral patterns for different **crisis** domains. Thus, there is no overall generic and all-purpose Resource (OOI) Management **Model**, but a set distinct models for different types of resources (e.g. ambulances, patients) and different situations. However, such domain and crisis specific Resource Management Models can be implemented on basis of the general **Agent**-Oriented **Simulation** Models **Building Block**, with its functionalities described in D312 .

Resource Management Training Dispatch and Monitor View

Authors: **Peter Kutschera**
Manuel Warum

BB Description:

The Dispatch and Monitor view provides a high-level overview over the **resource management** **simulation**'s **world state**. Its purpose is to display one world state at a time and allow the user to distribute resources (ambulances, etc.) among different areas where the **crisis** plays out.

Trainee operators are presented with the number of resources available at the current time in the resource management simulation, and also to some degree about the incidents and **OOI** states (depending on whether the administrator chose not to hide these details from trainees).

The map contains a predefined number of incidents depending on how **it** was set up. The operator can dispatch resources to scenes if they are available, or recall already dispatched resources. The operator is also always informed about the state of each resource at his disposal as well as the **incident** evolvement (aggregated patient states, severity of incidents, etc).

Trainee operators can perform micromanagement and observation tasks related to a smaller sub-area of the overarching **crisis scenario** map using the resources assigned to him/her by a trainer.

The operator is able to see all resources that were assigned to his site and their individual state, as well as the location and state of incidents (patients, fires, etc.).

The figure above illustrates the functionality of the Monitor and Dispatch view **building block**. It displays objects of interests in a table as well as on the map and allows the user to issue commands to individual OOIs.

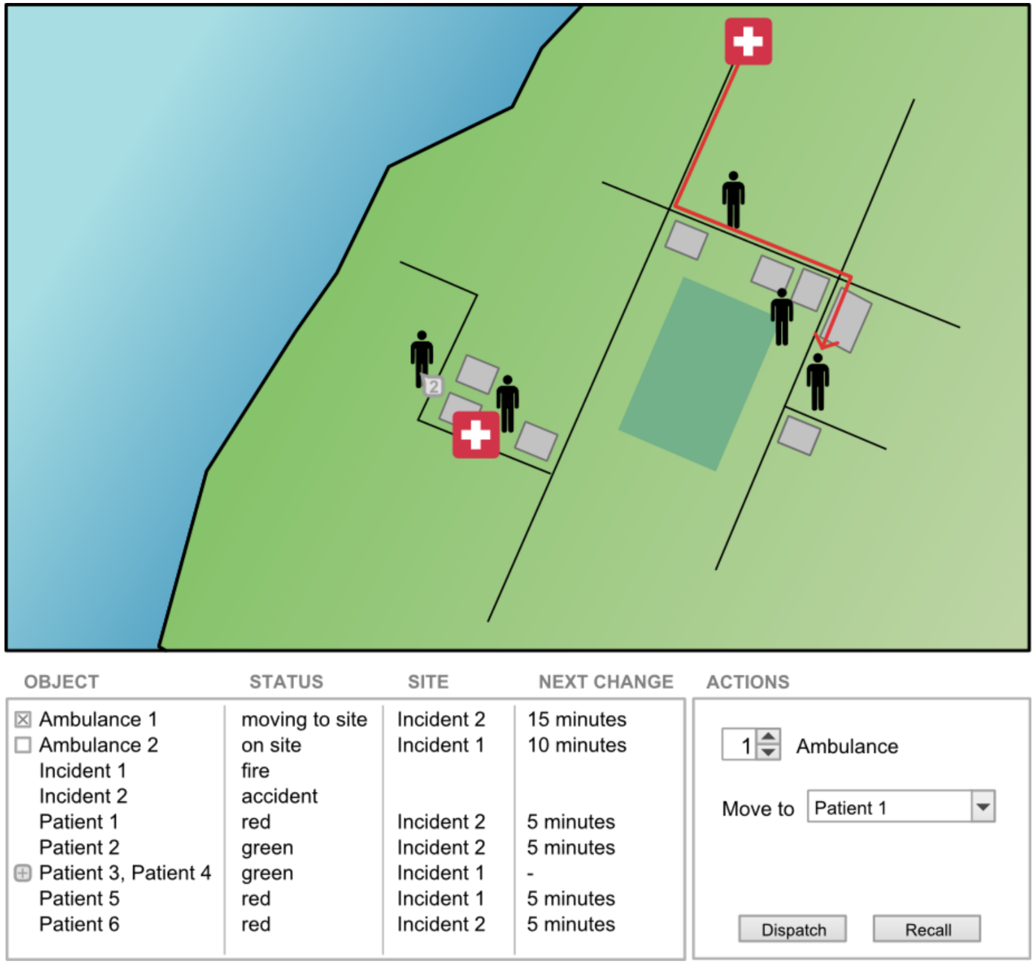
Rationale

This Building Block is one of the most central views for resource management training tasks.

Note that this is Building Block is merely a view and thus performs no calculations on its own. All processes and calculations are assumed to be provided to the view by other Building Blocks.

Key Functionality

1. The RM Training Dispatch and Monitor View Building Block is capable of rendering map data including overlays for objects of interest instances, hazards, and shapes.
2. The RM Training Dispatch and Monitor View Building Block allows the selection of OOI instances individually and issue orders
3. The RM Training Dispatch and Monitor View Building Block allows the selection of OOI instances as a group
4. The selection of OOI instances (FR-02, FR-03) can be achieved either via the map or the tabular view of OOI instances itself
5. Actions can be invoked on selected OOI instance(s) if their type supports it



6. Individual OOI instances (resources, patients, etc.) are represented in a table depicting their current state and properties associated with their OOI type.
7. The user can select a simulation instance.
8. The user can advance the time in the current simulation to proceed to the next worldstate
9. The user can navigate to the RM Training Indicators and Statistics View **BB**, or the RMT I&S View BB is already embedded in this view

Documentation:

Specifications

1. The view must present the current **world state** to the operator. This includes a table listing each **resource** available in the current world state. The table also contains information about the resource's availability, current status, how long they have been on the scene or when they will arrive; in addition, the table also lists patients and their current condition.
2. In addition to (1), a visual representation on a map indicating each **object**'s geographical position is required as well. This map includes incidents, patients, hazards as well as resources currently part of the world state.
3. The operator can assign, reassign, or unassign resources. Depending on the type of a selected resource, additional actions may be available.

Realization:

Wirecloud OOI GIS Map

A map **widget** showing **Object Of Interest** (**OOI**) on a background map.

Wirecloud WorldState Picker

This **widget** lets the user pick and load a **world state** from the Objects of Interest World State Repository (**OOI**-WSR).

It displays all world states and their hierarchy to the **end user**, allowing him/her to select the one that should be continued.

Wirecloud OOI Table

This **widget** shows all OOIs associated with a worldstate in a tabular form. **It** is possible to filter which elements should be visible (eg. only OOIs of certain types such as Ambulances and Hospitals). In addition, it also allows the user to group any OOIs for convenience; groups are persisted on the client-side using HTML5 storage mechanics and are automatically restored inbetween sessions.

Wirecloud Simulation Picker

A **widget** showing all simulations available on the **Object of Interest** **World State** Repository (**OOI**-WSR). **It** allows the user to select one of these simulations and load it, which will initialize other connected widgets and gadgets of the Wirecloud **application** with relevant data.

Wirecloud OOI Commands

Widget that allows to issue commands to one or more OOIs. The commands available depend on the OOIs' types; for instance, Ambulances can treat in a specific area. In addition, there is also a support for "global" commands, ie. commands that do not require an **OOI** (such as "create area").

Related Models and BBs:

UI Mashup Platform

The **Mashup Platform** **Building Block** acts as a runtime environment for Mashable Composite UI Modules (widgets) and provides inter-**widget** communication, persistent per-widget configuration as well as proxy capabilities. The platform allows the user (in this case the person configuring a **CRISMA application**) to create a HTML5-based front-end by selecting certain user components (widgets and operations) that can be combined into a mashup. These user components are the **interface** for users to underlying back-end services. The user components themselves are provided by a 3rd party platform through a **catalogue**, from which a range of components can be installed into a mashup and connected (“wired”) using a graphical editor.

Resource Management Training Indicators and Statistics View

BB Description:

The **Resource Management** Training Indicators and Statistics View **Building Block** is a User Interaction Building Block that will be realised as Mashable Composite UI Module. **It** focuses on the visual presentation of statistics and key indicators of a given Worldstate in order to provide a quick overview of the **situation** and to allow for comparison between any two given Worldstates.

Note that the component performs no complex calculations by itself. It only takes data either directly from the Worldstate, or from other building blocks providing any such indicators unless they are not already part of the Worldstate (e.g. the Indicators Building Block).

The figure above illustrates a comparison between two world states where patient health as well as utilization of dispatchable resources are compared side-by-side.

Key Functionality:

1. The Resource Management Training Indicators and Statistics View **BB** creates a visualization of **OOI** states.
2. The Resource Management Training Indicators and Statistics View BB creates a visualization of resource states.
3. The Resource Management Training Indicators and Statistics View BB allows the side-by-side comparison of indicators from different Worldstates.
4. The creation of diagrams is triggered manually through user interaction.
5. The creation of diagrams is triggered automatically by events (e.g. Worldstate changes).
6. Data is visualized e.g. as bar diagrams.

Rationale:

The CRISMA resource management **concept** demands an easy-to-grasp representation of the state and the evolution of resources in a given **simulation** by visually highlighting the most important details. Resource Management tasks require benefit from the visualization of patient resource states at a given point in time as well as the distribution of available resources within a **simulation run**. There is no need for complex number aggregation or arithmetic operations, which allows these visualizations to run on the client and on demand.

Documentation:

Specifications

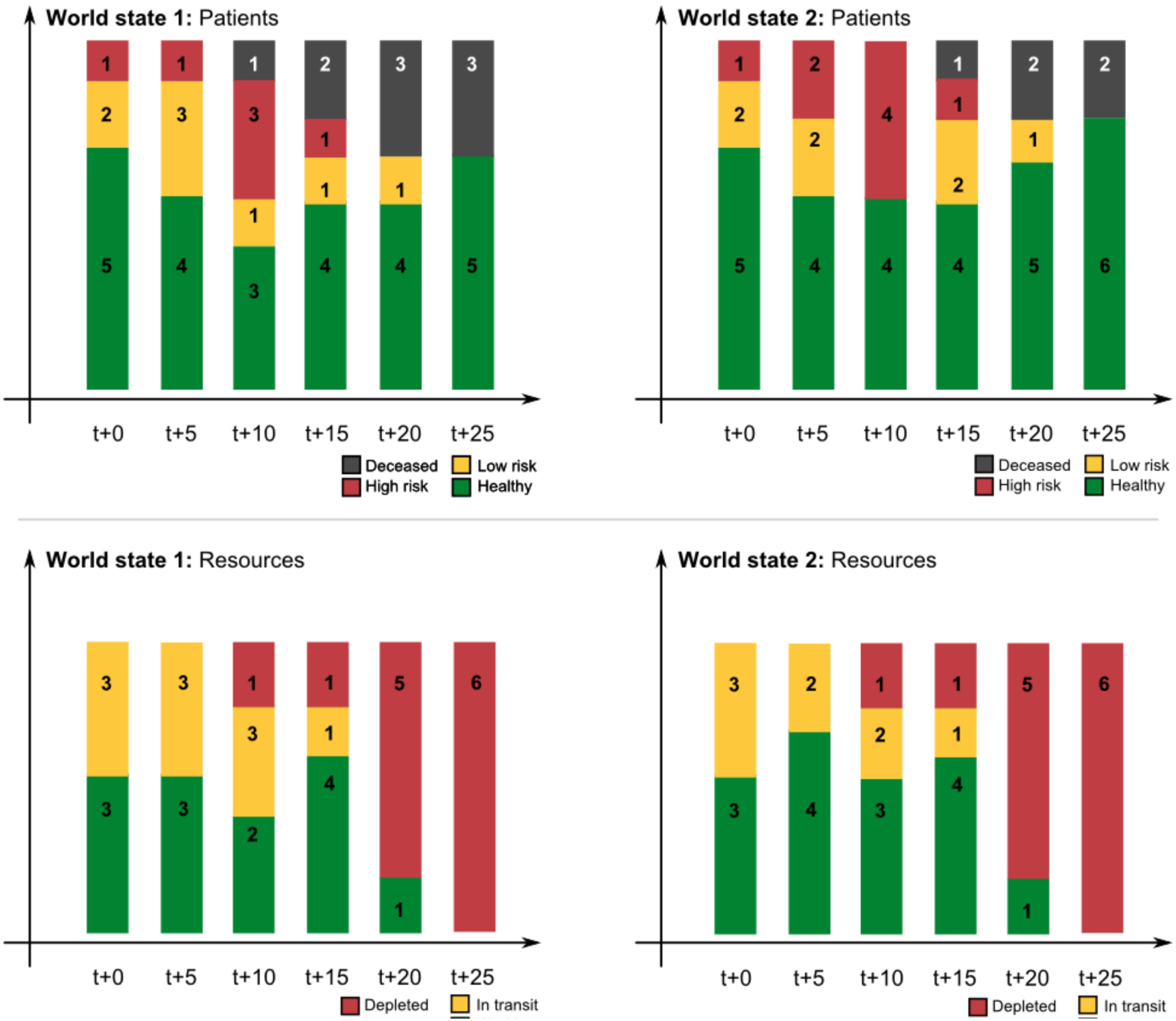
Representation of two different sets of indicators:

1. The state of OOIs
2. The state of resources.

Realization:

Wirecloud Charts

This **widget** displays stacked bar charts for **indicator** values for a given worldstate.



Wirecloud WorldState Picker

This **widget** lets the user pick and load a **world state** from the Objects of Interest World State Repository (**OOI**-WSR).

It displays all world states and their hierarchy to the **end user**, allowing him/her to select the one that should be continued.

Working Working

Wirecloud Simulation Picker

A **widget** showing all simulations available on the **Object of Interest** **World State** Repository (**OOI**-WSR). **It** allows the user to select one of these simulations and load it, which will initialize other connected widgets and gadgets of the Wirecloud **application** with relevant data.

Related Models and BBs:

Indicator Building Block

The Indicators **Building Block** (formerly Algebraic **Evaluation**) is a component that allows definition, storage and evaluation of "simple" algebraic models in order to evaluate consequences of decisions made by **CRISMA** users during a training session. In this context "simple" means that there is no large number crunching needed. Usage is for calculation of key performance indicators, heuristic models and to some extend the implementation of models actually running in interactive **GIS** environments in order to make them usable as services in the **CRISMA framework**.

Integrated Planning View

The Integrated Planning View **BB** is a generic integrated view for the configuration and inspection of arbitrary **crisis management** scenarios in planning situations. The **CRISMA** planning use cases (most use cases of CRISMA) intend to realize functionalities that are related to opening **world** states, changing them and storing them. Further running simulations, inspecting them and comparing results is addressed in all of them and the user work for planning purposes in a calm office environment.

UI Mashup Platform

The **Mashup Platform** **Building Block** acts as a runtime environment for Mashable Composite UI Modules (widgets) and provides inter-**widget** communication, persistent per-widget configuration as well as proxy capabilities. The platform allows the user (in this case the person configuring a **CRISMA application**) to create a HTML5-based front-end by selecting certain user components (widgets and operations) that can be combined into a mashup. These user components are the **interface** for users to underlying back-end services. The user components themselves are provided by a 3rd party platform through a **catalogue**, from which a range of components can be installed into a mashup and connected ("wired") using a graphical editor.

Resource Management Training Simulation Scenario Setup View

BB Description:

The **Resource Management** Training **Simulation** **Scenario** Setup View is a User Interaction **Building Block** that will be realised as Mashable Composite UI Module. **It** allows the creation of new resource management simulations or modification of existing ones. It allows the user - most likely a trainer - to create incidents and scenes as well as the creation and management of objects of interest (**OOI**) instances as shown in the figure below.

It consists of a map **widget** displaying the **world** map as well as layers for items that have been added through the scenario toolbox.

In addition, there is also a toolbox widget that contains controls to create OOI instances and other map elements as well as the ability to manage and edit properties of elements created this way and the scenario itself (name, description and other relevant metadata).

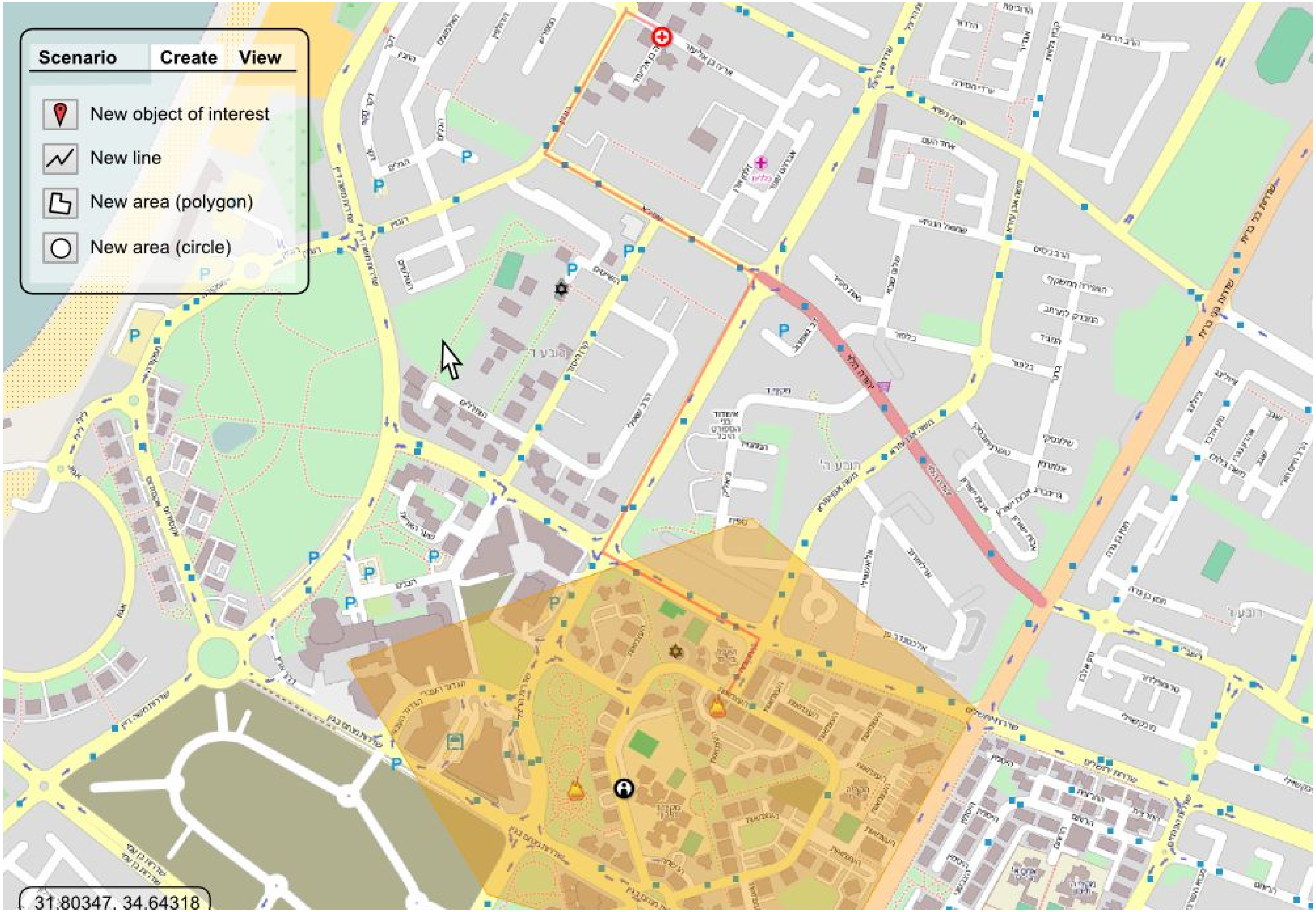
Resource Management Training Simulation Scenario Setup View Mockup

Key Functionality

1. Creation of new, modify existing, and store altered simulations
2. Placement and configuration of seismic events
3. Placement and configuration of floods
4. Placement and configuration of chemical plume hazards
5. Placement and configuration of air pollution hazards
6. Placement and configuration of fire and smoke hazards
7. Placement and configuration of pandemics
8. Placement and configuration of radioactive plume hazards
9. Placement and configuration of water contamination
10. Placement and configuration of extreme weather conditions
11. Evolution of states can be set manually
12. Ability to control when resources become available if they are not available right from the beginning
13. Ability to place and configure patients and people affected by hazards
14. Ability to place resources that can handle hazards locally (e.g. fire trucks, police officers, etc.)
15. Ability to place resources that can transport patients to places where they can be treated (e.g. ambulances)

Rationale

This is an important view focusing on the creation and management of simulations that are later used as configurations for initial world states. It allows the definition


Scenario
Create
View
New object of interest
New line
New area (polygon)
New area (circle)
31.80347, 34.64318

of simulation parameters, placement and configuration of **object of interest** instances, **scenario evolution** and other details that form a fully configured simulation for training purposes.

Note that this **BB** does not allow the creation or modification of new OOI types altogether as this is achieved by administrators of the systems in the backend of the system.

Realisation

The Resource Management Training Simulation Scenario Setup View is a User Interaction Building Block that will be realised as a Mashable Composite UI Module.

Documentation:

Specifications

1. The **BB** allows creating a new (empty) **resource management** **simulation**.
2. The BB allows altering existing simulations.
3. The BB allows deleting existing simulations.
4. The BB allows saving simulations.
5. The view allows the instantiation and placement of **object** of interests as well as setting their individual properties (as defined by their **OOI** type).
6. The view allows placing hazards and hazardous conditions.

Realization:

Wirecloud OOI Table

This **widget** shows all OOIs associated with a worldstate in a tabular form. **It** is possible to filter which elements should be visible (eg. only OOIs of certain types such as Ambulances and Hospitals). In addition, it also allows the user to group any OOIs for convenience; groups are persisted on the client-side using HTML5 storage mechanics and are automatically restored inbetween sessions.

Wirecloud Simulation Picker

A **widget** showing all simulations available on the **Object of Interest** **World State** Repository (**OOI**-WSR). **It** allows the user to select one of these simulations and load it, which will initialize other connected widgets and gadgets of the Wirecloud **application** with relevant data.

Related Models and BBs:

UI Mashup Platform

The **Mashup Platform** **Building Block** acts as a runtime environment for Mashable Composite UI Modules (widgets) and provides inter-**widget** communication, persistent per-widget configuration as well as proxy capabilities. The platform allows the user (in this case the person configuring a **CRISMA application**) to create a HTML5-based front-end by selecting certain user components (widgets and operations) that can be combined into a mashup. These user components are the **interface** for users to underlying back-end services. The user components themselves are provided by a 3rd party platform through a **catalogue**, from which a range of components can be installed into a mashup and connected (“wired”) using a graphical editor.

Integrated Planning View

The Integrated Planning View **BB** is a generic integrated view for the configuration and inspection of arbitrary **crisis management** scenarios in planning situations. The **CRISMA** planning use cases (most use cases of CRISMA) intend to realize functionalities that are related to opening **world** states, changing them and storing them. Further running simulations, inspecting them and comparing results is addressed in all of them and the user work for planning purposes in a calm office environment.

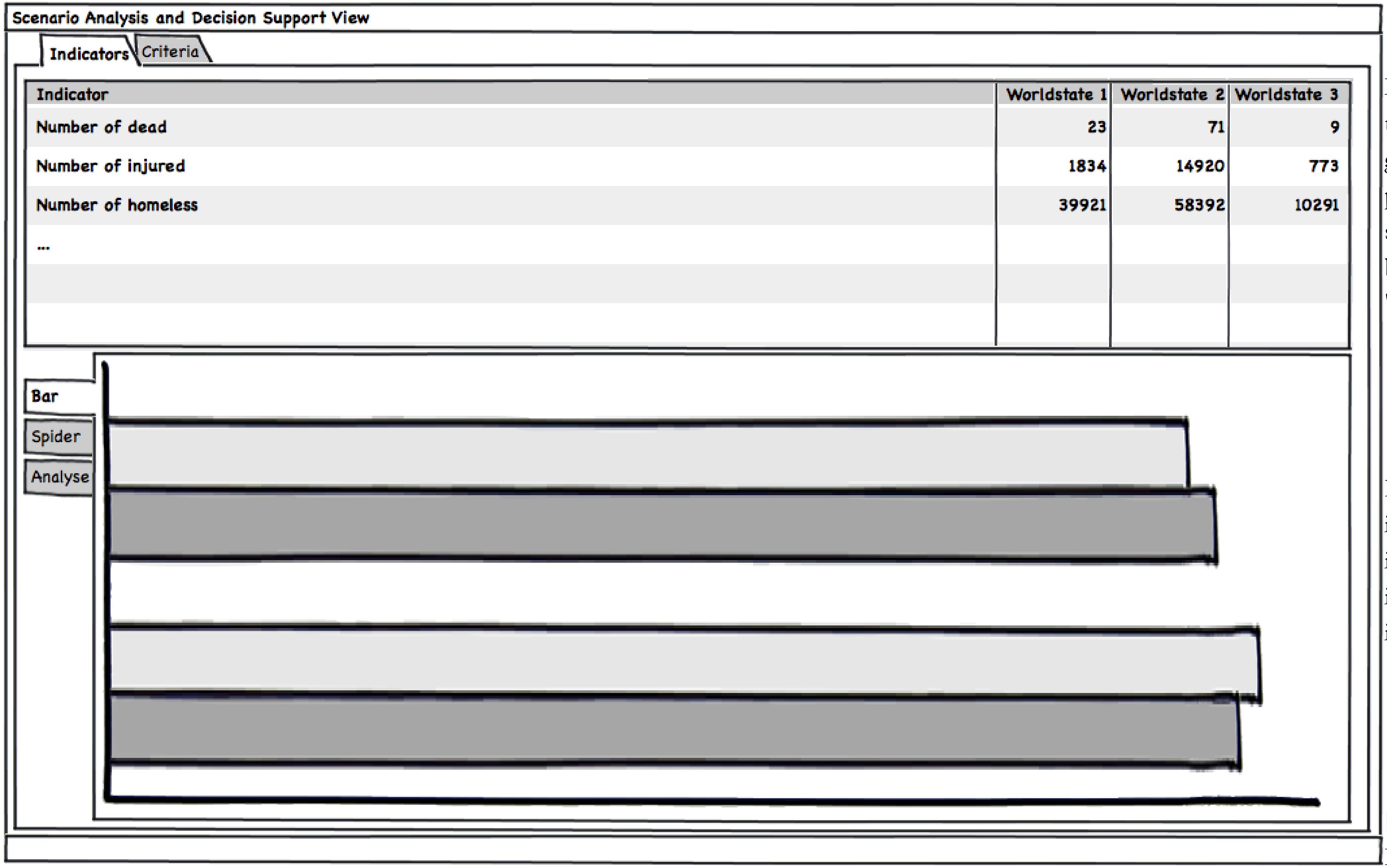
Scenario Analysis and Comparison View

BB Description:

The **Scenario** Analysis and Comparison View is able to visualise **Indicator** and **Criteria** data in a way that users are able to analyse and compare different Simulated **Crisis Management** Scenarios and ultimately come to a decision which fits the **simulation objective** best for a specific **Simulation Case**.

Indicators and Criteria are part of the key **concept** for performing uniform analysis and decision support within a **CRISMA** **hazard** domain. CRISMA Indicators as well as Criteria are synthetical data that is computed for every CRISMA **world** states using rules that suite the hazard domain. The computation is done by the Indicators **Building Block** and results in a vector of scalar values. However, the Indicators vector is plain data mainly based on quantities (e.g. number of dead) calculated from a **world state** without any attempt of qualification whereas Criteria are qualified thus put into perspective. This qualification basically means scaling the indicator data. Criteria vectors have the same amount of values as Indicator vectors. So there exists a One-to-one mapping between any Indicator and Criteria. Thus the qualification process may not do any indicator aggregation but maps one Indicator to a Criteria at a time without taking other Indicators into account. However, the qualification process may take other non-Indicator-relevant data of the underlying world state into account to provide a mapping as accurate as possible.

As Indicators and Criteria data has the same format both of them use the same visualisation facilities which are implemented by the Indicator Visualisation and Comparision and the Criteria Visualisation and Comparision **Widget** Software Components. This view provides a plain table-like visualisation for all values of all select world states. Additionally, **it** provides a bar chart visualisation to give an impression on how much the single values differ.



Indicator Bar Chart

Moreover, the view is able to visualise the data as spider charts in order to give a fast impression of the overall performance of the different world states. Even better comparability can be achieved by the selection of a "reference world state".

Indicator Spider Chart

Finally the view allows users to relate individual values to each other which is especially useful to get an impression of whether and how individual values correlate.

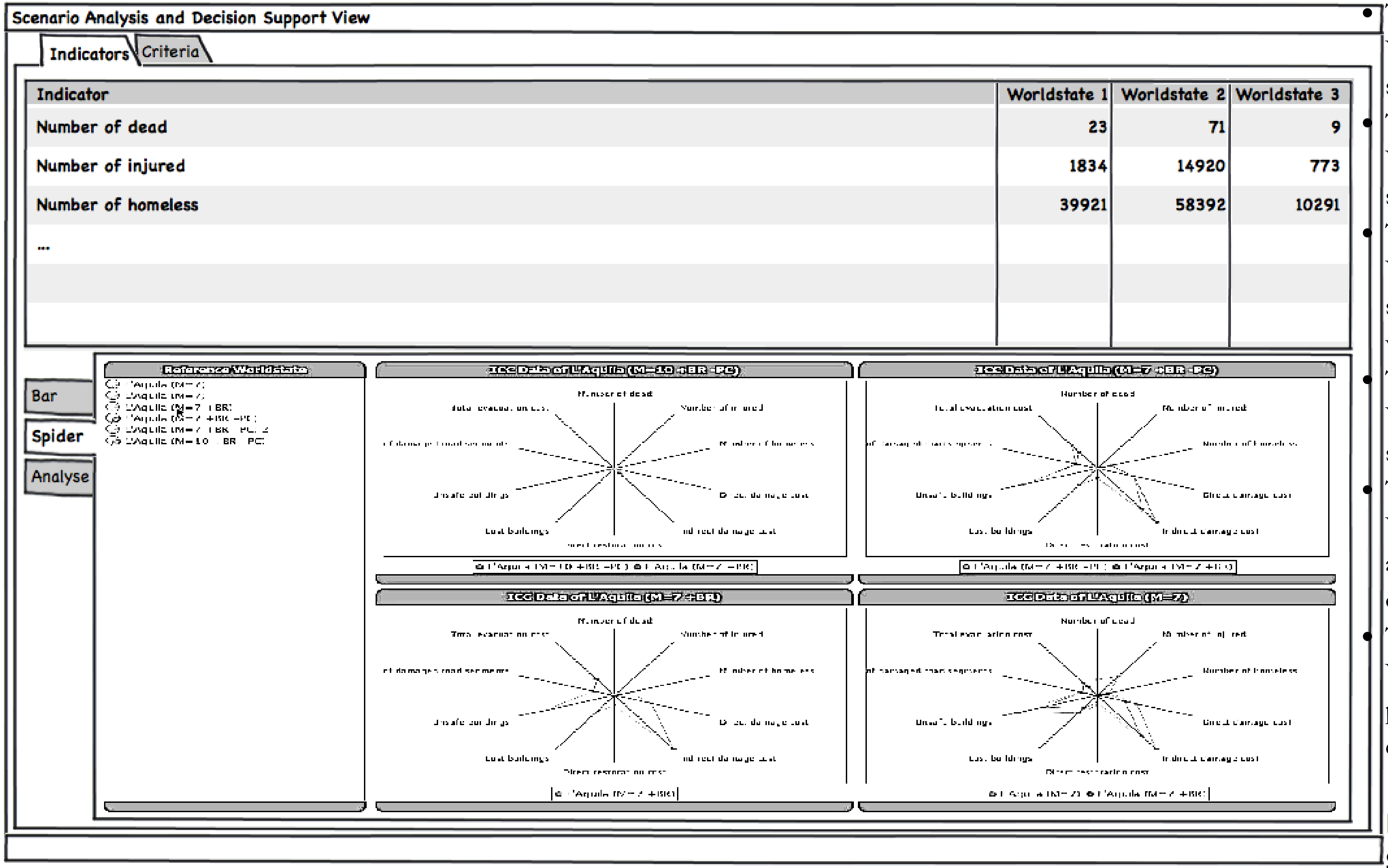
Indicator Analysis

Rationale

End users need to be able to analyse their different decisions to ultimately find a solution that suites the **situation** best. Thus the end users have to be presented with the decision relevant artefacts, Indicators and Criteria.

Key functionality

- The Scenario Analysis and Comparison View shall be able to visualise world state Indicators and Criteria in a uniform way.
- The Scenario Analysis and Comparison View shall be able to visualise world state Indicators in a plain table like view.
- The Scenario Analysis and Comparison View shall be able to visualise world state Criteria in a plain table like view.



- The Scenario Analysis and Comparison View shall be able to visualise world state Indicators in a bar chart like view.
- The Scenario Analysis and Comparison View shall be able to visualise world state Criteria in a bar chart like view.
- The Scenario Analysis and Comparison View shall be able to visualise world state Indicators in a Spider chart like view.
- The Scenario Analysis and Comparison View shall be able to visualise world state Criteria in a Spider chart like view.
- The Scenario Analysis and Comparison View shall be able to let the user choose a reference world state for Spider chart comparison.
- The Scenario Analysis and Comparison View shall provide the user the possibility to relate individual values to each other.

Documentation:

Specifications

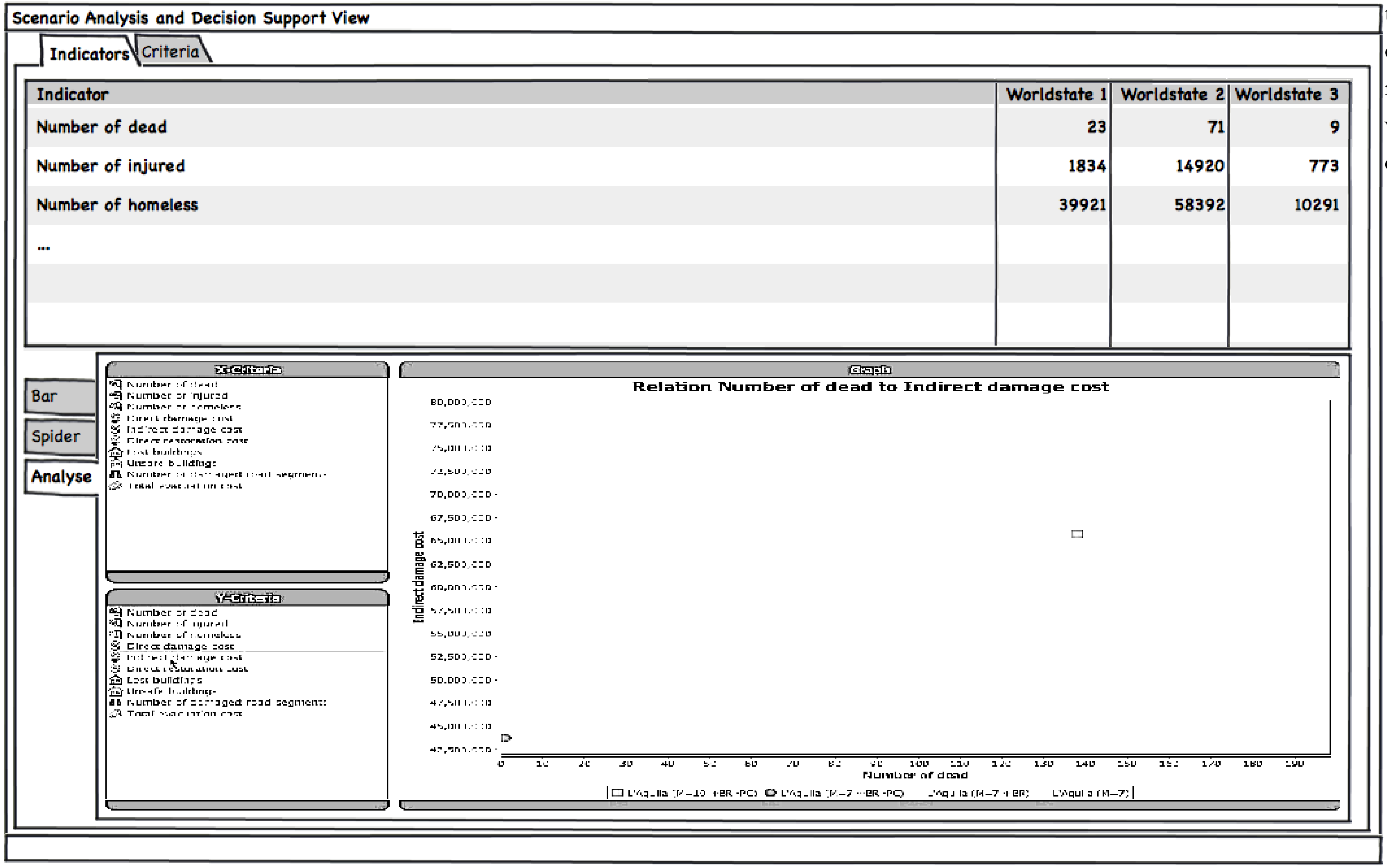
Architecture document that provides info on the overall relationships of Building Blocks

- D32.2 ICMS Architecture Document V2 **http://www.crismaproject.eu/deliverables/CRISMA_D322_public.pdf (http://www.crismaproject.eu/deliverables/CRISMA_D322_public.pdf)**

Realization:

Scenario Analysis and Comparison Widgets (Java)

This is the Java implementation of the **Scenario** Analysis and Comparison View. **It** provides a Table visualisation for both the indicators and the **criteria**. Moreover



the **indicator** values are visualised via bar charts and the single indicators can be related to each other so that a graph visualises the correlation between the different criteria.

Scenario Analysis and Comparison Widgets (JavaScript)

This is the JavaScript implementation of the **Scenario** Analysis and Comparison View **Building Block**.

Related Models and BBs:

Indicator Building Block

The Indicators **Building Block** (formerly Algebraic **Evaluation**) is a component that allows definition, storage and evaluation of “simple” algebraic models in order to evaluate consequences of decisions made by **CRISMA** users during a training session. In this context "simple" means that there is no large number crunching needed. Usage is for calculation of key performance indicators, heuristic models and to some extend the implementation of models actually running in interactive **GIS** environments in order to make them usable as services in the **CRISMA framework**.

Integrated Crisis Management Middleware BB

The **ICMM** is a central **Building Block** in every **CRISMA Application**. **It** connects **Crisis Management** Simulations with the Analysis and Decision Support functionality of CRISMA by providing a central repository for harmonized **world state** and **indicator** information. The ICMM is fed by simulations providing the basic information to be used for world state analysis and decision support Building Blocks. On an ICT conceptual level the ICMM is a generic distributed **resource**-oriented Control and Communication Information Management **System**. Thereby it is important to note, that the term 'resource-oriented' in the ICMM refers to the **concept** of generic resources as used in the context of the Resource Oriented **Architecture** (ROA).

Integrated Planning View

The Integrated Planning View **BB** is a generic integrated view for the configuration and inspection of arbitrary **crisis management** scenarios in planning situations. The **CRISMA** planning use cases (most use cases of CRISMA) intend to realize functionalities that are related to opening **world** states, changing them and storing them. Further running simulations, inspecting them and comparing results is addressed in all of them and the user work for planning purposes in a calm office environment.

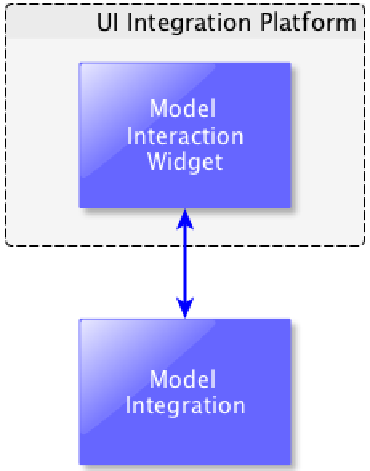
Worldstate View

The Worldstate View **Building Block** is a set of generic Widgets (Mashable Composite UI Modules) that allow to visualise Control and Communication Information (**CCIM**) related to Worldstates and Worldstate Transitions. Thus, **it** operates mainly on common **meta-information** about the **world** rather than the real data of the world (**World State** Data Slots).

Simulation Model Interaction View

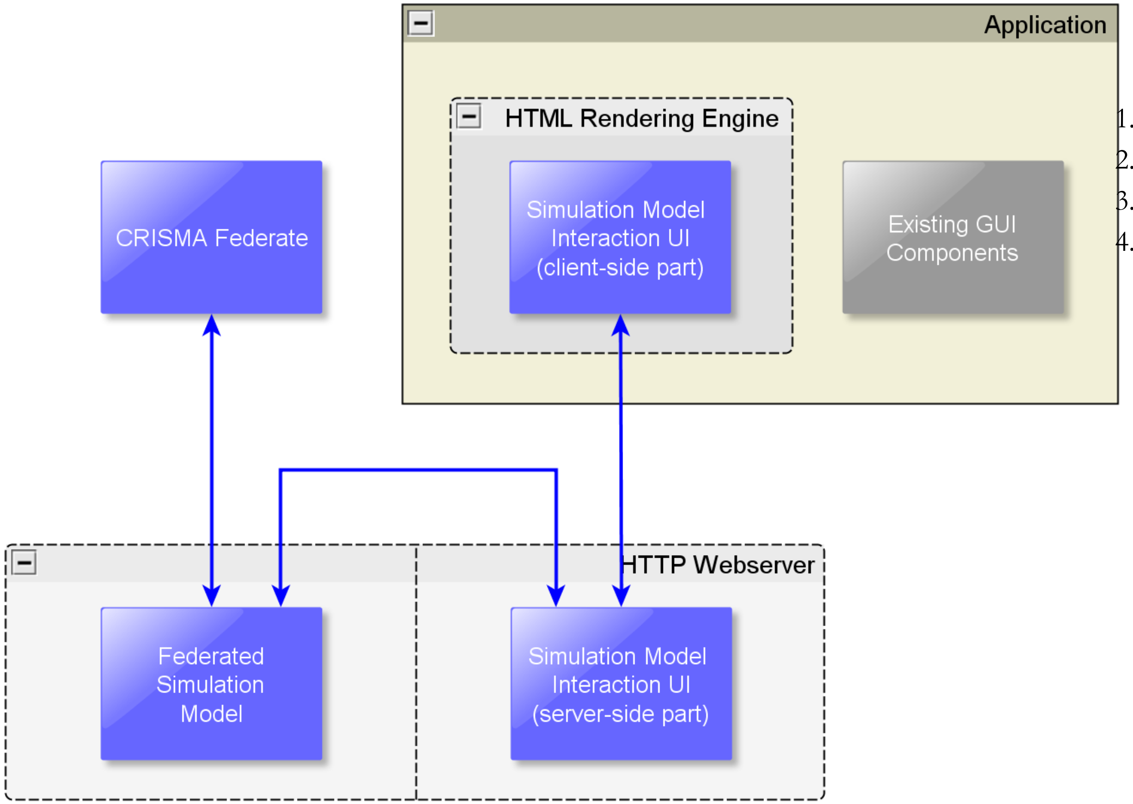
BB Description:

The **Simulation Model** Interaction **Widget** **Building Block** is a Composite UI Module that lets end users interact with the various simulation models exposed by a Simulation Model Integration Building Block. **It** is responsible for collecting simulation model parameters, launching the respective simulation model, showing status information and retrieving simulation model results. It also constitutes the **GUI** to the Simulation Model Integration Building Block.



Since the current model integration approach described in the **CRISMA Framework Architecture** is based on **OGC Web Processing Service**, the Simulation Model Interaction Widget is essentially a CRISMA-aware **WPS** client.

The Simulation Model Interaction Widget Building Block must be able to configure, run and monitor a simulation given a Worldstates and a SimulationDescriptors **entity** as input (see CRISMA Framework Core **CCIM**). Thus, the Simulation Model Interaction Widget should be split into a generic **business logic** part and a configurable user **interface** part. Thereby the generic business logic part should manage the communication with the **ICMM** Worldstates Repository and the Simulation Model Integration Building Block (OGC WPS) and should be the same for all types of simulation models. The user interface part may be different for each type of simulation model depending on the configuration and parameterization options of the simulation model. The business logic part should be realised as extended ICMM **API**.

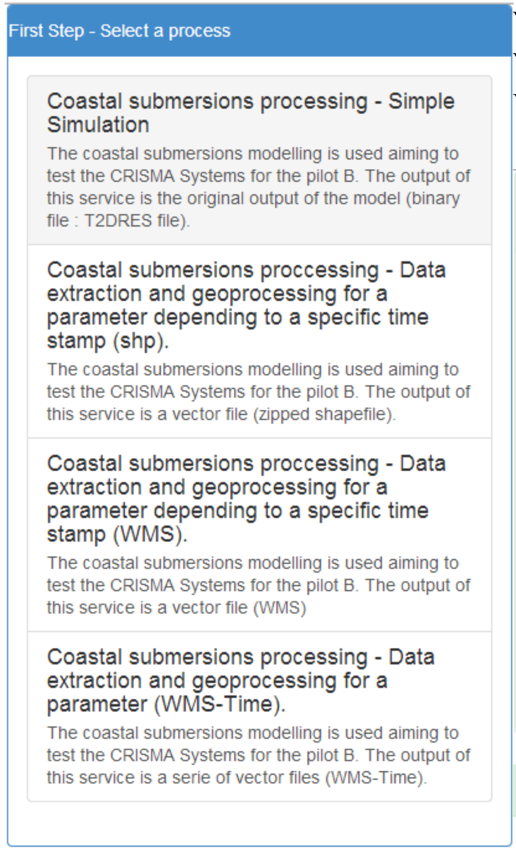


The Simulation Model Interaction View consists of the following four frames:

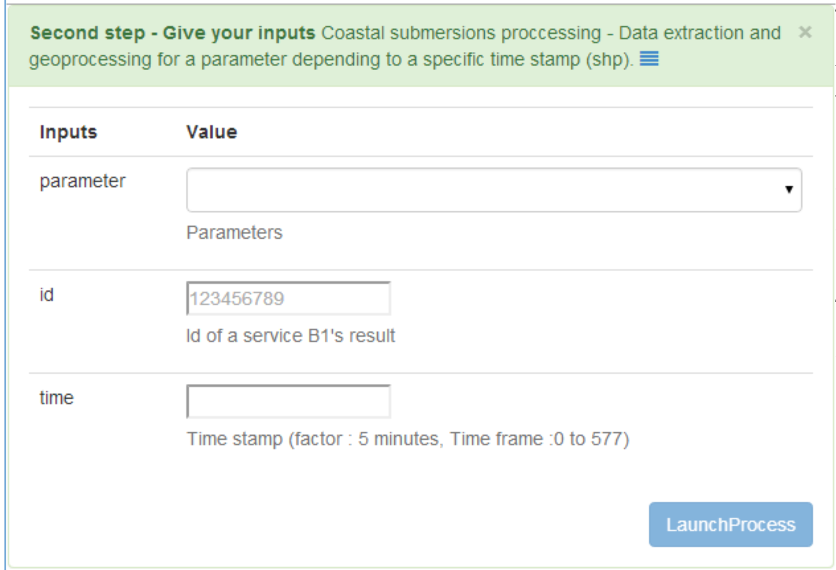
1. Simulation Model List
2. Simulation Model Configuration
3. Simulation Model monitoring
4. Simulation Model Result Widget

The **Simulation Model List** lists all of the available processes. This widget is generic and based on WPS GetCapabilities request. This widget displays the following information extracted from Getcapabilities : the title and the abstract of each process referenced on the WPS server.

The **Simulation Model Configuration** provides input parameters descriptions, input forms. This widget allows also users to run simulations. As with the Simulation Model List Widget, this

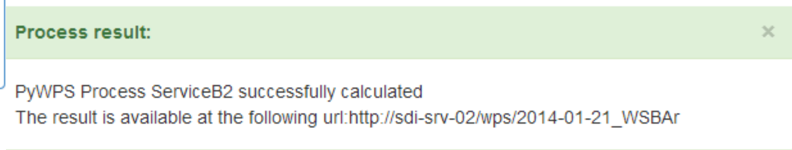


widget is generic thanks to, in this case, the use of the WPS DescribeProcess request. Indeed, information listed in this widget is directly extracted from this request: titles assigned to each input, input description, allowed values of inputs, default value of inputs… The information included in this widget depends directly of the completeness of the WPS.



The **Simulation Model monitoring** provides information regarding to the status of the simulation. This widget depends directly on the configuration of the WPS. If the status **parameter** is set, the Simulation Model monitoring widget can poll the process status aiming to monitor progress.

The **Simulation Model Result** provides information regarding to the result of the simulation.



Rationale:

End users need to be able to interact with the various CRISMA simulation models.

Key Functionality:

The key functionalities of the Simulation Model Integration Building Block are defined by the following functional requirements.

- The Simulation Model Interaction Widget shall be able to show to the user a list of all supports simulations that can be applied to the current worldstate.
- The Simulation Model Interaction Widget model shall be able to show to the user information like general model descriptions and required and optional model parameters.
- The Simulation Model Interaction Widget model shall provide a per-model configurable user interface (e.g. wizard, forms, …) to collect simulation model input and control parameters.
- The Simulation Model Interaction Widget shall be able to initialize the model with model input data (i.e. the currently selected Worldstate data) by uploading the data to a Data Integration Building Block (e.g. OGC **SOS**) or an integrated data store (e.g. **FTP** or **WebDAV** Server) - We distinguish between simulation model parameters and simulation model input data. (Simple, small) parameters are provided by the user directly while (huge, complex) input data has to be uploaded to some service.
- The Simulation Model Interaction Widget shall be able to initiate a new simulation model run.

- The Simulation Model Interaction Widget shall be able to monitor the **simulation run**, show status information to the user as well as publish status information to the ICMM Repository (Transitions status).
- The Simulation Model Interaction Widget shall be able to collect and store model results (i.e. create a new Worldstate including the results and the selected model control parameters).
- The Simulation Model Interaction Widget should be split into a generic business logic and a configurable user interface part.
- The generic business logic part of the Simulation Model Interaction Widget should manage the communication with the CRISMA Middleware (ICMM) and the Simulation Model Integration Building Block (OGC WPS) and should be the same for all types of simulation models. It should be realised as extended ICMM API.
- The user interface part of the Simulation Model Interaction Widget shall manage the configuration and parameterization options of the individual simulation models.
- The Simulation Model Interaction Widget shall provide convenient configuration and extension possibilities that leverage easy integration of many different types of simulation models.

Documentation:

Specifications

- D32.2 ICMS Architecture Document V2 **http://www.crismaproject.eu/deliverables/CRISMA_D322_public.pdf (http://www.crismaproject.eu/deliverables/CRISMA_D322_public.pdf)**

Realization:

Simulation Model Interaction Widget

The **Simulation Model** Interaction view is a generic **Web Processing Service** client. **It** generates an user-friendly **interface** aiming to discover, describe and run processes issued from **WPS** instances.

Related Models and BBs:

Cascade Events Configuration and Interaction View

The **Cascade Effects** View is a User interaction Building that allows a user to configure and run a Cascade Effects **Scenario**. The user can select a triggering **event** (for example, an earthquake) and provide may either specify the characterisation of the event (**Simulation Control Parameter**) and thus initiate a new **Simulation Model** Run for this particular event, or select (if available) the output of a past event or an event already simulated.

Data Integration

The Data Integration **Building Block** provides components that can be used to easily serve data in a **CRISMA**-compliant (**OGC** open **standard** compatible) way so that other Building Blocks may use them for further processing like viewing or editing. That way data can be made accessible for CRISMA components.

Integrated Planning View

The Integrated Planning View **BB** is a generic integrated view for the configuration and inspection of arbitrary **crisis management** scenarios in planning situations. The **CRISMA** planning use cases (most use cases of CRISMA) intend to realize functionalities that are related to opening **world** states, changing them and storing them. Further running simulations, inspecting them and comparing results is addressed in all of them and the user work for planning purposes in a calm office environment.

Simulation Model Integration BB

The **Simulation Model** Integration **Building Block** provides components that can be used to easily enable simulation models to participate in a **CRISMA Application**. Due to the heterogeneity of existing simulation models CRISMA has to ensure that they can be integrated in a standardized way and ultimately become a **CRISMA Federate**.

That way they are made CRISMA-aware. The envisioned technique to be used is that of so called wrapping so that this Building Block provides a piece of software that can be used to make (existing) simulation models CRISMA-aware with as little effort as possible with respect to the individualities of the different simulation models.

UI Mashup Platform

The **Mashup Platform** **Building Block** acts as a runtime environment for Mashable Composite UI Modules (widgets) and provides inter-**widget** communication, persistent per-widget configuration as well as proxy capabilities. The platform allows the user (in this case the person configuring a **CRISMA application**) to create a HTML5-based front-end by selecting certain user components (widgets and operations) that can be combined into a mashup. These user components are the **interface** for users to underlying back-end services. The user components themselves are provided by a 3rd party platform through a **catalogue**, from which a range of components can be installed into a mashup and connected (“wired”) using a graphical editor.

Building impact model

Model for the assessment of expected damage on building classes due to earthquakes

---

Road network vulnerability model (RNV)

Model for the assessment of probability of road link interruption due to earthquakes.

Time Dependent Vulnerability model (TDV)

Model for the assessment of time-dependent damage on **elements at risk**.

Cascading Effects Model

The cascading effect **model** for dynamic **scenario** assessment calculates the probability of attainment of cascading events scenarios, given an initial triggering **event**, or estimate consequence paths given the occurrence of selected scenarios.

Worldstate View

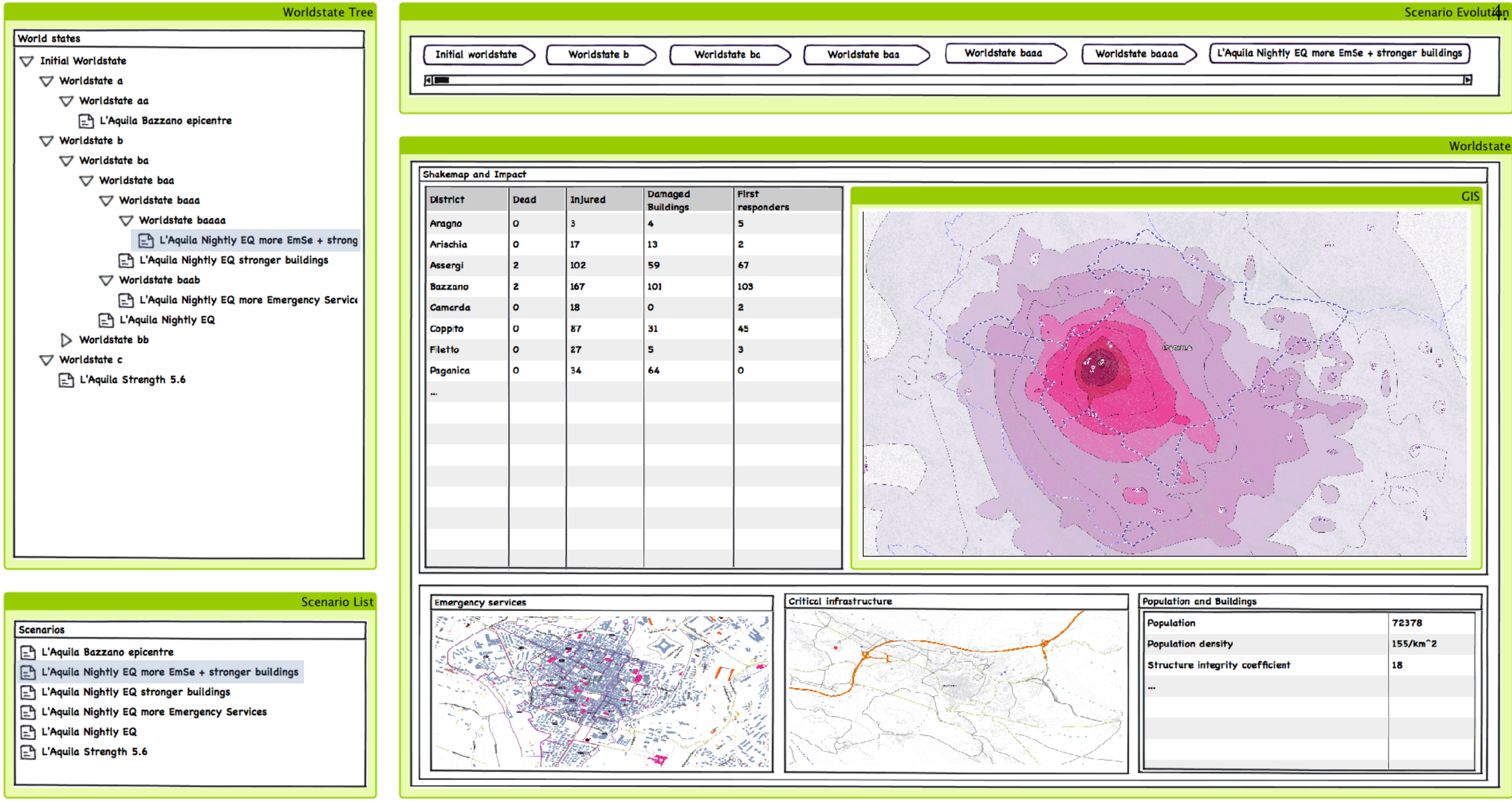
BB Description:

The Worldstate View **Building Block** is a set of generic Widgets (Mashable Composite UI Modules) that allow to visualise Control and Communication Information (**CCIM**) related to Worldstates and Worldstate Transitions. Thus, **it** operates mainly on common **meta-information** about the **world** rather than the real data of the world (**World State** Data Slots).

Worldstate View Mockup

As can be seen in the figure above, the Worldstate View consists of the following four independent and generic Widgets:

1. **Scenario Evolution** **Widget**
2. Scenario List Widget
3. Worldstate Tree Widget



4. Worldstate Widget

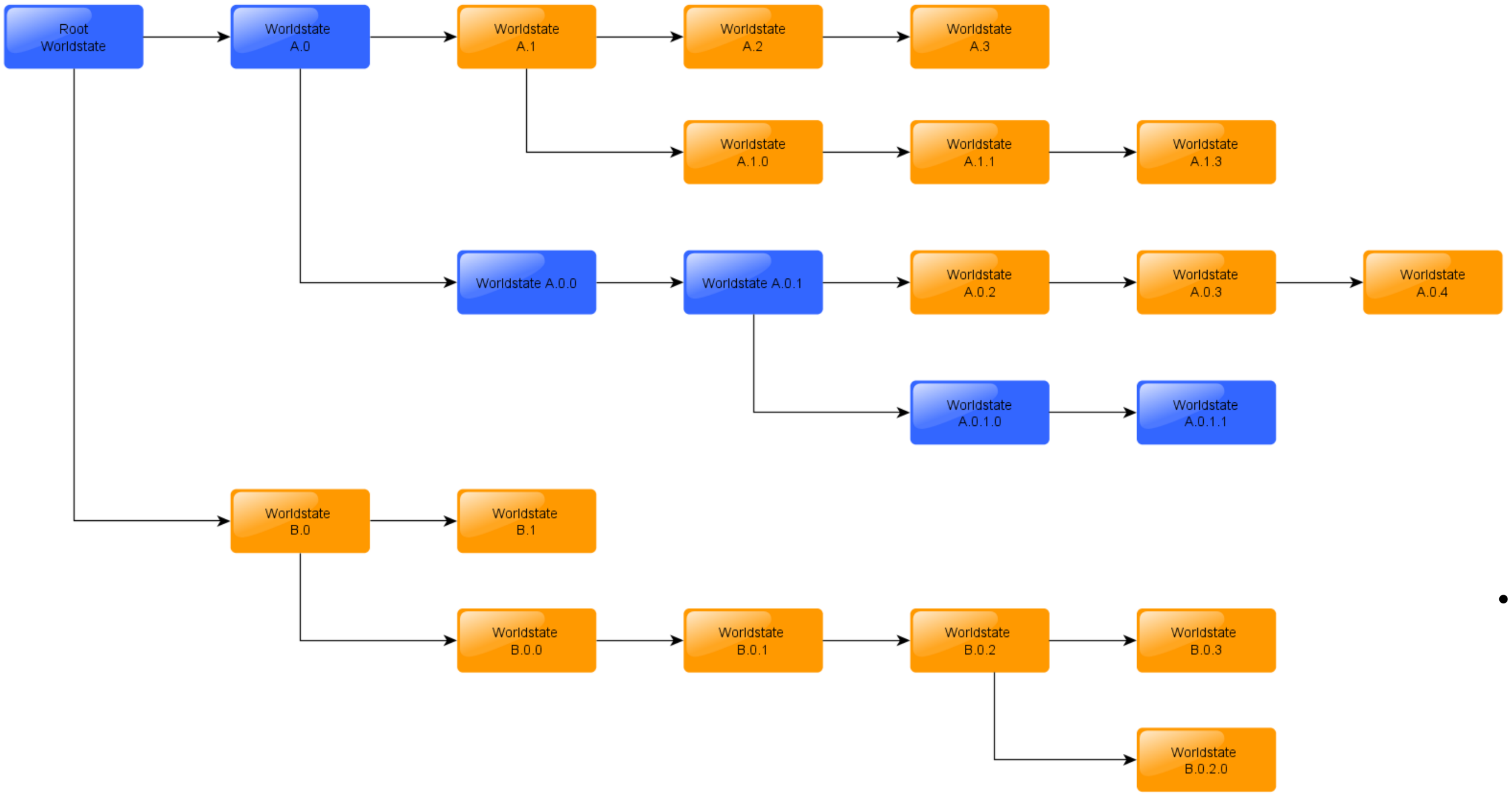
Scenario Evolution Widget

The Scenario Evolution Widget is a Mashable Composite UI Module of the Worldstate View Building Block that visualises the evolution of a specific scenario from the beginning (the “initial” Worldstate or Root Worldstate) to the currently selected Worldstate (leaf in the worldstates tree).

A scenario is thereby defined as a path of a worldstate tree. One worldstate can therefore be considered as a **decision point**. Since the Worldstate Repository of the **ICMM** replaced former **Simulation Case** and Versioned Reality Repositories the Scenario Evolution Widget therefore replaces the former Versioned Reality **GUI** Component.

A Scenario (blue) is a path in a worldstate tree

The Scenario Evolution Widget is also able to provide access to any Worldstate that is part of the currently selected Worldstate’s predecessors. It can also be used to select one or worldstates (decision points) of the scenario to trigger worldstate transitions, worldstate analyses, etc. within this context.



Scenario Evolution Widget Mockup

Rationale

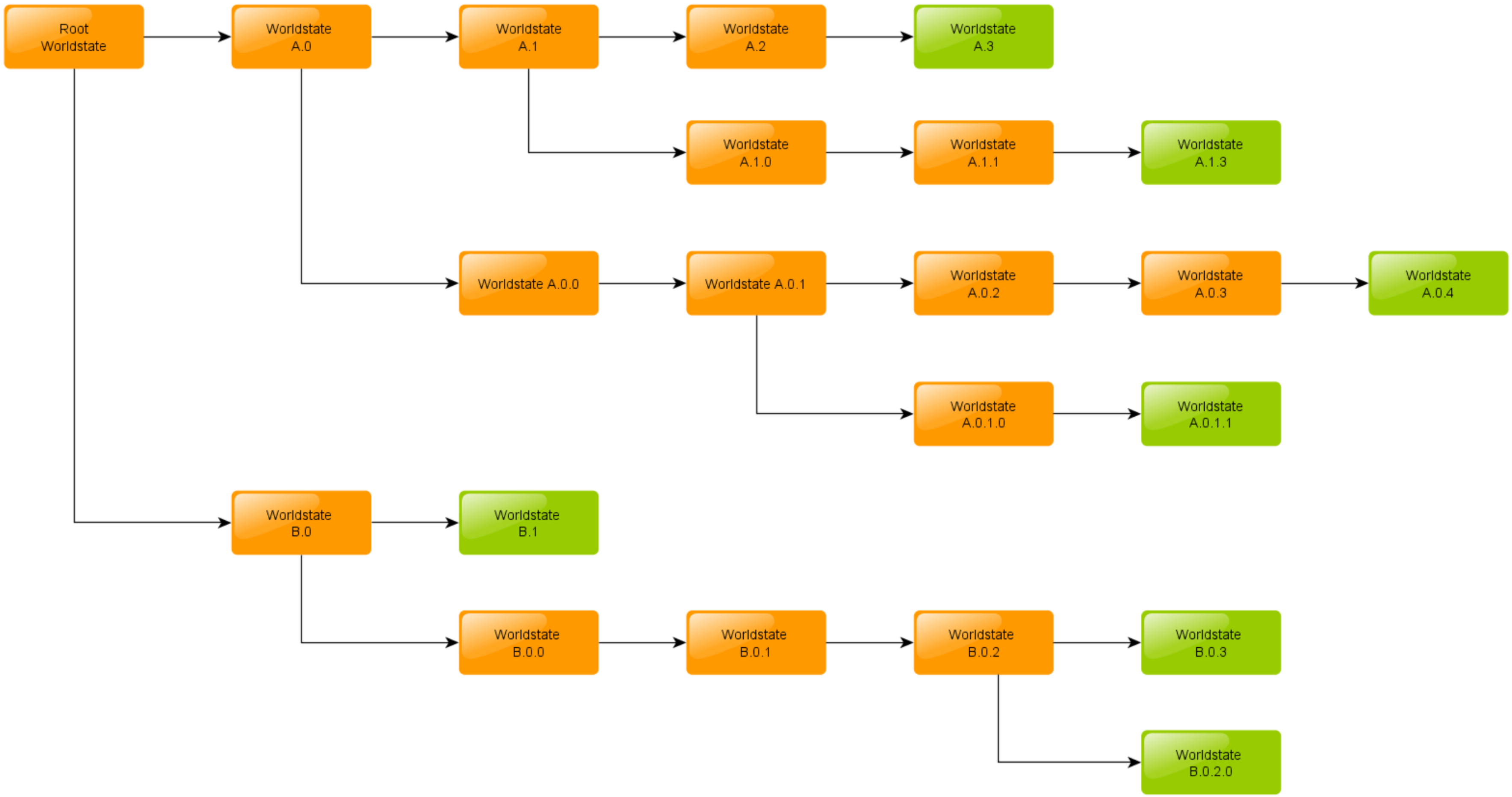
End users need to be able to investigate history of origins (versioned reality) of a specific Worldstate and perform several functions on one or more selected worldstates.

Key Functionality

- The Scenario Evolution Widget shall visualise every Worldstate that is part of a Worldstate's history of origins (that is effectively the path of a Worldstate to its topmost parent, the "initial" Worldstate).
- The Scenario Evolution Widget shall provide an **API** function that allows selecting a specific Worldstate. This worldstate is then used to visualise the scenario evolution (tree path).
- The Scenario Evolution Widget shall support single selection of a Worldstate.
- The Scenario Evolution Widget shall provide an API function that allows propagating worldstate selection change events.
- The Scenario Evolution Widget shall be able to visualise generic properties of a worldstate, e.g. name and description (as tooltip).
- The Scenario Evolution Widget shall support to select different worldstates to trigger a comparison of the worldstates.

Scenario List Widget

The Scenario List Widget is a Composite UI Module of the Worldstate View Building Block that visualises the currently available Scenarios.



Scenarios as leaves (green) in a Worldstate Tree

As a scenario is composed by a series of Worldstates the Scenario List Widget effectively visualises all leafs in the current Worldstate Tree so that the **end user** may not only get an overview of all available scenarios but also may easily access them.

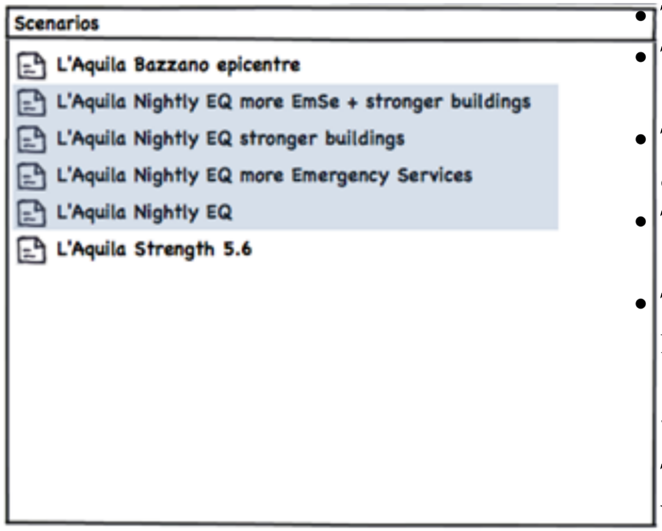
Scenario List Widget Mockup

Rationale

End users need to be able to easily find and access all currently available scenarios.

Key Functionality

- The Scenario List Widget shall visualise every currently available scenarios (that is effectively all leafs of the current Worldstate Tree).
- The Scenario List Widget shall provide an API function that allows specifying a specific Root Worldstate. This worldstate is then used to construct the list of worldstate tree leaves (scenarios).



- The Scenario List Widget shall support single and multiple selection of a Worldstate.
- The Scenario List Widget shall provide an API function that allows to specify the currently selected scenarios (programmatically perform a selection).
- The Scenario List Widget shall provide an API function that allows propagating worldstate selection change events.
- The Scenario List Widget shall be able to visualise generic properties of a worldstate, e.g. name and description (as tooltip).
- The Scenario List Widget shall provide convenient configuration mechanisms to specific the receiver (Building Blocks, Composite UI Module) of an **event**.

Worldstate Tree Widget

The Worldstate Tree Widget is a Composite UI Module of the Worldstate View Building Block that visualises the Worldstate Trees of a **CRISMA Application**. As defined by the CRISMA **Conceptual Business logic** Worldstates have exactly one parent Worldstate of which they where create through some kind of **transition** (e.g. a **model** execution, a manual change, etc.).

Worldstates Tree

Thus a Worldstate may have multiple children. Because of that inherent nature of Worldstates a tree-like visualisation of the Worldstate relationships is favourable and provides the user with a good overview over all currently available decision point.

The Worldstate Tree Widget can also be used to execute specific actions on a certain worldstate which in turn may trigger specific Building Blocks, e.g. a Simulation Interaction Widget. The definition of those actions as well as the integration with the respective API calls are a configuration **task** and thus part of the development of an individual CRISMA Application.

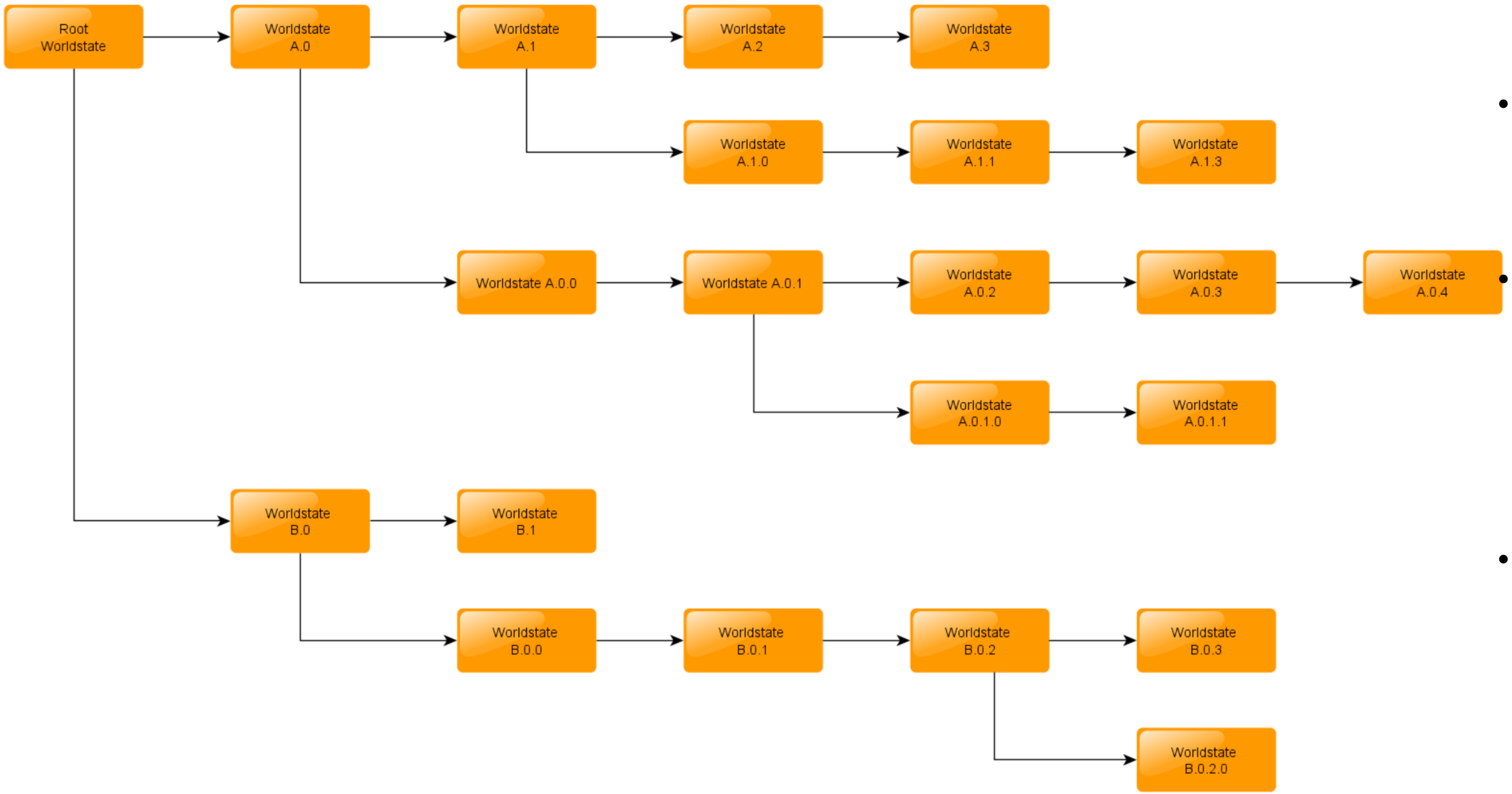
Worldstate Tree Widget Mockup

Rationale

End users have to have access to all Worldstates that were produced. Additionally they need to be able to investigate their history of origins.

Key Functionality

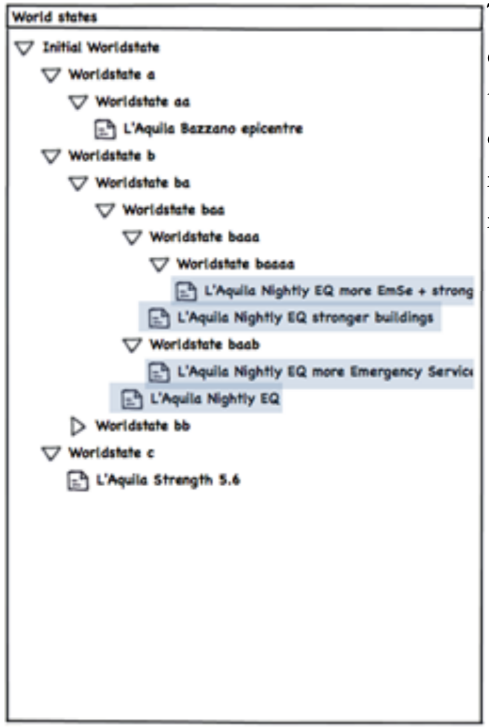
- The Worldstate Tree Widget shall visualise all available Worldstate as tree.
- The Worldstate Tree Widget shall offer a refresh functional that allows the user to get an updated list of Worldstates from the ICMM.
- The Worldstate Tree Widget shall implement automatic refresh functionality that either periodically checks the ICMM for updates to the worldstates tree or uses



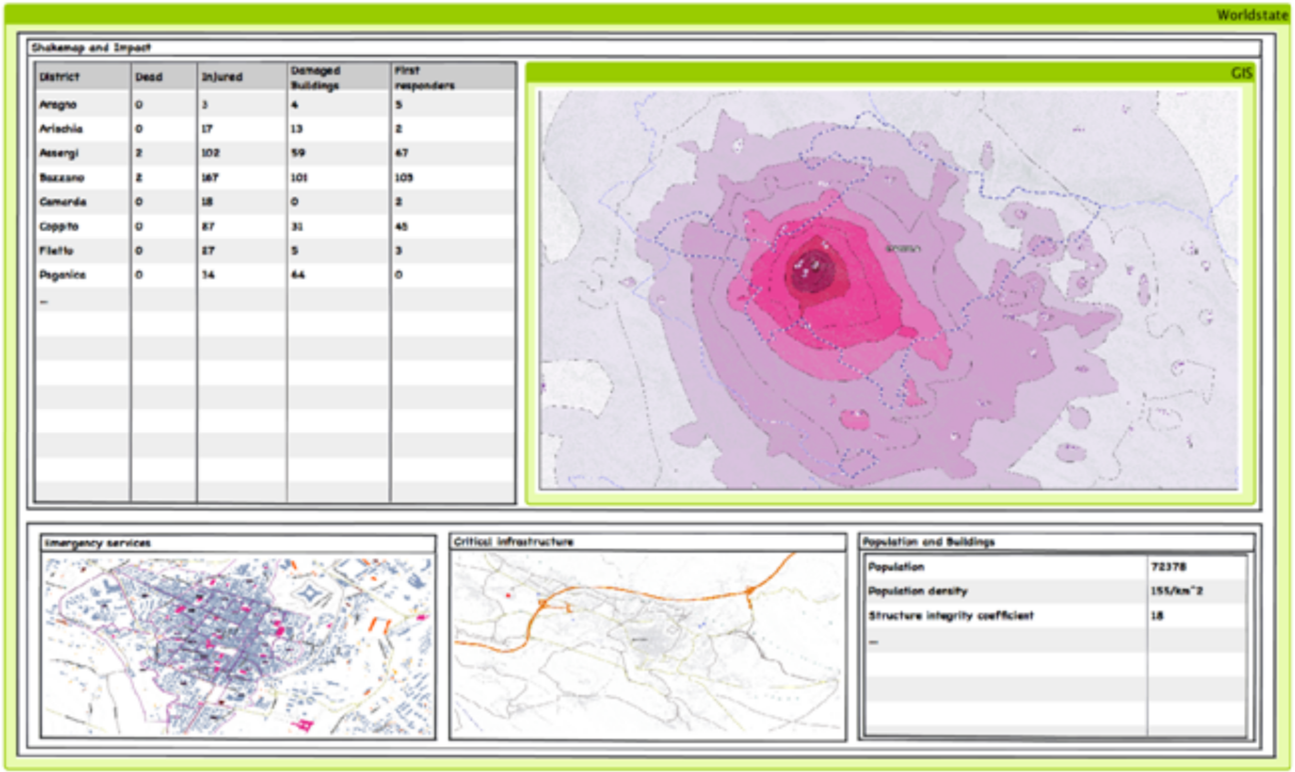
the Event Propagation API of the ICMM.

- The Worldstate Tree Widget shall support multiple and single selection of a Worldstate.
- The Worldstate Tree Widget shall provide an API function that allows to specify the currently selected scenarios (programmatically perform a selection).
- The Worldstate Tree Widget shall provide an API function that allows propagating worldstate selection change events.
- The Worldstate Tree Widget shall be able to visualise generic properties of a worldstate, e.g. name and description (as tooltip).
- The Worldstate Tree Widget shall provide a context menu which allows to execute certain actions on a specific worldstate, e.g. to initiate a **transition** (manipulation or simulation).
- The Worldstate Tree Widget shall provide convenient configuration mechanisms to specific the receiver (Building Blocks, Composite UI Module) of an event or an action.
- The Worldstate Tree Widget shall provide convenient configuration mechanisms to specific the actions that can be executed on worldstates.

Worldstate Widget



The Worldstate Widget is a Composite UI Module of the Worldstate View Building Block that is capable of visualising and editing the details of a Worldstate, in fact the actual Worldstate data. As a Worldstate may actually contain lots of information the Worldstate Widget creates a logical grouping of several aspects and visualises a single group with full detail whereas all the other groups are visualised via miniature views. The user may switch the current "full detail" group with any other of the miniature views so that the particular group becomes visible with full detail. Moreover the user may switch to an edit mode to manipulate any Worldstate data producing a new Worldstate.



Worldstate Widget Mockup

It is important to highlight, that the Worldstate Widget consist of a generic application independent part and a worldstate specific part. Since a single generic widget cannot anticipate all possible types of worldstate data and worldstate visualisation and manipulation possibilities, worldstate specific views have to be provided. The Worldstate Widget can therefore be considered as client-side container for hosting specific worldstate (data) manipulation and visualisation Composite UI Modules. In contrary to the UI **Mashup Platform** that allows dynamic composition of arbitrary Mashable Composite UI Modules the Worldstate Widget is a simple container with predefined layout and fixed slots for Composite UI Modules. In principle, any Composite UI Module (e.g. the **GIS** Widget) can be used as manipulation or visualisation view inside of the Worldstate Widget. Integrating a specific Composite UI Module into the Worldstate Widget is a configuration task that has to be carried out as part of the development of an individual CRISMA Application.

The Worldstate Widget is the main means to interact with Worldstate data. It provides suitable visualisations of all the data of a Worldstate and groups them in an appropriate way. This can mean that the Worldstate Widget uses other Building Blocks (e.g. the GIS Widget Building Block) to provide the data visualisation and edit functionality. This Building Block is by nature a slave to some other Building Block (e.g. the Worldstate Tree Widget Building Block).

Key Functionality

- The Worldstate Widget shall be able to visualise the data of a worldstate with help of specific worldstate viewer UI Modules.
- The Worldstate Widget shall be able to edit (manipulate) the data of a worldstate with help of specific worldstate editor UI Modules.
- The Worldstate Widget shall be able to show a group of worldstate editor or viewer UI Modules.
- The Worldstate Widget shall be able to switch between different worldstate editor and viewer UI Modules within one group.
- The Worldstate Widget shall be able to switch between a view and an edit mode which initiates a worldstate transition.
- The Worldstate Widget shall be able to initiate a worldstate transition and store the respective information about the transition in the ICMM Worldstates repository.
- The Worldstate Widget shall provide a detail and a several miniature views. The detail view shows the currently active worldstate editor or viewer UI Module.
- The Worldstate Widget shall be able to swap detail and miniature view on click on the miniature view.
- The Worldstate Widget shall create a new worldstate when a worldstate is edited with help of a specific editor module (worldstate transition).
- The Worldstate Widget shall provide an API function that allows setting a specific worldstate for visualisation.
- The Worldstate Widget shall provide an API function that allows setting a specific worldstate for manipulation.
- The Worldstate Widget shall provide an API function that allows propagating worldstate change events to other Building Blocks.
- The Worldstate Widget shall provide convenient configuration mechanisms to specific the viewer and editor Composite UI Modules of a worldstate.

Documentation:

Functional Description

Functional Descriptions of the Worldstate View Widgets. Located on the **VTT** WS.

- Scenario Evolution Widget Functional Description
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20Functional%20Building%20Blocks/v2.0/Scenario%20Evolution%20Widget.doc
- Scenario List Widget Functional Description
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20Functional%20Building%20Blocks/v2.0/Scenario%20List%20Widget.doc
- Worldstate Tree Widget Functional Description
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20Functional%20Building%20Blocks/v2.0/Worldstate%20Tree%20Widget.doc

- Worldstate Widget Functional Description **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20Functional%20Building%20Blocks/v2.0/Worldstate%20Widget.doc**

Realization:

Worldstate Tree Widget (Java)

This is a java-based prototype of the Worldstate Tree **Widget** Functional **Building Block**.

Worldstate Tree Widget (JavaScript)

This is a JavaScript-based implementation of the Worldstate View Functional **Building Block**. The **widget** uses the **ICMM** as backend and visualises the Worldstates in their natural tree structure. **It** lets users navigate the tree and can propagate selection and activation of worldstates so that other components may synchronise their views.

Scenario Evolution Widget (Java)

This is a java-based prototype of the **Scenario Evolution** **Widget** Functional **Building Block**.

Scenario Evolution Widget (JavaScript)

This is a JavaScript-based implementation of the Worldstate View Functional **Building Block**.

Scenario List Widget (Java)

This is a java-based prototype of the **Scenario** List **Widget**.

Scenario List Widget (JavaScript)

This is a JavaScript-based implementation of the Worldstate View Functional **Building Block**. The **widget** uses the **ICMM** as backend and visualises the **Scenario** Worldstates in a list-like view. **It** can propagate activation of worldstates so that other components may synchronise their views.

Worldstate Widget (Java)

This is a java-based prototype of the Worldstate **Widget** Functional **Building Block**. The Worldstate Widget is a simple container with predefined layout and fixed slots for further **application** specific widgets that are capable of visualising the actual data of Worldstate, including also **ICC** data.

Worldstate Widget (JavaScript)

This is a javascript-based prototype of the Worldstate View Functional **Building Block**. The Worldstate **Widget** is a simple container with predefined layout and fixed slots for further **application**-specific widgets that are capable of visualising the actual data of Worldstate, including also **ICC** data.

Related Models and BBs:

GIS View Building Block

The **GIS** View is a User Interaction **Building Block** that enables the visualisation and manipulation of geospatial data. Geospatial data plays a predominant **role** in all **crisis management** related applications, because most if not all information playing a role in crisis management has a geospatial component.

Integrated Crisis Management Middleware BB

The **ICMM** is a central **Building Block** in every **CRISMA Application**. **It** connects **Crisis Management** Simulations with the Analysis and Decision Support functionality of CRISMA by providing a central repository for harmonized **world state** and **indicator** information. The ICMM is fed by simulations providing the basic information to be used for world state analysis and decision support Building Blocks. On an ICT conceptual level the ICMM is a generic distributed **resource**-oriented Control and Communication Information Management **System**. Thereby it is important to note, that the term ‘resource-oriented’ in the ICMM refers to the **concept** of generic resources as used in the context of the Resource Oriented **Architecture** (ROA).

Integrated Planning View

The Integrated Planning View **BB** is a generic integrated view for the configuration and inspection of arbitrary **crisis management** scenarios in planning situations. The **CRISMA** planning use cases (most use cases of CRISMA) intend to realize functionalities that are related to opening **world** states, changing them and storing them. Further running simulations, inspecting them and comparing results is addressed in all of them and the user work for planning purposes in a calm office environment.

Multi Criteria Analysis and Decision Support View

The Multi **Criteria** Analysis View and Decision Support View is a User Interaction **Building Block** that allows performing a ranking of different **Crisis Management** Scenarios with respect to specific Criteria.

Publish Subscribe Context Broker BB

The Publish Subscribe Context Broker **Building Block** is a cross-over between an **event** broker which accepts events and dispatches them to subscribers and an access **service** providing the information on current state of the “world”. Thereby, the Publish Subscribe Context Broker Building Block can be used to realize the event subscription and event delegation functionality of the **ICMM**. Thus, the Building Block is both suitable for dispatching events related to Control and Communication Information (**CCI**) managed by the ICMM and events related to (Worldstate) Data managed by arbitrary data access services, e.g. the **OOI** **World State** Repository **BB** or implementations of the Data Integration BB (**WMS**, **WFS**, …).

Scenario Analysis and Comparison View

The **Scenario** Analysis and Comparison View is able to visualise **Indicator** and **Criteria** data in a way that users are able to analyse and compare different Simulated **Crisis Management** Scenarios and ultimately come to a decision which fits the **simulation objective** best for a specific **Simulation Case**.

UI Integration Platform BB

The UI Integration Platform **Building Block** is a component that is able to host Composite UI Modules. Composite UI Modules are User Interaction Building Blocks that are realised as HTML5 and JavaScript widgets. **It** constitues the Runtime Environment of the Composite UI Modules as they - by their nature - cannot be used as stand-alone applications.

UI Mashup Platform

The **Mashup Platform** **Building Block** acts as a runtime environment for Mashable Composite UI Modules (widgets) and provides inter-**widget** communication, persistent per-widget configuration as well as proxy capabilities. The platform allows the user (in this case the person configuring a **CRISMA application**) to create a HTML5-based front-end by selecting certain user components (widgets and operations) that can be combined into a mashup. These user components are the **interface** for users to underlying back-end services. The user components themselves are provided by a 3rd party platform through a **catalogue**, from which a range of components can be installed into a mashup and connected (“wired”) using a graphical editor.

4.4 Models

Model specifications can be seen as a special type of building blocks. Realizations thereof can be found in the Components section.

Building impact model

Model for the assessment of expected damage on building classes due to earthquakes

The model allows the assessment of the probability of attaining established damage levels in a given damage scale for suitable defined building classes The model is based on the SAVE methodology (Zuccaro et al., 2008). This **vulnerability model** is basically founded on the statistical elaboration of historical data on the damage occurred during past events in Italy in the last 30 years: Irpinia '80, Abruzzo '84, Sicilia '90, Parma '83, Umbria-Marche '97, Etna '02, Molise-Puglia '02. The procedure is based on "poor" data easily detectable and, far from giving detailed **evaluation** on building safety, allows to well point out the most vulnerable structures and to draw up "lists" of buildings to be investigated.

References

Zuccaro G., Albanese V., Cacace F., Mercuri C., Papa F. et al., (2008) Seismic Vulnerability Evaluations Within The Structural And Functional Survey Activities Of The COM Bases In Italy, AIP Conf. Proc. 1020, pp. 1665-1674; doi: **http**://dx.doi.org/10.1063/1.2963797

Documentation:

Functional Description

Functional description

The SAVE **Model** deals with the seismic **vulnerability** of a building portfolio by means of a building classification into several classes. The building classification is based on 5 interval ranges of a Synthetic **Parameter** of Damage (SPD) varying from 0 to 5. The SPD represents the average value of the distribution of damage defined according to the **damage classification** proposed in the European Macroseismic Scale (EMS 98). For each building SPD may be evaluated initially considering the structural material for vertical structures, as in EMS98 classification (type of masonry, RC, etc.) and next **it** may be "corrected" on the basis of parameters influencing seismic behavior (type of horizontal structures, number of storeys, construction age etc.).

The analysis is performed at the Minima territorial Units (MU) level, whose dimension has to be suitably established in a context-specific manner. A reasonable option is to discretize the entire territory with a square grid (e.g. 500x500 m), with the grid cell being adopted both for analysis at the MU level and for the results representation. Adopting a uniform grid the most critical areas may be localized with higher precision, while the global results may be also aggregated and represented at a larger spatial level (e.g. the Municipality level).

Input data

Elements vulnerability data

- For each class and for each relevant intensity level suitable Damage Probability Matrices (**DPM**) have to be provided. DPM express in a discrete form, for a given element class, the conditional probability of obtaining a damage level k , due to an **event** of intensity I.

Other input data are given at the Minima territorial Units (MU) level (e.g. grid of 500x500 m)

Inventory data

- Building inventory: distribution of vulnerability classes (number of **elements at risk** belonging to each class)

Hazard data

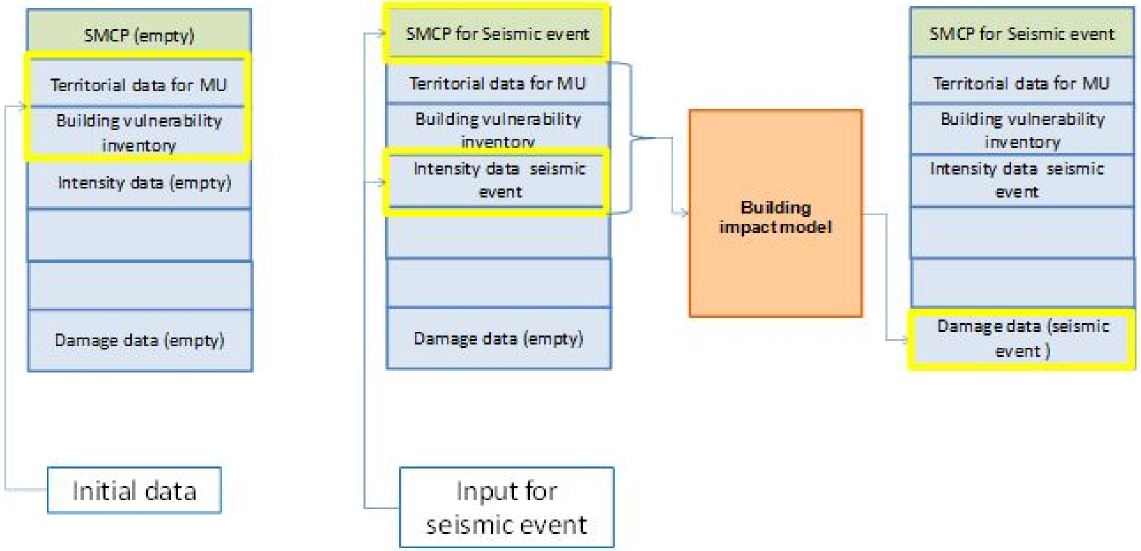
- Hazard intensity distribution (relevant value of the macro-seismic intensity –or alternatively of Peak Ground Acceleration PGA - for each MU); these data can be given:
    - directly as input file (e.g. in case of historic scenarios with shake-maps)
    - as result of **application** of hazard attenuation laws that are included in the building **impact model**; in such a case the necessary input is
        - type of attenuation law (among the ones available)
        - Event ID
        - Date and time of event
        - Epicenter location (point coordinates lat/lon)
        - Magnitude
        - Depth

Process

The next figure evidences the sequential steps that are followed for building impact analysis and showing the kind of data that are changed in the **World State** (WS) (from input to output, and also considering intermediate WS).

In particular, the initial WS contains only the basic territorial **GIS** data, as well as the inventory of the elements at risk (Building seismic vulnerability inventory).

Once WS is assigned, the next step is to assign the **Simulation Model** Control Parameters (**SMCP**) for the seismic event and the relative intensity distribution (hazard input); this way, the first updating of WS is obtained. Next, the Building seismic vulnerability inventory +the intensity distribution are used as input for the DPM (or fragility curves) in order to calculate the probable damage distribution. This way the final WS (after application of building impact model) is obtained.



Output data

- Damage distribution (number of buildings belonging to each class performing the different damage states within each MU)

Realization:

PostgreSQL stored procedure

PostgreSQL stored procedures allows users to extend a database with user-defined functions by using various procedural languages.

Related Models and BBs:

Simulation Model Integration BB
Simulation Model Interaction View
Earthquake casualty model
Road network vulnerability model (RNV)
Time Dependent Vulnerability model (TDV)
Cascading Effects Model

Cascading Effects Model

Authors: Miguel Almeida
Luis Mario Ribeiro

Domingos Viegas
Valeria Reva
Alexander Garcia
Maria Polese

The cascading effect **model** for dynamic **scenario** assessment calculates the probability of attainment of cascading events scenarios, given an initial triggering **event**, or estimate consequence paths given the occurrence of selected scenarios. Within the model, a **database of scenarios** and a **transition matrix** are defined as the two fundamental pieces of information required to assess the effects of possible **cascading effects**.

A **hazard** **crisis** **situation** may be due to the occurrence of a single hazard event with large impacts or due to several hazard events that occur simultaneously. Hazard events occurring at the same time may have independent causes or may result from a sequence of triggering hazard events. The outcome of a situation for which an adverse event triggers one or more sequential events (synergetic event) is called “cascading effects” (Marzocchi et al., 2009, 2012).

The perception and understanding of the potential occurrence of cascading effects is of great relevance for planning and **response** activities since a surprising situation in a hazard **crisis scenario** may endanger people and goods, and may nullify a strategy that was developed accounting for a scenario in which the triggering event was a single occurrence.

Possible sequences of cascading events triggered by seismic activities (Figure 1), forest fires, floods, extreme weather conditions and chemical incidents can be modeled, with associated **transition** matrices. The model allows to customize the sequence of events and visualize data of transition matrix for each node of the chain of event.

Figure 1: Diagram of the identified cascade event chains for Earthquake.

Figure 2 shows an example of a probability map of fire ignition triggered by electric cable failure due to a seismic event with the characteristics presented in Figure 3. To produce the map of Figure 2 other inputs such as the fuel map of the area, the voltage distribution across the power line, the electric pole fragility curve and the ignition probability function were used as well.

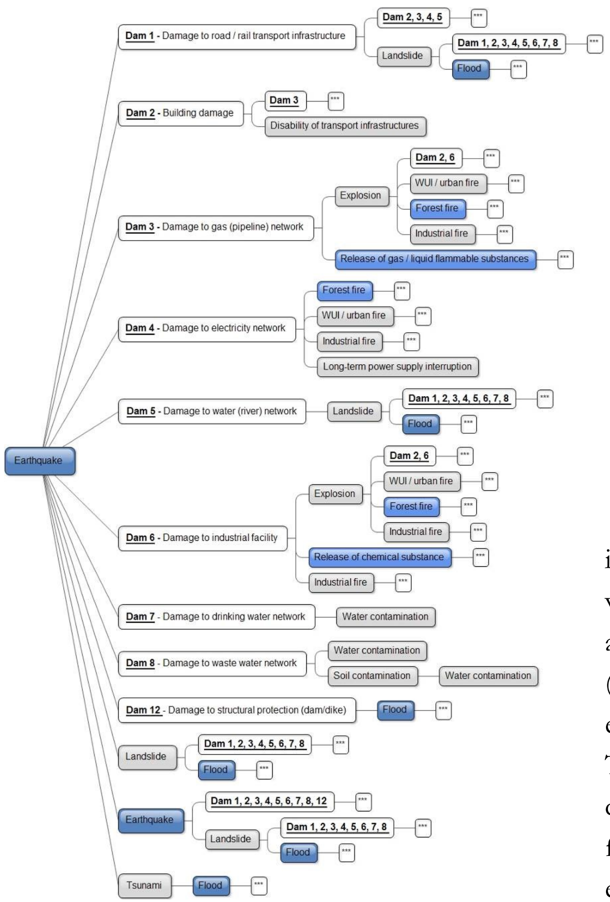
Figure 2: Map for probability of fire ignition by electric cable failure due to earthquake.

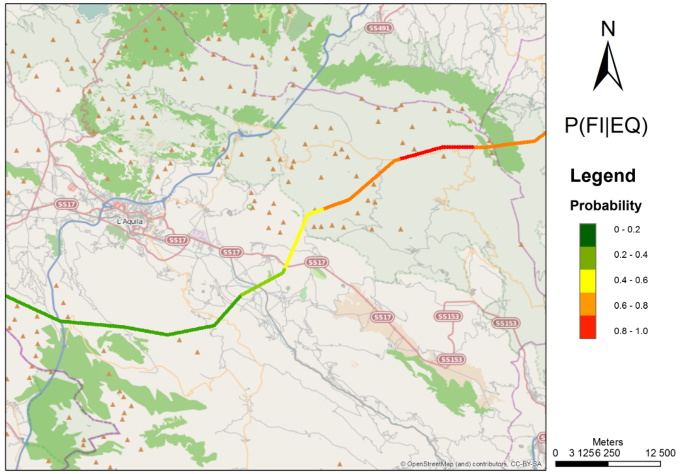
Figure 3: Map of the triggering earthquake intensity distribution with location of the electric power network (distribution lines - smaller poles).

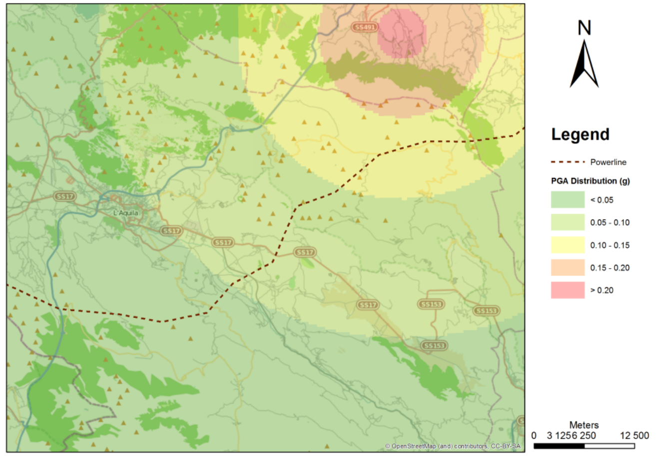
Documentation:

Presentation

The **cascading effects** **model** is based on the use of two fundamental concepts: (1) a database of cascading scenarios, and (2) a repository of **transition** matrices.







The database of cascading effects contains all the information about the identified scenarios of cascading effects developed for seismic **crisis**, forest fire, flood, extreme weather conditions and release of chemical substances. For example, the sequences of events that can be triggered (in a cascade sequence) after the occurrence of an earthquake (given triggering **event**) are: Earthquake -> Damage to gas (pipeline) network -> Explosion -> Forest fire -> etc.. The structure for the cascading effects is built on an event-tree like scheme, in which each of the nodes is quantitatively represented by a conditional probability.

The repository of **transition** matrices contains the probabilistic information necessary to quantify the expected damages produced by the potential cascading effects, and is a representation of the conditional probabilities of the form P(T1|E1), which means the probability of having the triggered event T1 given the occurrence of a triggering event E1. The **transition matrix** is N x M matrix, in which there are N classes of the intensity measures of the triggering event and M classes of intensity measure of the triggered event.

The triggering event is an event happening at a given time that is likely to produce a chain of adverse events. The direct effects of the triggering event (assessed e.g., using the **CRISMA** platform), are assessed in order to compute the direct consequences. Using the information from the database of cascading effects and the respective transition matrices (TM), the expected consequences of the chains of events can be quantified.

The probability values used in the transition matrices require different kinds of expertise, and the use of different sources of information. Not all the scenarios in the database may have respective quantitative analyses due to lack of the data available or models for theoretical analysis.

Realization:

TDV Python package

Collection of modules written in Python **object**-oriented programming language.

Related Models and BBs:

Cascade Events Configuration and Interaction View
Data Integration
GIS View Building Block
OOI World State Repository BB
Simulation Model Interaction View
Building impact model
Dikes vulnerability model
Earthquake casualty model
Forest fire behaviour model
Population exposure model
Road network vulnerability model (RNV)
Time Dependent Vulnerability model (TDV)

Coastal Submersion Model

Authors: **Mehdi Pierre Daou**
Armonie Cossalter

The Costal Submersion **Model** is a 2D-hydrodynamic model based on the **open source** TELEMAC-MASCARET **system**. In TELEMAC-MASCARET system, we use mainly TELEMAC2D to calculate the time and space dependent hydrodynamic characteristics such as water levels, velocities, discharges.

The 2D model takes into acount or compute the different phenomena:

- Hydrodynamics flow (maritime flow and land flow)
- Swell propagation
- Singularity treatment: weirs, bridge (including calculation of overflow on dikes and the dike failures)
- Meterological influence: wind, Mean Sea Level variation due to pressure...
- Tides

Documentation:

Functional Description

Input data:

The coastal submersion **model** uses the TELEMAC-MASCARET **System** and more particularly the Telemac-2D module (and possibly the TOMAWAC module) to run the coastal submersion **simulation**. **It** needs informations on the the **storm** **event**, the hydraulic structures and the elevation

- Geometry file (Mesh with projected bathymetry).
  - The global mesh is composed of a maritime mesh and a land mesh.
- Bondary conditions file:
  - is a file containing the geolocalization of the boundaries and their type (land, maritime).
- Hydrodynamics structure file:
  - The bridge and hydrodynamics structures are described in this file. This file is used to described the transfer of flow on each side of the structure. It indicates the nodes where the flow rate exchange is applied, at which level the water enters the hydraulic work and at which level the water goes out the hydraulic work, the width of the flow, and the flow transfer coefficients and types (check valve or normal flow).
- Dikes file:
  - During a storm, the water can overflow the dikes due to the swell and the phenomenum of waves run-up. In order to model this overflow, flow rate transfers between the two meshes (maritime mesh and land mesh) are used. The dike file includes the geolocalization and the height of the dikes, the swell overflow coefficients, and the nodes on both side of the singularity (a node on the maritime mesh and another on the land meh) where the exchange of flow is done.
- Dike failures file:
  - It indicates the failure position, its height and width, with start time and duration of the failure.
- Swell file :
  - It is a result file from the TOMAWAC module. The time-dependent results are given on the same maritime mesh introduced above. At each node of the mesh, the following variables are indicated:
    - the spectral significant wave height
    - the mean wave direction
    - the peak period
    - the driving force along X and along Y due to the radiation stresses
  - Note: it is possible to couple TELEMAC-2D and TOMAWAC. In that case the the swell file is a simple time series indicating the caracteristics of the swell at the boundary conditions. This swell will be applied at the maritime boundary indicated in the boundary conditions file.
- Sea area file:
  - It indicated the maritime area where the swell is applied. (used only if the TELEMAC-2D and TOMAWAC modules are not coupled)
- Wind file:
  - It is a time series indicating the wind direction and wind intensity. The described wind is used uniformly on the whole studied area.
- Mean Sea Level variation file:

  - It is a time series indicating the sea elevation due to the change of the atmospheric pressure during the storm. It is used on maritime boundaries that are indicated in the boundary conditions file.
- A folder containing a database to calculate the astronomical tides
- **Parameter** file:
  - It is the file where the calculation parameters are indicated (numerical schemes for calculation, time of simulation, name of the different required files, etc.)

Process:

After gathering all the data, the model is run thanks to an executable file.

If the simulated event is modified, the input files should be changed and should correspond to the description of the simulated event.

The results are written in the path indicated in the file parameter.

Output data:

The outputs are stored on the nodes of the mesh. They are the time-dependant hydrodynamic data (that are indicated in the parameter file). They are mainly:

- Water depth
- Free surface
- Water velocity
- Maximum velocity
- Maximum water depth
- Diffrence between water level and dikes

difference

- Open TELEMAC-MASCARET website **http://www.opentelemac.org/ (http://www.opentelemac.org/)**

Realization:

TELEMAC MASCARET System

TELEMAC-MASCARET is an integrated suite of solvers for use in the field of free-surface flow. Having been used in the context of many studies throughout the **world**, **it** has become one of the major standards in its field.

Related Models and BBs:

Simulation Model Integration BB

Dikes vulnerability model

Dikes vulnerability model

The dikes **vulnerability model** is a model programmed in python which allow to calculate the potential statistical **impact** on dikes depending on their status.The model is based on the damage probability matrix.

In view of results of dike vulnerability model, the user could make an informed choice on break or failure dikes for local **simulation**.

Damage probability matrix.

Documentation:

Functional Description

INPUT DATA:

- Dike status file: This shapefile contains the different status of dike sections .
- Link file : A text file which contains the link between dike sections and dike nodes of **simulation**.
- File of the diffrence between water level and dikes: which is a output of Coastal Submersion **Model**

PROCESS:

We run the dike **vulnerability model** with a button on user **interface** before the local simulation.

OUTPUT DATA:

- New dike status file: which contains the of damage probability on the dike sections.

Realization:

PyShp

The Python Shapefile Library (pyshp) provides read and write support for the Esri Shapefile format.

Water level above the dike	Status		
	Good	Medium	Poor
<20 cm	99,9% No failure 0,1% Breach 0% Total Failure	99% No failure 1% Breach 0% Total Failure	10% No failure 80% Breach 10% Total Failure
20 to 50 cm	99% No failure 1% Breach 0% Total Failure	10% No failure 80% Breach 10% Total Failure	5% No failure 15% Breach 80% Total Failure
>50 cm	98% No failure 2% Breach 0% Total Failure	5% No failure 15% Breach 80% Total Failure	0,1% No failure 4,9% Breach 95% Total Failure
>1 m	10% No failure 80% Breach 10% Total Failure	0,1% No failure 4,9% Breach 95% Total Failure	0,0% No failure 0,1% Breach 99,9% Total Failure

Related Models and BBs:

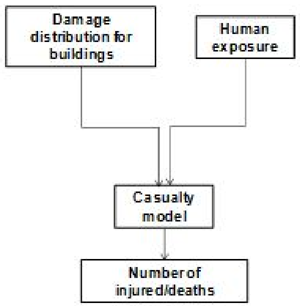
Simulation Model Integration BB
Coastal Submersion Model
Cascading Effects Model

Earthquake casualty model

Model for the assessment of expected number of injured and deaths due to an earthquake

The probability of injury or death of the building occupants is generally evaluated as a function of the damage level of the building, and **it** can be assumed that ratio of injured and deaths are significant only for higher damage levels (D4 and D5 of the European macro-seismic EMS-98 scale).

The earthquake casualty model is logically inserted (as regarding the overall **CRISMA** model workflow) after the seismic building **vulnerability model** that allows calculation of expected damage for building vulnerability classes, and also takes into account information on human **exposure** (as derived from the human exposure model), see figure below.



Documentation:

Functional Description

Functional description

Considering that the results of the **physical vulnerability model** (in terms of potential damage distribution) and of the human **exposure** model are evaluated at the Minima territorial Units (MU) level, the casualty model is applicable for each single MU; the results for all the MUs in the study area represent the territorial distribution of injured/deaths.

Input data

Input data are given at the Minima territorial Units (MU) level (e.g. grid of 500x500 m²)

- estimated number of persons in each cell (MU) during variable time stamps (e.g. day-time night-time, week days, week-end days etc.) (from Human exposure model HEM)
- distribution of occupants among different building vulnerability classes (from enhanced HEM)
- damage distribution for each class, with particular reference to damage levels D4 and D5 (from building **impact model**)

Process

The number of deaths (Nd) and injured (Ni) are then determined by the following expressions:

$$N_d = TI_c \cdot \sum_{t=1}^{4} \sum_{j=1}^{5} N_{t,j} NO_t QD_{t,j}$$

$$N_i = TI_c \cdot \sum_{t=1}^{4} \sum_{j=1}^{5} N_{t,j} NO_t QI_{t,j}$$

where:

t = building type (t = 1, … 4)

j = damage level (j = 1, …… 5)

Nt,j = number of buildings of type t having damage level j

NOt = number of occupants (at the time of the **event**) by building type

TIc = Touristic Index by city

QDt,j = proportion of deaths by building type and damage level

QIt,j = proportion of injured by building type and damage level

The casualty **evaluation** is obtained as a proportion of the occupants of the building, according to damage level, classified by vertical building structure type:

Reinforced Concrete (R.C.) or Masonry (see Table below)

	Damage level							
Casualty percentage	D_0	D_1	D_2	D_3	D_4	D_5	Vertical structure	Vulnerability class
QD	0	0	0	0	0.04	0.15	Masonry	A or B or C
QD	0	0	0	0	0.08	0.3	R.C.	C or D
QI	0	0	0	0	0.14	0.7	Masonry	A or B or C
QI	0	0	0	0	0.12	0.5	R.C.	C or D

Nt,j is evaluated as a result of the physical (structural) vulnerability analysis.

NOt is evaluated as a result of human exposure analysis, that has to be performed with extended **feature** allowing, in addition to the estimation of population during day/night in each cell, to evaluate their distribution among building vulnerability classes.

Output data

- number of deaths (Nd) and injured (Ni) for each MU

Further info may be found in Chapter 4 and 5 of D43.1 (downloadable at **https://workspace.vtt**.fi/sites/eu_crisma/Deliverables/Sent%20to%20the%20Commission/CRISMA_D431_final.pdf)

Realization:

PostgreSQL stored procedure

PostgreSQL stored procedures allows users to extend a database with user-defined functions by using various procedural languages.

Related Models and BBs:

Building impact model
Population exposure model
Resource Management Model
Cascading Effects Model

Economic impacts model

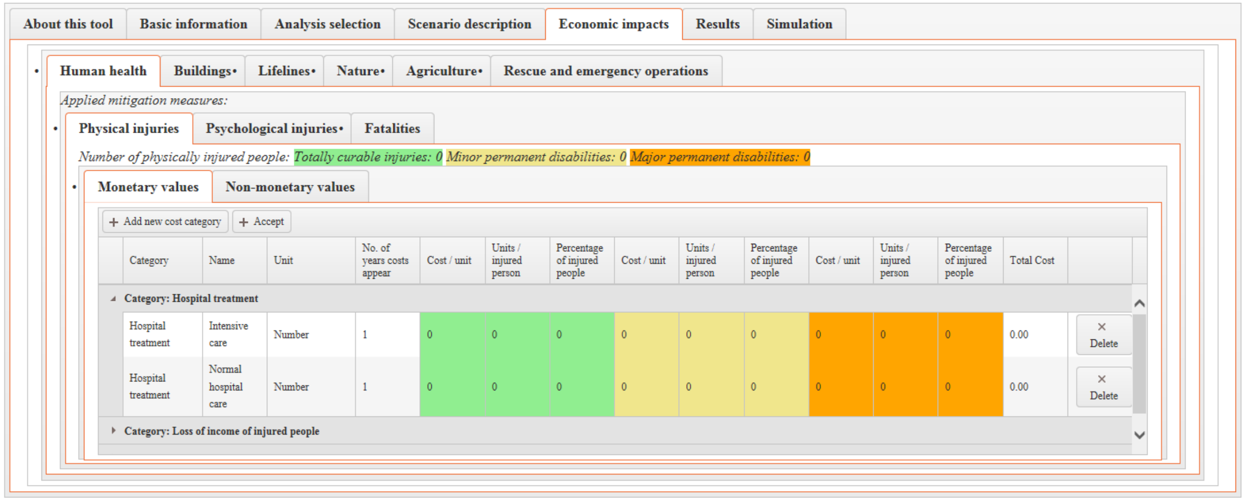
Authors: **Susanna Kunttu**
Minna Raikkonen
Markus Jähi

Model for:

- presenting economic impacts arising from crises (ex post performance) and
- assessing different **mitigation** proposals and their costs/benefits (ex ante planning).

The economic impacts **evaluation** model is intended to be used in the **preparedness** phase of **crisis management** to support long-term strategic decision-making.

The model uses data on alternative scenarios (e.g. base line **scenario** and **crisis scenario** after implementing a mitigation measure) in order to make the economic assessment. The assessment is done by determining economic **losses** of a crisis and costs and benefits linked to different mitigation investments. The procedure follows a passage from a **vulnerability** analysis and potential damage estimation to a loss assessment, focusing on discovering how such "damage" may be converted into economic losses.



Example view of CRISECON GUI component which provides a graphical user interface (GUI) for the model

Documentation:

Functional Description

Input data (other than economic data)

Economic assessment is conducted on a **crisis situation** simulated by other **CRISMA** models, e.g. **resource management model**. Required input data from those other CRISMA models can be classified into three categories:

1. Damages

The **damage cost** is dependent on the quantity of damaged items, the type of damaged items and the level of occurred damages. The total number of the damaged items should be classified into homogenous categories in a way that costs per an item (e.g. house, affected people) within a category are equal. In practice costs per an item cannot be equal but they need to be equal enough to be able to make sure that one average cost value is a sufficient approximation for all items of the category in question. That is because the total cost of a category is calculated by multiplying number of units by the average cost. For example in the table below cost for hospital treatment could be on average 500 euros in the row one and 800 euros in the category on the row 2. Total cost for slightly injured is 500 €/person x 150 persons = 75000€ and respectively total cost for more seriously injured is 800 x 100=80000€. Fixed damage categories in the model are human health, buildings, lifelines, nature and agriculture. Sub-categories can be defined according to the case at hand. The damage level is assumed to have three levels. An example of the possible input data table is given below:

2. Resources used during **emergency** and rescue operations

Row id	Main damage category	Sub damage category	Damage level	Unit	Number of units
1	Human health	Physical injuries	Level 1	Number	150
2	Human health	Physical injuries	Level 2	Number	100
3	Buildings	Residential	Level 1	Number	300
4	Buildings	Hospitals	Level 1	Number	3
5	Buildings	Commercial	Level 3	Number	50
6	Lifelines	Power supply system	Level 3	Km	5
7	...				
...					

Row id	Main resource category	Sub resource category	Number of units	Active work [Average hour/ unit]	Reserve work [Average hour/ unit]
1	People	Police force	50	24	30
2	People	Medical	80	15	15
3	People	Volunteers	60	30	30
4	Vehicle	Ambulance	5	40	20
5	Vehicle	Transporter	20	50	10
6	Other	Evacuation centre	2	60	0
7	....				
...					

The costs of used resources are dependent on used resource types, the quantity of used resource units and the quantity of active and reserve working hours during emergency and rescue operations. The total number of resources used should be classified into categories in which cost parameters and values are equal enough (the same procedure as in the context of crisis damages). Main resource categories in the model are personnel, vehicles, equipment and other resources. An example of possible input data table is given below:

3. **Mitigation** measures (applicable for assessment of economic impacts of mitigation interventions).

When assessing the economic impacts of the mitigation interventions, applied mitigation measures have **to be defined** before the crisis **simulation** either in the economic impacts **evaluation** model or as an input data from other CRISMA models.

Process

Cost assessment and Cost and benefit analysis (**CBA**) are the main procedures the assessment is following. The model calculates damage cost, costs of an emergency and rescue operations and costs and benefits of mitigation measures aimed at preventing or reducing crisis effects. Steps for the economic assessment are:

1. Basic information about the decision situation is filled in by the user
2. Analysis selection where the user selects the type of the analysis. There are two different types of the analysis:

(i) economic impacts of consequences of a crisis (cost assessment) and

(ii) economic impacts of mitigation interventions (**cost-benefit analysis** CBA).

3. Entering cost (and benefit) related data comprises three phases:

a. Input data (see input data definition above) transferring from CRISMA repositories.

b. The definition of a cost (and benefit) structure for all damage items and resource types and for mitigation investments (user input)

c. The estimation of monetary value for defined cost (and benefit) factors (user input)

4. The results of the current simulated crisis situation are calculated by the model and are presented by figures and tables.
5. Main results are transferred to a repository to be used, for example, in multi-**criteria** analysis

Output data

As the output of an economic assessment, the model provides selected economic result indicators (e.g. monetary values such as Present Value of Costs PVC, Net Present Value NPV …). The results are summarized both in numerical and in graphical forms and they can be further used as an input for other CRISMA models, summarized together with the results from other CRISMA models (i.e. aggregated results) or used as such to support decision-making.

Downloads:

- https://crisma-cat.ait.ac.at/system/files/uid_67/CRISECON_model_and_software_D442_1.doc

Realization:

CRISECON Service

CRISECON **Service** implements, together with the related UI component CRISECON **GUI**, the CRISECON **model** developed in SP4.

The key functionalities CRISECON Service are:

Related Models and BBs:

Economic impacts analysis view

Evacuation Model

The **Evacuation** **Model** is a prototypical model that represents how the population can be evacuated from the **hazard** area(s) to the safety zone(s). The purpose of the evacuation model is to determine how fast and in which health condition(s) the population is able to leave the hazard area(s).

The Evacuation Model enables to represent units of the terrain and units of the population. Each unit of the population is different in terms of its location and population density and other parameters pertaining to the given social group - elderly, children, people at home, people in offices, etc. Each unit of the terrain is characterized by the roads within **it** and the type of the terrain - plain, mountains, etc. The roads can have "signs" pointing to the directions in which the units of the population should evacuate and which roads the population should take. In case there are no "signs" on roads, the units of the population in the **simulation** should have some "knowledge" of what is the required direction of evacuation in case of a particular type of hazard in particular location(s). The speed of evacuating a given unit of the population from a particular location depends on the distance to the safety zone(s), the roads available, the kind of the social group in question, and the type of hazard under consideration.

There can be one or more hazard areas and one or more safety zones on the terrain. Please note that not all features described above have been implemented, as the Evacuation Model is a prototypical **demonstration**. However, should the need arise, the missing features can be relatively easily added to the existing Evacuation

Model demonstration.

Documentation:

Demo

1. Introduction

The **evacuation** **model** was built using an example **scenario** from Austria. However, the model can be easily applied to any **crisis scenario** addressed in the project, where evacuation is needed. The idea of the evacuation model is to determine how fast and in which conditions population is able to leave the **hazard** area.

The evacuation model is conceptually an **agent**-based **simulation**. Each **object** in the simulation is viewed as an agent, no matter if **it** describes a unit of the terrain or represents a unit of the population. Each agent of this kind has a specific set of properties and is able to execute a specific set of actions. For instance, the agent that represents a unit of the population is able to move on the map. On the other hand, the agent that represent a unit of the terrain on the map has properties representing if the given part of the map is within the simulation boundaries and if it contains roads.

A single agent that represents a unit of the population represents a group of people at a given place at some specific time. Naturally, each particular place on the map can have a different population density depending on the time of the day. For example, if we consider a business district in the center of a city, it will be crowded with people during the office hours. In contrast, during the evening and night hours the whole place will be deserted.

2. **Simulation tool**

NetLogo is a highly customizable tool that can be used for agent-based simulation. With a small learning curve it is quite easy to implement basically any kind of simulation of crowd behavior. Taking advantage of this, we were able to successfully import a simplified scenario from Austria and execute the basic evacuation simulation.

The NetLogo environment offers a basic object-oriented programming language. The language allows to operate with different objects and change their behaviors. In our simulation these objects are viewed as agents, even though NetLogo does not offer a **concept** of autonomous social software agents. As a result, there is also no concept of communication between agents in NetLogo. The only concept of interactions available in NetLogo is that of a direct physical interactions between agents that bump into each other. To some extent communication can be represented through variables shared by different agents. However, this is nearly not enough because communication between people lying in different locations by different kinds of information exchange means can strongly influence evacuation of any kind. This is the biggest deficiency of NetLogo.

3. Logic

The implemented simulation itself is very straightforward. The goal (represented by programming constructs) of each agent representing a unit of the population is to run away from the map. The map area is considered a danger zone, while the area outside of the map is considered a safe zone. Each agent starts moving forward in a random direction, as it does not know in which direction is the safe zone. At the moment the agent crosses the map boundary it enters the safe zone. If the agent finds a road on its way, it will try to follow the road, considering that roads in our simulation usually lead to safety. However, if the agent still does not know which way to move on the road, it randomly finds a direction and follows the road. If the road suddenly ends or if the agent finds itself outside of the road for some reason, it will choose the opposite direction and will follow the road again.

It is worthwhile to emphasize here that the implemented simulation is very straightforward because its purpose is to offer a proof of concept that the NetLogo tool can be used for simulating evacuations. In particular, the behaviour of the agents in the simulation is not overly realistic and serves mainly the **demonstration** purpose. However, this simulation has a great potential and should be definitely improved in the future.

4. Future development

The first goal for future development is to make agents representing units of the population smarter, so that they could choose a rational direction in which to evacuate. Also, units of the population should be “aware” of the location of the safe zones are and choose the direction accordingly. Currently, the safe zone is everything outside of the map boundaries, but there can be dedicated safe zones on the map instead of that.

Several improvements of the simulation require the agents to have a more detailed information of their location. In addition, the map should include even minor roads that lead to major thoroughfares in any given location. The simulation should also include "signs" pointing to the directions in which the units of the population should evacuate.

Secondly, the speed of the moving agents should also depend on various factors. For example, if a single population agent represents a unit of the population in the high population density zone, its speed should be slower. Likewise, we may want to consider different social groups and individuals.

Thirdly, partially related to the first goal for improvement described above, the knowledge by the agents of the location of the safe zones and the corresponding decisions on the evacuation direction should depend on the type of the scenario to be simulated. For example, in case of a flood scenario, it is easy to know where the hazard is and that one should run in more or less the opposite direction. Differently, in case of a radiation hazard one needs to know where a safe zone is even if it is in the direction of the hazard.

Realization:

Evacuation model (NetLogo)

Evacuation **model** prototype.

The idea of the evacuation model is to determine how fast and in which conditions population is able to leave the dangerous area. The model reads two input files with the data for the terrain and the population. After that we are ready to simulate:

- All civilians try to exit the dangerous area.
- They try to follow the roads, as they should lead to shelter.

Related Models and BBs:

Evacuation model for coastal submersion

Evacuation Resources Simulation Model

Evacuation Resources **Simulation Model** calcuates the **impact** of **resource** allocations chosen by the user (from the proposals based on the **preparedness** plan) to mitigate the **situation**.

For example, if the situation requires evacuation of 200 peoples, the user handles one bus (**capacity** 50 persons) to transport the people, the bus needs 4 round trips to evacuate them all. Depending on the duration of the round trip and the temporal interval between **world** states this may have an impact on several world states within a **scenario**. This model is used to calculate the progressing situation over the whole duration of the scenario.

Documentation:

Functional Description

Input data (from **OOI**-WSR/Input **world state**):

- input world state (e.g. first world state in the new **scenario** path)
- output world state (the next world state in the same scenario path)
- Number of people in need of **evacuation** in the selected area
- Availability and **capability** of the allocated **mitigation** resources

Output data (to OOI-WSR/Output world state)

- **Resource** usage
- Evacuation **impact** (number of persons evacuated)
- Number of people still in need of evacuation

Realization:

Insta EvacSim

Insta EvacSim is a simple tool simulating the **impact** of resources allocated to evacuate people from a given area. **It** calculates both the progress of the **evacuation** as well as the **resource** usage.

Related Models and BBs:

Preparedness Plan BB

Evacuation model (NetLogo)

Authors: **Stanislav Vasilyev**
Denis Havlik

Evacuation **model** prototype.

The idea of the evacuation model is to determine how fast and in which conditions population is able to leave the dangerous area. The model reads two input files with the data for the terrain and the population. After that we are ready to simulate:

- All civilians try to exit the dangerous area.
- They try to follow the roads, as they should lead to shelter.

Our **simulation** was built on Austrian **scenario**, but can be easily applied to any other available in the project. Underlying technology used to tealise the model is the one of software **agent** based simulation.

Documentation:

Demo

Downloads:

Team:

Stanislav Vasilyev

Development roles: development

Evacuation model for coastal submersion

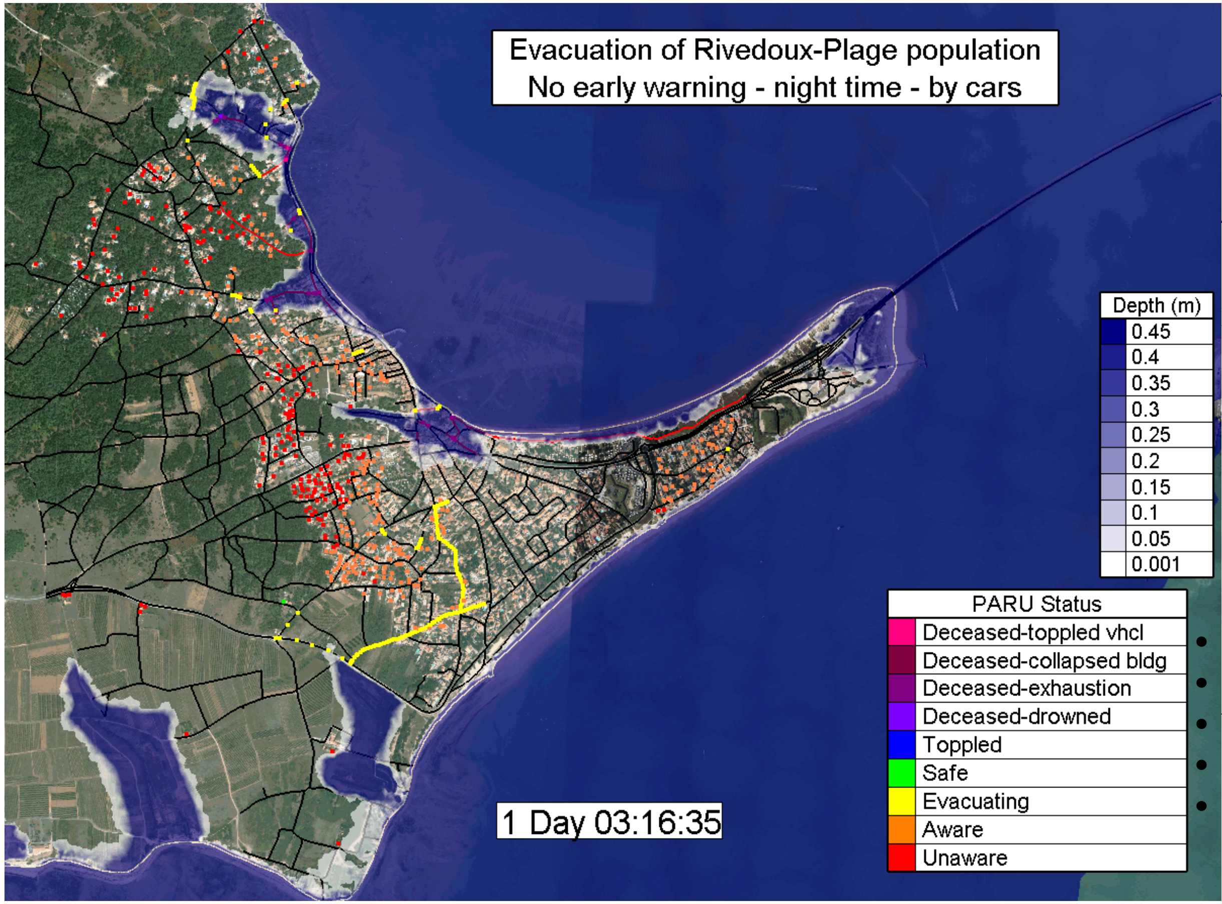
Authors: **Marianne Grisel**

Armonie Cossalter

The **evacuation model** developed for coastal submersion in Charente-Maritime (France) used the software LSM2D. This model cover the Rivedoux-Plage area in Ré Island as well as the area on the coast from Yves to Chatellaion. Different scenarios of evacuation could be simulated. The main results of this model are:

- the estimated time to evacuate the population
- the estimated closed roads
- the estimated casualties on population
- the estimated impacts on buildings due to the flood

The model results are illustrated below. A demo video showing the development of the **situation** over time is included in the documentation section.



Documentation:

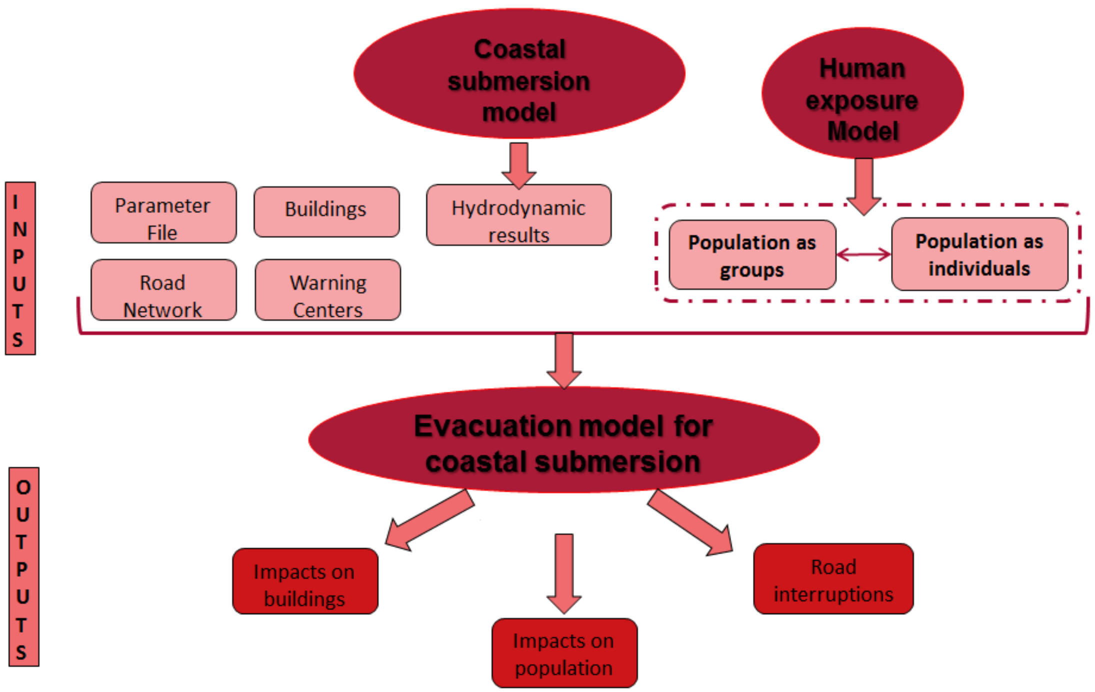
Demo

This video shows a demo of the **evacuation model** for coastal submersion.

The simulated **hazard event** is the histoirical storm surge **Xynthia** (27th and 28th of February 2010) with an additional 20cm of premium. Moreover, the hazard event also includes the failures of the different dikes. The people of Rivedoux-Plage (municipality of the Ré Island in Charente-Maritime in France) are evacuating by cars. There is no early **warning** and the **simulation** takes place during the night. The water depth are represented with a scale of blue. Further, the people are represented by the colored dots. The different colors correspond to different status:

- Red: the people are unaware of the **situation**
- Orange: the people are aware of the situation
- Yellow: the people are evacuating
- Green: the people are safe
- Purple: the people are dead

Functional Description



Input data

The evacuation model needs information on the the **hazard** **event** (hydrodynamic data) that is simulated during the evacuation time and on the different stationary (buildings, roads and **warning** centers) and moving objects (population).

The evacuation model requires 10 files:

- 4 files for the hydrodynamic data on the studied area.Each file is describing the water depth, free surface, water velocity and bathymetry. They are outputs of the coastal submersion model.
- 1 file describing and geolocating the road network.
- 1 file describing and geolocating the buildings.
- 2 files discribing and geolocating at time zero the population. A first file describes the population individually (People at **Risk** Unit = PARU), a second file describes the population by groups (People at Risk Group = PARG). The groups are used to defined the people that are evacuating together. Each person should belong to a group even if **it** is a group of one person.
- 1 file describing and geolocating the warning centers. They are the centers that will **alert** the population of the **situation**. This alert will trigger the start of the evacuation. The rate and initial warning time are set in this file.
- 1 **parameter** file that indicates the path of each input file and the different parameters' value.

To these 10 files can be added an event file describing the different changes of objects' status during the **simulation** (status of building, road or person). For example, it can be used to close a road during the simulation even if it is still not flooded.

Process

The Life Safety Model 2D (LSM2D) is launch in a shell indicating the parameter file. The LSM2D loads the different inputs file and runs the simulation.

The outputs are written in the indicated folder within the parameter file.

Output data

The outputs of the evacuation model are the evolution in time of the different **object** during the simulation:

- the evolution of buildings' status (standing or destroyed)
- the evolution of population status (unaware, aware, evacuating, safe, toppled or deceased). The description is made in two files, one for PARU and another for PARG
- the evolution of vehicles' status (parked, driving, safe, floating, toppled)

The outputs are summarised in a text file. It contains a brief summary of the simulation including the input file names and their directories, the duration and time step of the simulation, and the final status of the individuals, groups, vehicles and buildings at the end of the simulation.

- Life Safety Model website
  http://www.lifesafetymodel.net/ (http://www.lifesafetymodel.net/)

Realization:

Life Safety Model 2D

The Life Safety **Model** 2D (LSM2D) software is a **simulation tool** that performs **agent** based simulation and models the dynamical interaction between moving (pedestrians and vehicles) and stationary (buildings) environment objects with flood wave.

Related Models and BBs:

Data Integration
Simulation Model Integration BB
Evacuation Model
Population exposure model

Forest fire behaviour model

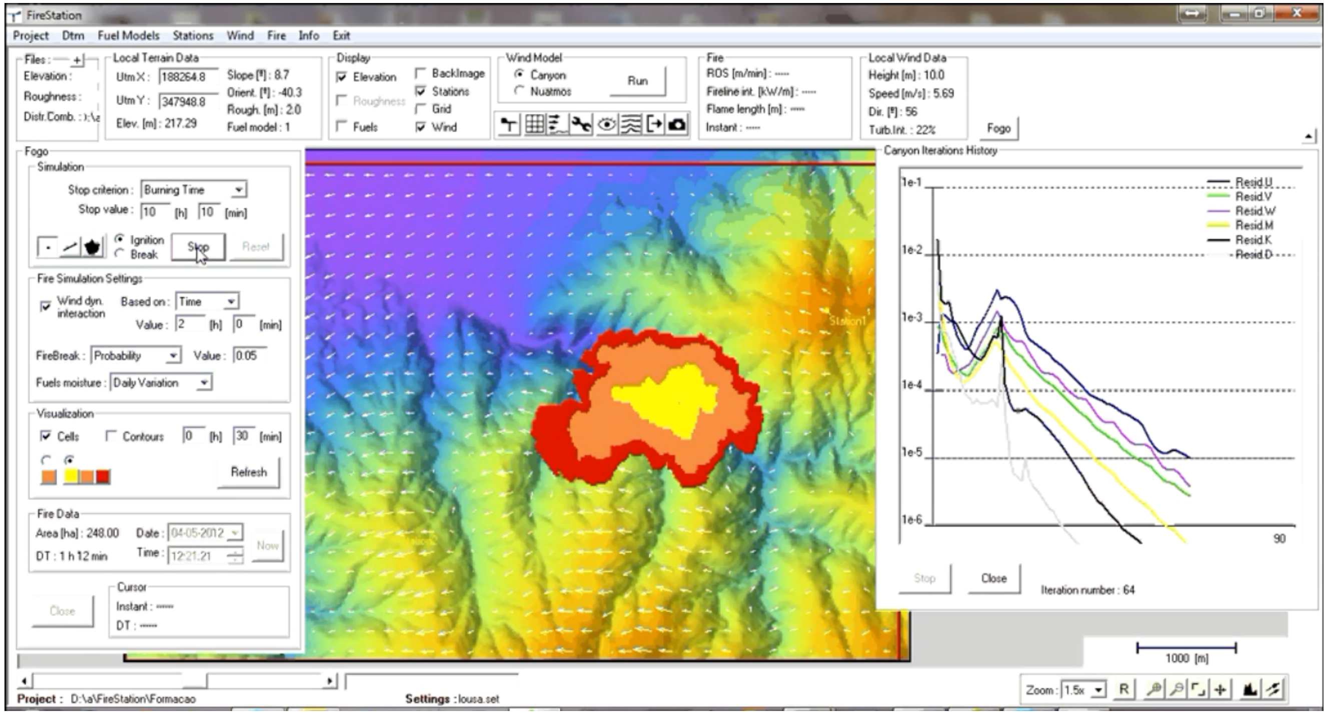
Authors: **Luis Mario Ribeiro**
Miguel Almeida

The Forest Fire Behaviour **model** is a deterministic integrated **system**, based on the Model of Rothermel (1972), for the spatial **simulation** of forest fire behaviour over complex topography and wind flows in areas with heterogeneous vegetation cover. Its main components are the fire behaviour predictions at local scale and wind field prediction at local and large scale taking into account different thermal and recirculation effects.

Firestation also has the **capability** to simulate smoke dispersion and particles concentration over the area afected by the simualted fire.

There are three main factors that can affect the intensity of fire, creating peaks in the rate of fire spread: high wind velocities, steep topography and vegetation dryness. All these parameters are considered in the model through topographic and fuel cover maps and data of local meteorological stations. However, extreme fire behaviour phenomena such as spot fires, tornados or blow-up fires, cannot be modelled.

The figure below is a **screenshot** of the Firestation software during a fire simulation.



Documentation:

Publications

Please follow the links to access Firestation publications:

http://www.sciencedirect.com/science/article/pii/S136481520100072X (http://www.sciencedirect.com/science/article/pii/S136481520100072X)

http://www2.dem.uc.pt/antonio.gameiro/ficheiros/artigos/FStation.pdf (http://www2.dem.uc.pt/antonio.gameiro/ficheiros/artigos/FStation.pdf)

- Paper describing Firestation **http://www2.dem.uc.pt/antonio.gameiro/ficheiros/artigos/FStation.pdf (http://www2.dem.uc.pt/antonio.gameiro/ficheiros/artigos/FStation.pdf)**

Realization:

Component to be defined

This is a placeholder which allows us to define building blocks and **CRISMA** models for which no components have been defined yet.

Related Models and BBs:

Cascading Effects Model

Patients model

Authors: **Kalev Rannat**

This **model** governs the behaviour of the patients in Israeli and German pilots. Full description shall be provided by **TTU**.

De-facto we have two different models, since the Israeli patients are affected by bromine poisoning and the German ones by mechanical damage (injury). And there is also the aspect of patients moving around on their own - at least in the israeli **pilot**. I am not sure how fine-grained we want to need to do this (is this 1 or 2 models? do we have "people moving arround" as another separate model?), so I'll leave this question for Kalev to answer. BUT: whatever the decision, please don't forget to update the "constituents" in related **application** descriptions!

Documentation:

User Guide

todo

Realization:

Dynamic Map Agents

This is a description of a software candidate that offers tools for **agent**-based simulations for the implementation of a functional **Building Block** (Agent-Oriented **Simulation** Models Building Block), together with **OOI** **World State** Storage **Service** (**WFS**) **BB** (**NICE**), **Resource management** integration and deployment (**WPS**) BB (CASS), Indicators BB (**AIT**), Tactical RM Training BB (Situator integration) (**NICE**) and RM Training Dispatch and Monitor View (**AIT**), RM Training Simulation **Scenario** Setup View (AIT), RM Training Indicators and Statistic View (AIT).

Related Models and BBs:

Agent Oriented Simulation Models

Population exposure model

Authors: **Klaus Steinnocher**
Christoph Aubrecht
Heinrich Humer
Hermann Huber

Model for distributing population in spatial and temporal dimensions.

This model uses temporal and spatial proxies in order to disaggregate the population from administrative units to spatio-temporal grids. The outcome is used in **CRISMA** as basis for time-dependent **exposure** assessment and in further steps as a base data for **evacuation** and casualty modeling (Aubrecht et al., 2014ab; Steinnocher et al., 2014).

Population data is usually available from census as totals per inhomogeneous spatial reference unit. For modelling population exposure, data is required that is independent from enumeration and administrative areas. Raster representations meet this demand but worldwide spatially and thematically consistent population rasters aren't available today.

Documentation:

Functional Description

Input data

Three types of input data are required to run the **model** :

Population data:

- total population per administrative unit (table)
- administrative borders (polygon layer)

Spatial data:

- density grid for residential areas - **H**ome (raster)
- density grid for transportation areas - **C**ommuting (raster)
- density grid for work areas - **W**ork (raster)
- depending on the availability of other types of location (e.g. touristic places, shopping places, etc.) corresponding density grids can be used

Temporal data:

- diurnal variation (e.g. hourly time steps) of residential population (H) (table)
- diurnal variation of commuting population (C) (table)
- diurnal variation of working population (W) (table)
- diurnal variation of other types of population (e.g. tourists, school kids) depending on the availability of corresponding data on location

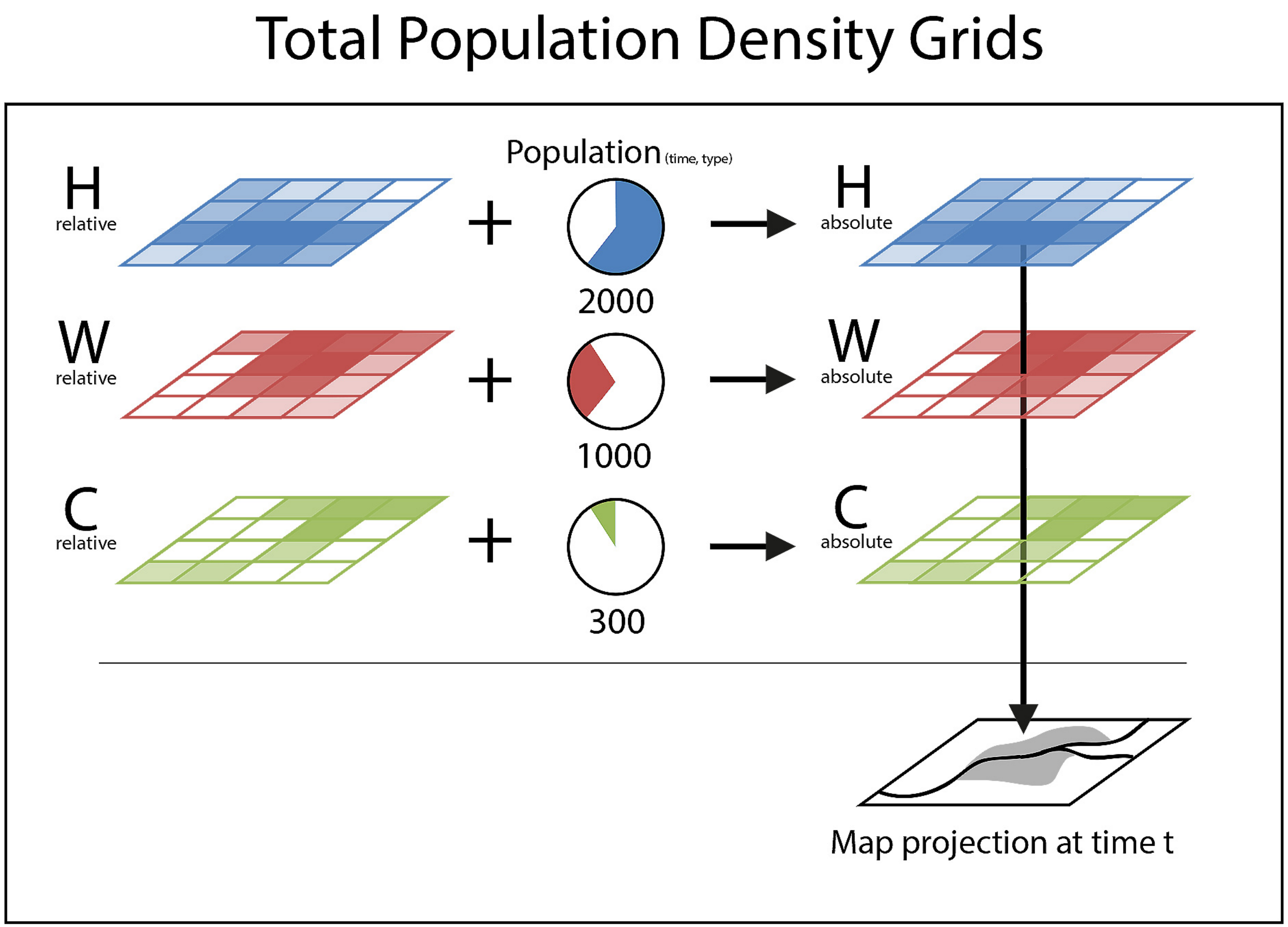
Process

Generally, the model to calculate time dependent spatial population disaggregation consists of two steps:

1. Calculate total amount of population per class (H, C, W) being present in a given area at a specific time slot (temporal domain).
2. Use invariant distribution grids to calculate population density in the respective area (spatial domain).

To fulfil spatial disaggregation, each population group (H, C, W) being present in a specific administrative unit at a specific time slot is distributed according to the density values of the corresponding grid.

Cell values are calculated by multiplying the total of a population group (in the given administrative unit at a specific time slot) with the density values of the corresponding grid. This calculation is performed for each population group per time slot. The resulting population grids (H, C, W) are added to result in one population distribution grid per time slot.

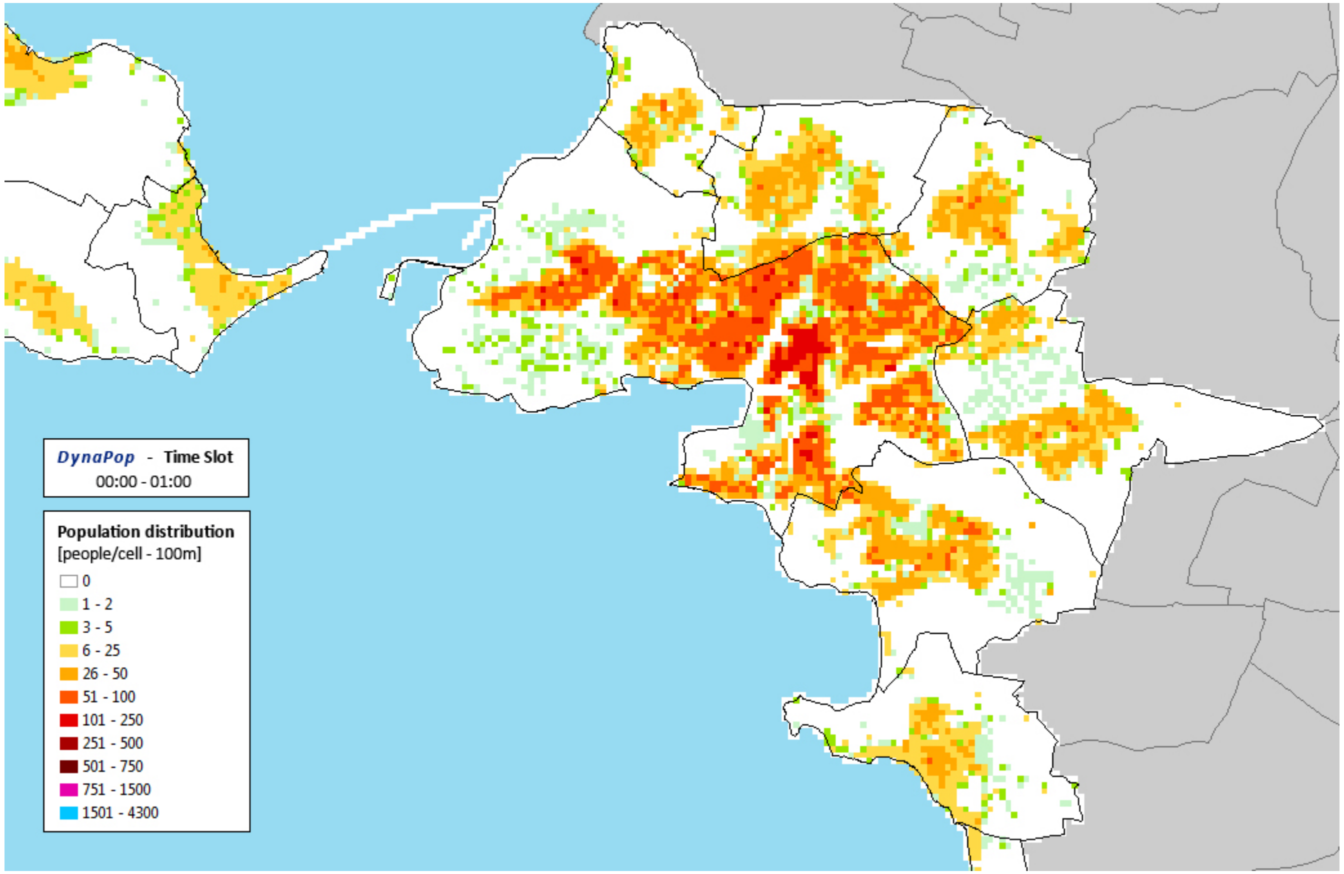


Output data

The model delivers one disaggregated population grid per time slot, so the resulting data set could be described as a data cube with two spatial dimensions and on temporal dimension. For use in **exposure** modelling the required time slot can be chosen and the modelled population distribution for this time slot is provided.

References

1. Aubrecht, C., K. Steinnocher, H. Huber (2014) *DynaPop – Population distribution dynamics as basis for social impact evaluation in crisis management*. In: S.R. Hiltz, M.S. Pfaff, L. Plotnick, A.C. Robinson



(Eds.): ISCRAM 2014 Conference Proceedings (Book of Papers), 11th International Conference on Information Systems for Crisis **Response** and Management. University Park, PA, USA, May, 2014. In review.

2. Steinnocher, K., C. Aubrecht, H. Humer, H. Huber (2014) *Modellierung raum-zeitlicher Bevölkerungsverteilungsmuster im Katastrophenmanagementkontext*. In: M. Schrenk, V. Popovich, P. Zeile, P. Elisei (Eds.): REAL CORP 2014: 19th International Conference on Urban Planning and Regional Development in the Information Society - Clever Solutions for Smart Cities. Proceedings. Vienna, Austria, May 21-23, 2014. In review.

Realization:

Emikat

Emikat is a **data management** and modelling **system** for spatially resolved data. The system allows the integration of emission-relevant activity data (e.g. from surveys, traffic models or statistics) as well as the definition and analysis of **model** scenarios. Scenarios allow a comparative examination of model results corresponding to different versions of data inventories – for example the overall effects of trends in emission generating activities and pollution control measures or the influence of different calculation models on estimated results.

Related Models and BBs:

Indicator Building Block
Cascading Effects Model
Earthquake casualty model
Evacuation model for coastal submersion
Road network vulnerability model (RNV)

Resource Allocation Tactic Model

This **model** replicates the decision making process of **crisis** managers in **resource management** planning applications. **It** allows us to **test** the planning scenarios with different management strategies, e.g. "prioritize transport to hospital" or "prioritize treatment on the field". It realizes a second layer above the Ambulance Model in order to being capable of running simulations with predefined strategy settings. These different strategy settings for resource planning allow users determining the best strategy in a specific **scenario**. Depending on the strategy ambulances in an idle state are assigned to certain commands before rerunning the ambulance model **simulation**.

Figure1: States of the ambulance vehicles relevant for the **Resource Allocation Tactic Model**

After and during running the ambulance model with specific commands assigned to specific vehicles, vehicles change their states as depicted in Figure 1. A soon as vehicles are in state "ready on site" or "ready" the **Resource Allocation Tactic Model** can assign new commands to them (write new **world state** enhanced with those commands) and restart the ambulance model.

Documentation:

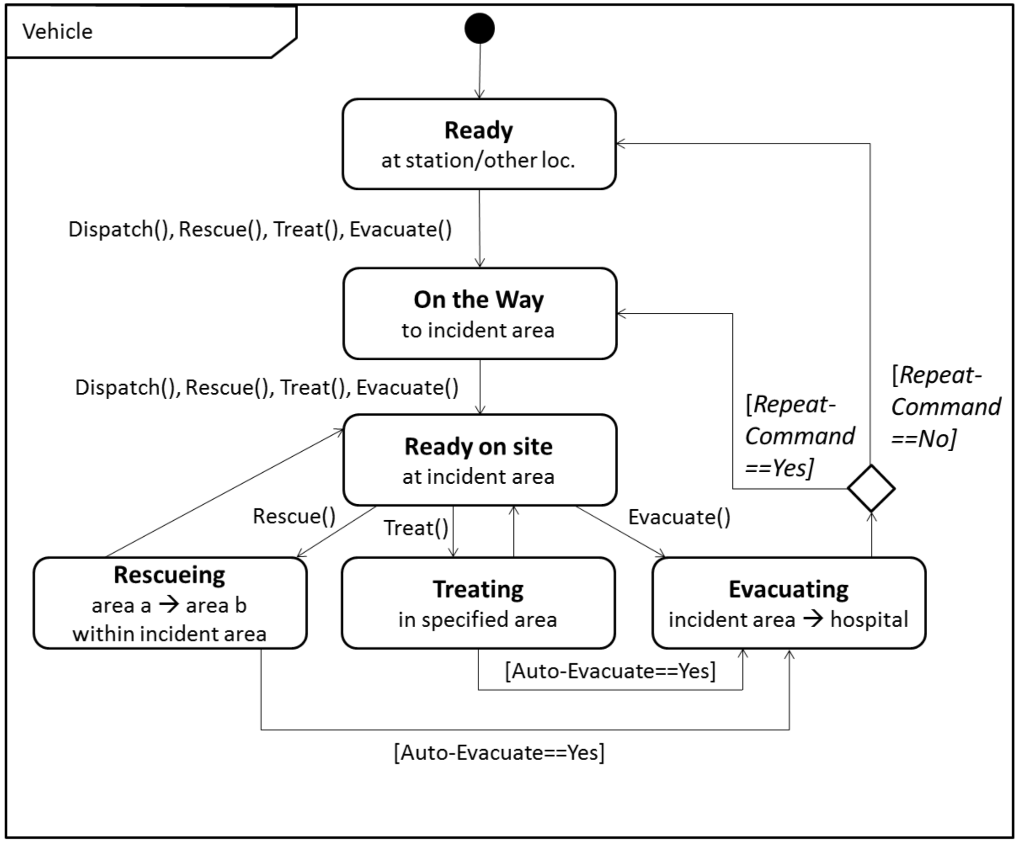
Functional Description

Figure 1 illustrates an overall **simulation run** controled by the **Resource Allocation Tactic Model**.

Figure 1: Overall simulation run of the resource planning **application** controlled by the **Resource Allocation Tactic Model**

The steps of the technical workflow can be described as follows:

1. The **Resource Allocation Tactic Model** starts the simulation



2. Depending on the exit criterium (time, available vehicle), the simulation stops
3. The **Resource Allocation Tactic Model** reads the latest **World State**
4. Depending on the strategy settings the **Resource Allocation Tactic Model** decides on the next actions
5. The **Resource Allocation Tactic Model** copies the latest World State and adds a new command
6. The **Resource Allocation Tactic Model** starts the simulation

Figure 2 shows the overall **architecture** of the resource planning application.The **Resource Allocation Tactic Model** plays an essential **role** and enables the user running planning simulations with given tactics, locations and vehicles to deploy that create immediate results.

Figure 2: Architecture of the **Reference Application** for Resource Planning

Related Models and BBs:

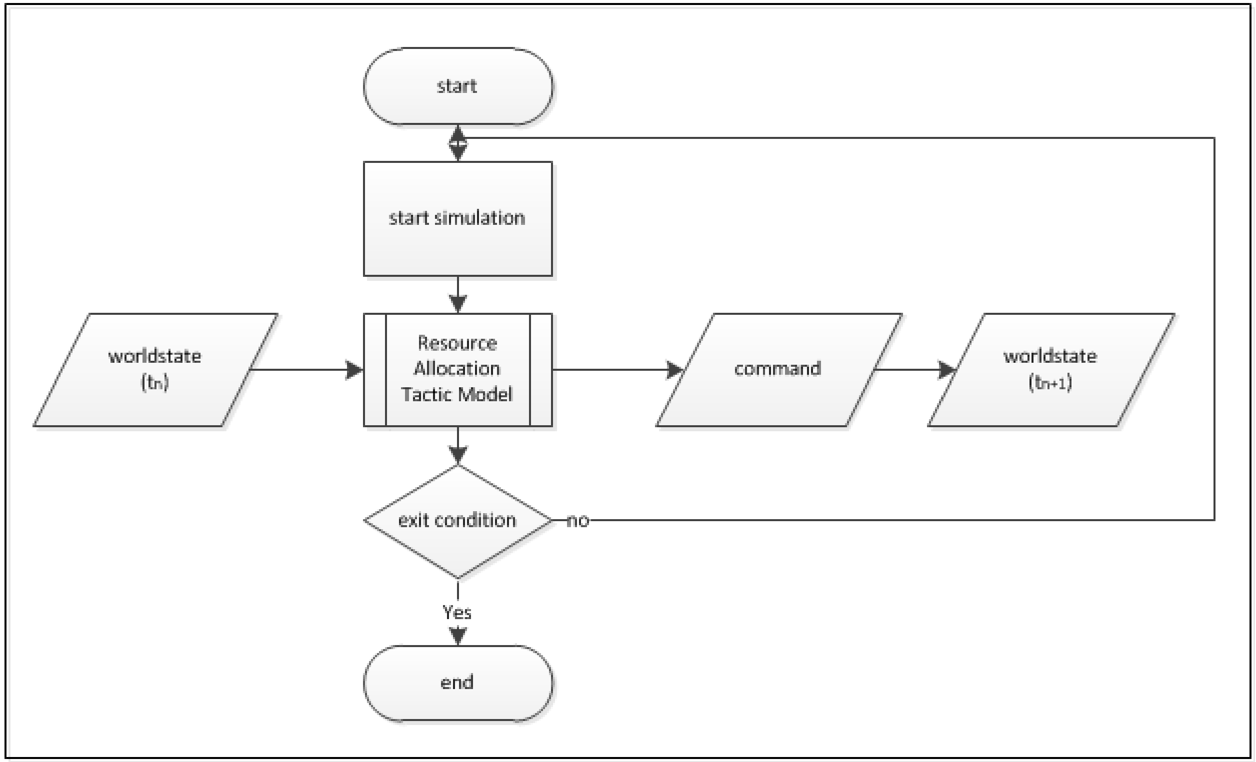
Agent Oriented Simulation Models

Resource Management Model

The **Resource Management** Models developed in the **CRISMA** project are built upon the **OOI** **concept** with different context dependent behavioral patterns for different **crisis** domains. Thus, there is no overall generic and all-purpose Resource (OOI) Management **Model**, but a set distinct models for different types of resources (e.g. ambulances, patients) and different situations. However, such domain and crisis specific Resource Management Models can be implemented on basis of the general **Agent**-Oriented **Simulation** Models **Building Block**, with its functionalities described in D312 .

Documentation:

Functional Description



The **Agent**-Oriented **Simulation** Models (AOSM), implemented with **Agent**-Oriented **Simulation** Models **Building Block**, may use several (often **pilot**-specific) sub-models. The Patient Life% and Population **Exposure Model** can serve as good examples of this kind. Different type of simulation-support can be used as external (stand-alone) models (for example, **VTT**-House), describing the effects of certain environmental processes on OOIs.

Additionally to the simulation models (and their sub-models), there exist models focussed on **resource**(s) management, but only implicitly linked to the simulations – like **Preparedness** Planner and **Economic Impact** Model, getting their input directly from **OOI**-WSR (the place the **simulation model** records the consecutive **World** States).

Additionally to the simulation models (and their sub-models), there exist models focussed on resource(s) management, but only implicitly linked to the simulations – like Economic Impact Model, getting its input directly from OOI-WSR (the place the simulation model records all consecutive World States).

Different use-cases of AOSM for **Resource Management** and simulations in CRISMA:

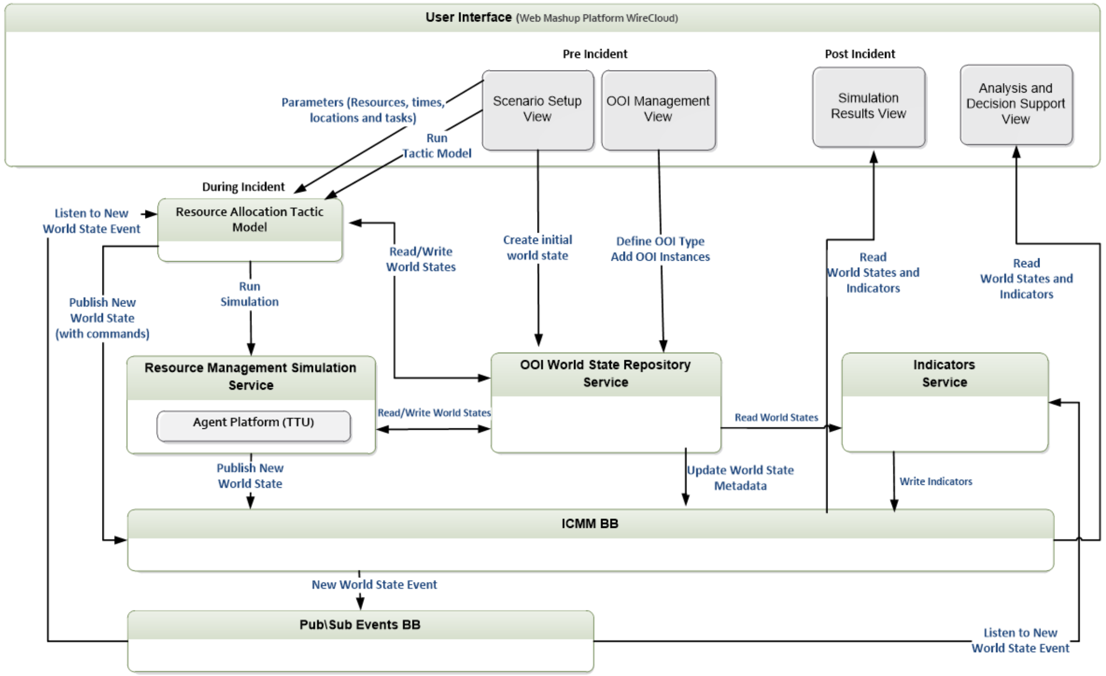
Agent-Oriented Simulation Model (ref. applications for Pilot C, E):

a) Input (WSx from OOI-WSR)

b) Output (WSx … WSn to OOI-WSR, **ICMM**)

c) Connections (Simulation Setup, OOI-Management, OOI-WSR, ICMM)

d) Sub-Models used (Vehicle, Areas, Patients, triage, pre-triage, Chemical Plume, + all **MDA**-specified models/(tables)



This Agent-Oriented Simulation Model (AOSM) uses common driving logic and generic types OOIs for both pilots . According to the pilot needs, the generic OOIs can be configured by assigning values to pilot-specific properties. This kind of implementation is possible due to many formal similarities between reference applications for the pilots C and E.

However, the behavioural logic for formally similar procedures in applications may be remarkably different. These issues are taken into consideration while designing the active OOIs (realized as Software Agents).

The simulation **scenario** for different applications (with common generic OOIs) can be realized by "specific usage" of the Command-Type OOIs. The appropriate usage is organized by scenario planning and driven over **GUI** by using pilot-specific OOIs and their properties values. Setup for both reference applications is independent and different.

The AOSM generates a consecutive series of World States from initial **World State** WS0 (or alternatively from some intermediate state WSx) up to WSn, with a fixed time-step for a simulation and pre-defined END-time. Optionally, the simulation can be terminated by an external command.

Each time the AOSM writes a new WS into OOI-WSR, **it** reports to ICMM .

Agent-Oriented Simulation Model (ref. **application** for Pilot A):

a) Input (WSx from OOI-WSR)

b) Output (WSx … WSn to OOI-WSR, ICMM)

c) Connections (UI, OOI-Management, OOI-WSR, ICMM)

d) Sub-Models used (Buildings – used and categorized in **VTT**-House), Life-condition Index (for a geo-cell)

e) Additional (external) models (VTT-House)

In contrary to AOSM for reference applications of Pilot C and E, in **reference application** for Pilot A we don’t represent individual entities – the entities the properties we control and follow. For example, the Buildings with 3 subtypes are aggregated to geo-cell type of OOIs as properties of a geo-cell. The same with Habitants- the habitants with 3 age-categories are represented with properties and property values of a geo-cell. OOI-descriptions .

Environmental effects and Power Grid ON/OFF are controlled/monitored at geo-cell level. The simulation plays with summary effects of the environment (meteorological conditions and power supply) on groups of Buildings and Habitants in these buildings located in a geo-cell. The AOSM calculates a “Life Condition Index” for each geo-cell, based on the cooling-**situation** of buildings and habitants’ category in “critical buildings”. The end-user can follow the situation at geo-cell level for a selected area of **crisis** (including a chosen number of geo-cells) – the **simulation results** are recorded in OOI-WSR and can be visualized by GUI.

No logistics, no **evacuation**, no cost-analysis by simulation model – these options are supported by additional (independent) models for a reference application.

The AOSM generates a consecutive series of World States from initial World State WS0 (or alternatively from some intermediate state WSx) up to WSn, with a fixed time-step for a simulation and pre-defined END-time. Optionally, the simulation can be terminated by an external command.

Each time the AOSM writes a new WS into OOI-WSR, it reports to ICMM

Realization:

Dynamic Map Agents

This is a description of a software candidate that offers tools for **agent**-based simulations for the implementation of a functional **Building Block** (Agent-Oriented **Simulation** Models Building Block), together with **OOI** **World State** Storage **Service** (**WFS**) **BB** (**NICE**), **Resource management** integration and deployment (**WPS**) BB (CASS), Indicators BB (**AIT**), Tactical RM Training BB (Situator integration) (**NICE**) and RM Training Dispatch and Monitor View (**AIT**), RM Training Simulation **Scenario** Setup View (AIT), RM Training Indicators and Statistic View (AIT).

Related Models and BBs:

OOI Management View
Resource Management Tactical Training BB
Simulation Model Integration BB
Earthquake casualty model
Road network vulnerability model (RNV)

Road network vulnerability model (RNV)

Model for the assessment of probability of road link interruption due to earthquakes.

The model was originally developed at **Plinivs** centre in order to assess the safety of possible escape routes in impacted areas. However, the model is here described in the logic of integration within the **CRISMA framework**.

According to the three levels of analysis for seismic **vulnerability** of road networks, described in section 3.2.2.1 of **D43.1**, here we are going to consider a "level I" model, aiming at assessing the connectivity **capacity** of a road network after a seismic **event**.

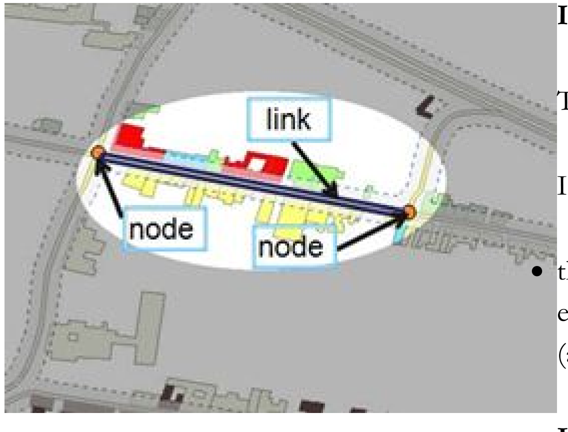
The model is based on the assumption that vulnerability of road networks is strictly connected with buildings collapses, therefore the model is usefully applied to roadways in urban areas. Indeed, **it** is reasonable to assume that the probability of interruption of a road is highly correlated to the seismic vulnerability of buildings along it.

Documentation:

Functional Description

Functional description

The Minima Unit of analysis (MU) for the case of road network **vulnerability** are identified by road segments (or branches, or links) that are formed by one or more selected road tracts and are comprised between two consecutive nodes (see figure below).



Input data

The input data are given at the Minima territorial Units (MU) level (links)

Inventory data

- the inventory for the present **model** necessitates the association, to each link, of the number of buildings belonging to each relevant vulnerability class that are built along to the link. In particular, buildings placed on both sides of the road (at a fixed maximum distance from the road axis) are considered.

Hazard data

- Similarly to the building **vulnerability model**, input for each MU (in this case the MU coincides with the link) is the seismic intensity.

Process

Given the seismic intensity, and after **application** of the seismic vulnerability model, the expected number of buildings affected by partial or total collapse Nc along each link can be computed. Then, the probability of link interruption Pi can be calculated with the following expression, that is based on the assumption that the number of blockages along each road follows a Poisson distribution

$$P_i = 1 - \frac{1}{e^{N_c}}$$

The figure below illustrates the logic flow of analysis for the road network seismic vulnerability model.

Having assigned the area of interest and having chosen the road links to be assessed (bottom map in Figure), the necessary inventory lists the number of buildings belonging to relevant vulnerability classes for each link. Given as input the earthquake intensity, the application of building seismic vulnerability model allows the assessment of Nc for each link and, by the use of above equation, the computation of road link interruption probability.

Output data

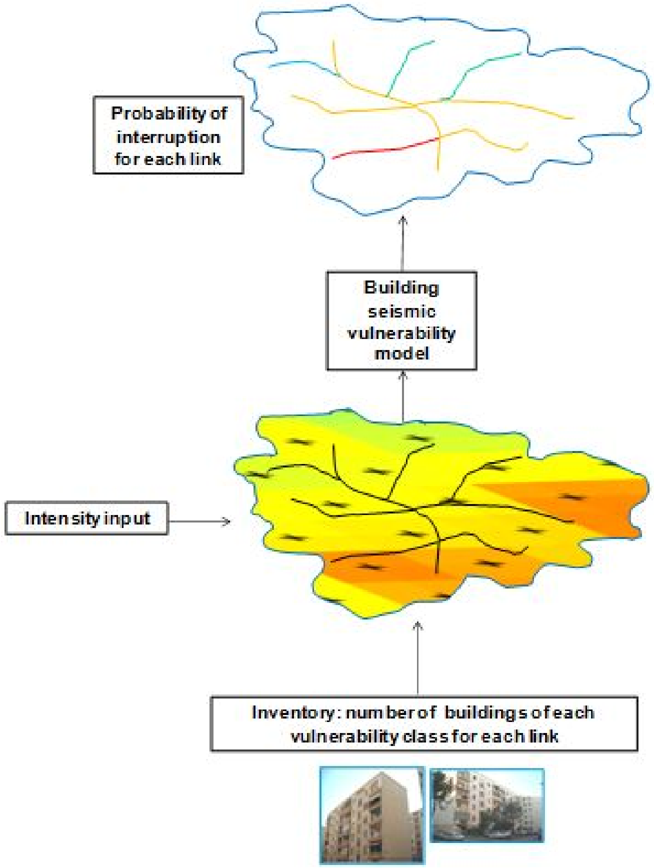
- probability of interruption Pi for each link

Further info may be found in Chapter 5 of D43.1 (downloadable at **https://workspace**.**vtt**.fi/sites/eu_crisma/Deliverables/Sent%20to%20the%20Commission/CRISMA_D431_final.pdf) and in Chapter 3 & Appendix B of D43.2

Realization:

PostgreSQL stored procedure

PostgreSQL stored procedures allows users to extend a database with user-defined functions by using various procedural languages.



Related Models and BBs:

Simulation Model Integration BB

Simulation Model Interaction View

Building impact model

Population exposure model

Resource Management Model

Cascading Effects Model

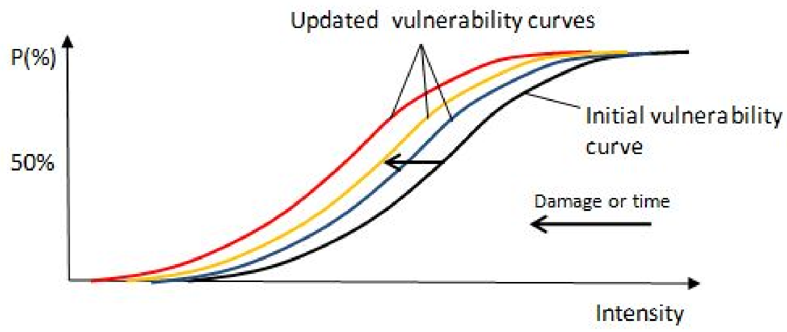
Time Dependent Vulnerability model (TDV)

Model for the assessment of time-dependent damage on **elements at risk**.

Vulnerability of elements at risk may be considered to be affected by time-dependency for the following reasons: - continuous deterioration of material characteristics or ageing (in the long term) - cumulating damage because of repeated overloading due to adverse events - inherent dependency on time of the damaging phenomenon (as for the case of the cooling of houses with time from black-out in the extreme weather case) Given the initial vulnerability of an element (or class) at risk, in order to determine **time dependent vulnerability** the effects of time (or damage caused by an initial **impact**) have to be properly considered in order to allow consistent computation of time dependent damage and/or **losses**.

The TDV model allows to perform consistent computation of time dependent damage by the use of suitably updated vulnerability functions. The updating can be done following two approaches:

• 1st option - variation of vulnerability functions This approach entails the explicit consideration of the variation of vulnerability functions describing the propensity of the exposed assets to suffer damage due to an hazardous **event**; the change of vulnerability functions may be directly determined as a function of the damage level that the generic element has suffered during the previous event (in case of vulnerability variation due to damage by previous impact) or as a function of time (in case of ageing or, as for the extreme weather example, when the phenomenon is inherently dependent on time)



• 2nd option, update of inventory This approach, without changing the vulnerability functions, entails the re-classification of the exposed assets (in pre-fixed vulnerability classes) considering the worsening of their behaviour due to damage.

The TDV model is "domain-independent" in the sense that the logic scheme is the same for different **hazard** domains (e.g. earthquake, flood, extreme weather …), but for using the model in order to compute time dependent losses (in terms of damages in the established damage scale) there is the need to suitably feed the model for each hazard-domain.

Documentation:

Functional Description

Functional description

The description given below refers to the implementation of Option 2 for implementation of TDV **model** (update classification of **elements at risk**). Specifications for Option 1 will have to be updated

Input data

Elements **vulnerability** data

• Rules for inventory updating due to damage for previous **event** (for **application** of option 2)

Other input data are given at the Minima territorial Units (MU) level (e.g. grid of 500x500 m)

Inventory data

• Initial inventory: Initial distribution of vulnerability classes (number of elements at risk belonging to each class)

Damage data

• Damage distribution (number of elements at risk – or Objects Of Interest **OOI** - belonging to each class performing the different damage states within each MU (for application of option 2, or option 1 in case of damage dependent vulnerability)). The damage distribution is result of **Vulnerability Model** (stationary) e.g. the Building **impact model** for the case of earthquakes.

Process

The next figure evidences the sequential steps that are followed for **time dependent vulnerability** analysis (with option 2, i.e. update of inventory for elements at risk).

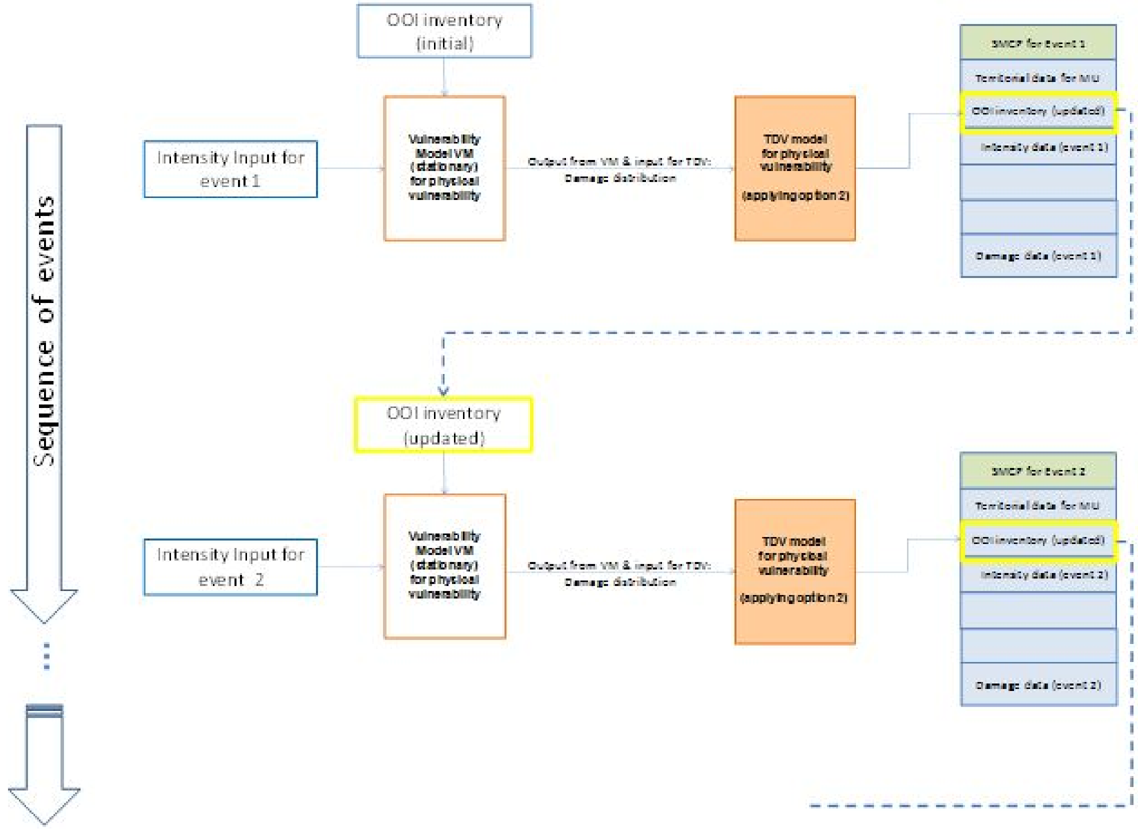
The TDV model logically works interacting with **standard** Vulnerability Model VM and for sequence of events (for option 2). For applying Option 2 the model needs as input the (initial) inventory and the damage distribution for the elements at risk (or OOI). Damage input derives from the application of a standard Vulnerability Model VM (e.g. the Building Impact Model) after first event (event 1). The (initial) inventory for the elements at risk is retrieved from the WS, that shall contain **it** due to the (former needed) application of standard VM. The application of TDV model with option 2, next, allows to update the OOI inventory in WS. The updated OOI inventory becomes the inventory input, together with intensity input for the next event, for the standard VM to be used within the sequential analysis.

Output data

• Updated inventory: Updated distribution of vulnerability classes (number of elements at risk belonging to each class) after each event

Further info may be found in Chapter 2 & Appendix A of D43.2 (downloadable at **http**://**www.crismaproject.eu/deliverables/CRISMA_D432_public_print.pdf (http://www.crismaproject.eu/deliverables/CRISMA_D432_public_print.pdf)**)

Realization:



TDV Python package

Collection of modules written in Python **object**-oriented programming language.

Related Models and BBs:

Cascade Events Configuration and Interaction View
Simulation Model Integration BB
Simulation Model Interaction View
Building impact model
Cascading Effects Model

VTT House model

Authors: **Markus Jähi**
Jussi Yliaho

Model for estimating the extreme cold weather related **vulnerability** curves for buildings.

VTT House is a **simulation model** for calculating extreme cold weather based cooling curves of different types of buildings. The model is based on EN ISO 13790 and EN 15241 standards in addition to models for estimating solar radiation. The model includes methods for a dynamic hourly-based calculation of building energy and thermal performance, including heating and cooling and airflow related energy **losses** due to the ventilation **system** and infiltration. In addition the model can be used to predict the speed of temperature **recovery** when heating is restored.

For **CRISMA** integration, **VTT** House exposes a **SOAP** based web **service** **API**.

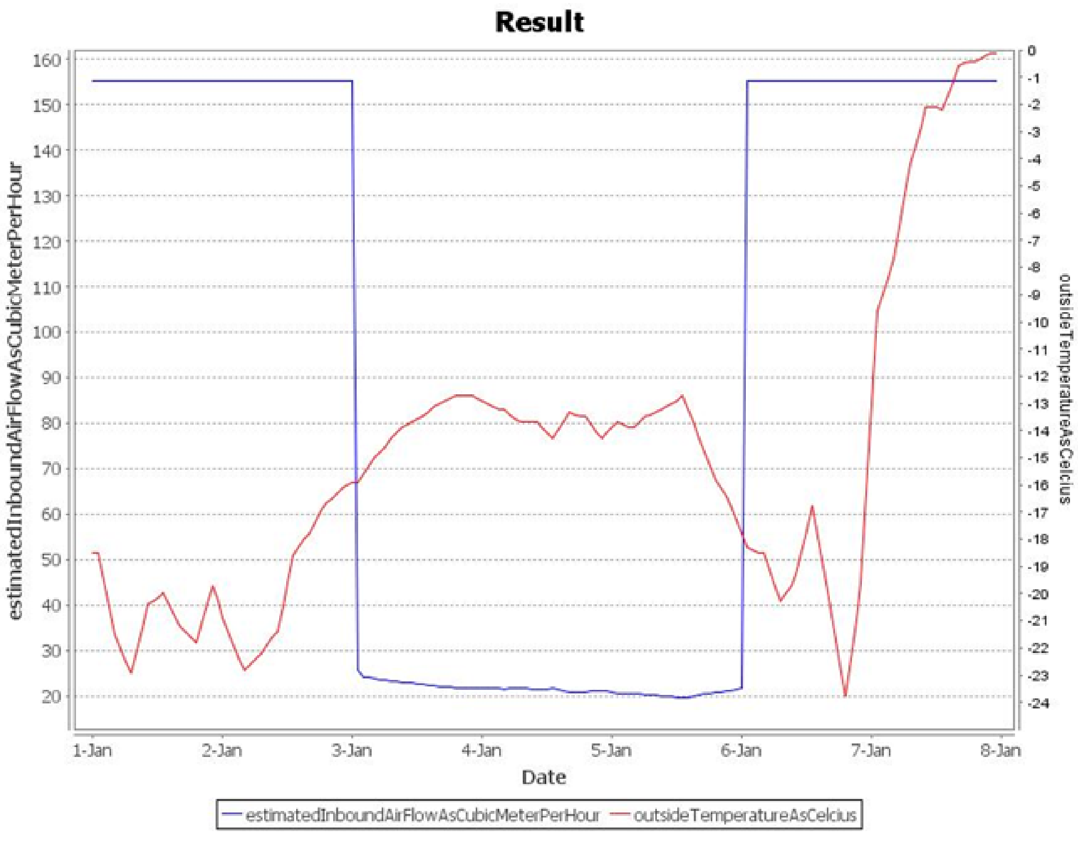


Figure 1. Example results: Inbound air flow, Outside temperature

Figure 2. Example results: Heating power, Inside temperature

Documentation:

Functional Description

Input data

Two types of input data are required to run the **model**:

1) **Authentication**

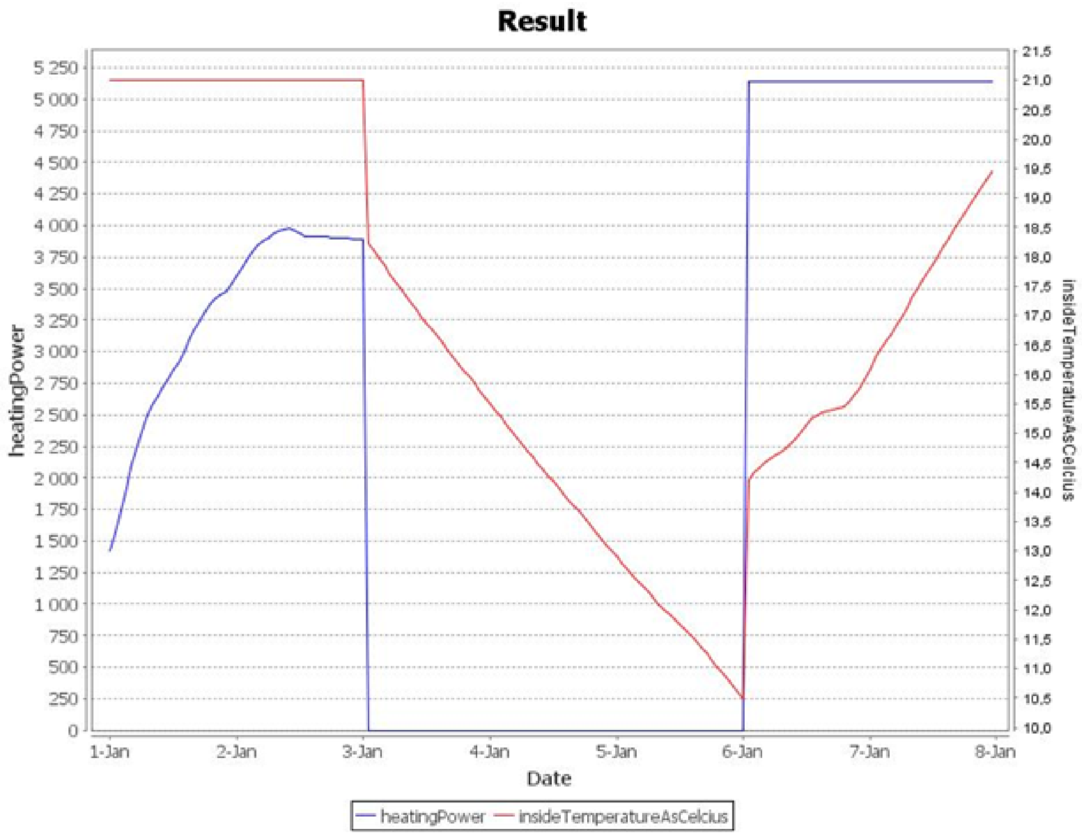
- username
- password

2) Model input data

- building data (typically available in existing building regulations)
- weather or weather forecast data
- schedule of heatingPower (on/off)

Process

The calculation of the extreme cold weather related **vulnerability** curves for buildings consists of four steps:



1. Select building from building type library (light one family house, medium one family house, medium 3 floor apartment building, heavy 2 floor apartment building etc.) and optionally change some building data if needed for some specific building.

2. Select nearest FMI's weather station ("locality"), start time and end time. If the end time is bigger than present time the FMI's weather forecast is used (max 48 hour weather forecast available). **It** is also possible to choose existing weather file or constant outdoor temperature.

3. Select the start times and end times (schedule) for the power failure (heating power on/off).

4. Calculate the extreme cold weather related vulnerability curves for buildings using the selected building data, weather data and heating on/of schedule.

Output data

The output of the model is time series of the result data including e.g. date and time, used outside temperature , used diffuse solar horizontal radiation , used direct solar horizontal radiation and calculated inside temperature .

Realization:

VTT House service

Web **service** to implement **VTT** House building **simulation model**.

Related Models and BBs:

Agent Oriented Simulation Models

5. Main Components

Components are the actual pieces of software used in **CRISMA**.

In most cases the components are materialization (reference implementations) of the CRISMA Building Blocks and models. However, the Components can also be libraries, platforms and any other software which is necessary to realize the **CRISMA framework** and applications.

BasicIndicators

Authors: **Peter Kutschera**
Version: 1.0
BasicIndicators is a reference implementation of the **Indicator** **BB**. **It**'s **role** is to calculate the some of the indicators needed in **CRISMA** applications and store the results in the **World State**.

According to a CRISMA **architecture**, the indicators should be calculated whenever a new WorldState is available and the results should be stored in the WorldState repository. In the case of BasicIndicators, the result is stored in the **ICMM** and optinally also as an **OOI** in the in the OOI WorldStateRepository **service**.

In order to integrate in the **CRISMA framework**, BasicIndicators offers a **WPS** **interface** to integrate the component into CRISMA listening to WPS requests.This interface is realised using **PyWPS** and therefore the BasicIndicators is licensed under GPL. The actual indicator calculation logic can be separated from the GPL code and licensed under different conditions if needed.

The BasicIndicators also offer a listener subscribed to a PubSub service and automatically executing the indicator calculations whenever a new WorldState is announced.

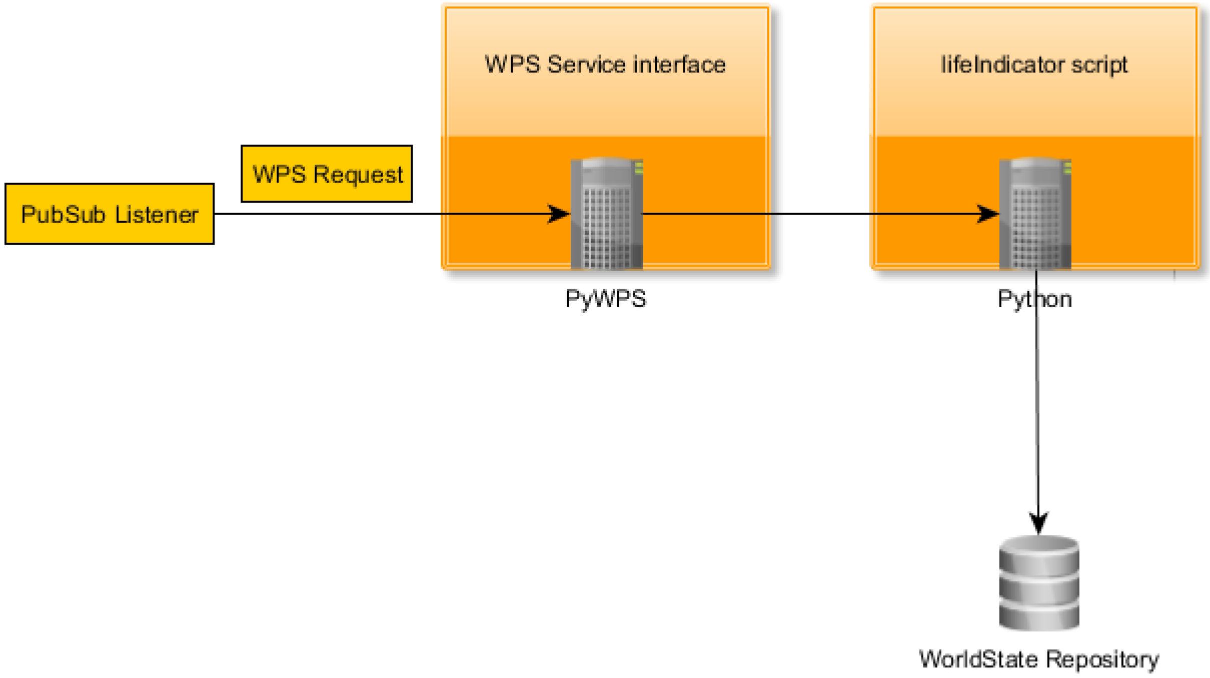
Documentation:

Tutorial

In **CRISMA** a wide range of indicators is defined. The idea of an **indicator** is to have one or few simple values to evaluate and compare situations and judge about actions taken.

The definitions of indicators is compleatly different in terms of used data (number and status of patients, **GIS** data) and complexity of calculation. Depending on that different indicator implementations are forseen in CRISMA. This BasicIndicators can be seen as the "low end" implementation (simple data, simple calculation) and acts also (because of the simplicity) as an example how to implement indicators in CRISMA.

The BasicIndicators calculation is done in the following steps:



1. A request to calculate comes in, specifying which worldstate has to be used as input.
   Here a **standard** **WPS** **interface** is used.
2. The WPS server (**PyWPS** in this case) runs one of the available indicator scripts
3. The indicator script fetches the required input data from the WorldState repository.
   This is done using CRISMA **REST** interface.
   A more general example using **WFS** is also available.
4. The actual indicator value is calculated.
   This value is a **JSON** data structure contaiong the actual value, a description, id and so on.
5. The indicator value is stored
   The indicator value is store in the **ICMM**. In general the indictor value can be stored anywhere, the ICMM need only to hold a reference in that case.
6. The WPS returns as result the location of the just calculated indicator as **response**.

The URLs below provide access to a web site providing links to the WPS description documants and allowing to trigger the indicator **service** using a web form. This is meant for **test** / developement purposens only

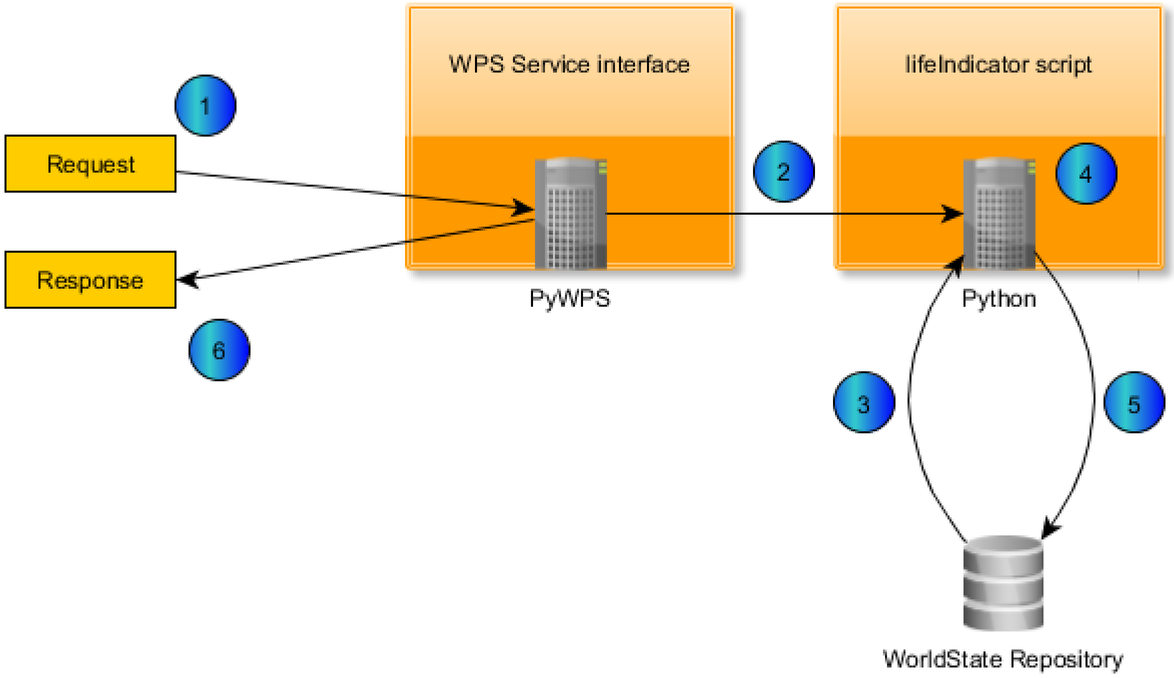
Normally the service is triggered by the Pub/Sub **event** listener for each new WorldState.

- Service used in Pilot C
  http://crisma.ait.ac.at/indicators/indicators.html (http://crisma.ait.ac.at/indicators/indicators.html)

Downloads:

The source code of the BasicIndicators services providing **Pilot** C and Pilot Ev1 indicators and KPIs is available on github

- Pilot C Indicators
  https://github.com/crismaproject/crisma-indicators-c
- Pilot E version 1 indicators
  https://github.com/crismaproject/crisma-indicators-e1



The ready-for-use binaries of BasicIndicators services are available as **Docker** images on docker hub.

In order to use them, you need to install docker first ("sudo apt-get install docker" on ubuntu or debian).

Example usage:
1. Start an orion context broker:
docker run -P -d –name e_orion peterkutschera/**crisma**-orion
2. Start the **indicator** **WPS**:
docker run -P -d –name e1_indicators –link e_orion:orion peterkutschera/crisma-indicator_e1

- Pilot C docker image **https://registry.hub.docker.com/u/peterkutschera/crisma-indicators-c/**
- Pilot Ev1 docker image **https://registry.hub.docker.com/u/peterkutschera/crisma-indicators-e1/**

Team:

Peter Kutschera
Can provide help using the existing **service** and implementing new indicators

CRISECON GUI

Authors: **Jussi Yliaho**
Markus Jähi

CRISECON **GUI** (see figures below) implements, together with the related web-**service** component CRISECON Service, the CRISECON **model** developed in SP4.

The purpose of the component is twofold. Firstly, its purpose is to collect cost related data (economic parameters) which is not readily available from the Worldstate description and is needed in **economic impact** calculation from a user. Secondly, its purpose is to show results related to economical **evaluation**.

The main functionalities to be provided by CRISECON GUI are:

- The user should be able to define the desired type of economic impact analysis.
- The user should be able to see Worldstate values related to economical calculations (e.g. number of destroyed buildings, used working hours for **emergency** personnel).
- The user should be able to fill in on a form cost data related to **response**, damages and **mitigation** measures.
- From the GUI user input values need to be transferred to the CRISECON Service.
- To present the economic impacts arising from crises (ex post performance).
- To assess different mitigation proposals and their costs/benefits (ex ante planning). The CRISECON model is implemented by two software components.

CRISECON GUI is a user interaction component and provides a **graphical user interface** (GUI) for end users. The other component is a federated calculation and **simulation model** (CRISECON Service) providing a service to calculate the costs of the **crisis** under investigation. The economic impact calculation view receives some of the parameters from an external storage and some of the parameters are entered by the user. The parameters are sent to the economic impact calculation service in a calculation request which will launch the calculation and/or simulation process implemented in the **Web Processing Service** **WPS** instance. When the requested economic impact calculation is dispatched, the calculation and simulation service provides the results back to the user interface. In accordance with the **CRISMA** architectural requirements for user interaction components, the economic impact calculation view is provided as a HTML5/JavaScript **application** where the user interface is based on HTML and the application logic is based on JavaScript. The jQuery JavaScript library and Kendo UI **framework** are used to build the application.

CRISECON software is installed on VTT's Windows Server 2008 R2 with IIS7 web server.

CRISECON GUI figures:

Documentation:

Functional Description

Description adapted from **CRISMA** deliverable D44.2

Downloads:

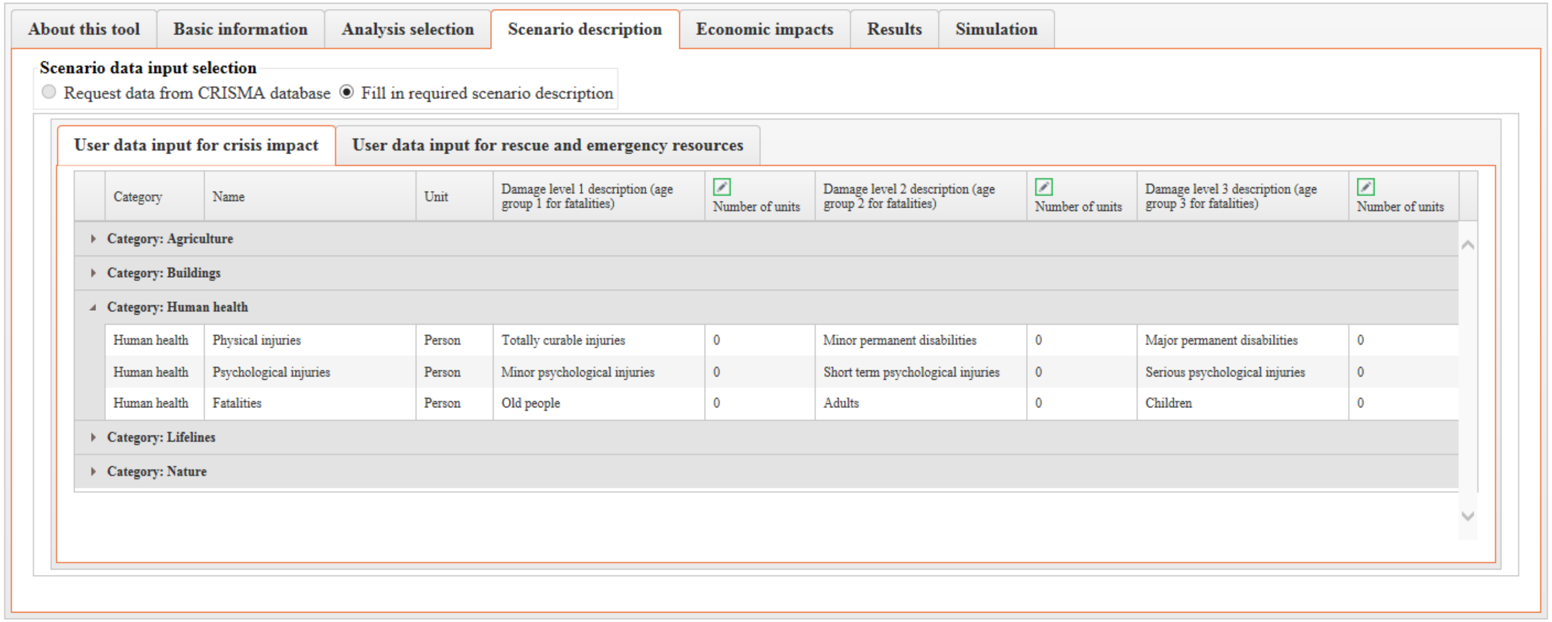
- https://crisma-cat.ait.ac.at/system/files/uid_67/CRISECON_model_and_software_D442.doc
- Mock up
  https://moqups.com/MarkusJ/6lOSjK4s/

Downloads:

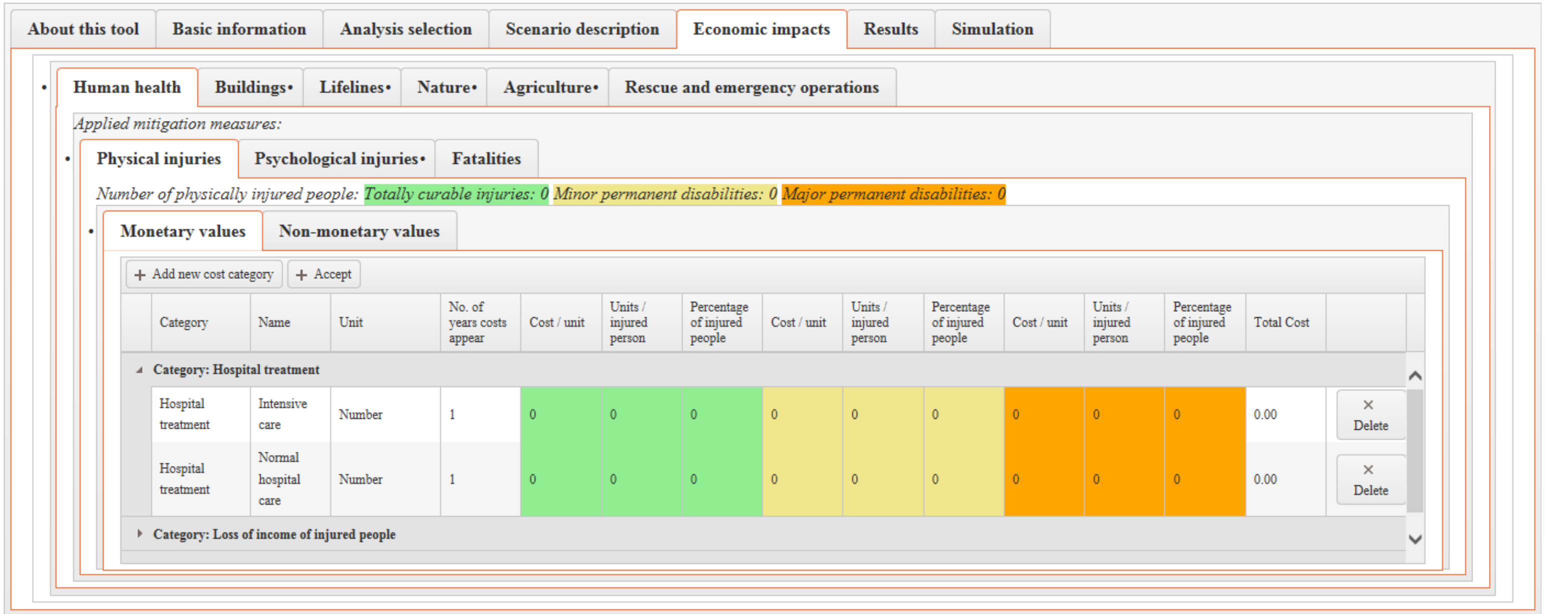
Team:

Jussi Yliaho
Markus Jähi

CRISECON Service



Authors: Jussi Yliaho
Markus Jähi



CRISECON **Service** implements, together with the related UI component CRISECON **GUI**, the CRISECON **model** developed in SP4.

The key functionalities CRISECON Service are:

- CRISECON Service should be able to take from CRISECON GUI numerical and text values related to **world state** and costs.
- CRISECON Service should be able to calculate pre-defined cost indicators for **response** phase and **crisis** damage according world state values.
- CRISECON Service should be able to transfer pre-defined values to CRISECON GUI and to save them to an external storage.

Software components that are developed as well as the interfaces within the **CRISMA** **architecture** are presented in the figure below. The CRISECON model is implemented by two software components. CRISECON GUI is a user interaction component and provides a **graphical user interface** (GUI) for end users. The other component in the lower part of the diagram is a federated calculation and **simulation model**, CRISECON Service, providing a service to calculate the costs of the crisis under investigation. As the diagram in the figure above indicates, the **economic impact** calculation view receives some of the parameters from an external storage and some of the parameters are entered by the user. The parameters are sent to CRISECON service in a calculation request which will launch the calculation and/or simulation process implemented in the **Web Processing Service** **WPS** instance. When the requested economic impact calculation is dispatched, the calculation and simulation service provides the results back to the user interface.

In accordance with the CRISMA architectural requirements for federated simulation models CRISECON Service is provided as an **OGC** (**Open Geospatial Consortium**) WPS (Web Processing Service). In spite of the fact that the OGC WPS **standard** defines how geospatial services can be executed **it** can be used to describe any process and execute it by using predefined input and output parameters. **PyWPS**, which is a Python implementation of OGC's WPS 1.0.0 standard, was chosen because it has received positive reviews and seemed appropriate for the purpose. For its simplicity and rapid deployment, **CGI** (Common Gateway Interface) was chosen. The objective is that the economic impact calculation service could be implemented as a single Python process.

CRISECON software is installed on VTT's Windows Server 2008 R2 with IIS7 web serv**er.**

CRISECON components:

Documentation:

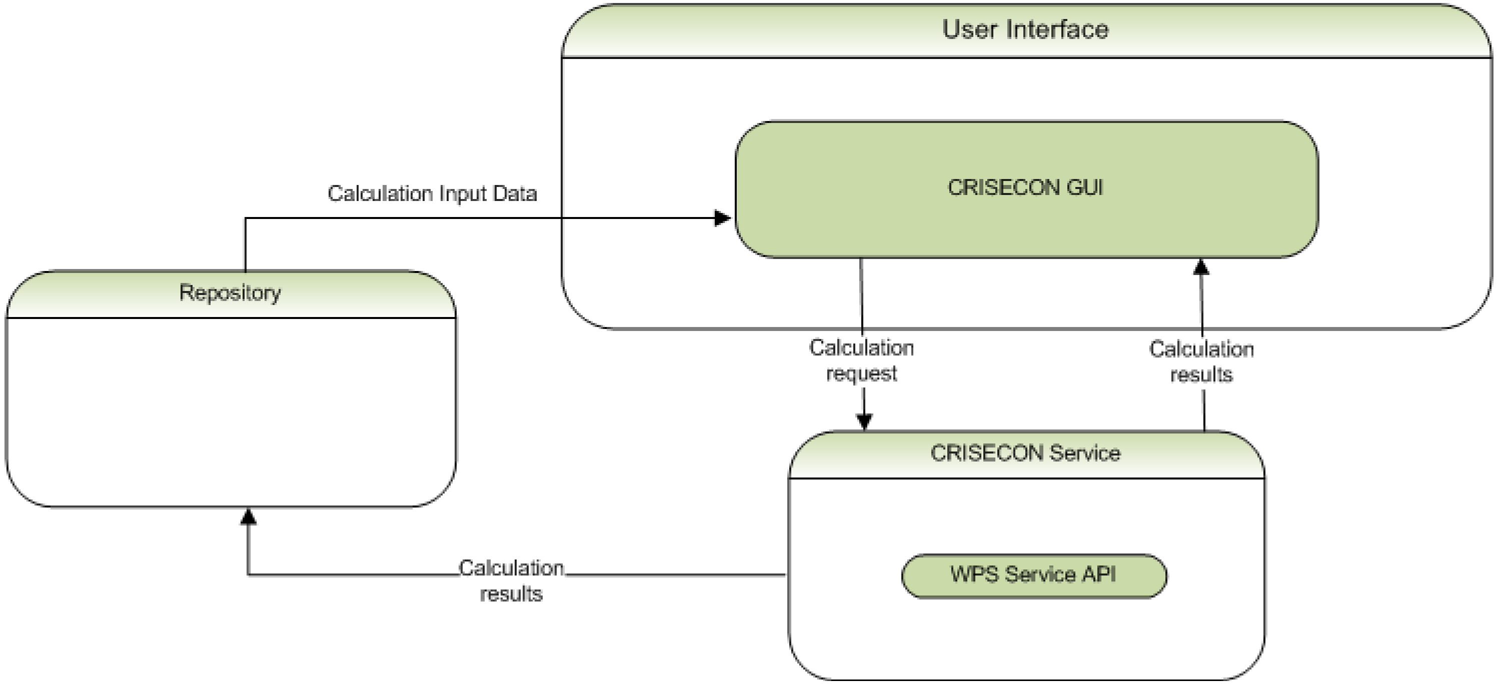
Functional Description

Downloads:

- https://crisma-cat.ait.ac.at/system/files/uid_67/CRISECON_model_and_software_D442_0.doc
- Mock up
  https://moqups.com/MarkusJ/6lOSjK4s/

Downloads:

Team:



Jussi Yliaho
Markus Jähi

Configuration Component

Version: 0.1.1.2

The configuration component is the core component of the Integrated Planning View. **It** has been exemplary implemented visualizing the Charts in the Debriefing View of the **Reference Application** for **Exercise**-Support in **Pilot** E.

Documentation:

Specifications

The configuration component is the core component of an interaction **concept** for navigating through hierarchical contents. The user iis capable of going a level deeper on the overview screen and he or she is capable of returing to the overview screen from every deeper level. Other components may synchronize with the selected level of granularity within the hierarchical contents (e.g. map, charts etc. visualizing **simulation results** or **exercise** data).

- D25.1 - Interaction Concept for Planning Use Cases **https://workspace.vtt.fi/sites/eu_crisma/Deliverables/Sent%20to%20the%20Commission/CRISMA_D251_final.pdf**

Functional Description

The configuration component is the core component of an interaction **concept** for navigating through hierarchical contents. The user iis capable of going a level deeper on the overview screen and he or she is capable of returing to the overview screen from every deeper level. Other components may synchronize with the selected level of granularity within the hierarchical contents (e.g. map, charts etc. visualizing **simulation results** or **exercise** data).

- D25.1 -

CRISMA Pilot E Debriefing Widget

Inspect

Response - Exercise Run 2

Key Performance Indicators

0.6	Ratio of medical responders per patient

Response Phases

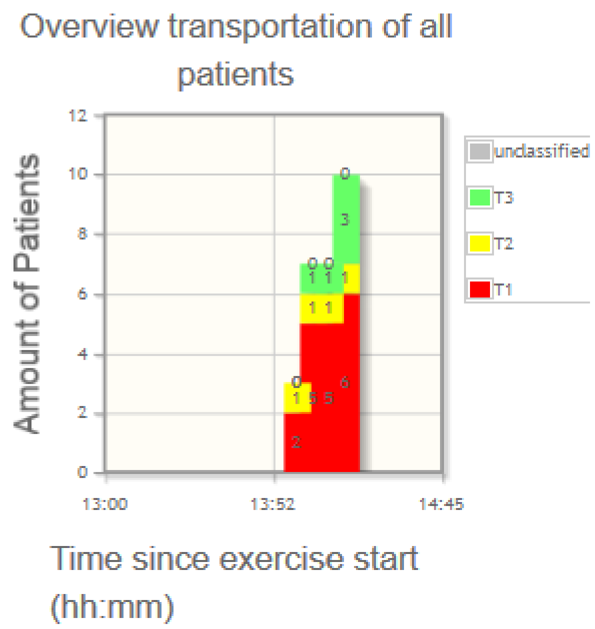
- Pre-Triage
- Triage
- Transportation of Patients

CRISMA Pilot E Debriefing Widget

Inspect

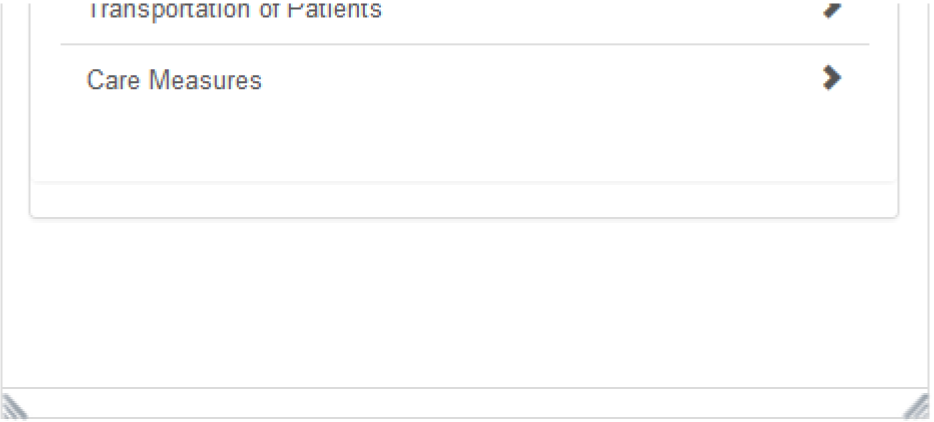
Response - Exercise Run 2

Transportation of Patients



General

17min	Time until last patient is transported to the hospital
17min	Time until red patients are away from the incident scene



Interaction Concept for Planning Use Cases
https://workspace.vtt.fi/sites/eu_crisma/Deliverables/Sent%20to%20the%20Commission/CRISMA_D251_final.pdf

Downloads:

- Source Code (Transportation Widget) **https://github.com/crismaproject/pilot-e-application/tree/feature/10-transportation-widget**

Team:

Johannes Sautter

Development roles: specifications, testing, validation

Specified in D25.1 and during the **pilot** e **exercise**-support specification.

Frank Jonat

Development roles: development
Developed by Christian Rinner, now represented by Frank Jonat from **Airbus** Defence and Space.

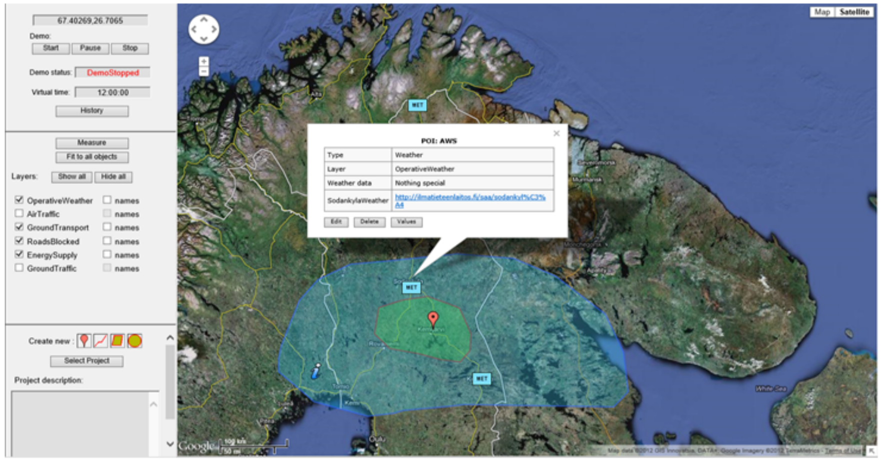
Dynamic Map Agents

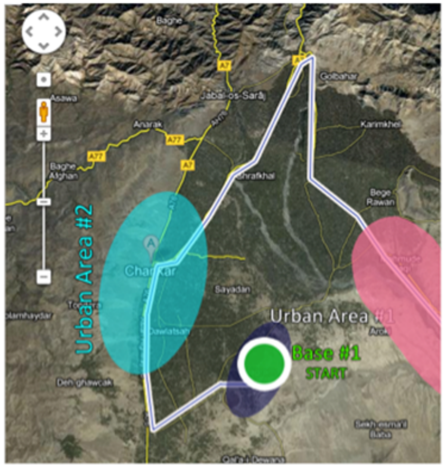
This is a description of a software candidate that offers tools for **agent**-based simulations for the implementation of a functional **Building Block** (Agent-Oriented **Simulation** Models Building Block), together with **OOI** **World State** Storage **Service** (**WFS**) **BB** (**NICE**), **Resource management** integration and deployment (**WPS**) BB (CASS), Indicators BB (**AIT**), Tactical RM Training BB (Situator integration) (**NICE**) and RM Training Dispatch and Monitor View (**AIT**), RM Training Simulation **Scenario** Setup View (AIT), RM Training Indicators and Statistic View (AIT).

The Agent-Oriented Simulation Models is implemented by **CRISMA** partner **TTU** on basis of their own Dynamic Map Agents Software.

Dynamic Map Agents (as an implementation of Agent-Oriented Simulation Models BB) is a self-organizing **system** of networking information and map agents. The multi-layer map is constructed/updated dynamically, on the basis of reciprocal interactions of its actors – the map emerges from activities of its agents (incl. users). The map could be coupled with specific information layers adjusted to particular users and collections of objects-of-interest (e.g. situational information items or (pro)active information providers) linked with **it**. Naturally, a dynamic map is also interactive - tuned to appropriate interactions patterns for its users. A dynamic map consists of a **model** of the environment (e.g. base map), its (pro)active agents positioned on the dynamic map (representing entities embodied in this environment) and its database of map objects and activities.

Agents constructing the dynamic map are spatially and temporally distributed natural and artificial actors (computing devices, vehicles, software agents and humans). For the map constructing peers only the observable behaviour of agents is known (in terms of their input-output streams). Software agents serve a dynamic map construction with (a) collecting/providing/validating situational data; (b) for smooth interactions with particular categories of physical agents embodied in the environment the map is modelling and, (c) for simulating the observable behaviour of physical agents or humans.





Dynamic Map Agents offer tools for construction of interactive map based simulations/visualization of particular aspects/threads of an evolving **crisis**. Offering a general overview about processes and actors in the crisis, it can serve as a convenient tool supporting resources management, especially for situations where the user can get additional and valuable information from agent's interactions and interactions with the simulation environment, analyzing situations that don't have a clear analytic solution.

The users can centrally follow the **situation** development in the crisis domain in near-real-time, according to their hierarchy. The users can work simultaneously on physically different devices and at different locations. Each user can interact with the server-**application** by adding/updating operative information if he/she is allowed to do so. The map agents (together with the dynamic map's middleware) take care about user activities, sharing/finding computational resources for different models used for crisis development, organizing data exchange between the computational resources, models and human actors, communicating with sensor networks for **situational awareness**.

Documentation:

Implementation Plan

Downloads:

- https://crisma-cat.ait.ac.at/system/files/uid_23/BB%20Implementation%20Plan%20Dynamic%20Map%20Agents_v3.doc

Demo

This video demonstrates some of the platform functionality. The video is pertinent to ressource management training **scenario**.

Downloads:

Team:

Kalev Rannat

Development roles: development
Development of user-specific **agent** models, integration

Emikat

Authors: **Hermann Huber**
Denis Havlik

Emikat is a **data management** and modelling **system** for spatially resolved data. The system allows the integration of emission-relevant activity data (e.g. from surveys, traffic models or statistics) as well as the definition and analysis of **model** scenarios. Scenarios allow a comparative examination of model results corresponding to different versions of data inventories – for example the overall effects of trends in emission generating activities and pollution control measures or the influence of different calculation models on estimated results.

Although initially developed to deal with emission data, air pollutant emission inventories and emission/immission modelling in regions and municipalities, the system is fully generic and capable of handling many types of data and models with spatial or temporal properties.

The **concept** of scenarios and "what-if" simulations meet the requirements defined for the Indicators **Building Block**, but Emikat can also be used to develop various models. This was demonstrated in CRISMA, by using the Emikat as a software base for the **population exposure model**.

Simulations are based on calculation models that use input data from different scenarios in order to facilitate **impact** comparison. The integrated metadata editor supports the definition of new data objects as well as respective domains and models. The light weight client can be used to specify **indicator** models. Technically, indicator models in Emikat consist of a set of cascaded SQL-Queries.

Documentation:

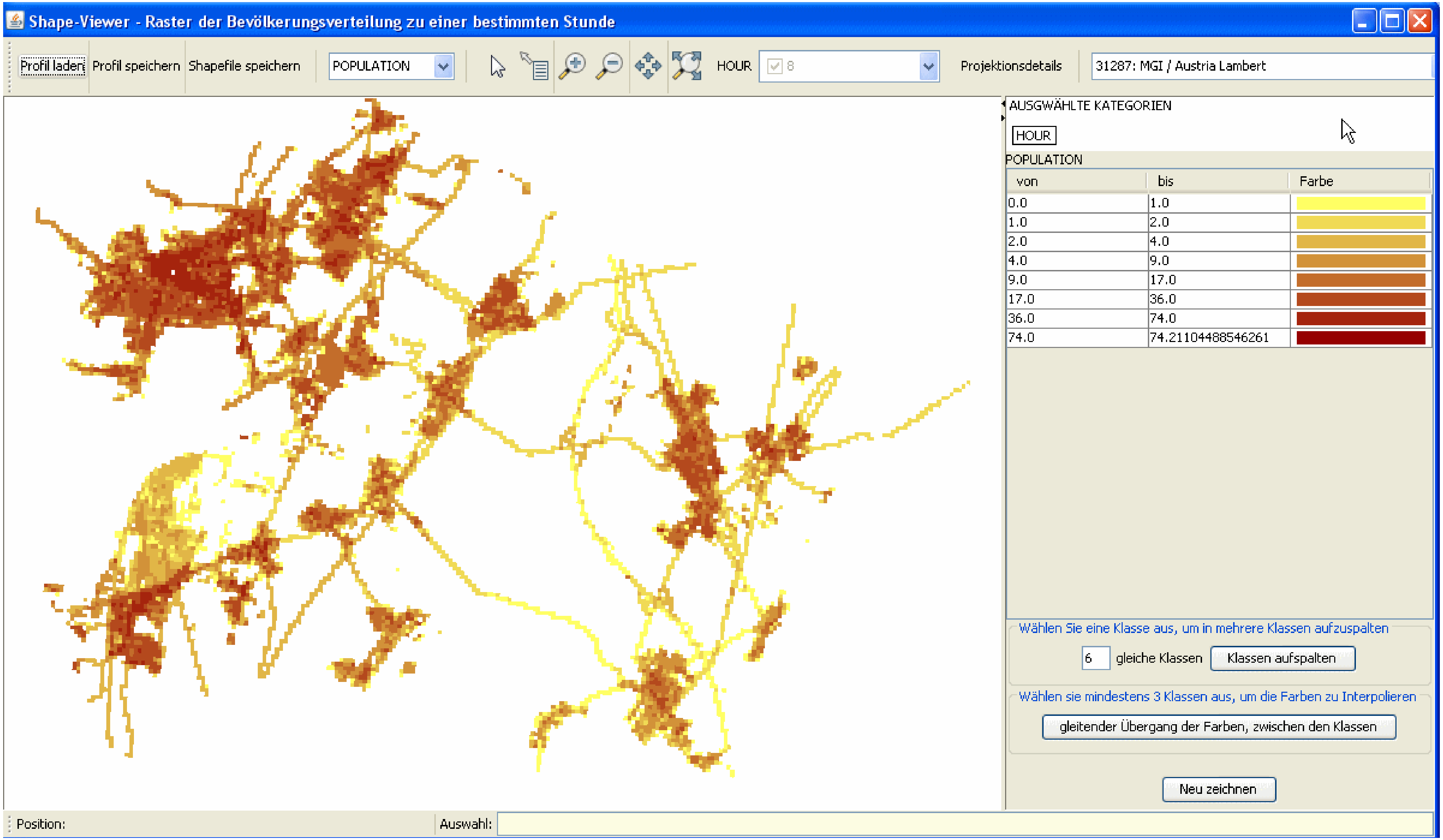
Publications

Main EMIKAT-related publications and other documentation can be found on the emikat.at web site.

- Main EMIKAT publications
  http://www.emikat.at/ (http://www.emikat.at/?page_id=384&lang=en)

FAQ

The **GUI** client which can be downloaded from the "download" section of this article can be used to develop models as well as to analyse their results. However, Emikat also offers two types of web-**service** APIs:



1) Emikat-specific **API** which is used by the client and provides access to full functionality of the backend.

2) **Standard** **OGC** services (**WMS**, **WFS**) for accessing the results of the **model** runs.

Standard access services simplify the use of Emikat in **SOA** applications such as the ones which were developed in **CRISMA** since the integrator does not need to know anything about the way Emikat is concieved and works in order to use the model results.

Downloads:

Emikat.at **system** is operated in **SaaS** **model** by the **AIT** for its customers. In this model, **AIT** assumes the secured operation of the entire emikat.at system, including the updates and the daily storage.

The access of authorized users to emikat.at takes place via Internet by means of a dedicated web-start Java **application**. In that way work teams from different operating departments can work on the emikat.at system simultaneously. The client installation is very easy and all users always have the latest software version (bug fixes, security updates, new features). Emikat client can be downloaded from the emikat.at web site **free of charge** (please do not use the **test** client - test means "development version" in this context!).

CRISMA demo account credentials

CRISMA **demonstration** account can be used to examine the population distribution model which was developed using EMIKAT in CRISMA.The account allows read-only access to this model, so feel free to give **it** a try.

Further information on this model, which is used in two of the CRISMA pilots, is **available here**.

> **mandate:** crisma
> **user:** test
> **password:** test

- Emikat client download
  http://www.emikat.at/ (http://www.emikat.at/?page_id=388&lang=en)

Team:

Hermann Huber

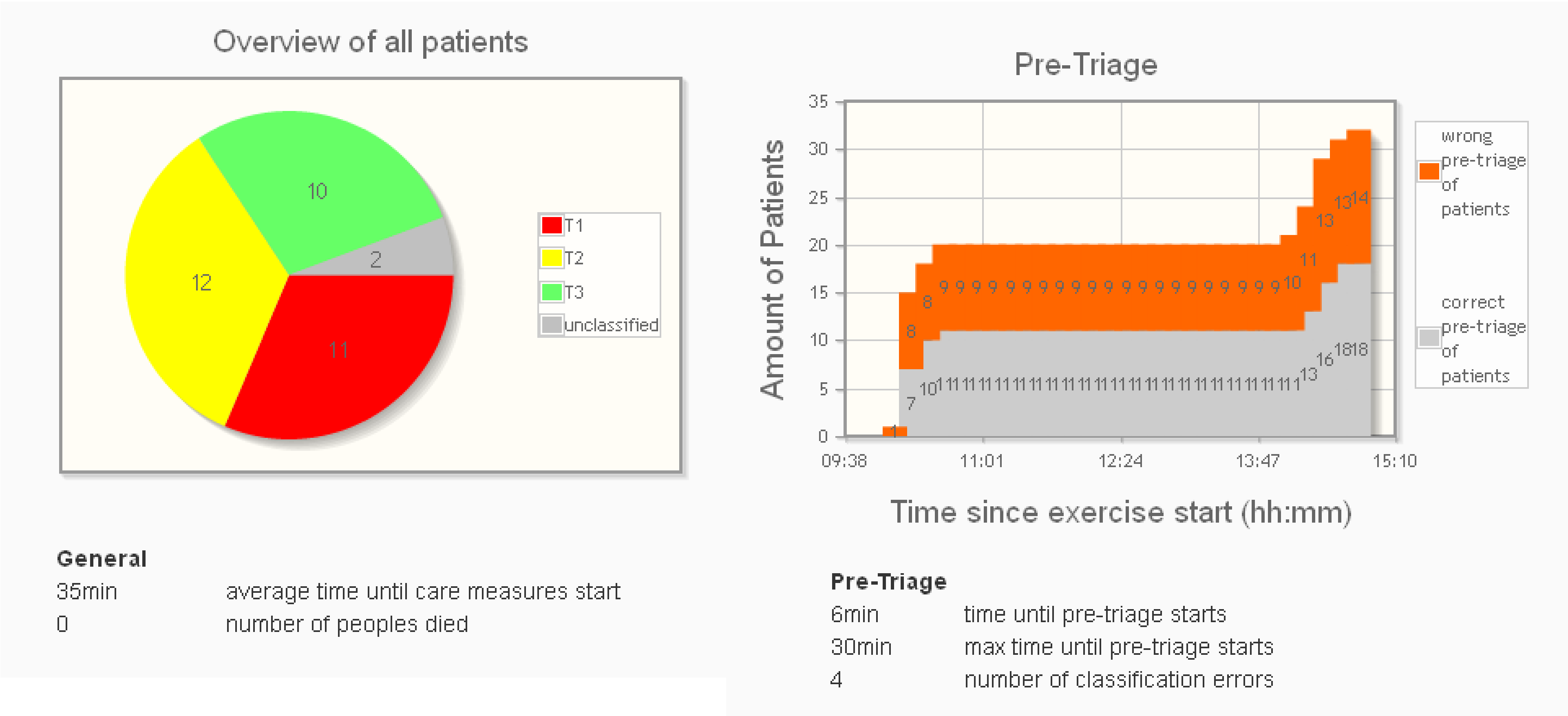
Development roles: configuration, development, integration
Hermann ist part of the EMIKAT core **developer** team. He is mainly responsible for data visualization and RESTful **API**.

Heinrich Humer

Development roles: administration, architecture, coordinaton, development, integration
Heinrich is the main point of contact regarding EMIKAT. He is responsible for data integration and modeling.

Exercise Worldstate Data Chart Widgets (Wirecloud)

Several widgets that display worldstate data from **ICMM** in graphical form (as pie charts and stacked bar charts) and in table form. Developed with AngularJS.



Documentation:

Functional Description

Care Measures Widget

The care measures widget contains a pie chart and a key-value list. **It** is used in **pilot** E **application** to provide an overwiew of the triage states of all patients. The generic key-value list is used to display a set of key performance indicators (KPIs).

Pre Triage Widget

The pre triage widget contains a stacked bar chart and a key-value list. It is used in pilot E application to show the amount of patients that have been pretriaged correctly or incorrectly at specific times during an **exercise**. The timesteps are configurable (i.e. 10min). The generic key-value list is used to display a set of key performance indicators (KPIs).

Triage Widget

The triage widget contains a stacked bar chart and a key-value list. It is used in pilot E application to show the amount of patients that have been triaged correctly or incorrectly at specific times during an exercise. The timesteps are configurable (i.e. 10min). The generic key-value list is used to display a set of key performance indicators (KPIs).

Transportation Widget

The transportation widget contains a stacked bar chart and a key-value list. It is used in pilot E application to show the amount and the correct triage classification of patients that have been evacuated (transported to an hospital) at specific times during an exercise. The timesteps are configurable (i.e. 10min). The generic key-value list is used to display a set of key performance indicators (KPIs).

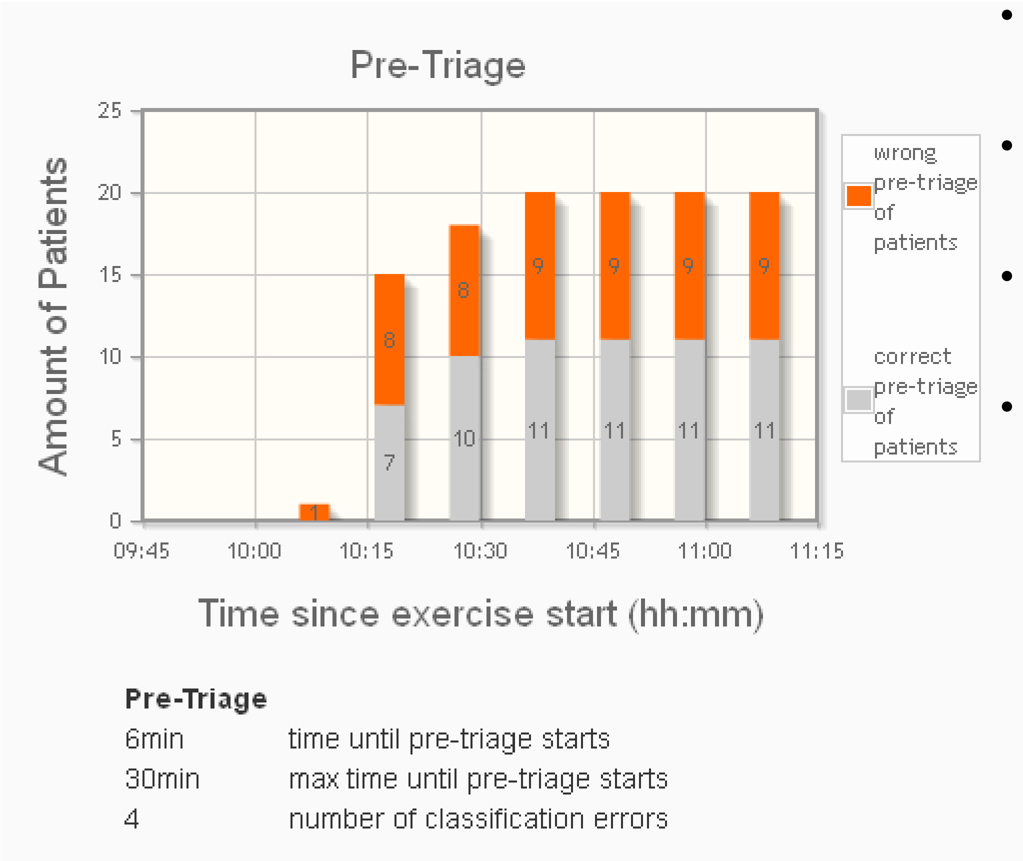
Alerts and Requests Widget

The alerts-and-requests widget contains a stacked bar chart and two key-value lists. It is used in pilot E application to show when an **alert** happened in general and in particular when a **resource** (i.e. an ambulance) has been requested during an exercise. The first generic key-value list is used to display a list of all requested vehicles including their amount and the total amount of vehicles that have been requested during an exercise. The second list shows the timestamps of the alerts/resource requests.

Downloads:

Source Code is available on Git Hub (see links).

- Care Measures Widget
  https://github.com/crismaproject/pilot-e-application/tree/feature/7-care-measures-widget



- Pre Triage Widget **https://github.com/crismaproject/pilot-e-application/tree/feature/8-pre-triage-widget**
- Triage Widget **https://github.com/crismaproject/pilot-e-application/tree/feature/9-triage-widget**
- Transportation Widget **https://github.com/crismaproject/pilot-e-application/tree/feature/10-transportation-widget**
- Alerts and Requests Widget **https://github.com/crismaproject/pilot-e-application/tree/feature/19-Alerts-and-Requests-Widget**

Team:

Frank Jonat

Development roles: development

Insta EvacSim

Insta EvacSim is a simple tool simulating the **impact** of resources allocated to evacuate people from a given area.
It calculates both the progress of the **evacuation** as well as the **resource** usage.

Documentation:

Specifications

Insta EvacSim is a simple tool modelling the **impact** of resources allocated to evacuate people from a given area. Other components can call Insta EvacSim using a **REST** **interface**. **It** gets the input data from **OOI**-WSR and stores the results there, respectively. In the REST call the tool gets the IDs of the current **world state** (for getting the **resource** and **situation** data), the new world state (for storing the calculation results) and the time step between the world states.

Downloads:

Team:

Kari Uusitalo

Development roles: development

Insta Response Preparedness Planner

Insta **Response** **Preparedness** Planner is a decision support tool, based on existing **situation** assessment functionalities of the Insta Response product family. Preparedness planning is based on creation of a pre-defined narrative (i.e. the Preparedness Plan) as a response to different kinds of **emergency** or otherwise exeptional situations.

Insta Response Preparedness Planner helps the decision maker to allocate the appropriate resources for response and **recovery** tasks as well as inform to the other relevant parties in case of a regional emergency situation, according to pre-planned patterns and up-to-date **resource** and situation data.

Insta Response Preparedness Planner is a subset of the toolset of the current Insta Response product family, tailored to suit the needs of the **CRISMA** project. Insta Response is a networked, integrated solution, which provides situation picture, **resource management**, integrated communications, computer-aided situation assessment, dispatching and field support functionalities. The product is targeted for emergency response centres as well as for other authorities in the command centres and on the field. The product also has a browser-based UI for administrative users not requiring the integrated communications and dispatching. Insta Response is based on client-server **architecture** and respects **SOA** principles. Services are provided for the clients from a distributed and clustered network enabling high availability and good scalability. The provided user interfaces include fat client (Insta Response desktop) **interface** as well as browser-based interfaces (Insta Response Portal).

Insta Response Preparedness Planner in CRISMA context

For CRISMA, the Insta Response Preparedness Planner concists of the Preparedness Planner **Service** and the web-based user-interfaces, which facilitate the creation of the preparedness plans and related **capability** needs, viewing and editing resource data as well as selecting and excecuting the preparedness plans which are appropriate to the situation. In CRISMA environment the Insta Preparedness Planner integrates into common situation data repository (**OOI**-WSR), which is also used as the communication bus between the different components. Insta Preparedness Planner gets the capability needs for each **world state** from the repository and provides several alternative combinations of resource units, which are suitable for the **mitigation** actions. When the user selects one of the proposed resource

combination alternatives, Insta Preparedness Planner sets these resources allocated in the repository. The allocation data is used by the other tools to simulate the progress and resource usage as the **scenario** progresses and to calculate the costs of the mitigation actions.

Insta Response Preparedness Planner: Plan Excecution View

Documentation:

Presentation

Insta **Preparedness** Planner presentation in the Links-section below.

A customer-oriented functional description of the "Insta **Response**" product is attached below.

Downloads:

- https://crisma-cat.ait.ac.at/system/files/Insta_Response_brochure_0.pdf
- Insta Preparedness Planner (presentation from Pilot A demo in Kemi, Finland, 2014-04-08) **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP5/WP52/02%20meetings/Pilot%20A%20demo%20in%20Kemi%207%20-%209%20April%202014/CRISMA%20Preparedness%20Planner%20in%20Pilot%20A.ppt**

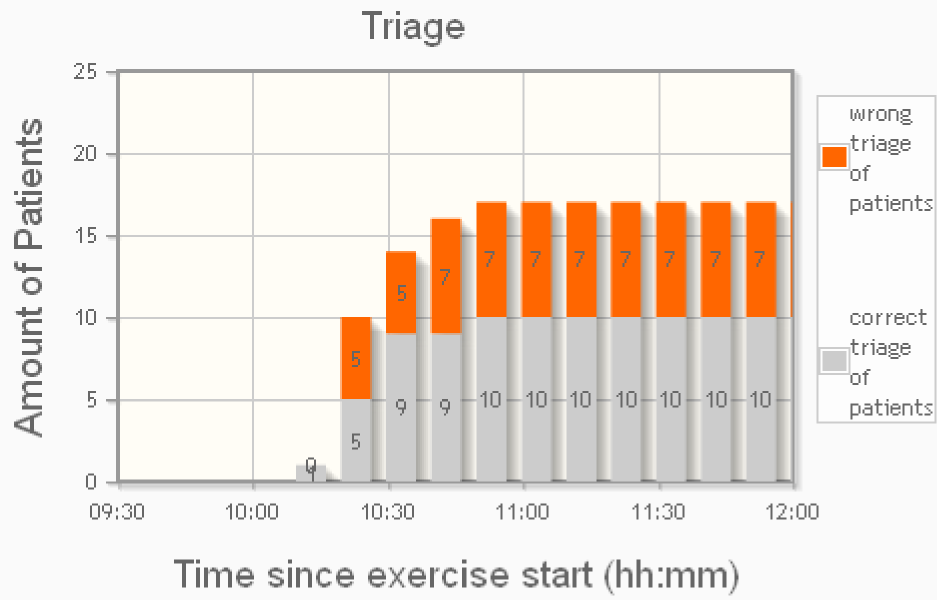
Implementation Plan

Implementation Plan for **CRISMA**-aware **Insta** **Preparedness** Planner is available at the CRISMA Work Space

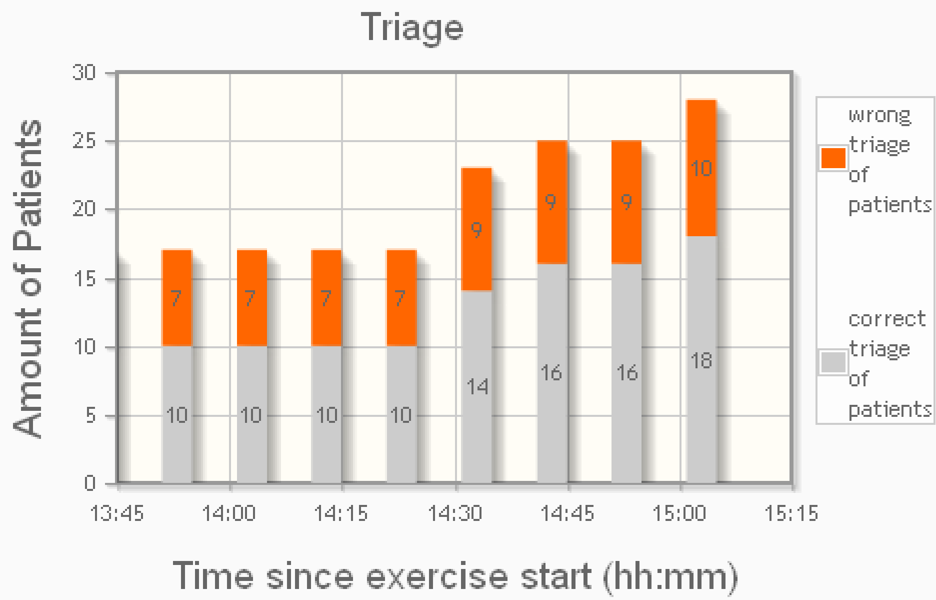
- Preparedness Planner Implementation Plan **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP34/03%20Deliverables/D34.1%20Software%20Components%20(final)/[X]%20BB%20Implementation%20Plan%20-%20Preparedness%20Plan.doc**

Downloads:

Team:



Triage	
11min	time until triage starts
60min	max time until triage starts
3	number of classification errors



Triage	
11min	time until triage starts
60min	max time until triage starts
3	number of classification errors

Ari Kosonen

Development roles: administration, architecture, coordinaton, requirements, specifications

Antti Järvinen

Development roles: development, integration

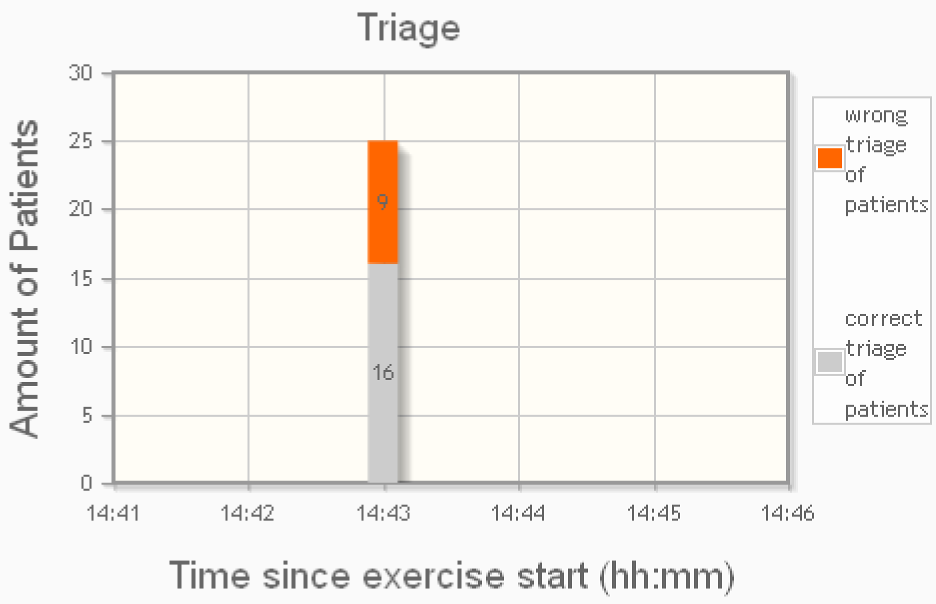
Technical Lead, development

Tuomo Soinio

Development roles: development, integration

Kari Uusitalo

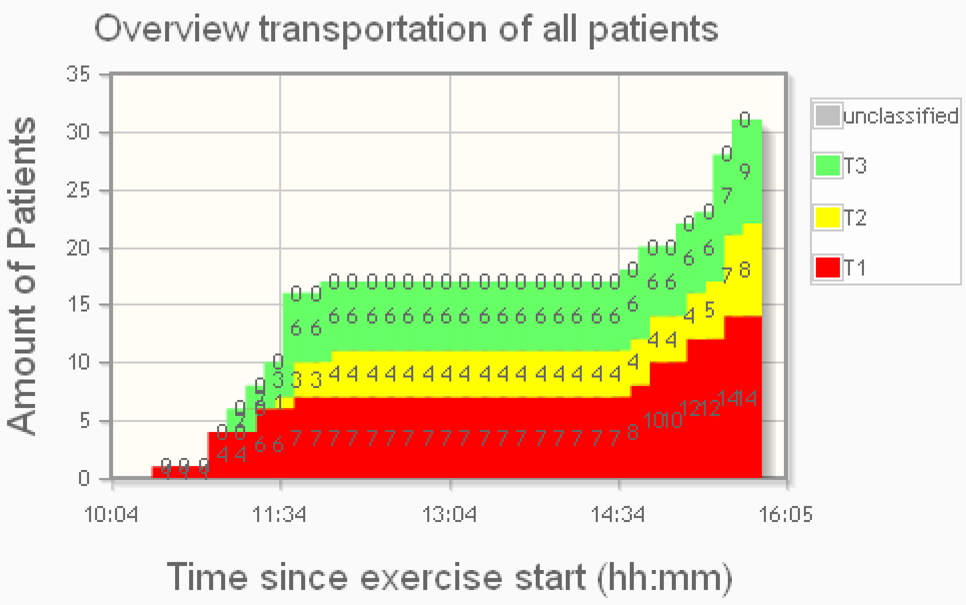
Development roles: development



Triage

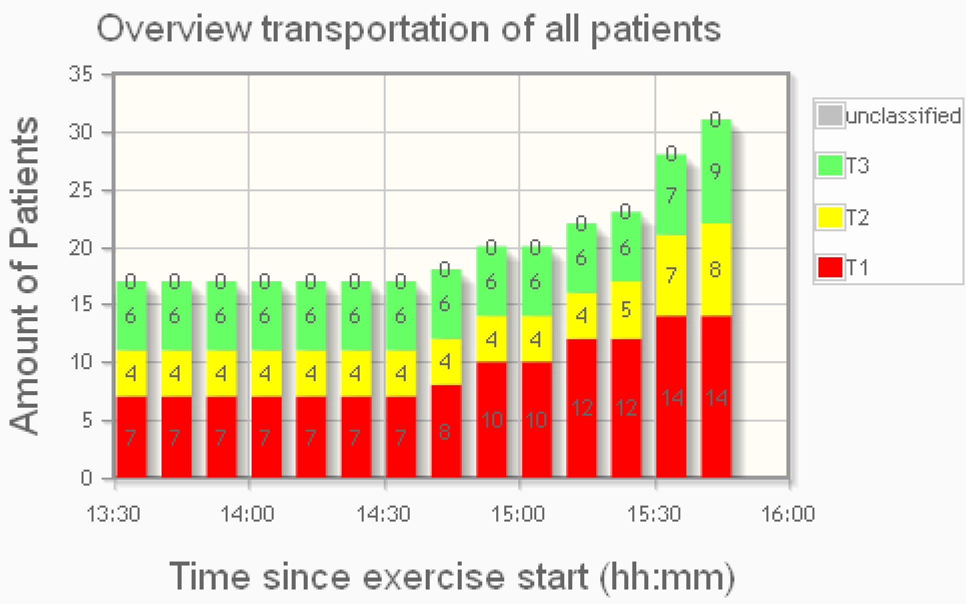
11min	time until triage starts
60min	max time until triage starts
3	number of classification errors

Leaflet.js



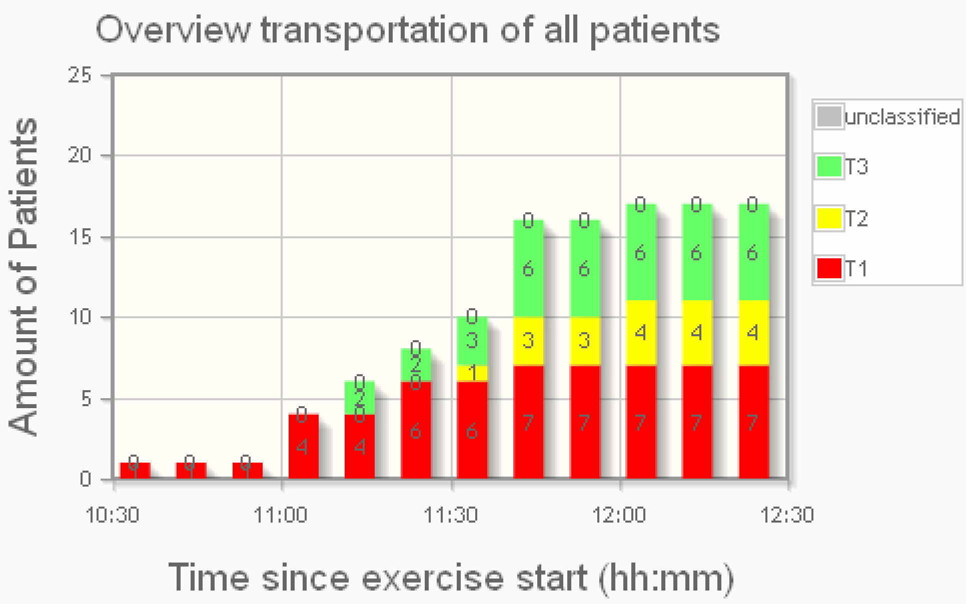
General

70min	max time last red patient is transported
43min	time first red patient is transported
0	number of people died



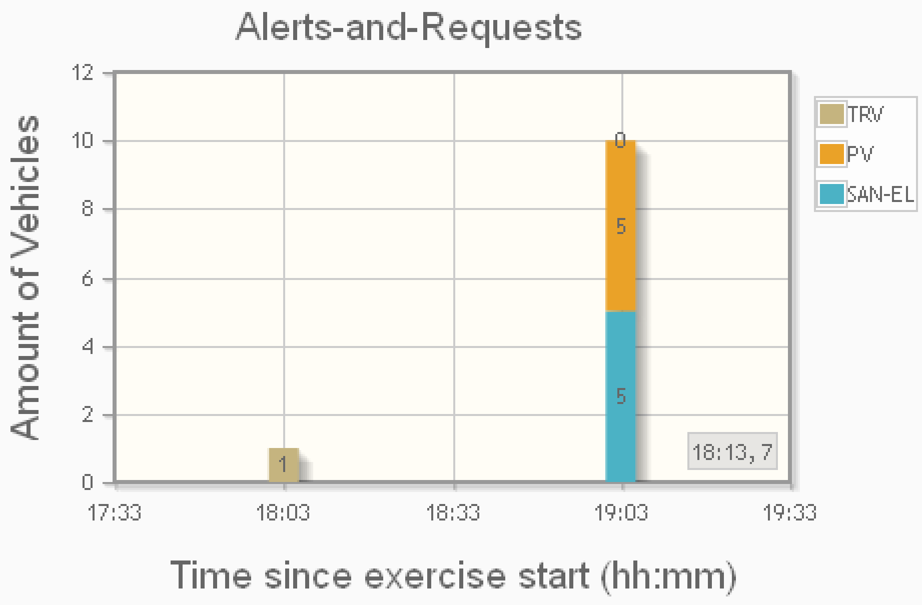
General

70min	max time last red patient is transported
43min	time first red patient is transported
0	number of people died



General

70min	max time last red patient is transported
43min	time first red patient is transported
0	number of people died

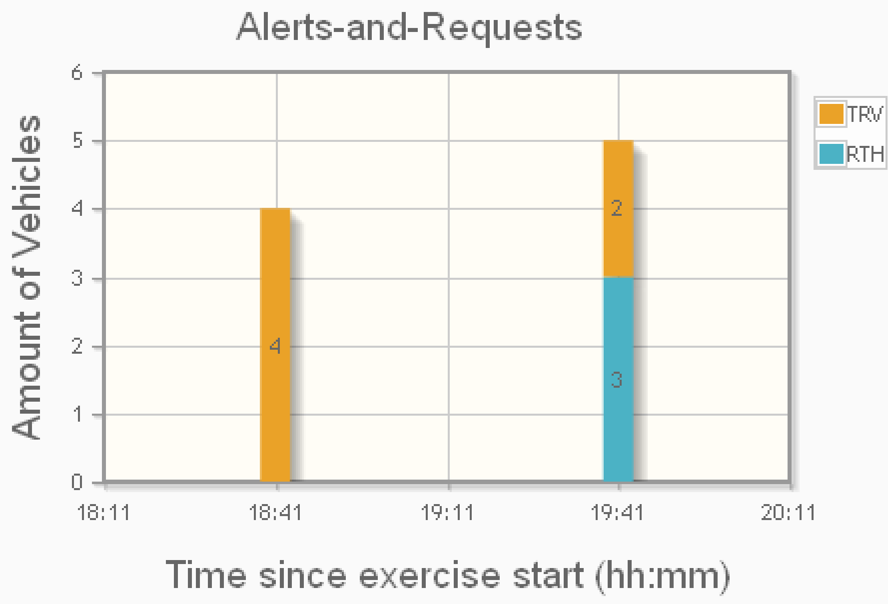


Requested Vehicles

5	SAN-EL
5	PV
1	TRV
11	total number

Alerts

18:03:03	hgfd
19:03:09	no mesage



Requested Vehicles

3	RTH
6	TRV
9	total number

Alerts

18:41:45	fdas
19:41:53	no mesage

Requested Vehicles

6	RTW
4	MTW
4	RTH
3	KDOW
3	PV

Alerts

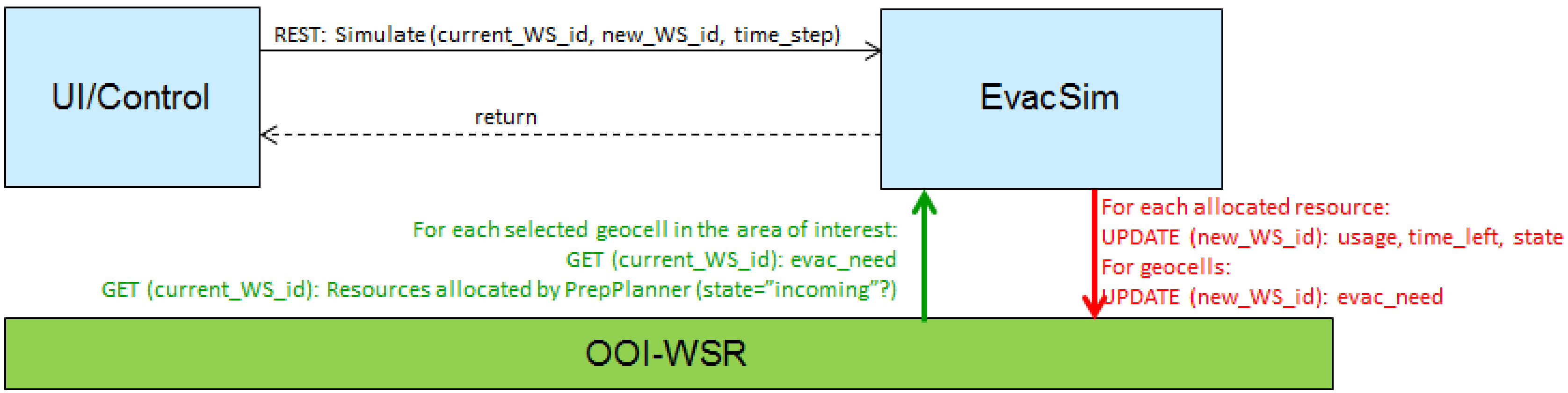
14:15:35	More responders
15:32:36	More responders
15:33:42	no mesage
15:34:11	no mesage
15:34:11	no mesage

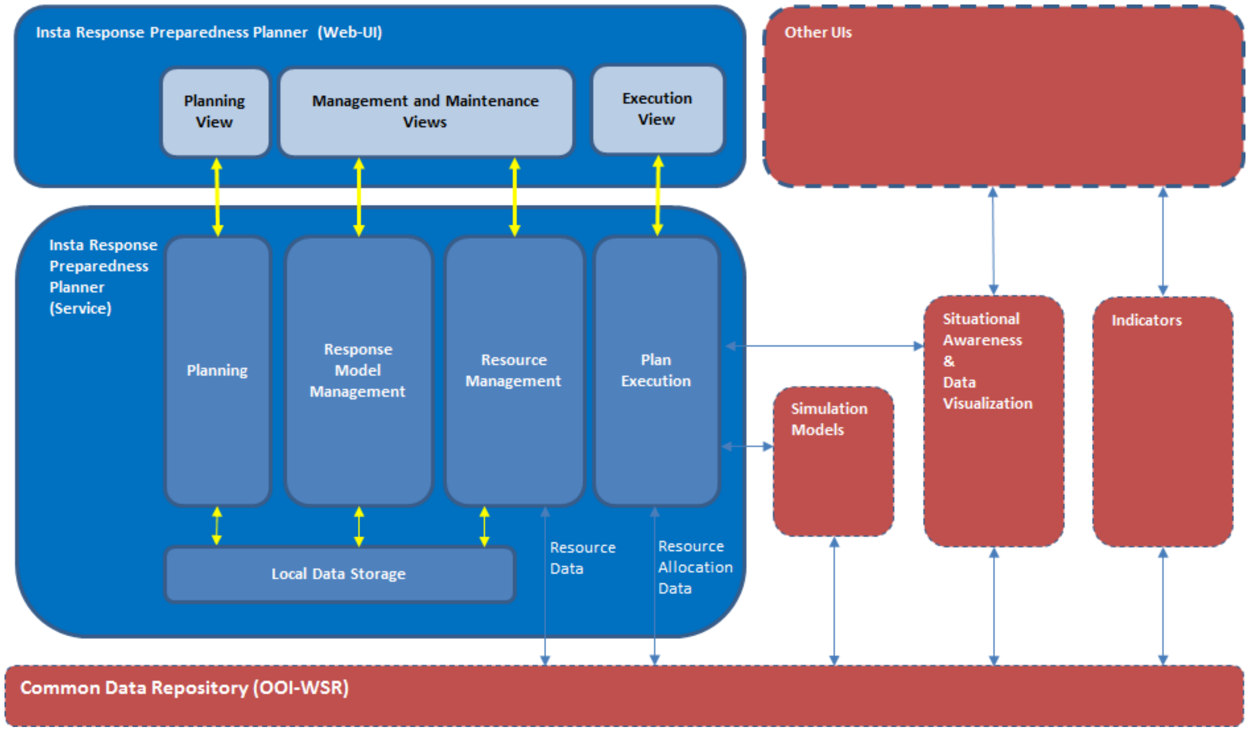
Requested Vehicles

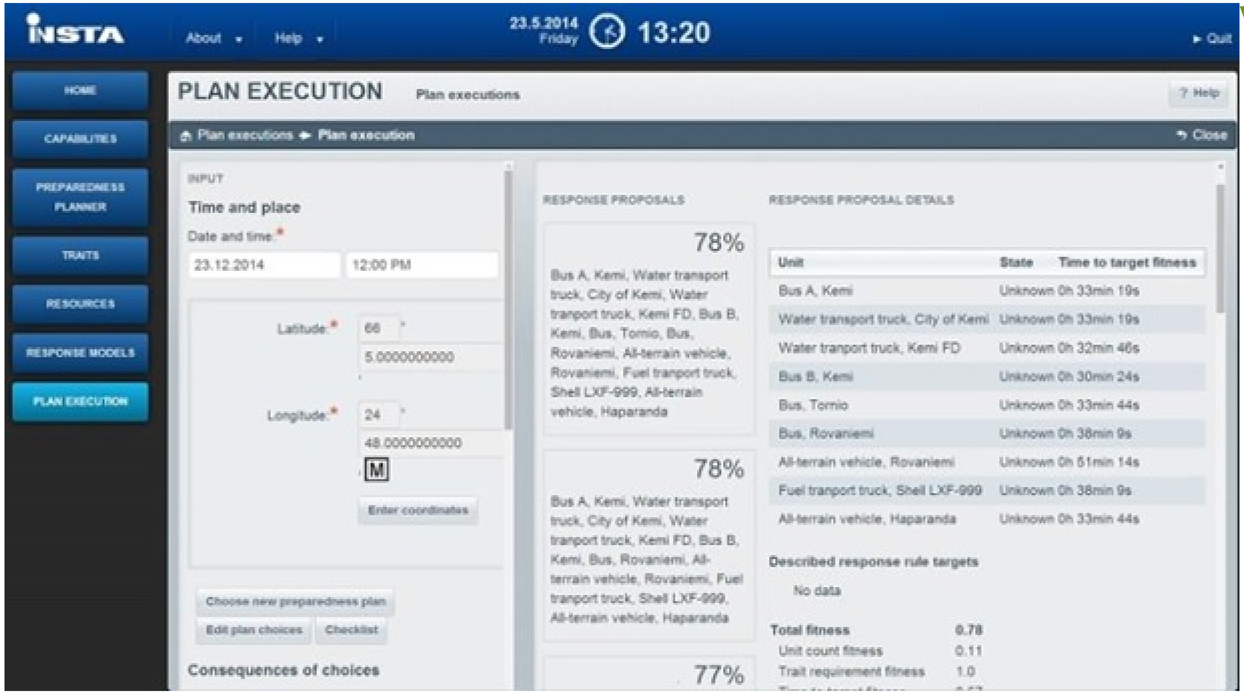
4	RTH
3	KDOW
3	PV
2	FT
22	total number

Alerts

15:34:20	no mesage
15:34:21	no mesage
15:34:21	no mesage
15:34:21	no mesage
15:34:21	no mesage


UI/Control
REST: Simulate (current_WS_id, new_WS_id, time_step)
return
EvacSim
For each selected geocell in the area of interest:
GET (current_WS_id): evac_need
GET (current_WS_id): Resources allocated by PrepPlanner (state="incoming"?)
For each allocated resource:
UPDATE (new_WS_id): usage, time_left, state
For geocells:
UPDATE (new_WS_id): evac_need
OOI-WSR


Insta Response Preparedness Planner (Web-UI)
Planning View
Management and Maintenance Views
Execution View
Other UIs
Insta Response Preparedness Planner (Service)
Planning
Response Model Management
Resource Management
Plan Execution
Situational Awareness & Data Visualization
Indicators
Simulation Models
Local Data Storage
Resource Data
Resource Allocation Data
Common Data Repository (OOI-WSR)



Version: 0.64

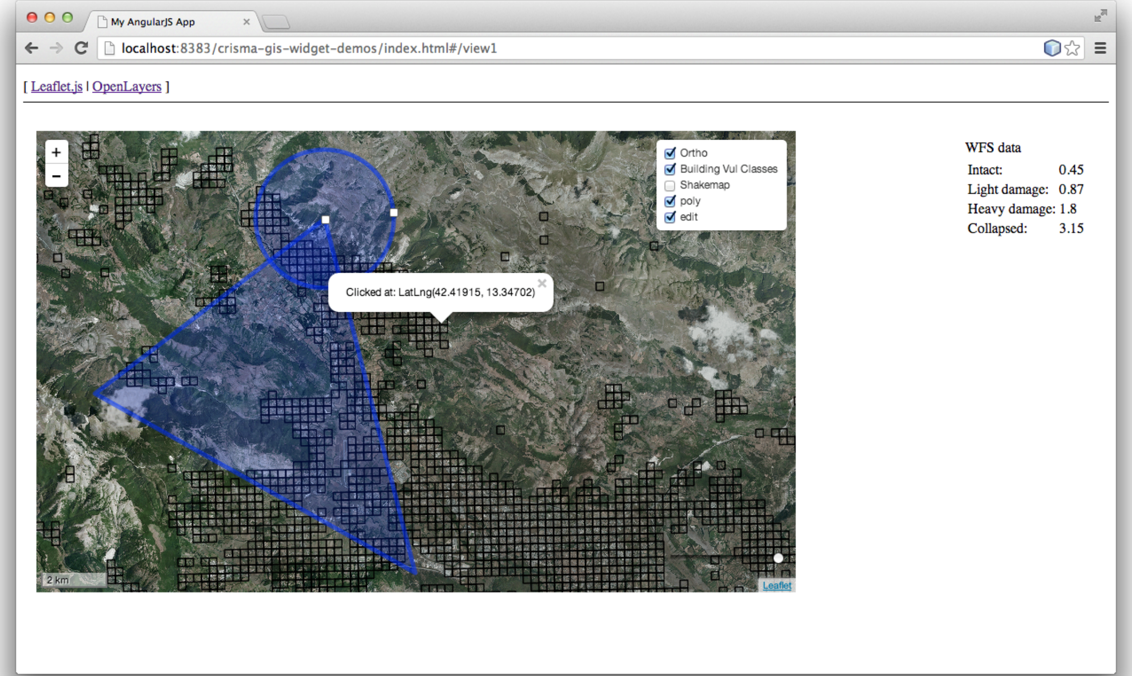
Leaflet.**js** is one of the software components that implement the Functional **Building Block** **GIS** **Widget** of the **CRISMA Framework**. **It** is used "as is" and thus not extended or modified for CRISMA.

“Leaflet is a modern open-source JavaScript library for mobile-friendly interactive maps. It is developed by **Vladimir Agafonkin** with a team of dedicated contributors. Weighing just about 31 KB of JS, it has all the features most developers ever need for online maps.
Leaflet is designed with simplicity, performance and usability in mind. It works efficiently across all major desktop and mobile platforms out of the box, taking advantage of HTML5 and **CSS3** on modern browsers while still being accessible on older ones. It can be extended with many plugins, has a beautiful, easy to use and well-documented **API** and a simple, readable source code that is a joy to contribute to.”

Leaflet.js in Action

The main features of Leaflet are:

- Supports **WMS**, Vector, Marker, Image and GeoJSON layers natively
- Asynchronous requests to load several layers in parallel
- Basic control themes for any devices (Desktop, Mobile)



- Support for nearly any well known modern browser (Desktop, Mobile)
- Smooth animations
- Pure CSS3 can be used for styling
- Hardware acceleration on iOS
- Extremely lightweight with no external dependencies
- Plugin **interface** and any Community developed plugins
- Pure JavaScript

Development Status

Since Leaflet is an **open source** project released under FreeBSD license, extensibility is explicitly given by the possibility to modify the source code. CRISMA specific modifications can either be contributed directly to the codebase of Leaflet on github or by a fork of the Leaflet github project. Additionally Leaflet also provides the possibility to create plugins. However, an extension of Leaflet.js by CRISMA Partner is currently not forseen.

Usage in CRISMA

Leaflet.js implements basic functionalities requested by the GIS Widget Building Block. Manipulation of GIS data is currently not the scope of this lightweight mapping component. Leaflet is an alternative to the cismap and OpenLayers software candidates for this Building Block. However, compared to both of the others it is not as powerful out of the box although compared to cismap it is more flexible when it comes to integration.

Documentation:

Developer's Guide

Leaflet.**js** Tutorials and Reference Documentation

- Leaflet.js Tutorials
  http://leafletjs.com/examples.html (http://leafletjs.com/examples.html)
- Leaflet.js Reference Documentation
  http://leafletjs.com/reference.html (http://leafletjs.com/reference.html)

Implementation Plan

Leaflet.**js** Implementation Plan

- Leaflet.js Implementation Plan
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP34/03%20Deliverables/D34.1%20Software%20Components%20(final)/%5BX%5D%20BB%20Implementation%20Plan%20-%20Leaflet.js.doc
- Leaflet.js Feature Matrix
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP34/03%20Deliverables/D34.1%20Software%20Components%20(final)/%5BX%5D%20BB%20Implementation%20Feature%20Matrix%20-%20Leaflet.js.xls

Downloads:

Download Leflet.**js** stable version, released on June 26, 2013 and last updated on July 25, 2013.

- Download Page
  http://leafletjs.com/download.html (http://leafletjs.com/download.html)
- Leaflet and OpenLayers Demos Sourcecode
  https://github.com/crismaproject/crisma-gis-widget-demos

Team:

Thorsten Hell

Supports Pilots & their technical stewards in the configuration of Leaflet.**js**, e.g. helping in the configuration of **WMS**/**WFS** data sources, layers, etc.
Supports Pilots & their technical stewards in the integration of Leaflet.js with other BBs, e.g. Data integration BBs (**OGC** Services), the **ICMM**, etc.
Supports Pilots & their technical stewards in the possible **adaptation** of Leaflet.js, this includes e.g. helping Pilots to implement functionalities needed in thier **CRISMA Application** that are not supported by Leaflet.js

Manuel Warum

Provides support related to the usage of Leaflet.**js** in the Wirecloud **Mashup Platform**.

Life Safety Model 2D

Authors: **Armonie Cossalter**
Version: v2.2

The Life Safety **Model** 2D (LSM2D) software is a **simulation tool** that performs **agent** based simulation and models the dynamical interaction between moving (pedestrians and vehicles) and stationary (buildings) environment objects with flood wave. **It** determines whether a population at **risk** can evacuate safely or find a safe shelter during an **event**.

The LSM2D software can be used to model flood **emergency management** scenarios and estimate **evacuation** times and loss of life. With it, planners can compare different emergency management strategies that can assist in reducing the loss of life and injuries during floods. It can also assist in simulating emergency scenarios and enable **resource** planning and the exploration consequences of different decisions, like:

- time needed for a safe evacuation against time available and
- loss of life and vehicles due to the characteristics of the flood.

The LSM2D software calculates fate of a set of simulation objects described by their position at each time step of the simulation, ability to withstand the effect of flood wave and how it would react to the approaching wave. It uses a generalized event logic to determine **object**'s parameters and what happens if it encounters the flood (as dictated by the Object Damage and Loss Function – ODLF).

This software is owned by **HR Wallingford and BC Hydro**.

Documentation:

Presentation

All documentation and publications related to Life Safety **Model** 2D can be found on the Life Safety Model website.

- Life Safety Model website
  http://www.lifesafetymodel.net/index.html (http://www.lifesafetymodel.net/index.html)

Downloads:

Team:

Marianne Grisel

Development roles: modelling

Marianne will use the LSM2D in order to create the **evacuation** **model** for Coastal submersion in Charente-Maritime, France.

Multi Criteria Analysis and Decision Support Widget (Java)

This is the Java implementation of the Multi **Criteria** Analysis and Decision Support View.

The implementation is **OWA**-based and enables users to define custom decision strategies by letting them choose the importance of the different criteria as well as an appropriate weight vector (andness/orness). The ranking result is then visualised for every selected worldstate. This component is used in the **CRISMA**

Demonstrator.

Features:

- CRUD operations for custom decision strategies
- Importance level of each criteria individually adaptable
- Five different preconfigured weight vectors to emphasis andness/orness
- On-The-Fly calculation of ranking results, every change is reflected immediately

Multicriteria and Decision Support Widget

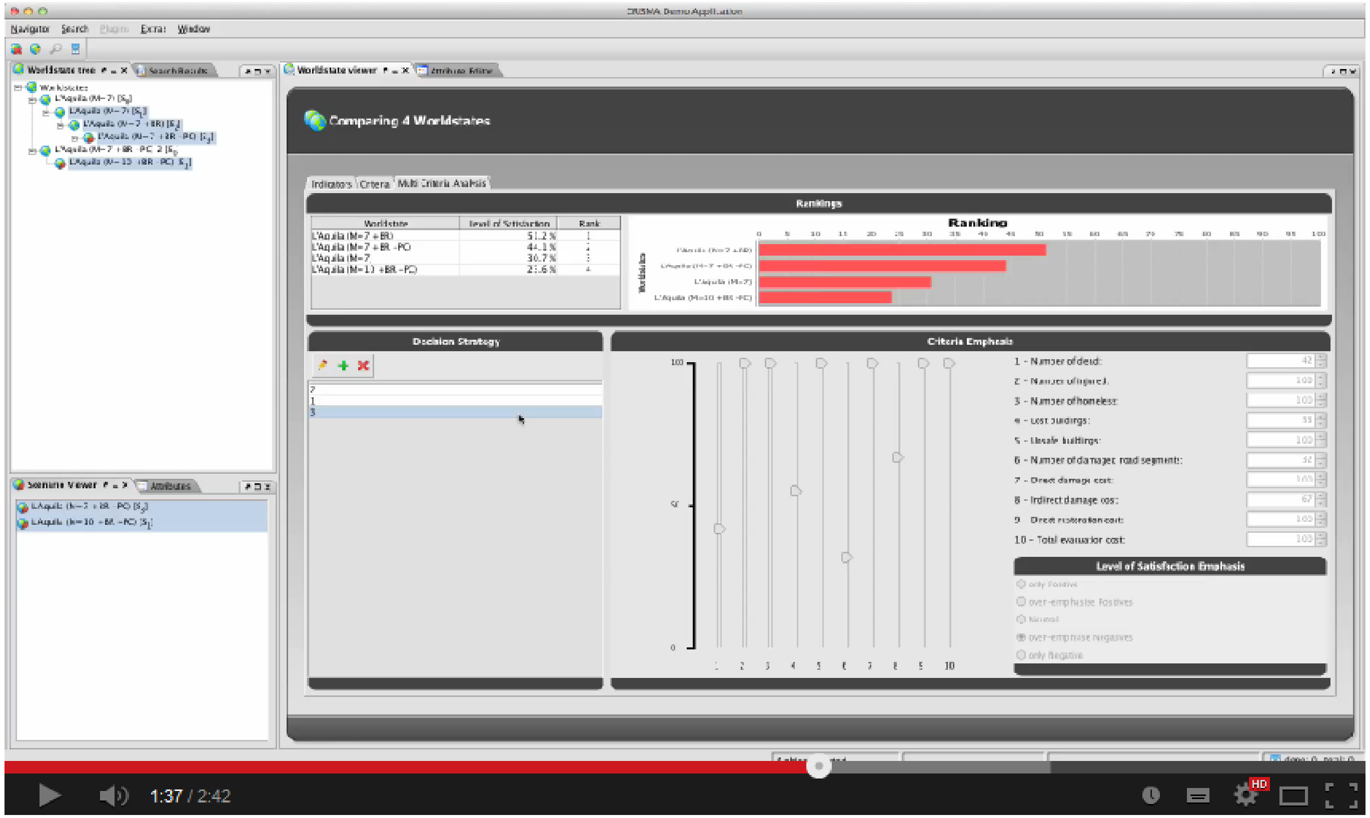
Documentation: Demo

This is a **demonstrator** video of the Multi **Criteria** Analysis and Decision Support **Widget** (Java) implementation. **It** contains no sound but subtitles which should be activated by default but can simply be switched on via the caption control of the youtube player. The video is available in up to FullHD resolution.

- Demonstrator Video for Scenario Analysis and DSS **https://www.youtube.com/watch**

Functional Description

- Description of the Multi Criteria Analysis and Decision Support View BB **https://crisma-cat.ait.ac.at/bb/Multi-Criteria-Analysis-and-Decision-Support-View**



Downloads:

The cids-custom-**crisma** binary is available from the **cismet** repository

- cismet repo
  https://repo.cismet.de/simple/cismet-libs-snapshots-local/de/cismet/cids/custom/crisma/

  The source code is available from the github repository

- GitHub repo
  https://github.com/crismaproject/cids-custom-crisma

  This is the **OWA** library that has been implemented and is used in this component. The Namespace library is needed at runtime.

- OWA library (JavaScript)
  http://crisma.cismet.de/lib/js/de/cismet/crisma/dev/owa.js (http://crisma.cismet.de/lib/js/de/cismet/crisma/dev/owa.js)
- Namespace library (JavaScript)
  http://crisma.cismet.de/lib/js/de/cismet/Namespace/snapshot/Namespace.js (http://crisma.cismet.de/lib/js/de/cismet/Namespace/snapshot/Namespace.js)

Team:

Martin Scholl

Development roles: configuration, development, installation, integration, support
Main **developer** of the **Building Block**.

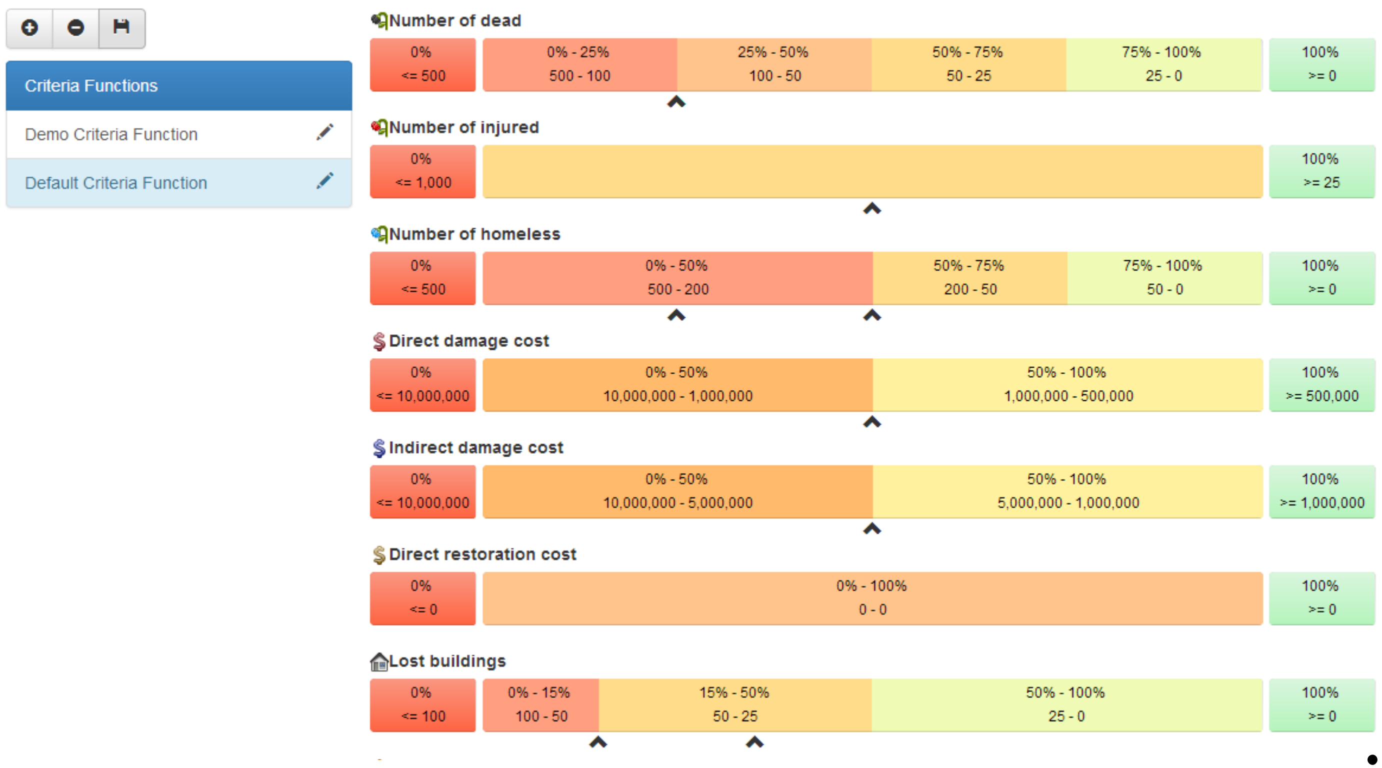
Multi Criteria Analysis and Decision Support Widgets (JavaScript)

Version: 1.0

This is the JavaScript implementation of the Multi **Criteria** Analysis and Decision Support View **Building Block**.

The Multi Criteria Analysis and Decision Support View Building Block is realised by several widgets, amongst others the **Criteria Function** Definition **Widget**, the Decision Strategy Widget and the Decision Ranking Widget. The Criteria Function Definition Widget allows defining functions that converts **indicator** values to criteria while the Decision Strategy Widget allows defining a weighting strategy for different Criteria while the Decision Ranking Widget allows selecting a previously defined Decision Strategy and to apply **it** to a set of **Crisis Management** Scenarios.

The screen shot below shows the Criteria Function Definition Widget.



Multi Criteria Analysis: Criteria function Widget

The Multi Criteria Analysis and Decision Support Widgets are new software components that are developed from scratch. They satisfy the technological requirements for new software developments as they are realized as Composite UI Module (HTML and JavaScript). Thus they can be used in any UI Integration Platform that supports HTML/CSS and JavaScript such as a Browser. Moreover, there is an additional build for the WireCloud platform effectively making it a Mashable UI Component.

Documentation:

Install Guide

See the GitHub projects for info.

- AngularJS implementation: Readme on GitHub **https://github.com/crismaproject/worldstate-analysis-widgets**
- WireCloud implementation: Readme on GitHub **https://github.com/crismaproject/worldstate-analysis-widgets-wirecloud**

Developer's Guide

See the GitHub projects for info.

- AngularJS implementation: Readme on GitHub **https://github.com/crismaproject/worldstate-analysis-widgets**
- WireCloud implementation: Readme on GitHub **https://github.com/crismaproject/worldstate-analysis-widgets-wirecloud**
- Filling ICMM with compatible data **https://github.com/crismaproject/worldstate-analysis-widgets/wiki/HowTo:-Enable-MCA-and-DSS-in-your-CRISMA-application**

Functional Description

- Multi Criteria Analysis and Decision Support View Functional Description **https://crisma-cat.ait.ac.at/bb/Multi-Criteria-Analysis-and-Decision-Support-View**

FAQ

Issue Tracking and FAQ on GitHub

- AngularJS implementation: GitHub Issue Tracker **https://github.com/crismaproject/worldstate-analysis-widgets/issues**
- WireCloud implementation: GitHub Issue Tracker **https://github.com/crismaproject/worldstate-analysis-widgets-wirecloud/issues**

Demo

(http://youtu.be/1DoGo2AOEOA)

This demo video shows how criteria functions can be defined.

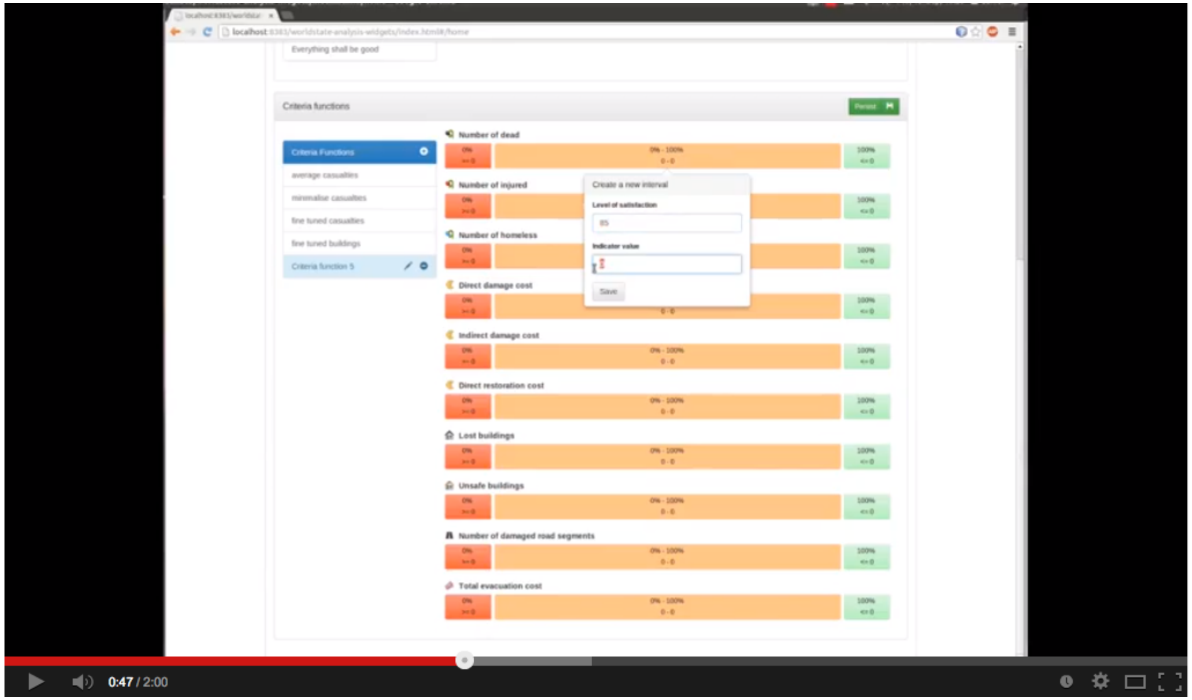
(http://youtu.be/URX4hFxElHQ)

This demo video shows how criteria can be analysed.

(http://youtu.be/TTPzA3SoP4I)

This demo video shows how decision strategies can be defined.

(http://youtu.be/5vCUX8k7P4s)



This demo video shows how DSS can be done.

Tutorial

(http://youtu.be/niQC0RbEtrQ)

This is a HowTo video that showcases how the Worldstate WireCloud widgets can be wired together.

- HowTo: Wire Worldstate WireCloud Widgets **http://youtu.be/niQC0RbEtrQ (http://youtu.be/niQC0RbEtrQ)**

Downloads:

This component is divided into several AngularJS directives that can be put together individually and as needed. The repository, however, contains a basic demo **application** that puts all available analysis and **DSS** components together in a meaningful way. Additionally, there is the repository that only contains wire code for WireCloud and thus effectively makes the analysis and DSS components available as mashable widgets.

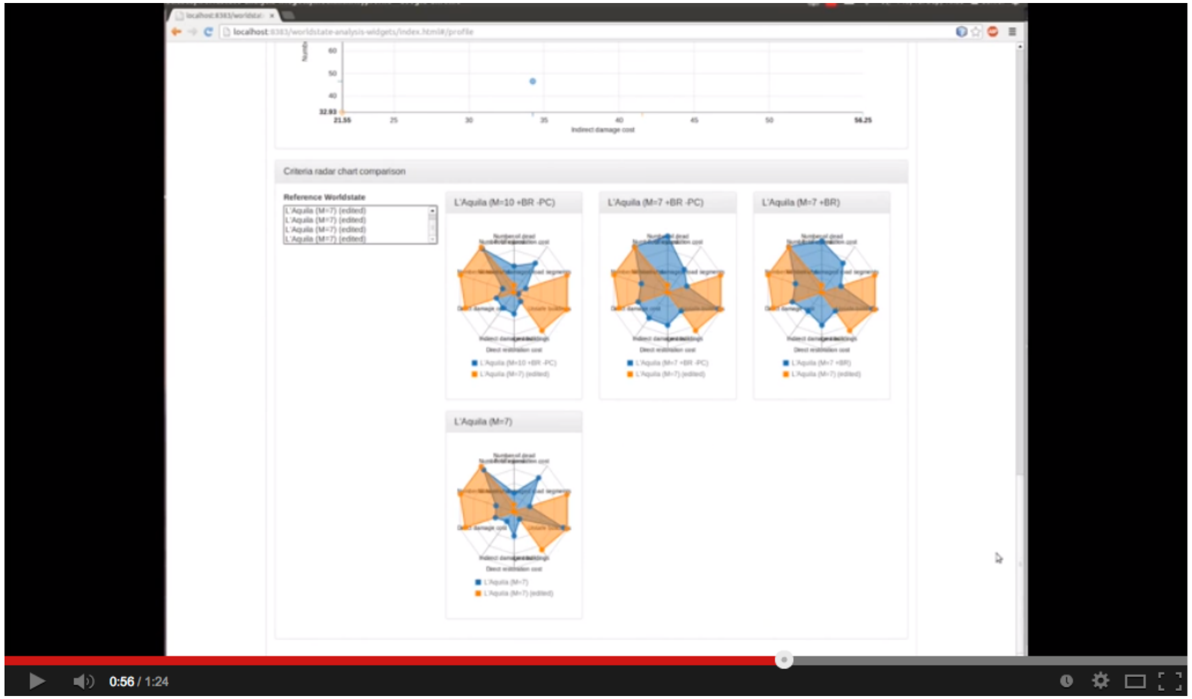
- AngularJS components **https://github.com/crismaproject/worldstate-analysis-widgets**
- WireCloud component **https://github.com/crismaproject/worldstate-analysis-widgets-wirecloud**

Ready-to-use build of the analysis and **DSS** mashable WireCloud **widget**. As soon as the **CRISMA** WireCloud platform marketplace is available the newest builds can be obtained from there. A minified version of the WireCloud widget will also be available in the future.

The AngularJS components can easily obtained via the dist folder of the project or via the bower dependency management tool

shell@computer:> bower install --save worldstate-analysis-widgets

- AngularJS components **https://github.com/crismaproject/worldstate-analysis-widgets/tree/v1.0.7/dist**



- Mashable WireCloud widget (not minified) **http://crisma.cismet.de/lib/wirecloud/worldstate-analysis-widgets-wirecloud.wgt (http://crisma.cismet.de/lib/wirecloud/worldstate-analysis-widgets-wirecloud.wgt)**

Team:

Martin Scholl

Development roles: architecture, coordinaton

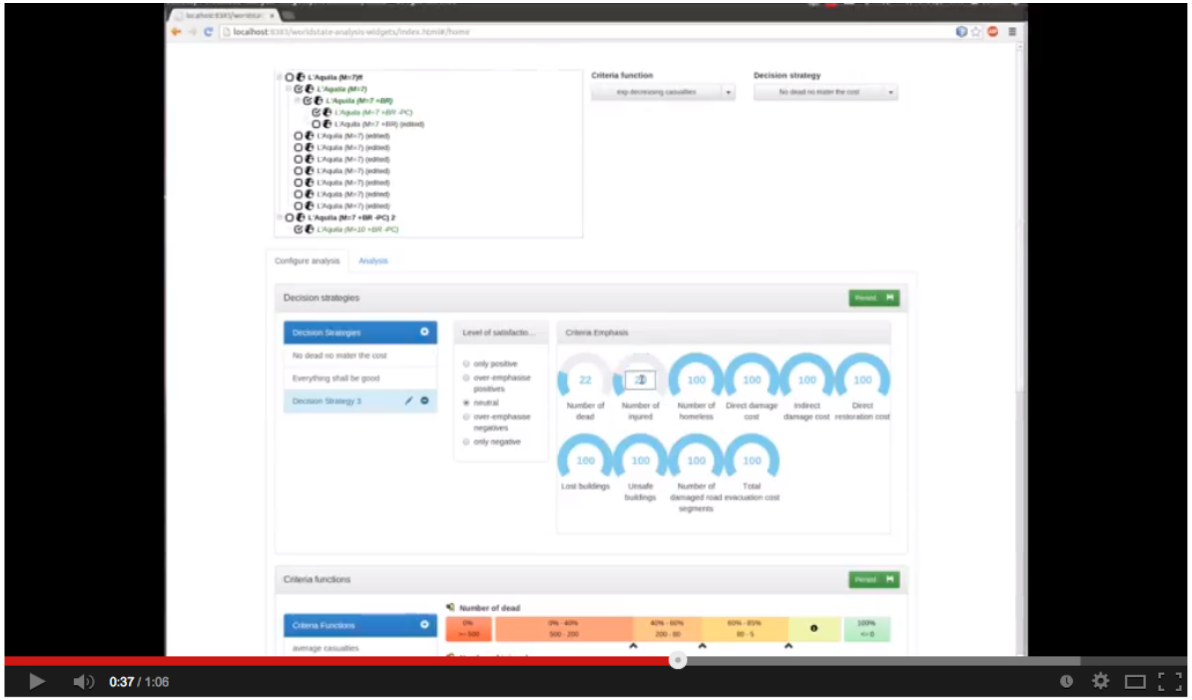
Coordinates development activities and provides general support.

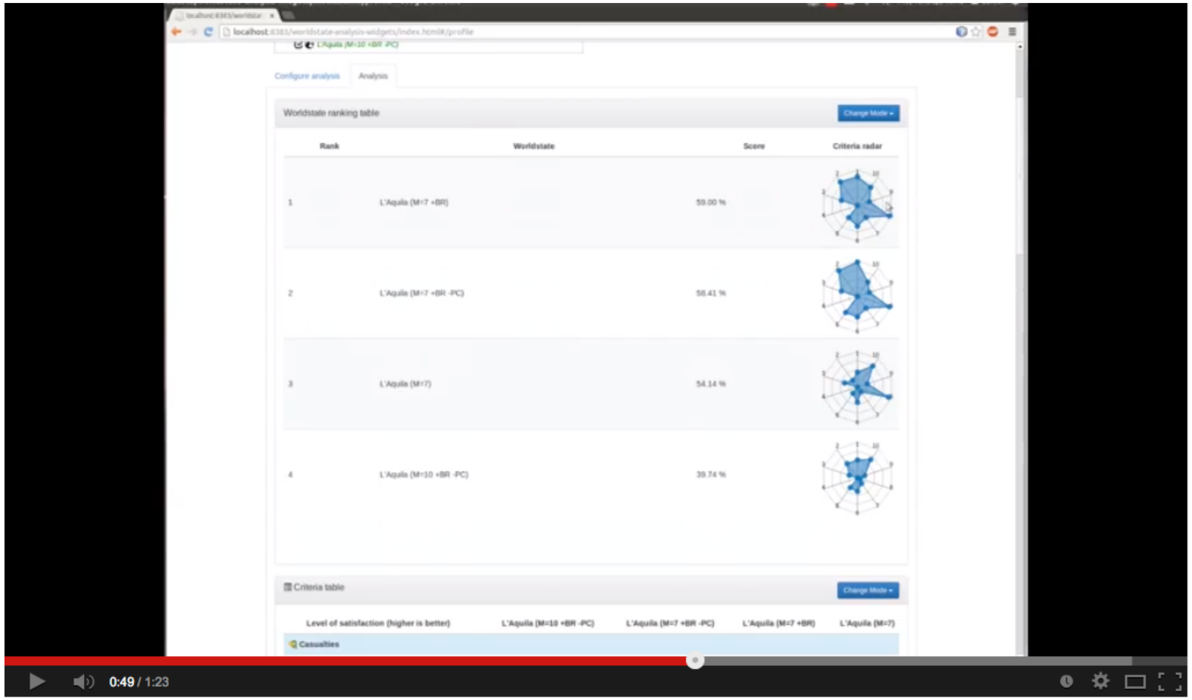
Daniel Meiers

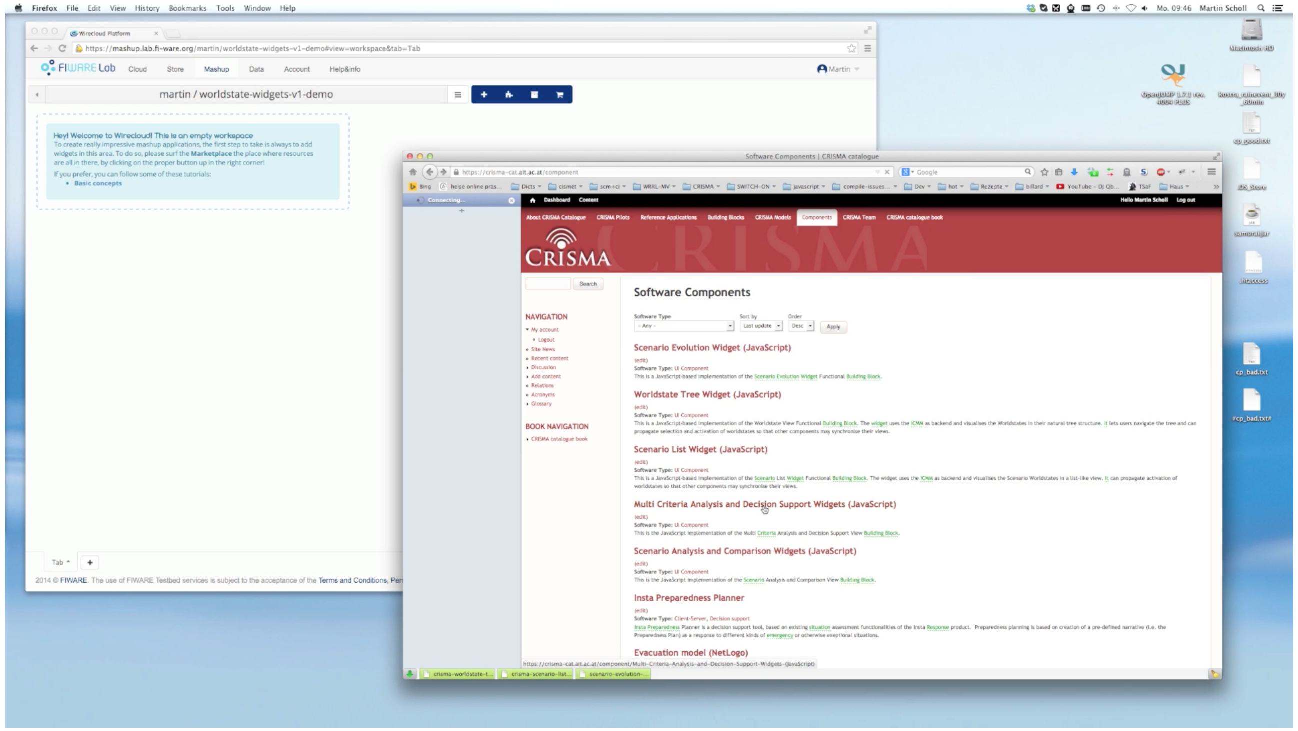
Development roles: development, installation, integration, support

Lead **developer** of this **widget** and also main contact point for any **Pilot** support regarding the embedding of this widget into **CRISMA** applications.

OOI World State Repository




Configure analysis
Analysis
Worldstate ranking table
Change Mode
Rank
Worldstate
Score
Criteria radar
1
L'Aquila (M=7 +BR)
59.00 %
2
L'Aquila (M=7 +BR -PC)
58.41 %
3
L'Aquila (M=7)
54.14 %
4
L'Aquila (M=10 +BR -PC)
39.74 %
Criteria table
Change Mode
Level of satisfaction (higher is better)
L'Aquila (M=10 +BR -PC)
L'Aquila (M=7 +BR -PC)
L'Aquila (M=7 +BR)
L'Aquila (M=7)
Casualties
0:49 / 1:23



Version: 1

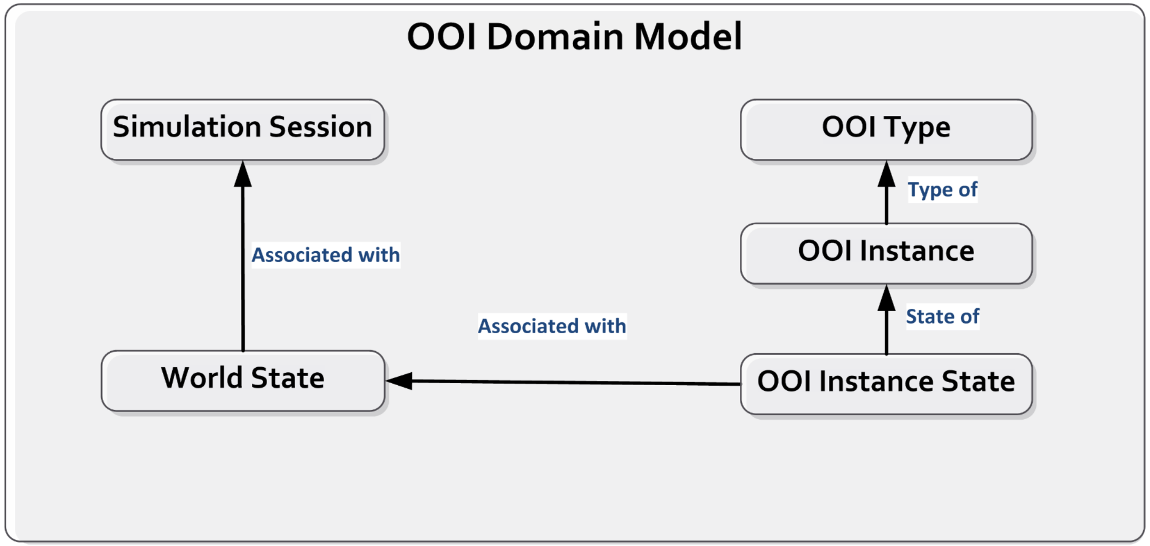
OOI-WSR is a **Resource Management** related module that enables archiving, querying and manipulation of OOI **world state** data. This module serves as a Repository **service** for OOI data that can be consumed or manipulated by other interaction or functional building blocks and resource management models.

In order to make this document self-contained, some basic definitions are needed:

- OOI – **Entity** (**object**) that is manipulated by the user or automatically by some resource management mathematical **model**. Contains a flat list of properties, that each construct from one of the following types: Number, Text, Date and Time, Geography and Geometry.
- **Simulation** Session – Represents a training or planning simulation session that is initiated manually by the user. **It** usually involves defining the initial world state and then applies user decisions, via running resource management models or manually updates the OOI state. Each OOI state must be associated to specific world state instance.
- World State – Represent a meaningful OOI data snapshot at specific time. The world state holds list of OOI instances with state that relevant to specific point in time. It usually represent the following:
    - Initial state of simulation Session
    - Output result of model execution at specific point in time.
    - Manual manipulation of user at specific point in time.

- Final state of Simulation session

The relation between the different entities described in the following diagram:



The information stored in the context of Simulation Session and World State OOI data and can be updated manually by user or via execution of resource management models.

The OOI-WSR exposes the following APIs types:

- **RPC** based **HTTP** APIs that follow basic **REST** principles.
- **WFS** **OGC** mainly for query world state data.
- ODBC in order to support external analysis services (e.g. Indicators and Analysis **BB**).

In a **CRISMA Federation**, data may be exchanged directly between federates (peer to peer) while the exchange of Control and Communication Information (information about simulation cases, control flows, events, etc.) must be performed using the APIs provided by the **ICMM**. Therefore OOI-WSR publishes all world state changes to the ICMM BB.

The OOI-WSR conceptual design is described in the following diagram:

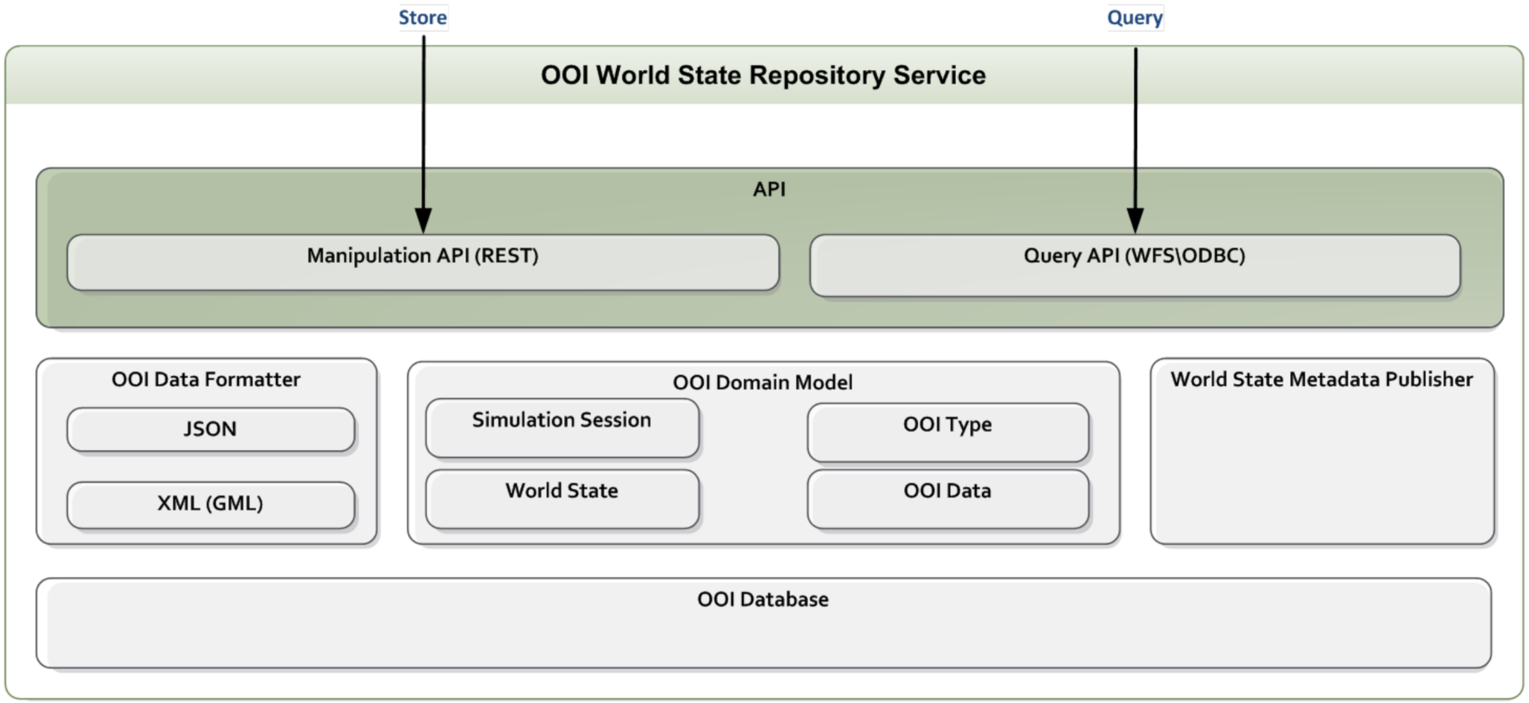
Documentation:

Implementation Plan

- Implementation Plan **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP34/01%20Work%20in%20Progress/Software%20Components/OOI%20World%20State%20Repository/OOI%20World%20State%20Repository%20-%20BB%20Implementation%20Plan.doc**

Developer's Guide

- REST API (including Swagger online documentation and JavaScript Samples) **http://54.213.161.17/App/index.html (http://54.213.161.17/App/index.html#/api/rest)**
- WFS API (Including JavaScript Samples) **http://54.213.161.17/App/index.html (http://54.213.161.17/App/index.html#/api/wfs)**



Demo

Demo video that illustrate the **service** usage and **API** documentation

Downloads:

- Online Service **http://54.213.161.17/App/index.html (http://54.213.161.17/App/index.html#/)**

Team:

Oren Deri

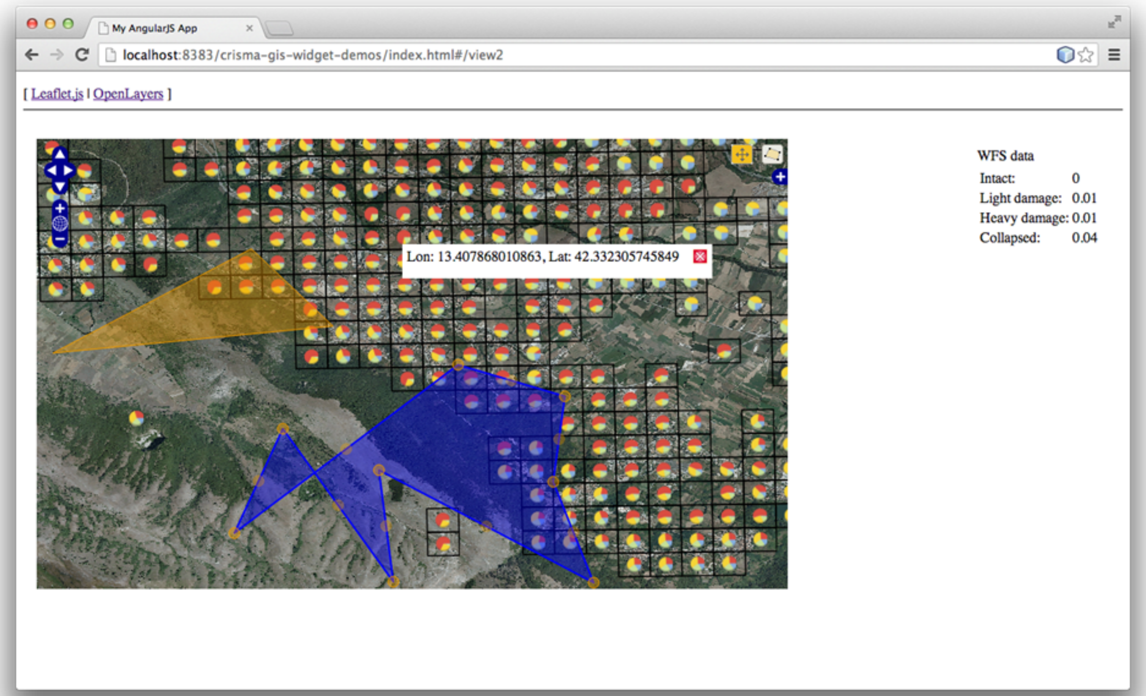
OpenLayers

Version: 2.13

OpenLayers is one of the software components that implement the Functional **Building Block** **GIS** **Widget** of the **CRISMA Framework**. **It** is used "as is" and thus not extended or modified for CRISMA.

OpenLayers is a JavaScript-based map **application** specialised to access **OGC** compliant services like **WFS** and **WMS** for information from internal and external geospatial data sets. Thereby, it provides powerful visualisation and editing functionalities for OGC compliant geospatial information (e.g. **GML**). The main features of cismap are:

- Supports a vast array of standards such as WMS(-T), WFS(-T), GeoJSON, etc. natively
- Asynchronous requests to load several layers in parallel
- Provides many features such as Editing, Navigation history, UndoRedo
- Pure **CSS3** can be used for styling
- Pure JavaScript without dependencies
- Supports most Desktop and Mobile browsers
- Huge community
- Good and extensive documentation



OpenLayers

Development Status

OpenLayers is a powerful mapping component implementation that has grown over the years and thus provides a rather good variety of supported standards and tools. It is suited to provide a viable GIS View Building Block implementation. However, some functionalities and controls of the GIS View **BB** are not available yet. However, modification and extension of OpenLayers by CRISMA is currently not forseen.

Usage in CRISMA

It is a true alternative for both cismap and Leaflet because although it is not quite as powerful as cismap it makes up for it by its pure JavaScript nature and thus - opposed to cismap - complies with the technological requirements for Composite UI Modules and Mashable Composite UI Modules (pure HTML5 and JavaScript widgets). Compared to Leaflet it does not provide the same level of simplicity but makes up for it by providing more features and a bigger community.

Documentation:

Implementation Plan

OpenLayers Implementation Plan.

- BB Implementation Feature Matrix - OpenLayers
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP34/03%20Deliverables/D34.1%20Software%20Components%20(final)/%5BX%5D%20BB%20Implementation%20Feature%20Matrix%20-%20OpenLayers.xls
- BB Implementation Plan - OpenLayers
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP34/03%20Deliverables/D34.1%20Software%20Components%20(final)/%5BX%5D%20BB%20Implementation%20Plan%20-%20OpenLayers.doc

Developer's Guide

OpenLayers has begun maintaining a set of prose documentation. Currently, **it** is minimal, but in the long term, most stable wiki documentation should be migrated into this website.

- OpenLayers Documentation
  http://trac.osgeo.org/openlayers/wiki/Documentation (http://trac.osgeo.org/openlayers/wiki/Documentation)

Downloads:

OpenLayer Sourcecode Repository @github

- Sourcecode Repository
  https://github.com/openlayers/openlayers
- Leaflet and OpenLayers Demos Sourcecode
  https://github.com/crismaproject/crisma-gis-widget-demos

OpenLayers Download Page

- Downloading OpenLayers
  http://trac.osgeo.org/openlayers/wiki/HowToDownload (http://trac.osgeo.org/openlayers/wiki/HowToDownload)

Team:

Thorsten Hell

Supports Pilots & their technical stewards:

- in the configuration of OpenLayers, e.g. helping in the configuration of **WMS**/**WFS** data sources, layers, etc.
- in the integration of OpenLayers, with other BBs, e.g. Data integration BBs (**OGC** Services), the **ICMM**, etc.
- in the possible **adaptation** of OpenLayers, this includes e.g. helping Pilots to implement functionalities needed in their **CRISMA Application** that are not supported by OpenLayers.

Manuel Warum

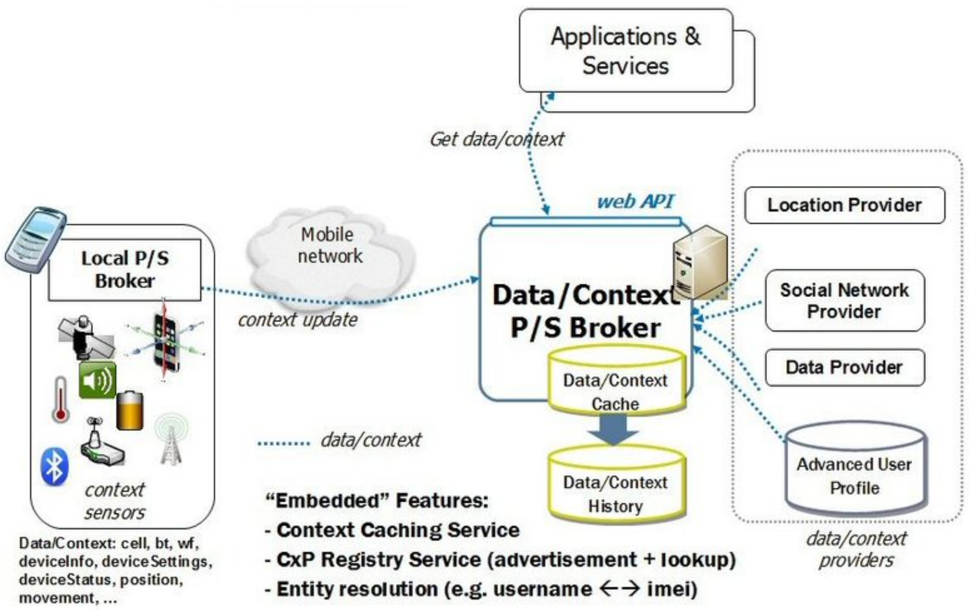
Provides support related to the usage of OpenLayers in the Wirecloud **Mashup Platform**.

Orion Context Broker

Authors: **Hermann Huber**

Peter Kutschera
Denis Havlik

The Orion Broker is a reference implementation of the FI-Ware Publish/Subscribe Context Broker GE, providing the NGSI9 and NGSI10 interfaces. In **CRISMA**, **it** is used as a reference implementation of the CRISMA Publish Subscribe Context Broker **BB**. The Context Broker is used by other building blocks to exchange events and process updates. It acts as a single point of contact and allows to create structured communication topologies. Building blocks may subscribe for specific events and will be notified if another **building block** reports such an **event**.



Using the Orion Context Broker you are able to register context elements and manage them through updates and queries. In CRISMA for instance a "world state" or a "resource" can be seen as a typical context element. Subscriptions can be used to track changes of context elements with respect to some predefined conditions.

The Orion Broker will enable publication of context information by entities, referred as Context Producers, so that published context information becomes available to other entities, referred as Context Consumers, which are interested in processing the published context information. In case of CRISMA building blocks may play the **role** of Context Producers, Context Consumers or both. The Orion Broker supports two ways of communications: push and pull towards both the Context Producer and the Context Consumer. That means that a Context Producer with a minimal or very simple logic may continuously push the context information into the Context Broker. In a similar way, Context Consumers can pull the context information from the Context Broker (on-request mode). The Context Broker can push the information to Context Consumer interested in it (subscription-mode).

A fundamental principle is the total decoupling between Context Producers and Context Consumers. This means that Context Producers publish data without knowing which, where and when Context Consumers will consume published data. On the other hand, Context Consumers consume context information of their interest, without knowing which Context Producer has published a particular event: they are just interested in the event itself but not in who generated it.

Documentation:

User Guide

Orion Public/Subscribe broker comes with a wealth of documentation, including the Users and Programmers guide which is linked below.

The short "primer" is provided by **AIT** "as is" and in the hope **it** might be useful, but with absolutely no warranty. Please inform us if you update this document, we would be happy to provide the updated version here. Licensing conditions for this work are the same as for the main **catalogue** context (**cc Attribution-ShareAlike (http://creativecommons.org/licenses/by-sa/3.0/)**)

Downloads:

- https://crisma-cat.ait.ac.at/system/files/uid_2/Orion%20Broker%20-%20Primer.doc
- User and Programmers Guide **https://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Publish/Subscribe_Broker_-_Orion_Context_Broker_-_User_and_Programmers_Guide**
- NGSI Context Management - Open Mobile Alliance **http://technical.openmobilealliance.org/Technical/release_program/docs/NGSI/V1_0-20101207-C/OMA-TS-NGSI_Context_Management-V1_0-20100803-C.pdf (http://technical.openmobilealliance.org/Technical/release_program/docs/NGSI/V1_0-20101207-C/OMA-TS-NGSI_Context_Management-V1_0-20100803-C.pdf)**

Install Guide

- Installation and Administration Guide **https://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Publish/Subscribe_Broker_-_Orion_Context_Broker_-_Installation_and_Administration_Guide**

Tutorial

Example: Usage of PubSub in PilotC(V1)

1. Mapping of PilotC to PubSub

In PilotC a **resource management** **simulation model** creates a series of WorldStates showing **situation** envolvment.
For this new WorldStates the BasicIndicators are used to calculate the required indicators
All open user interfaces (WireCloud) need to be informed about the availibility of new WorldStates or **indicator** values to eventually update the display

PubSub uses two concepts to handle events: The "Context" and the "Attributes" within a context. Whenever an **attribute** changes an **event** is generated (There are more events and other things, but they are not used here).

The mapping of **CRISMA** needs to the Orion Context Broker is as follows:
WorldStates are mapped to contexts.
Information about new data (e.g. WorldState created, indicator value available) related to the WorldState are mapped to attributes.
Whenever a new WorldState is created in CRISMA (e.g. by the resource model) a new context with an attribute "creation" is created in Orion.

There is a listener for this to trigger the calculation of the BasicIndicators (via the BasicIndicators **WPS**)
The **GUI** listens for both to offer view updates.

Each context has an type (Always use "CRISMA.worldstates") and an id (An **ICMM** internal id is used). The attributes are as follows:

Description	Attribute type	Attribute name	Attrubute value
New WorldState is available	Type of Worldstate	worldstate or worldstate_Baseline	{"operation":"created", "time":1399545321,"URI":"**http**://crisma.**cismet**.de/pilotC/icmm_**api**/CRISMA.worldstates/248"}
Some data, e.g. indicator value, available	<empty>	dataslot_<name of dataslot>	{"operation":"created","time":1399545321,"URI":"**http://crisma.cismet.de/pilotC/icmm_api/CRISMA.dataitems/6114 (http://crisma.cismet.de/pilotC/icmm_api/CRISMA.dataitems/6114)** "}

The context is created implicit when setting the first attribute. This is needed to avoid mixing NGSI9 and NGSI10 API calls, which does not work as expected sinve version 0.9.0.

2. Access to Orion Context Broker

The Orion Context Broker offers a **REST** **interface** ad described in the USER GUIDE.
All operations accept **XML**, since version 0.9.0 most operations also accept **JSON** data.

There is a Orion instance available for PilotC at the endpoint *http://crisma.ait.ac.at/orion (http://crisma.ait.ac.at/orion/)*

For instances of the reference applications see the **reference application** documentation.

The usage of REST is possible by programming language dependent tools.

Downloads:

The source code of the Orion Broker is hosted on Github driven by **Telefonica**.

- Orion Github Repository

https://github.com/telefonicaid/fiware-orion/

CRISMA instance of the Orion Context Broker is hosted on **AIT** servers and can be reached at the following address. Currently the broker can be used by anyone who knows the IP address; access controll will be added at a later time.

FI-Ware public instance of the context broker is part of the Open Lab. **It** can be used for development, but features frequent downtimes (our experience, your mileage may vary).

- CRISMA instance of the Orion Context Broker
  http://crisma.ait.ac.at/orion/ (http://crisma.ait.ac.at/orion/)
- Test-Page
  http://crisma.ait.ac.at/PubSub/PubSubContent.html (http://crisma.ait.ac.at/PubSub/PubSubContent.html)

Docker image providing orion instances for **CRISMA**

There is no need to download the image, this will be done automatically when needed..

Usage:
docker run -P -d --name e_orion peterkutschera/crisma-orion

- Docker image source on github
  https://github.com/crismaproject/crisma-orion
- Docker image on hub.docker.com
  https://registry.hub.docker.com/u/peterkutschera/crisma-orion/

Team:

Peter Kutschera

Uses **CRISMA** Orion Broker in several services. Able to give feedback on modeling and integration.

Denis Havlik

Development roles: coordinaton
Strategic coordination within **AIT**, on **CRISMA** level and with the FI-Ware/FI-PPP

Hermann Huber

Development roles: administration, configuration, development

Responsible for maintaining the **CRISMA** instance. Able to help with data models and programming libraries.

Scenario Analysis and Comparison Widgets (Java)

This is the Java implementation of the **Scenario** Analysis and Comparison View. **It** provides a Table visualisation for both the indicators and the **criteria**. Moreover the **indicator** values are visualised via bar charts and the single indicators can be related to each other so that a graph visualises the correlation between the different criteria.

This analysis tool is available for critera as well. Additionally, criteria provide a spider chart visualisation that enables comparison of all criteria in a easy way. A **reference scenario** can be chosen to compare the individual worldstates against a reference scenario.

Scenario Comparision

Indicator Visualisation

Criteria Visualisation

Documentation:

Demo

This is a **demonstrator** video of the Scenario Analysis and Comparison **Widget** (Java) implementation. **It** contains no sound but subtitles which should be activated by default but can simply be switched on via the caption control of the youtube player. The video is available in up to FullHD resolution.

- Demonstrator Video for Scenario Analysis and DSS
  https://www.youtube.com/watch

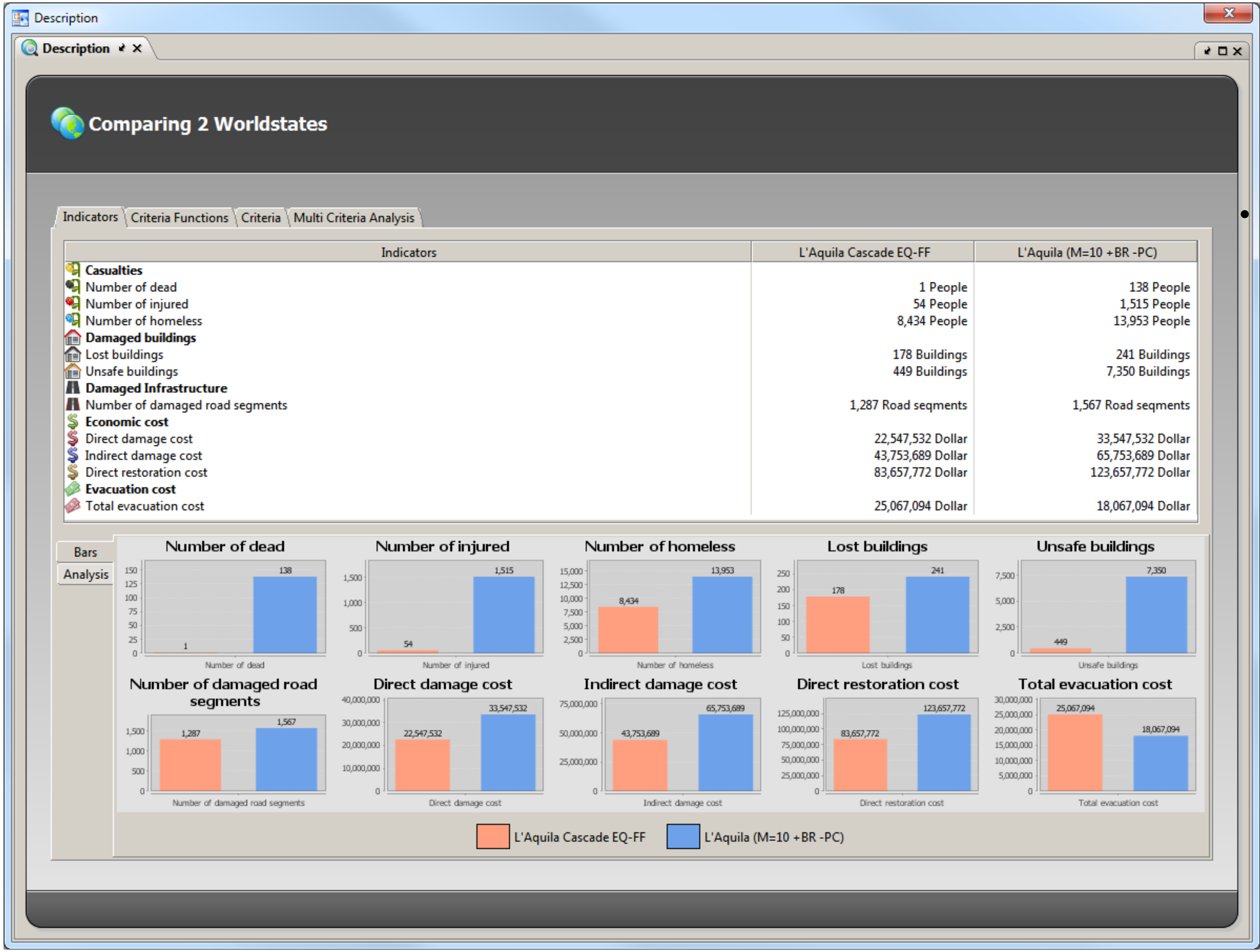
Functional Description

- Scenario Analysis and Comparison View Description
  https://crisma-cat.ait.ac.at/bbs/scenario-analysis-and-comparison-view

Downloads:

The cids-custom-**crisma** binary is available from the **cismet** repository

- Snaphots in the Maven Repository



https://repo.cismet.de/simple/cismet-libs-snapshots-local/de/cismet/cids/custom/crisma/

The source code is available from the github repository

- Sourcecode on GitHub **https://github.com/crismaproject/cids-custom-crisma**

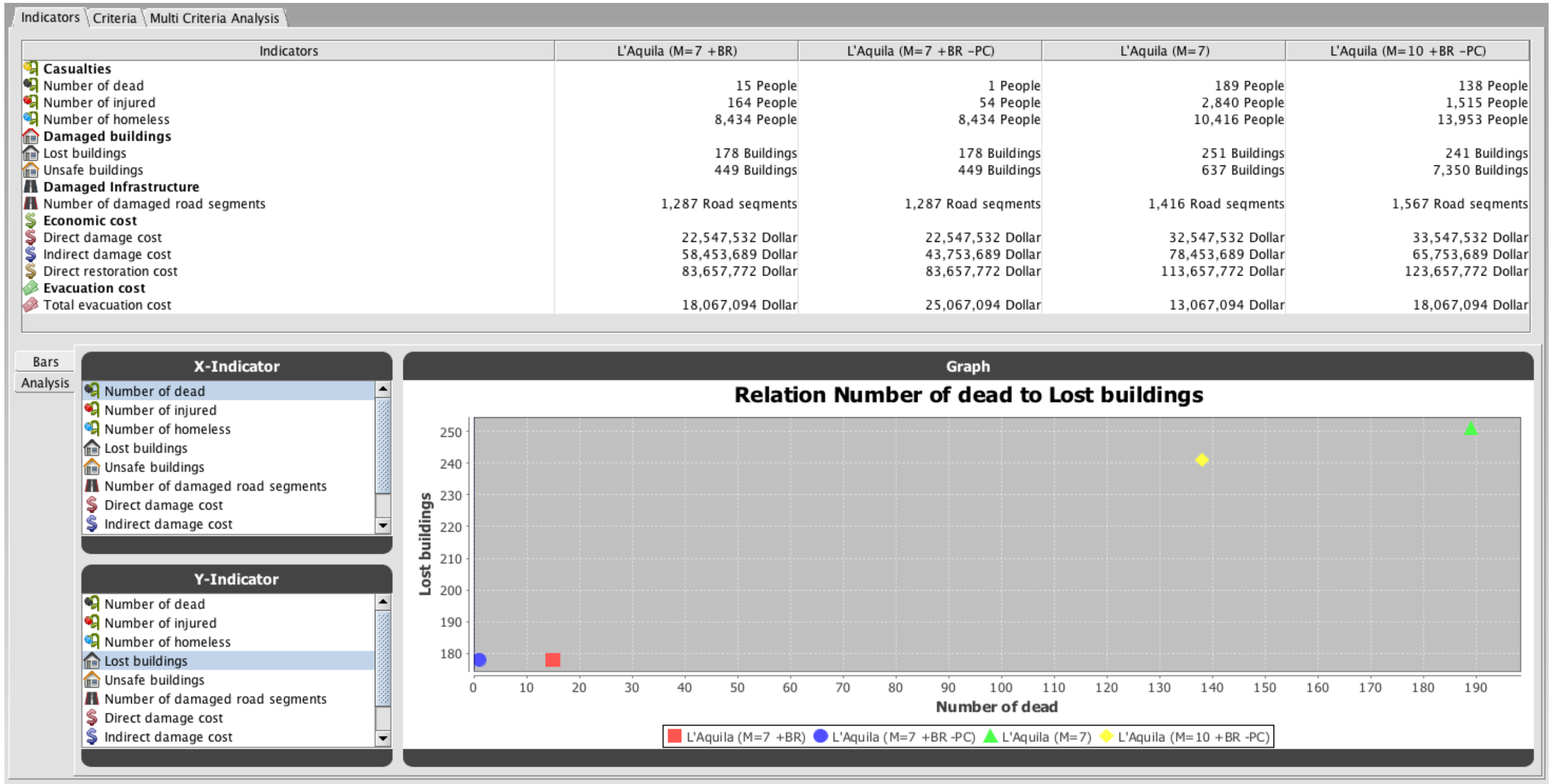
Team:

Martin Scholl

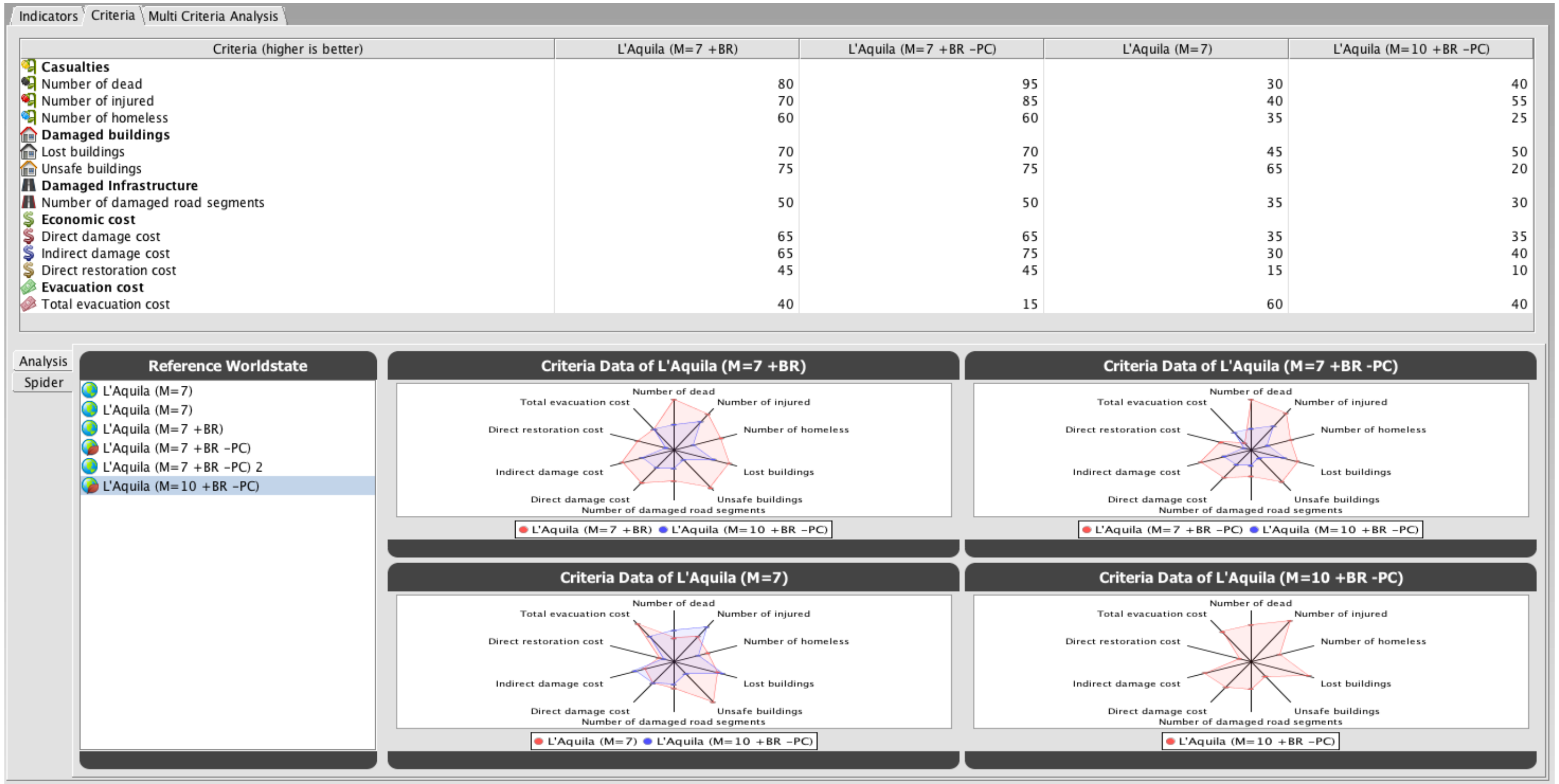
Development roles: configuration, development, installation, integration, support

Main **developer** of the **Widget**.

Scenario Analysis and Comparison Widgets (JavaScript)



Indicators	L'Aquila (M=7 +BR)	L'Aquila (M=7 +BR -PC)	L'Aquila (M=7)	L'Aquila (M=10 +BR -PC)
Casualties				
Number of dead	15 People	1 People	189 People	138 People
Number of injured	164 People	54 People	2,840 People	1,515 People
Number of homeless	8,434 People	8,434 People	10,416 People	13,953 People
Damaged buildings				
Lost buildings	178 Buildings	178 Buildings	251 Buildings	241 Buildings
Unsafe buildings	449 Buildings	449 Buildings	637 Buildings	7,350 Buildings
Damaged Infrastructure				
Number of damaged road segments	1,287 Road segments	1,287 Road segments	1,416 Road segments	1,567 Road segments
Economic cost				
Direct damage cost	22,547,532 Dollar	22,547,532 Dollar	32,547,532 Dollar	33,547,532 Dollar
Indirect damage cost	58,453,689 Dollar	43,753,689 Dollar	78,453,689 Dollar	65,753,689 Dollar
Direct restoration cost	83,657,772 Dollar	83,657,772 Dollar	113,657,772 Dollar	123,657,772 Dollar
Evacuation cost				
Total evacuation cost	18,067,094 Dollar	25,067,094 Dollar	13,067,094 Dollar	18,067,094 Dollar



Criteria (higher is better)	L'Aquila (M=7 +BR)	L'Aquila (M=7 +BR -PC)	L'Aquila (M=7)	L'Aquila (M=10 +BR -PC)
Casualties				
Number of dead	80	95	30	40
Number of injured	70	85	40	55
Number of homeless	60	60	35	25
Damaged buildings				
Lost buildings	70	70	45	50
Unsafe buildings	75	75	65	20
Damaged Infrastructure				
Number of damaged road segments	50	50	35	30
Economic cost				
Direct damage cost	65	65	35	35
Indirect damage cost	65	75	30	40
Direct restoration cost	45	45	15	10
Evacuation cost				
Total evacuation cost	40	15	60	40

Version: 1.0

This is the JavaScript implementation of the **Scenario** Analysis and Comparison View **Building Block**.

It provides a Table visualisation for both the indicators and the **criteria**. Moreover single indicators can be related to each other so that a scatter plot graph visualises the correlation between the different criteria. This analysis tool is available for criteria as well. Additionally, criteria provide a radar chart visualisation that enables comparison of all criteria in a easy way. A **reference scenario** can be chosen to compare the individual worldstates against a reference scenario.

Indicator and Criteria Table Widget

Indicator and Criteria Scatter Plot Widgets

Criteria Radar Chart Widget

Criteria Table   Change Mode

Criteria	L'Aquila (M=7 +BR)	L'Aquila (M=7)	L'Aquila (M=7) (edited)
Casualties			
Number of dead	80 Percent	90 Percent	54 Percent
Number of homeless	9 Percent	38 Percent	83 Percent
Number of injured	68 Percent	67 Percent	79 Percent
Economic cost			
Direct damage cost	76 Percent	86 Percent	53 Percent
Indirect damage cost	67 Percent	70 Percent	65 Percent
Direct restoration cost	47 Percent	5 Percent	65 Percent
Damaged buildings			
Lost buildings	73 Percent	7 Percent	1 Percent
Unsafe buildings	8 Percent	97 Percent	8 Percent
Damaged Infrastructure			
Number of damaged road segments	30 Percent	45 Percent	96 Percent
Evacuation cost			
Total evacuationcost	28 Percent	45 Percent	84 Percent

The Scenario Analysis and Comparison Widgets are new software components that are developed from scratch. They satisfy the technological requirements for new software developments as they are realized as Composite UI Module (HTML and JavaScript). Thus they can be used in any UI Integration Platform that supports HTML/CSS and JavaScript such as a Browser. Moreover, there is an additional build for the WireCloud platform effectively making this a Mashable UI Component.

Documentation:

Install Guide

See the GitHub projects for info.

- AngularJS implementation: Readme on GitHub **https://github.com/crismaproject/worldstate-analysis-widgets**
- WireCloud implementation: Readme on GitHub **https://github.com/crismaproject/worldstate-analysis-widgets-wirecloud**

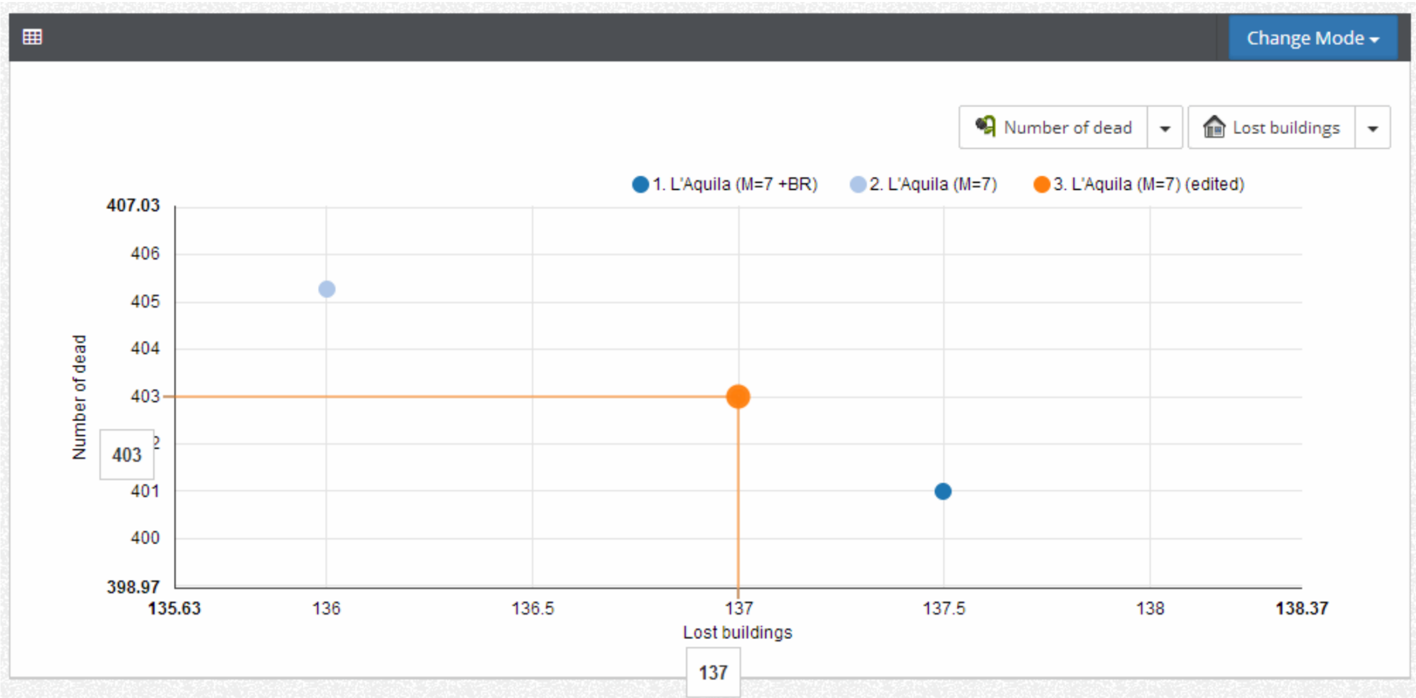
Developer's Guide

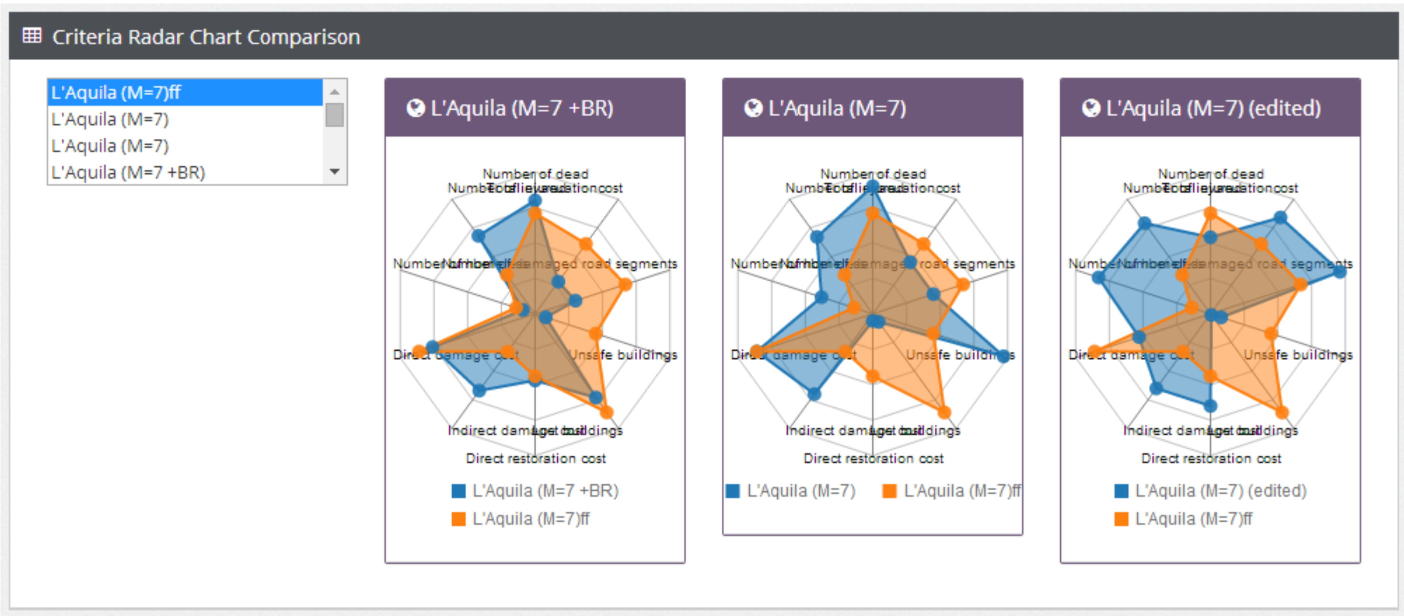
See the GitHub projects for info.

- AngularJS implementation: Readme on GitHub **https://github.com/crismaproject/worldstate-analysis-widgets**
- WireCloud implementation: Readme on GitHub **https://github.com/crismaproject/worldstate-analysis-widgets-wirecloud**
- Filling ICMM with compatible data **https://github.com/crismaproject/worldstate-analysis-widgets/wiki/HowTo:-Enable-MCA-and-DSS-in-your-CRISMA-application**

Functional Description

- Scenario Analysis and Comparison View Functional Description





https://crisma-cat.ait.ac.at/bb/Scenario-Analysis-and-Comparison-View

Demo

(http://youtu.be/J2ZEDAQf7A8)

This demo video shows how scenarios can be compared and analysed by their indicators.

FAQ

Issue Tracking and FAQ on GitHub

- AngularJS implementation: GitHub Issue Tracker **https://github.com/crismaproject/worldstate-analysis-widgets/issues**
- WireCloud implementation: GitHub Issue Tracker **https://github.com/crismaproject/worldstate-analysis-widgets-wirecloud/issues**

Tutorial

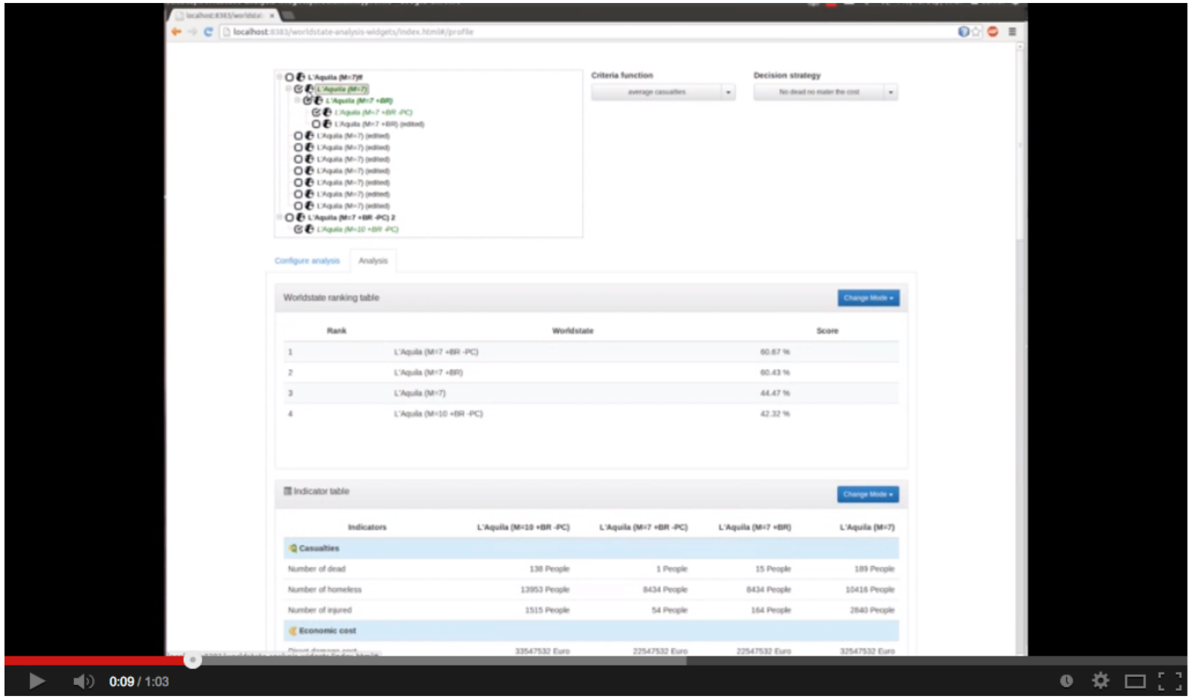
(http://youtu.be/niQC0RbEtrQ)

This is a HowTo video that showcases how the Worldstate WireCloud widgets can be wired together.

- HowTo: Wire Worldstate WireCloud Widgets **http://youtu.be/niQC0RbEtrQ (http://youtu.be/niQC0RbEtrQ)**

Downloads:

This component is divided into several AngularJS directives that can be put together individually



and as needed. The repository, however, contains a basic demo **application** that puts all available analysis and **DSS** components together in a meaningful way. Additionally, there is the repository that only contains wire code for WireCloud and thus effectively makes the analysis and DSS components available as mashable widgets.

- AngularJS components **https://github.com/crismaproject/worldstate-analysis-widgets**
- WireCloud component **https://github.com/crismaproject/worldstate-analysis-widgets-wirecloud**

  Ready-to-use build of the analysis and **DSS** mashable WireCloud **widget**. As soon as the **CRISMA** WireCloud platform marketplace is available the newest builds can be obtained from there. A minified version of the WireCloud widget will also be available in the future.

The AngularJS components can easily obtained via the dist folder of the project or via the bower dependency management tool

shell@computer:> bower install --save worldstate-analysis-widgets

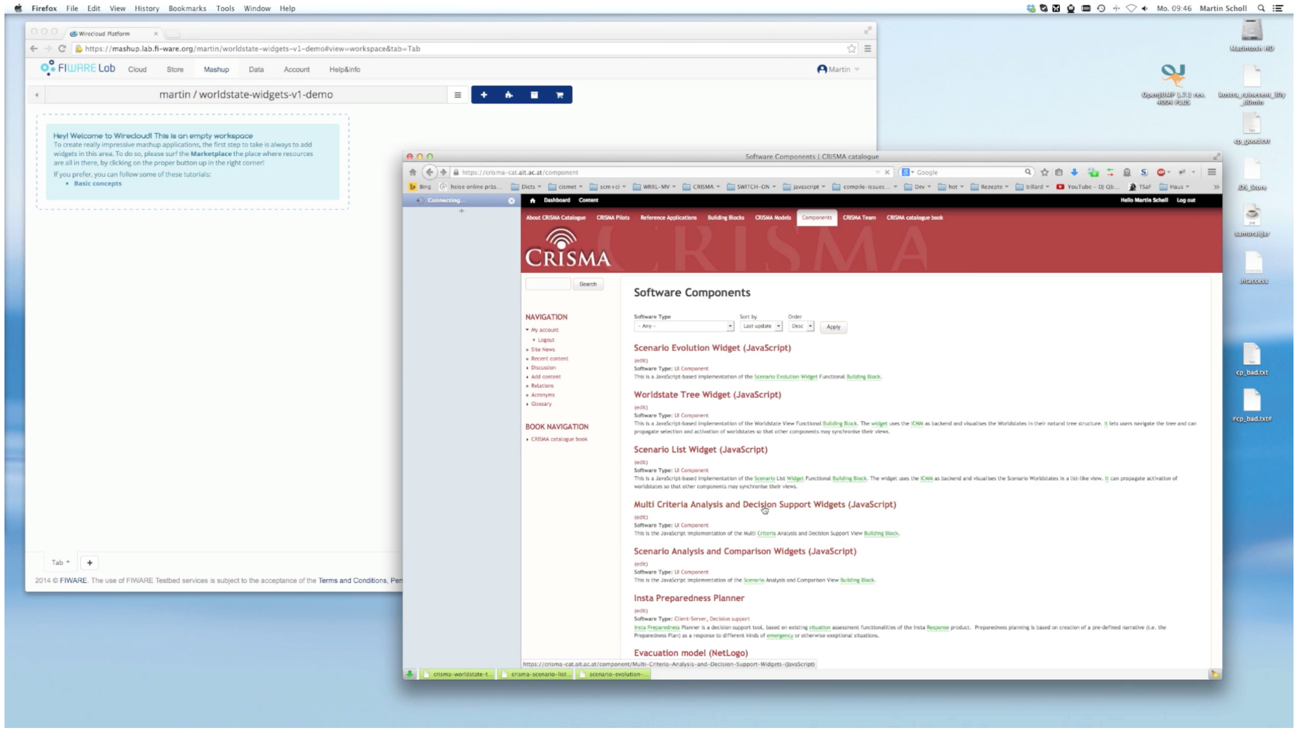
- AngularJS components **https://github.com/crismaproject/worldstate-analysis-widgets/tree/v1.0.7/dist**
- Mashable WireCloud widget (not minified) **http://crisma.cismet.de/lib/wirecloud/worldstate-analysis-widgets-wirecloud.wgt (http://crisma.cismet.de/lib/wirecloud/worldstate-analysis-widgets-wirecloud.wgt)**

Team:

Martin Scholl

Development roles: configuration, coordinaton

Coordinates development activities and provides general support.



Daniel Meiers

Development roles: development, installation, integration, support

Lead **developer** of this **widget** and also main contact point for any **Pilot** support regarding the embedding of this widget into **CRISMA** applications.

Scenario Evolution Widget (Java)

Version: 1.0

This is a java-based prototype of the **Scenario Evolution** **Widget** Functional **Building Block**.

The Scenario Evolution Java Widget is a new software component that is developed from scratch in phase one of WP34. In contrast to the technological requirements for new software developments that demands the implementation of new User Interaction Building Blocks as Composite UI Module (HTML and JavaScript), the Scenario Evolution Java Widget is developed in Java in the context of ongoing prototyping activities to be able to demonstrate the capabilities of the **CRISMA Framework** towards the end of the first implementation phase.

Scenario Evolution Widget Java Implementation

The Scenario Evolution Java Widget is implemented as cids navigator (UI Integration Platform) Renderer and Editor module which required considerable less development effort than the development of a HTML component.

The development of the Scenario Evolution Java Widget is finished. The Scenario Evolution Java Widget is replaced by a software component that satisfies all technological requirements of Mashable Composite UI Modules.

Documentation:

Demo

The demo Video of the CRISMA Technolog **Demonstrator** v1 shows, among others, the **Scenario Evolution** **Widget** in action.

- Technology Demo Video
  https://www.youtube.com/watch

Functional Description

- Worldstate View Building Block Description
  https://crisma-cat.ait.ac.at/bbs/worldstate-view

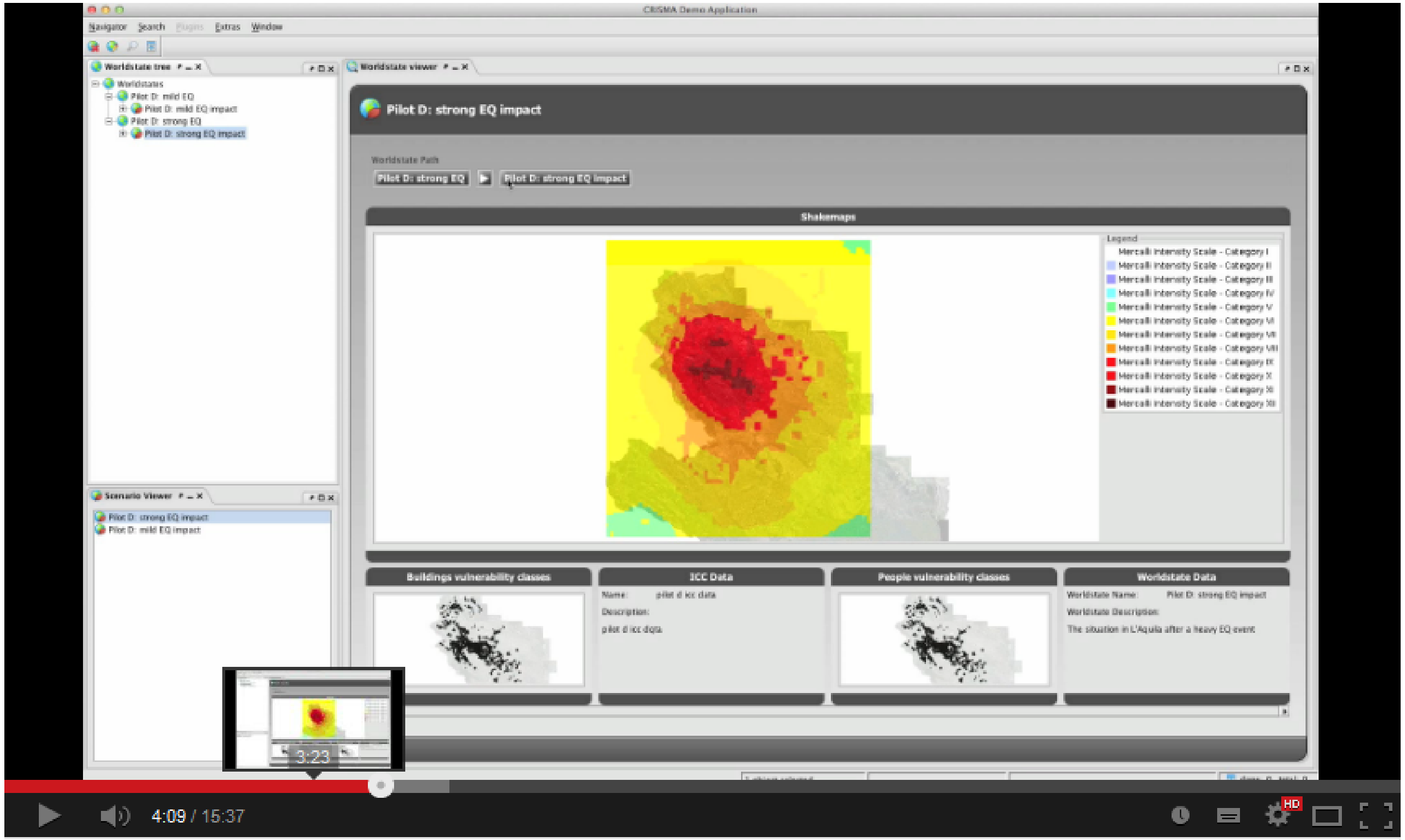
Downloads:

- Sourcecode on GitHub
  https://github.com/crismaproject/cids-custom-crisma
- Snapshots in the Maven Repository
  https://repo.cismet.de/simple/cismet-libs-snapshots-local/de/cismet/cids/custom/crisma/

Team:

Martin Scholl

Development roles: development, installation, integration, support
Main **developer**



Scenario Evolution Widget (JavaScript)

Version: 1.0

This is a JavaScript-based implementation of the Worldstate View Functional **Building Block**.

The **widget** uses the **ICMM** as back end and visualises the path of worldstates according to the Core **CCIM** of the **CRISMA Framework** from the currently selected one to the root of the worldstate tree. **It** can propagate selection of worldstates so that other components may synchronise their views.

Scenario Evolution Widget (JavaScript)

The Scenario Evolution JavaScript Widget is a new software component that is developed from scratch. It satisfies the technological requirements for new software developments as it is realized as Composite UI Module (HTML and JavaScript). Thus it can be used in any UI Integration Platform that supports HTML/CSS and JavaScript such as a Browser. Moreover, there is an additional build for the WireCloud platform effectively making it a Mashable UI Component.

Documentation:

Install Guide

See the GitHub projects for info.

- AngularJS implementation: Readme on GitHub
  https://github.com/crismaproject/scenario-evolution-widget-angular
- WireCloud implementation: Readme on GitHub
  https://github.com/crismaproject/scenario-evolution-widget-wirecloud

Developer's Guide

See the GitHub projects for info.

- AngularJS implementation: Readme on GitHub
  https://github.com/crismaproject/scenario-evolution-widget-angular
- WireCloud implementation: Readme on GitHub
  https://github.com/crismaproject/scenario-evolution-widget-wirecloud

Functional Description

- Worldstate View Building Block Description
  https://crisma-cat.ait.ac.at/bbs/worldstate-view

Demo

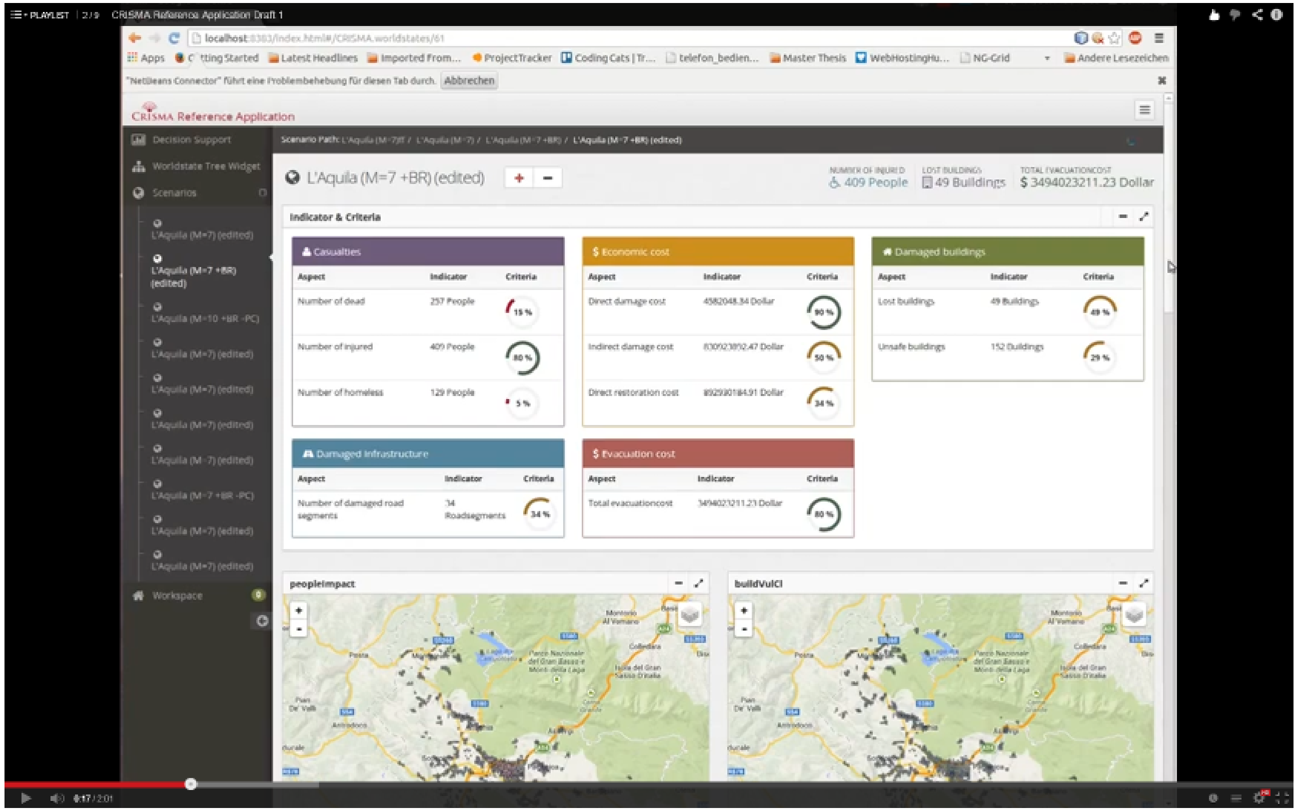
This demo Video of the CRISMA Reference Application for the Earthquake and Forest Fire Domains shows, among others, the **Scenario Evolution** **Widget** in action.

- Reference Application Demo Video
  https://www.youtube.com/watch

FAQ

Issue Tracking and FAQ on GitHub

- AngularJS implementation: GitHub Issue Tracker
  https://github.com/crismaproject/scenario-evolution-widget-angular/issues
- WireCloud implementation: GitHub Issue Tracker



https://github.com/crismaproject/scenario-evolution-widget-wirecloud/issues

Tutorial

(http://youtu.be/niQC0RbEtrQ)

This is a HowTo video that showcases how the Worldstate WireCloud widgets can be wired together.

- HowTo: Wire Worldstate WireCloud Widgets **http://youtu.be/niQC0RbEtrQ (http://youtu.be/niQC0RbEtrQ)**

Downloads:

This component is provided in form of a AngularJS directive that can be easily embedded and configured as needed. The repository, however, contains a basic demo **application** that shows the capabilities of the component. Additionally, there is the repository that only contains wire code for WireCloud and thus effectively makes the tree component available as mashable **widget**.

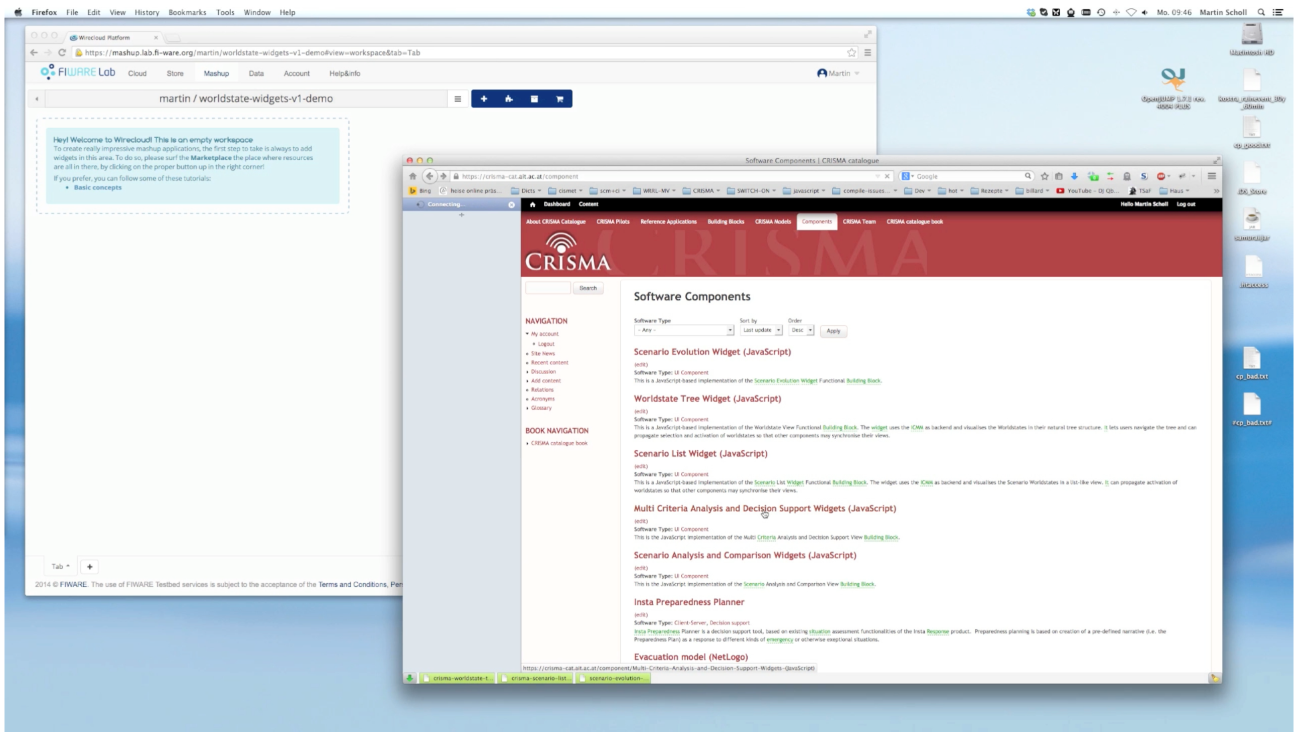
- AngularJS component **https://github.com/crismaproject/scenario-evolution-widget-angular**
- WireCloud component **https://github.com/crismaproject/scenario-evolution-widget-wirecloud**

Ready-to-use build of the evolution WireCloud **widget**. As soon as the **CRISMA** WireCloud platform marketplace is available the newest builds can be obtained from there. A minified version of the WireCloud widget will also be available in the future.

The AngularJS components can easily obtained via the dist folder of the project or via the bower dependency management tool

shell@computer:> bower install --save worldstate-tree-widget-angular

- AngularJS component



https://github.com/crismaproject/scenario-evolution-widget-angular/tree/v1.1.3/dist

- Mashable WireCloud widget (not minified) **http://crisma.cismet.de/lib/wirecloud/scenario-evolution-widget-wirecloud.wgt (http://crisma.cismet.de/lib/wirecloud/scenario-evolution-widget-wirecloud.wgt)**

Team:

Martin Scholl

Lead **developer** of this **widget** and also main contact point for any **Pilot** support regarding the embedding of this widget into **CRISMA** applications.

Manuel Warum

Main contact point for all questions related to Wirecloud and development of the Wirecloud widgets, connectors and mashup applicaitons within **CRISMA**. He is also the author of **AIT**'s Wirecloud widgets and mash-up applications used in "**Resource Management** Training" **technology demonstrator**.

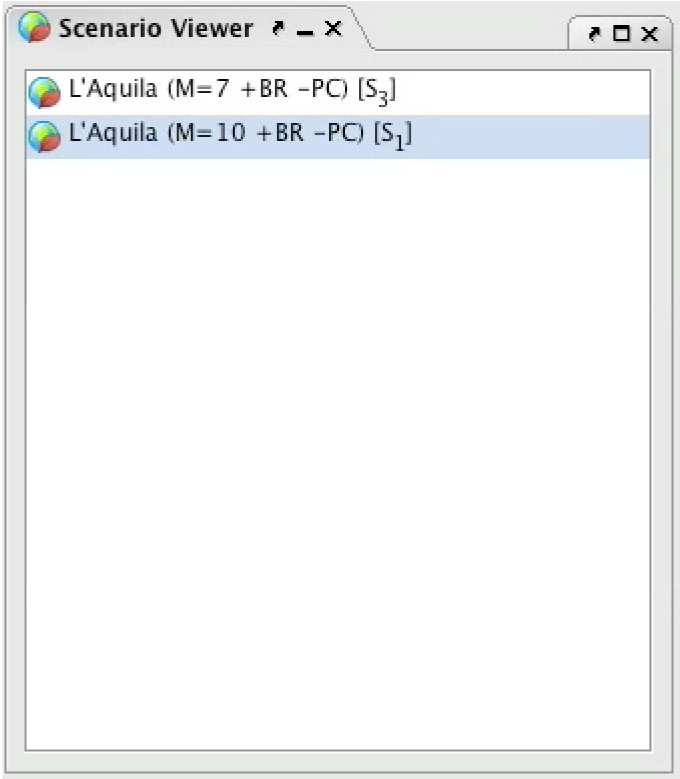
Scenario List Widget (Java)

This is a java-based prototype of the **Scenario** List **Widget**.

The Scenario List Java Widget is a new software component that is developed from scratch in phase one of WP34. In contrast to the technological requirements for new software developments that demands the implementation of new User Interaction Building Blocks as Composite UI Module (HTML and JavaScript), the Scenario List Java Widget is developed in Java in the context of ongoing prototyping activities to be able to demonstrate the capabilities of the **CRISMA Framework** towards the end of the first implementation phase.

Scenario List Widget Java Implementation

The Scenario List Java Widget is implemented as cids navigator (UI Integration Platform) Renderer and Editor module which requires considerable less development

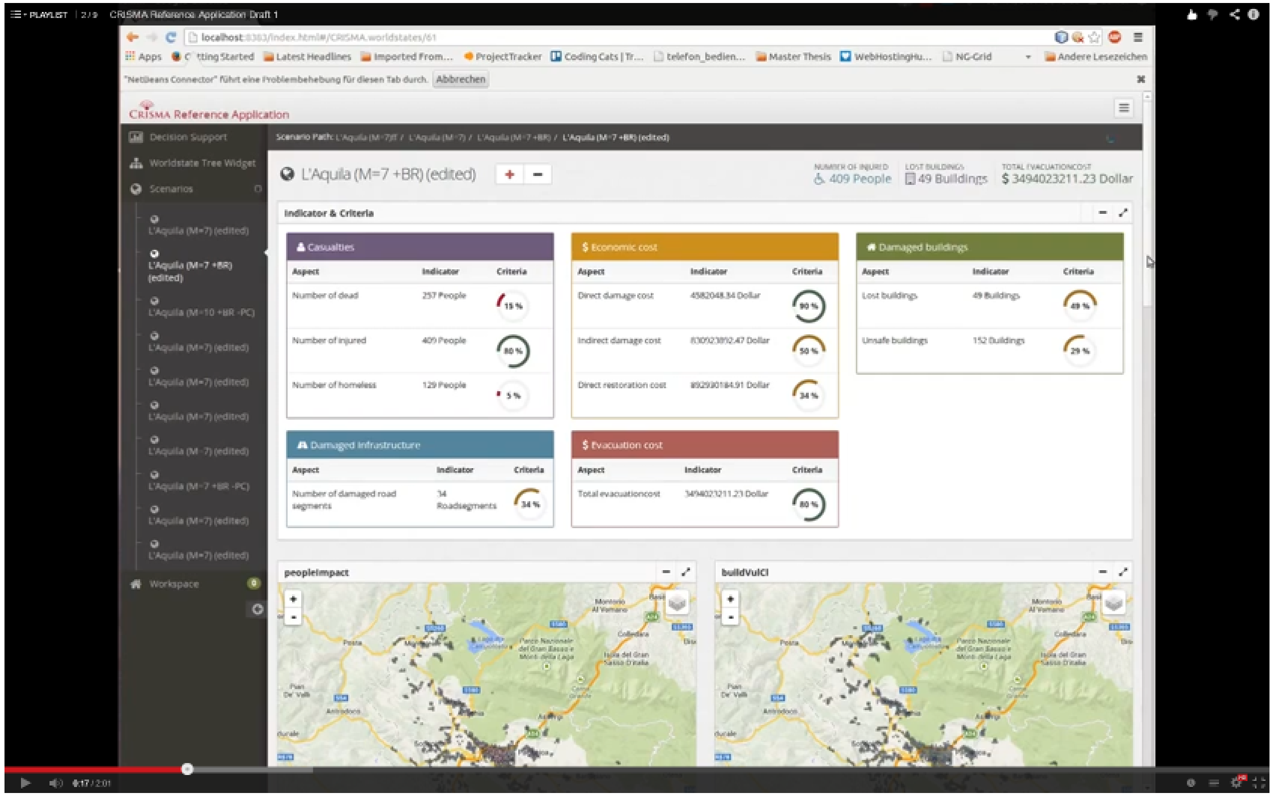


effort than the development of a HTML component.

The development of the java-based Scenario List Java Widget is finished. The Scenario List Java Widget is replaced by a software component that satisfies all technological requirements of Mashable Composite UI Modules.

Documentation:

Demo



The demo Video of the CRISMA Technolog Demonstrator v1 shows, among others, the **Scenario** List **Widget** in action.

- Technology Demonstrator Application Demo Video

https://www.youtube.com/watch

Functional Description

- Worldstate View Building Block Description
  https://crisma-cat.ait.ac.at/bbs/worldstate-view

Downloads:

- Sourcecode on GitHub
  https://github.com/crismaproject/cids-custom-crisma
- Snaphots in the Maven Repository
  https://repo.cismet.de/simple/cismet-libs-snapshots-local/de/cismet/cids/custom/crisma/

Team:

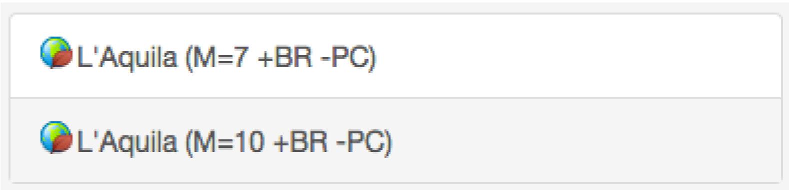
Martin Scholl

Development roles: development, installation, integration
Main **developer** of the **Widget**.

Scenario List Widget (JavaScript)

Version: 1.0

This is a JavaScript-based implementation of the Worldstate View Functional **Building Block**. The **widget** uses the **ICMM** as backend and visualises the **Scenario** Worldstates in a list-like view. **It** can propagate activation of worldstates so that other components may synchronise their views.



Scenario List Widget (JavaScript)

The Scenario List JavaScript Widget is a new software component that is developed from scratch. It satisfies the technological requirements for new software developments as it is realized as Composite UI Module (HTML and JavaScript). Thus it can be used in any UI Integration Platform that supports HTML/CSS and JavaScript such as a Browser. Moreover, there is an additional build for the WireCloud platform effectively making it a Mashable UI Component.

Documentation:

Install Guide

See the GitHub projects for info.

- AngularJS implementation: Readme on GitHub
  https://github.com/crismaproject/scenario-list-widget-angular
- WireCloud implementation: Readme on GitHub
  https://github.com/crismaproject/scenario-list-widget-wirecloud

Developer's Guide

See the GitHub projects for info.

- AngularJS implementation: Readme on GitHub
  https://github.com/crismaproject/scenario-list-widget-angular
- WireCloud implementation: Readme on GitHub
  https://github.com/crismaproject/scenario-list-widget-wirecloud

Functional Description

- Worldstate View Building Block Description
  https://crisma-cat.ait.ac.at/bbs/worldstate-view

Demo

This demo Video of the CRISMA Reference Application for the Earthquake and Forest Fire Domains shows, among others, the **Scenario** List **Widget** (JavaScript) in action.

FAQ

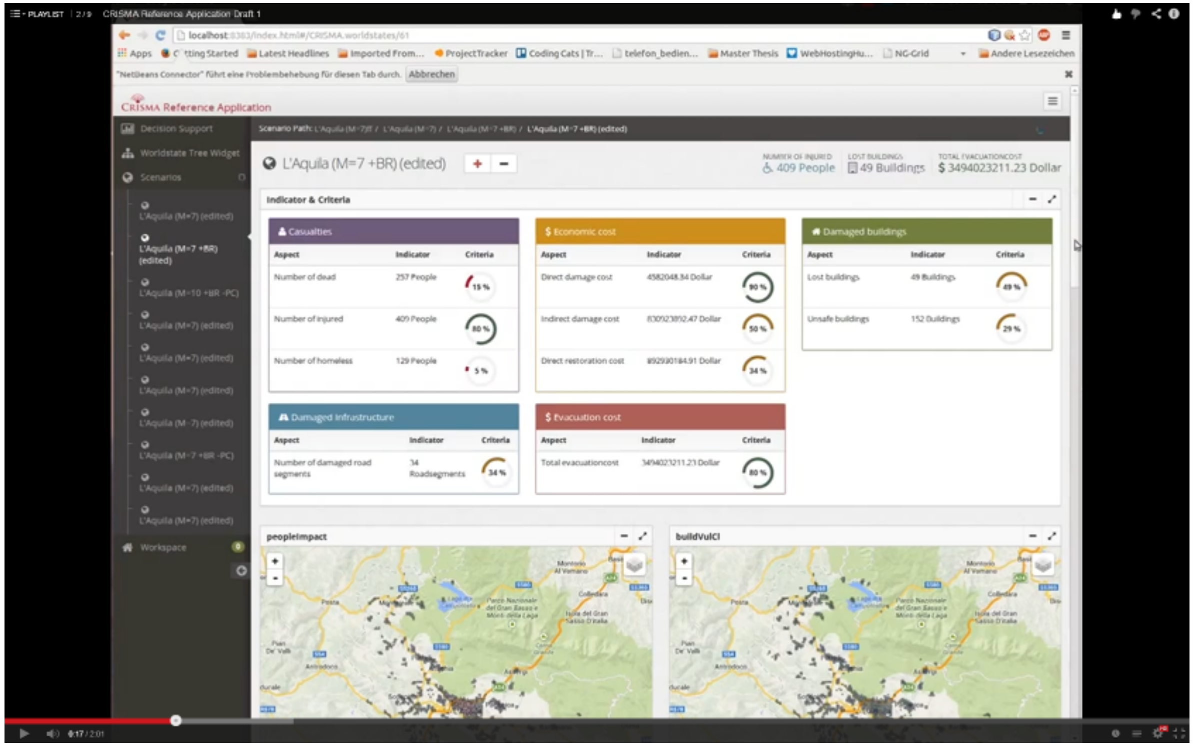
Issue Tracking and FAQ on GitHub

- AngularJS implementation: GitHub Issue Tracker
  https://github.com/crismaproject/scenario-list-widget-angular/issues
- WireCloud implementation: GitHub Issue Tracker
  https://github.com/crismaproject/scenario-list-widget-wirecloud/issues

Tutorial

(http://youtu.be/niQC0RbEtrQ)

This is a HowTo video that showcases how the Worldstate WireCloud widgets can be wired together.



- HowTo: Wire Worldstate WireCloud Widgets **http://youtu.be/niQC0RbEtrQ (http://youtu.be/niQC0RbEtrQ)**

Downloads:

This component is provided in form of a AngularJS directive that can be easily embedded and configured as needed. The repository, however, contains a basic demo **application** that shows the capabilities of the component. Additionally, there is the repository that only contains wire code for WireCloud and thus effectively makes the list component available as mashable **widget**.

- AngularJS component **https://github.com/crismaproject/scenario-list-widget-angular**
- WireCloud component **https://github.com/crismaproject/scenario-list-widget-wirecloud**

  Ready-to-use build of the list WireCloud **widget**. As soon as the **CRISMA** WireCloud platform marketplace is available the newest builds can be obtained from there. A minified version of the WireCloud widget will also be available in the future.

The AngularJS components can easily obtained via the dist folder of the project or via the bower dependency management tool

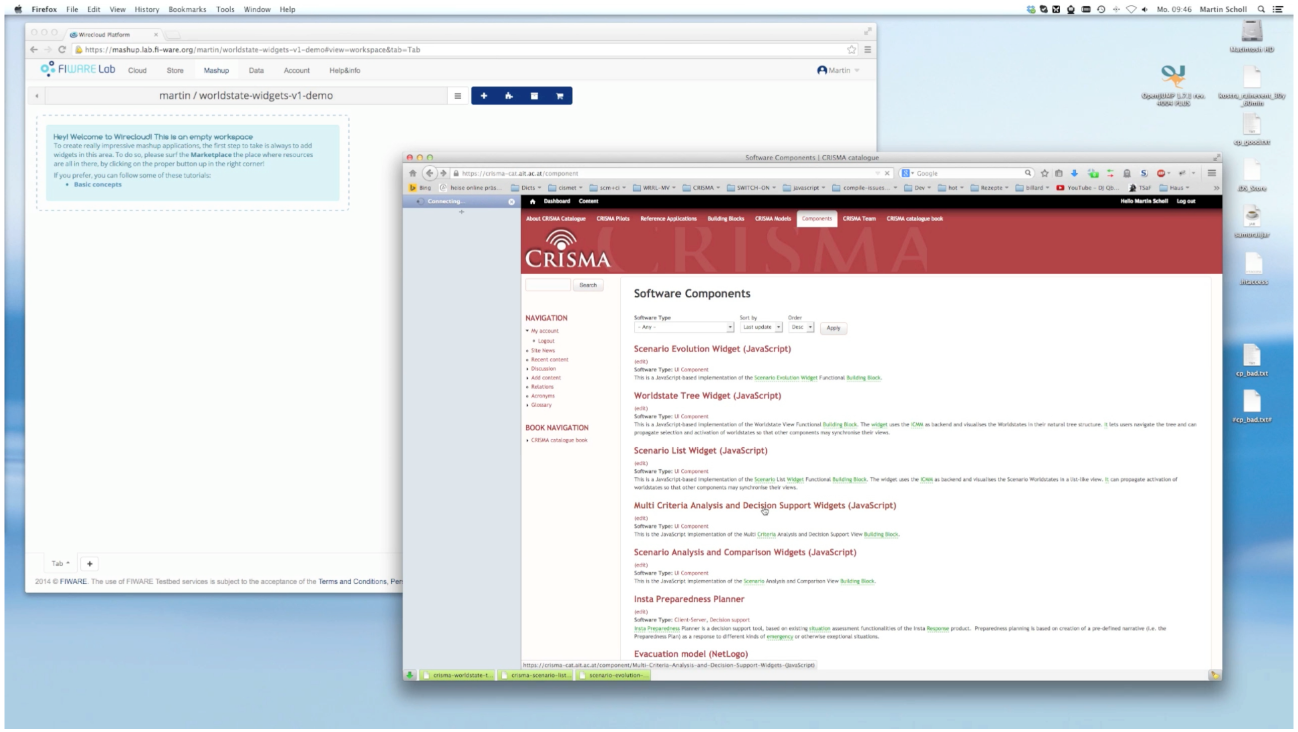
shell@computer:> bower install --save **scenario**-list-widget-angular

- AngularJS component **https://github.com/crismaproject/scenario-list-widget-angular/tree/v1.1.2/dist**
- Mashable WireCloud widget (not minified) **http://crisma.cismet.de/lib/wirecloud/crisma-scenario-list-widget-wirecloud.wgt (http://crisma.cismet.de/lib/wirecloud/crisma-scenario-list-widget-wirecloud.wgt)**

Team:

Martin Scholl

Development roles: coordinaton, development, installation, integration, support



Lead **developer** of this **widget** and also main contact point for any **Pilot** support regarding the embedding of this widget into **CRISMA** applications.

Manuel Warum

Development roles: support
Main contact point for all questions related to Wirecloud and development of the Wirecloud widgets, connectors and mashup applicaitons within **CRISMA**. He is also the author of **AIT**'s Wirecloud widgets and mash-up applications used in "**Resource Management** Training" **technology demonstrator**.

Daniel Meiers

Development roles: development, integration, support
Provides integration support.

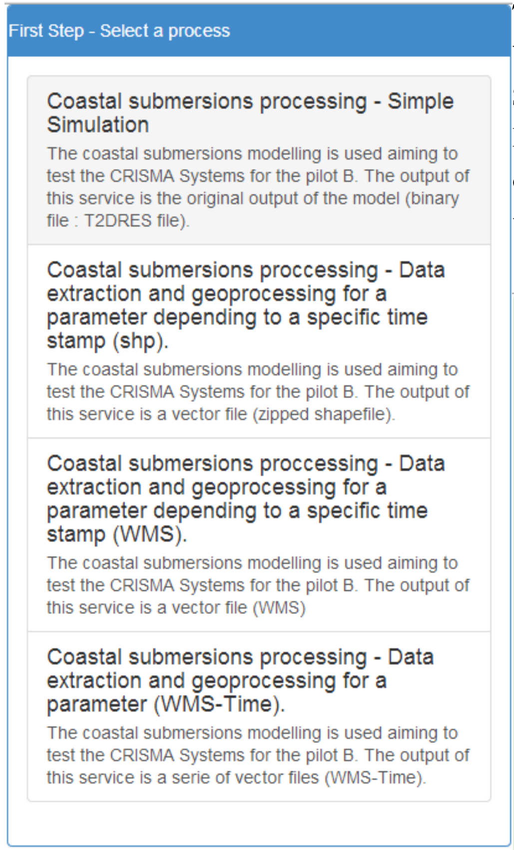
Simulation Model Interaction Widget

The **Simulation Model** Interaction view is a generic **Web Processing Service** client. **It** generates an user-friendly **interface** aiming to discover, describe and run processes issued from **WPS** instances. Thanks to this **widget**, the user can easily interact with the WPS server and the processes. Several models used in the **Crisma** project (coastal submersion, FireStation, Time-dependant model,...) and wrapped in WPS are available towards this widget.

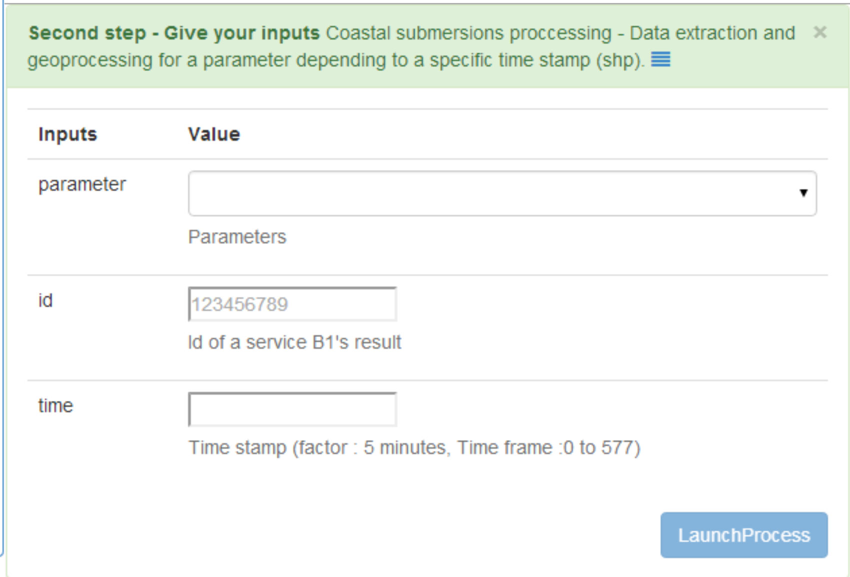
The Simulation Model Interaction View consists of the following four generic Widgets:

1. Simulation Model List Widget
2. Simulation Model Configuration Widget
3. Simulation Model monitoring widget
4. Simulation Model Result Widget

The **Simulation Model List Widget** is a Mashable Composite UI Module of the Simulation interaction **Building block** that lists all of the available processes. This widget is generic and based on WPS GetCapabilities request. This widget displays the following information extracted from Getcapabilities : the title and the abstract of each process referenced on the WPS server.



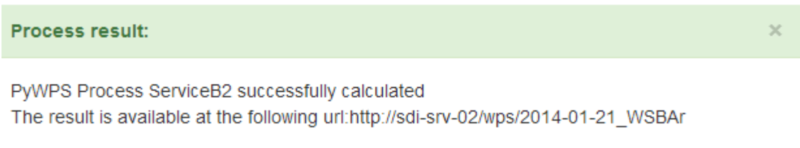
The **Simulation Model Configuration Widget** is a Mashable Composite UI Module of the Simulation interaction Building block that provides input parameters descriptions, input forms. This widget allows also users to run simulations. As with the Simulation Model List Widget, this widget is generic thanks to, in this case, the use of the WPS DescribeProcess request. Indeed, information listed in this widget is directly extracted from this request: titles assigned to each input, input description, allowed values of inputs, default value of inputs… The information included in this widget depends directly of the completeness of the WPS.



The **Simulation Model monitoring widget** is a Mashable Composite UI Module of the Simulation interaction Building block that provides information regarding to the status of the simulation. This widget depends directly on the configuration of the WPS. If the status **parameter** is set, the Simulation Model monitoring widget can poll the process status aiming to monitor progress.

The **Simulation Model Result Widget** is a Mashable Composite UI Module of the Simulation interaction Building block that provides information regarding to the result of the simulation.

Documentation:



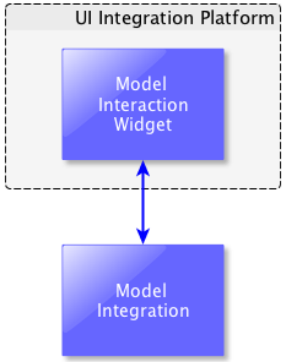
Demo

Aiming to discover the **Simulation Model** Interaction **Widget**, please visit the following demo site. Log in with the following predefined user: demo/smi

- Simulation Model Interaction Widget - Demonstrator **http://crisma.spacebel.be/smidemo (http://crisma.spacebel.be/smidemo)**

Specifications

The **Simulation Model** Interaction **Widget** **Building Block** is a Composite UI Module (Figure 1) that lets end users interact with the various simulation models exposed by a Simulation Model Integration Building Block. **It** is responsible for collecting simulation model parameters, launching the respective simulation model, showing status information and retrieving simulation model results. It also provides a **GUI** for the Simulation Model Integration Building Block.



Since the current model integration approach described in the **CRISMA Framework** **Architecture** is based on **OGC** **Web Processing Service**, the Simulation Model Interaction Widget is essentially a CRISMA-aware **WPS** client.

The Simulation Model Interaction Widget Building Block must be able to configure, run and monitor a simulation given a Worldstates and a SimulationDescriptors **entity** as input (see CRISMA Framework Core **CCIM**). Thus, the Simulation Model Interaction Widget should be split into a generic **business logic** part and a configurable user **interface** part. Thereby the generic business logic part should manage the communication with the **ICMM** Worldstates Repository and the Simulation Model Integration Building Block (OGC WPS) and should be the same for all types of simulation models. The user interface part may be different for each type of simulation model depending on the configuration and parameterization options of the simulation model. The business logic part should be realised as extended ICMM **API**.

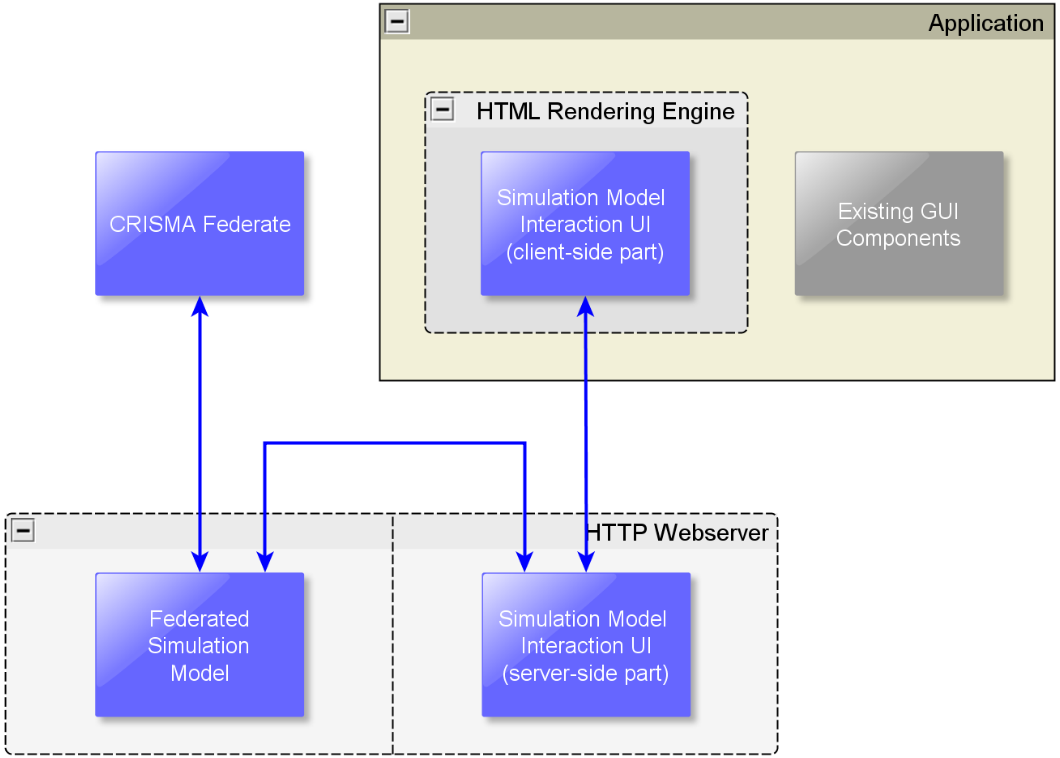
Rationale:

End users need to be able to interact with the various CRISMA simulation models.

Key Functionality:

The key functionalities of the Simulation Model Integration Building Block are defined by the following functional requirements.

- The Simulation Model Interaction Widget shall be able to show to the user a list of all supports simulations that can be applied to the current worldstate.
- The Simulation Model Interaction Widget model shall be able to show to the user information like general model descriptions and required and optional model parameters.
- The Simulation Model Interaction Widget model shall provide a per-model configurable user interface (e.g. wizard, forms, …) to collect simulation model input and control parameters.
- The Simulation Model Interaction Widget shall be able to initialize the model with model input data (i.e. the currently selected Worldstate data) by uploading the data to a Data Integration Building Block (e.g. OGC **SOS**) or an integrated data store (e.g. **FTP** or **WebDAV** Server) - We distinguish between simulation model parameters and simulation model input data. (Simple, small) parameters are provided by the user directly while (huge, complex) input data has to be uploaded to some service.
- The Simulation Model Interaction Widget shall be able to initiate a new simulation model run.
- The Simulation Model Interaction Widget shall be able to monitor the **simulation run**, show status information to the user as well as publish status information



to the ICMM Repository (Transitions status).

- The Simulation Model Interaction Widget shall be able to collect and store model results (i.e. create a new Worldstate including the results and the selected model control parameters).
- The Simulation Model Interaction Widget should be split into a generic business logic and a configurable user interface part.
- The generic business logic part of the Simulation Model Interaction Widget should manage the communication with the CRISMA Middleware (ICMM) and the Simulation Model Integration Building Block (OGC WPS) and should be the same for all types of simulation models. It should be realised as extended ICMM API.
- The user interface part of the Simulation Model Interaction Widget shall manage the configuration and parameterization options of the individual simulation models.
- The Simulation Model Interaction Widget shall provide convenient configuration and extension possibilities that leverage easy integration of many different types of simulation models.

Downloads:

The source code of **Simulation Model** Interaction **Widget** will be available on the **crisma**'s gihub (September 2014).

In the meantime, if you will **test** your **WPS** processes with the Simulation Model Interaction Widget, **Spacebel** can create specific instance for **it** (see documentation). Currently, several WPS clients have been deployed for several partners (**Artelia**, **AMRA** and **ADAI**).

Team:

Arnaud De Groof

Development roles: development

Responsible for development of the **Simulation Model** Interaction **Widget**.

TDV Python package

Collection of modules written in Python **object**-oriented programming language.

Documentation:

Presentation

TDV Python package: a collection of modules written in Python **object**-oriented programming language.
It implements the Time Dependant phisical **Vulnerability** modeling based on the logic working scheme and description of TDV.
Integrates with the Models **Interface** Package (**AMRA**) for accessing models as **WPS** processes.
Interfaces with **Simulation Model** Integration and Simulation Model Interaction **Widget** BBs.

Downloads:

Team:

Salvatore Larosa

Development roles: architecture, configuration, development, installation, integration, support, specifications, testing, validation

Stefano Nardone

Development roles: development, specifications

TELEMAC MASCARET System

TELEMAC-MASCARET is an integrated suite of solvers for use in the field of free-surface flow. Having been used in the context of many studies throughout the **world**, **it** has become one of the major standards in its field. TELEMAC-MASCARET is managed by a consortium of core organisations: **Artelia** (formerly Sogreah, France), BundesAnstalt für Wasserbau (BAW, Germany), Centre d'Etudes Techniques Maritimes et Fluviales (CETMEF, France), Daresbury Laboratory (United Kingdom), Electricité de France R&D (EDF, France), and **HR Wallingford** (United Kingdom).

The TELEMAC-MASCARET **system** is a powerful tool integrated by modelling of free-surface flow.
Having been used in the context of very many studies throughout the world, it has become one of the major standards in its field.
The various **simulation** modules use high-**capacity** algorithms based on the finite-element method. Space is discretised in the form of an unstructured grid of triangular elements, which means that it can be refined particularly in areas of special interest. This avoids the need for systematic use of embedded models, as is the case with the finite-difference method.
All the numerical algorithms are gathered into a single library that is shared by all the simulation modules. Thefore, it makes the TELEMAC-MASCARET system consistent.
TELEMAC-MASCARET has numerous applications in both river and maritime hydraulics.

Documentation:

Presentation

The full presentation of the TELEMAC MASCARET **system** is available on the Open TELEMAC MASCARET website.

- Presentation of TELEMAC MASCARET System
  http://www.opentelemac.org/index.php/presentation (http://www.opentelemac.org/index.php/presentation)

Install Guide

The install guide is available on the Open TELEMAC MASCARET website.

- Install guide for TELEMAC MASCARET System
  http://www.opentelemac.org/index.php/installation (http://www.opentelemac.org/index.php/installation)

Downloads:

TELEMAC MASCARET pre-compiled binaries for Windows and Linux can be downloaded from the opentelemac web site. The easiest way to install the program is with the automatic installer (windows only for now).

- TELEMAC MASCARET binary downloads
  http://www.opentelemac.org/index.php/binaries (http://www.opentelemac.org/index.php/binaries)

TELEMAC MASCARET source code is available from the SVN repository, as well as in the form of tarballs for windows and (opensuse) linux.

- TELEMAC MASCARET tarballs
  http://www.opentelemac.org/index.php/binaries/viewcategory/39-manual-installation-sources (http://www.opentelemac.org/index.php/binaries/viewcategory/39-manual-installation-sources)
- TELEMAC MASCARET SVN instructions
  http://www.opentelemac.org/index.php/sources-svn (http://www.opentelemac.org/index.php/sources-svn)
- TELEMAC MASCARET Licenseing conditions
  http://www.opentelemac.org/index.php (http://www.opentelemac.org/index.php?option=com_content&view=article&id=80&Itemid=48&lang=en)

Team:

Mehdi Pierre Daou

Development roles: modelling

Mehdi uses the TELEMAC MASCARET **System** to develop the coastal submersion **model**.

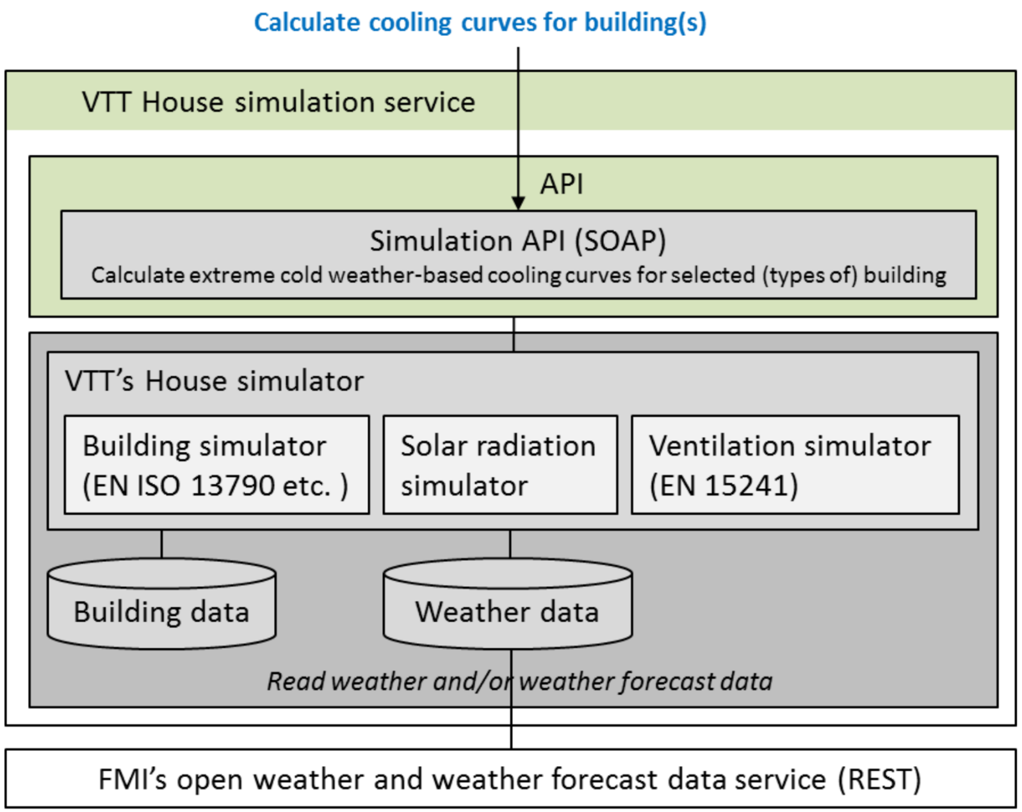
VTT House service

Authors: **Markus Jähi**

Web **service** to implement **VTT** House building **simulation model**.

VTT House Service is a web service (**SOAP**) based simulation service for calculating extreme cold weather based cooling curves of different types of buildings. The model is based on EN ISO 13790 and EN 15241 standards in addition to models for estimating solar radiation. The model includes methods for a dynamic hourly-based calculation of building energy and thermal performance, including heating and cooling and airflow related energy **losses** due to the ventilation **system** and infiltration. In addition the model can be used to predict the speed of temperature **recovery** when heating is restored.

VTT House simulation service exposes the SOAP based web service **API** as follows:



Documentation:

Downloads:

Team:

Markus Jähi

Jussi Yliaho

Wirecloud Application Mashup Platform

Authors: **Denis Havlik**
Manuel Warum
Peter Kutschera

Version: 1.1.0

Wirecloud is a reference implementation of the **FI-Ware (http://www.fi-ware.eu/)** (Web) **Application** Mashup Generic Enabler. **It** allows users with no programmers experience to rapidly build web applications by "wiring" the available widgets and operators and positioning them on a screen. In **CRISMA**, the Wirecloud is used as a reference implementation of the **UI Mashup Platform** **Building Block** and as a platform for development of the CRISMA web widgets and Views (Mashup applicaitons) based on these widgets.

Currently available CRISMA-specific widgets and operators for this platform:

- **Charts**
- **OOI** Commands
- **OOI GIS Map**
- **OOI Table**
- **Simulation** Picker
- **Worldstate Loader** (to be replaced by an interactive **Worldstate Picker** in future iterations)
- Worldstate Saver

Documentation:

Specifications

Wirecloud is a reference implementation of the FI-Ware "**Application** Mashup" Generic Enabler (GE). Full specifications of this GE is available on the FI-Ware Wiki and linked below.

- FI-WARE Application Mashup GE specifications
  https://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Apps.ApplicationMashup

Developer's Guide

Wirecloud web site offers the "**Developer**'s guide", and the FI-Ware wiki also provides a document called "User and Programmer Guide".

Both documents target end users interested in using Wirecloud to compose their own customized **application** mashups from gadgets, operators and mashuplets available in the marketplace (e.g. to create a customized configuration, control, or administration cockpit), as well as the developers interested in programming new gadgets and operators and making them available through the marketplace.

- Wirecloud Developer's guide
  http://conwet.fi.upm.es/wirecloud/developer (http://conwet.fi.upm.es/wirecloud/developer)
- Wirecloud Platform User and Programmer Guide (FI-Ware version)
  https://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Composition_Editor_-_Wirecloud_Mashup_Platform_-_User_and_Programmer_Guide

Install Guide

Wirecloud Platform Installation and Administration Guide is targeted at technical users and administrators responsible of installing and administrating the Wirecloud platform. (Wirecloud version is probably more up to date than the version offered on the FI-Ware wiki.)

- Wirecloud Installation and Administration Guide (Wirecloud version) **http://conwet.fi.upm.es/wirecloud/install (http://conwet.fi.upm.es/wirecloud/install)**
- Wirecloud Platform Installation and Administration Guide (FI-Ware version) **https://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Composition_Editor_-_Wirecloud_Mashup_Platform_-_Installation_and_Administration_Guide**

Presentation

Following youtube video shows how the wirecloud was used to implement a simple application in ENVIROFI project.

- ENVIROFI Wirecloud demo video on Youtube **http://youtu.be/yEXlLQYq7s4 (http://youtu.be/yEXlLQYq7s4)**

Tutorial

This video illustrates how to deploy widgets and mashups. In this example, the **Resource Management** Training Monitor & Dispatch view is being deployed to FI-Ware's Lab instance of Wirecloud, then altered a bit and subsequently downloaded again for redistribution or archiving.

User Guide

This video illustrates how to deploy widgets and mashups. In this example, the **Resource Management** Training Monitor & Dispatch view is being deployed to

FI-Ware's Lab instance of Wirecloud, then altered a bit and subsequently downloaded again for redistribution or archiving.

Tutorial

This video illustrates how to offer widgets and mashups on a store. In this example, an offering for a simple workspace is created and then subsequently bought and installed by another user.ls -lah

User Guide

This video illustrates how to offer widgets and mashups on a store. In this example, an offering for a simple workspace is created and then subsequently bought and installed by another user.ls -lah

FAQ

We maintain a list of known issues and workarounds for the Wirecloud developers on **Crisma Catalogue**. Please take a look if you run in strange issues...

- Wirecloud workarounds
  https://crisma-cat.ait.ac.at//node/177

Developer's Guide

We maintain a list of known issues and workarounds for the Wirecloud developers on **Crisma Catalogue**. Please take a look if you run in strange issues...

- Wirecloud workarounds
  https://crisma-cat.ait.ac.at//node/177

Downloads:

FI-Ware provides a **general-purpose Wirecloud instance** as a part of their "Open Innovation Lab". This instance can be used **free of charge** for development purpose.

CRISMA-specific Wirecloud instance for development and testing of the **pilot** and reference applicaitons is hosted by **AIT**. Please contact **Maria** or **Manuel** to get an account on this server.

- FI-Ware open laboratory (open for all developers and public testing)
  http://lab.fi-ware.eu/ (http://lab.fi-ware.eu/)
- CRISMA wirecloud instance (CRISMA development and testing)
  http://wirecloud.ait.ac.at/ (http://wirecloud.ait.ac.at/)

Source code of the WireCloud platform and of some some basic widgets is available on Github

- Wirecloud Github repositories
  https://github.com/Wirecloud

Quick install:

```
$ pip install wirecloud
$ wirecloud-admin startproject your_wirecloud_instance_name --quick-start
$ cd your_wirecloud_instance_name
$ python manage.py runserver --insecure
```

Use admin/admin to log into the platform

The **AIT** team maintains a set of virtual images with pre-installed and pre-configured WireCloud instances which can be easily installed and used "as is" in **CRISMA** reference applicaitons and pilots. These images can be downloaded from the links below.

As the project progresses, we shall provide updated versions of the virtual images. Please always use the latest image version, unless **it** is broken (in which case you should contact **AIT** so that we can fix it).

- Generic WireCloud VM(s) - if any
  https://crisma-cat.ait.ac.at/some%20URL%20-%20directory%20with%20VMs
- Application X - specific VM(s) - repeat as needed
  https://crisma-cat.ait.ac.at/some%20URL%20%20-%20directory%20with%20VMs

Team:

Manuel Warum

Development roles: development, support
Main **developer** of the Wirecloud widgets at **AIT**. First contact for all questions related to development of the Wirecloud widgets, operators and mashup applications within **CRISMA**.

Denis Havlik

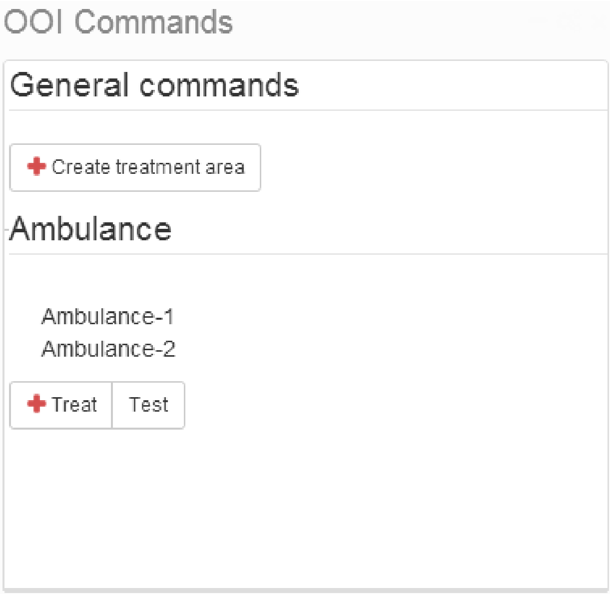
Development roles: coordinaton
Strategic positioning of Wirecloud in **CRISMA**; contact with Fi-PPP, FI-Ware project and the Wirecloud owners.

Wirecloud OOI Commands

Authors: [Manuel Warum](#)
[Oren Deri](#)
Version: 0.5

Widget that allows to issue commands to one or more OOIs. The commands available depend on the OOIs' types; for instance, Ambulances can treat in a specific area. In addition, there is also a support for "global" commands, ie. commands that do not require an **OOI** (such as "create area").

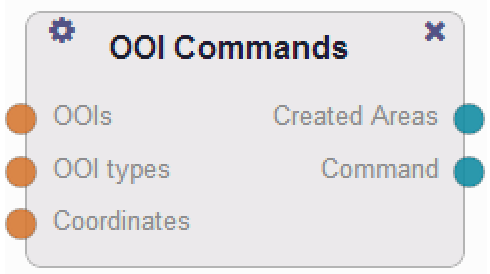
In addition, this widget has the capabilities to accept additional information for commands through interaction with other widgets. For instance, selecting the "Treat" command of an ambulance will prompt the user to select a point on the map. These command arguments can also be limited to some degree, for instance some commands will only except certain OOI types as an argument.



Documentation:

User Guide

This component is a Wirecloud **widget** and as such has a visual representation in the mashup. **It** is designed to grow vertically.



Inputs:

- a list of OOIs that can be commanded: these will be displayed by the component
- a list of **OOI** types (optional): ideally, a complete list of OOI types is sent, and this only needs to be done *once*. The widget will remember this data throughout its lifetime.
- coordinates for command arguments (optional iff there are no commands that require arguments): these coordinates should be **JSON** objects with a lat and a lon property, e.g. { "lat": 41.14, "lon": 12.21 }

Outputs:

- an array of OOIs that have been created by a command that was currently executed, iff a command has the **capability** to do so
- commands that have been issued

Which commands are available is entirely regulated through a JavaScript file, but could be loaded from different sources (eg. a RESTful **service**) as well.

Downloads:

The source code for this component can be found here: **https://github.com/crismaproject/wirecloud-widgets/tree/master/ooi_command**

Team:

Manuel Warum

Main contact point for all questions related to Wirecloud and development of the Wirecloud widgets, connectors and mashup applications within **CRISMA**. He is also the author of **AIT**'s Wirecloud widgets and mash-up applications used in "**Resource Management** Training" **technology demonstrator**.

Denis Havlik

Development roles: coordinaton, requirements

Principal contact between **CRISMA** and FI-PPP programme/FI-Ware project. He can discuss the strategic issues with the WireCloud owners if/when needed.

Wirecloud OOI GIS Map

Version: 0.5

A map **widget** showing **Object Of Interest** (**OOI**) on a background map.

This OpenLayers-based implementation allows to load any kind of map from external services, including but not limited to OpenStreetMaps, Google Maps, **WMS**, etc.

In addition, **it** allows OOIs to be selected by the **end user**, and selections can be synchronized with other widgets supporting this behaviour (such as the OOI Table widget).

The widget should work on mobile devices with limited computational or memory **capacity** but its functionality or performance is likely not comparable to a desktop system, especially under load.

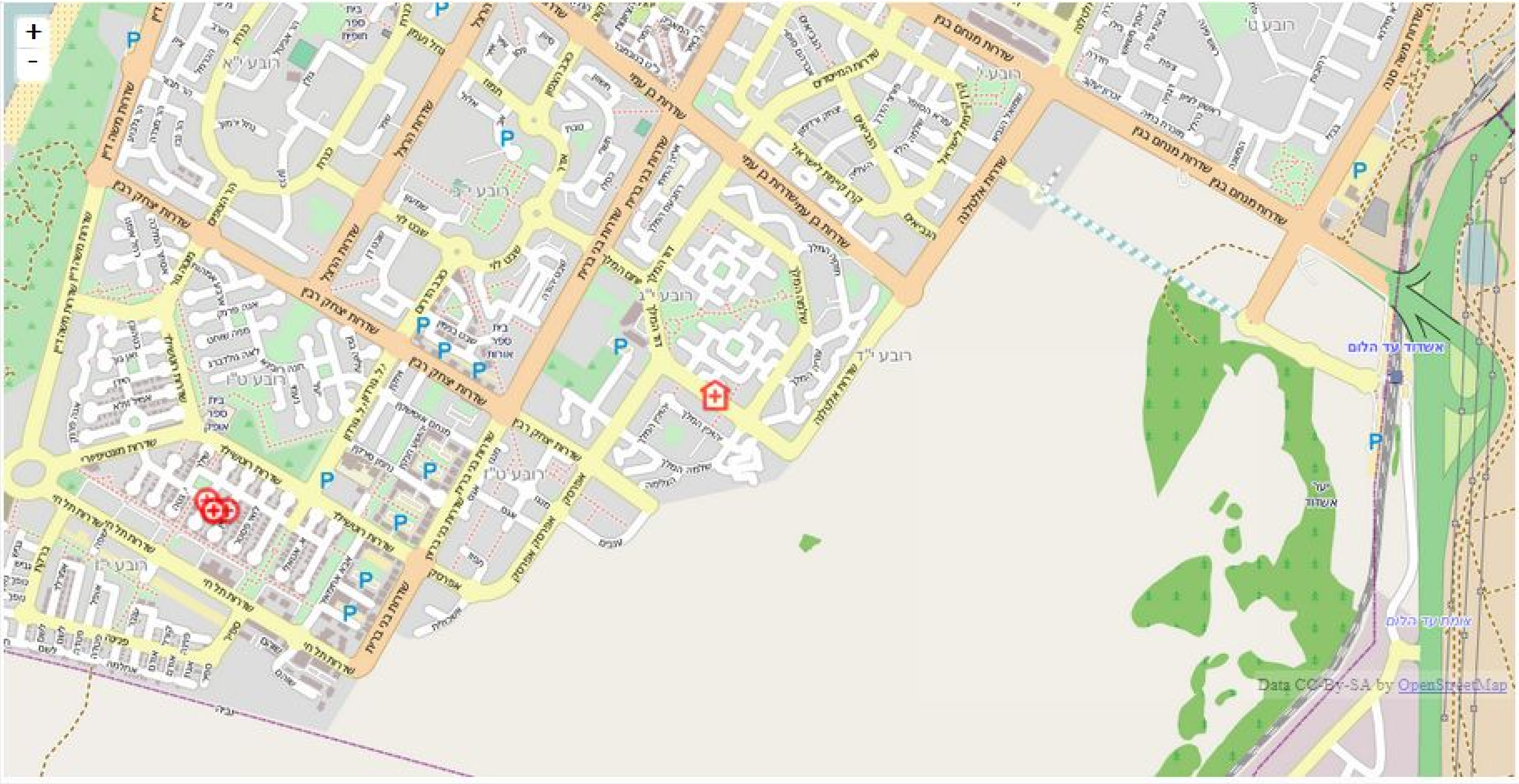
Documentation:

User Guide

TBD:

Inputs: OOIs, Selected OOIs

Outputs: Selected OOIs, Center, Click



Configuration:
Background
WMS

Downloads:

The source code for this component can be found here: **https://github.com/crismaproject/wirecloud-widgets/tree/master/ooi_gis_map**

Team:

Manuel Warum

Main contact point for all questions related to Wirecloud and development of the Wirecloud widgets, connectors and mashup applicaitons within **CRISMA**.

He is also the author of **AIT**'s Wirecloud widgets and mash-up applications used in "**Resource Management** Training" **technology demonstrator**.

Denis Havlik

Development roles: coordinaton

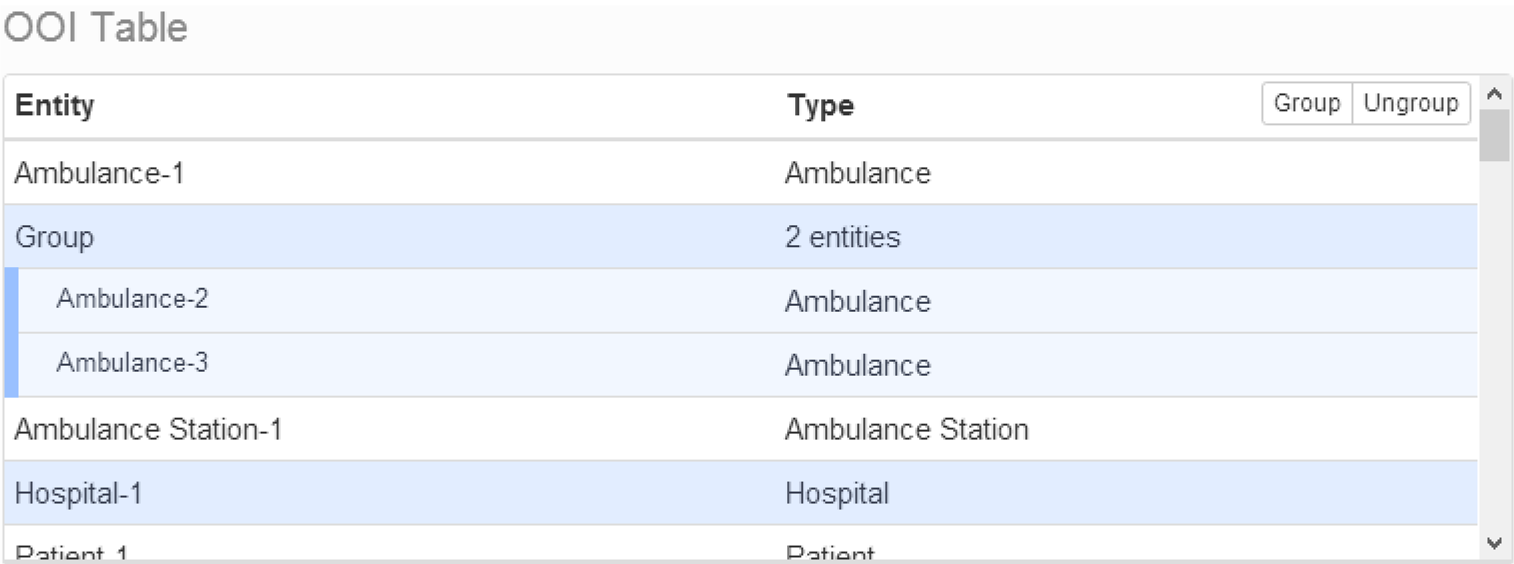
Principal contact between **CRISMA** and FI-PPP programme/FI-Ware project. He can discuss the strategic issues with the WireCloud owners if/when needed.

Wirecloud OOI Table

Version: 0.5

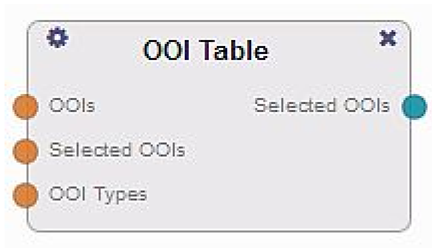
This **widget** shows all OOIs associated with a worldstate in a tabular form. **It** is possible to filter which elements should be visible (eg. only OOIs of certain types such as Ambulances and Hospitals). In addition, it also allows the user to group any OOIs for convenience; groups are persisted on the client-side using HTML5 storage mechanics and are automatically restored inbetween sessions.

Furthermore, this widget synchronizes selections with the **OOI** **GIS** widget, ie. anything selected on one end will be selected on the other.



Documentation:

User Guide



Downloads:

The source code for this component can be found here: **https://github.com/crismaproject/wirecloud-widgets/tree/master/ooi_table**

Team:

Manuel Warum

Main wirecloud **developer**

Wirecloud Simulation Picker

Authors: Manuel Warum
Version: 0.5

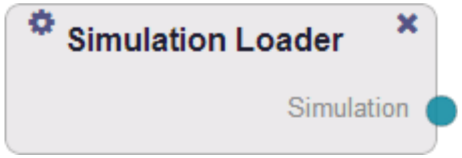
A **widget** showing all simulations available on the **Object of Interest** **World State** Repository (**OOI**-WSR). **It** allows the user to select one of these simulations and load it, which will initialize other connected widgets and gadgets of the Wirecloud **application** with relevant data.

Note that this does not yet set a world state by itself; it only limits appropriate widgets capable of doing so to the selected **simulation**.

Documentation:

User Guide

TBD



Output: **simulation**

Configuration: **OOI**-WSR URI

Downloads:

The source code for this component can be found here: **https://github.com/crismaproject/wirecloud-widgets/tree/master/simulation_picker**

Team:

Manuel Warum

Main contact point for all questions related to Wirecloud and development of the Wirecloud widgets, connectors and mashup applicaitons within **CRISMA**. He is also the author of **AIT**'s Wirecloud widgets and mash-up applications used in "**Resource Management** Training" **technology demonstrator**.

Denis Havlik

Development roles: coordinaton, requirements

Principal contact between **CRISMA** and FI-PPP programme/FI-Ware project. He can discuss the strategic issues with the WireCloud owners if/when needed.

Wirecloud WorldState Picker

Version: 0.6.0.1

This **widget** lets the user pick and load a **world state** from the Objects of Interest World State Repository (**OOI**-WSR).

It displays all world states and their hierarchy to the **end user**, allowing him/her to select the one that should be continued.

After a world state is selected by the end user, it will load any and all associated data, including associated OOIs, OOI types as well as world state metadata.

Documentation:

User Guide

Downloads:

The source code for this component can be found here: **https://github.com/crismaproject/wirecloud-widgets/tree/master/worldstate_loader**

Note that **it** will be superceded by an interactive version soon.

Team:

Manuel Warum

Main Wirecloud **developer**

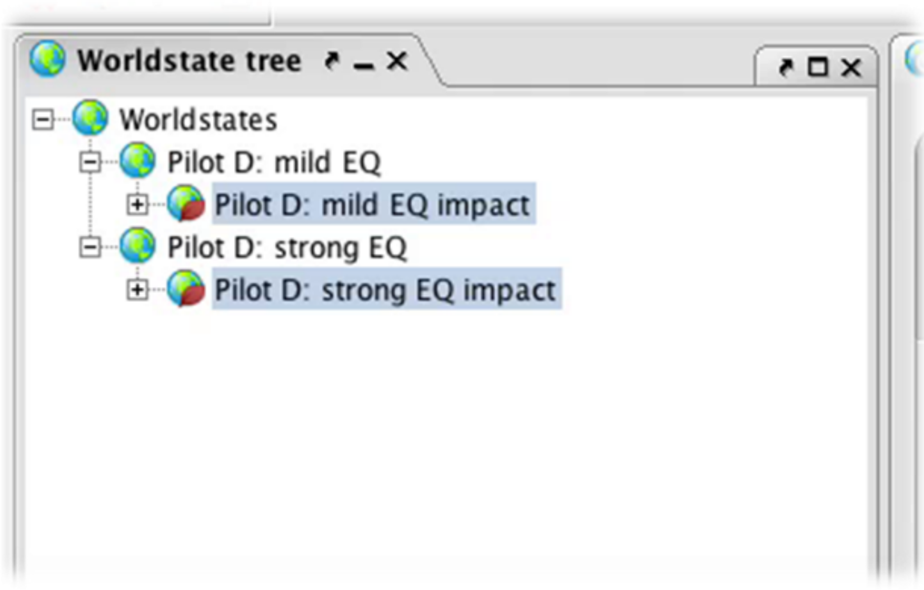
Worldstate Tree Widget (Java)

Version: 1.0

This is a java-based prototype of the Worldstate Tree **Widget** Functional **Building Block**.

The Worldstate Tree Java Widget is a new software component that is developed from scratch in phase one of WP34. In contrast to the technological requirements for new software developments that demand the implementation of new User Interaction Building Blocks as Composite UI Module (HTML and JavaScript); the Worldstate Tree Java Widget is developed in Java in the context of ongoing prototyping activities to demonstrate the capabilities of the **CRISMA Framework** towards the end of the first implementation phase.

Worldstate Tree Widget Java Implementation

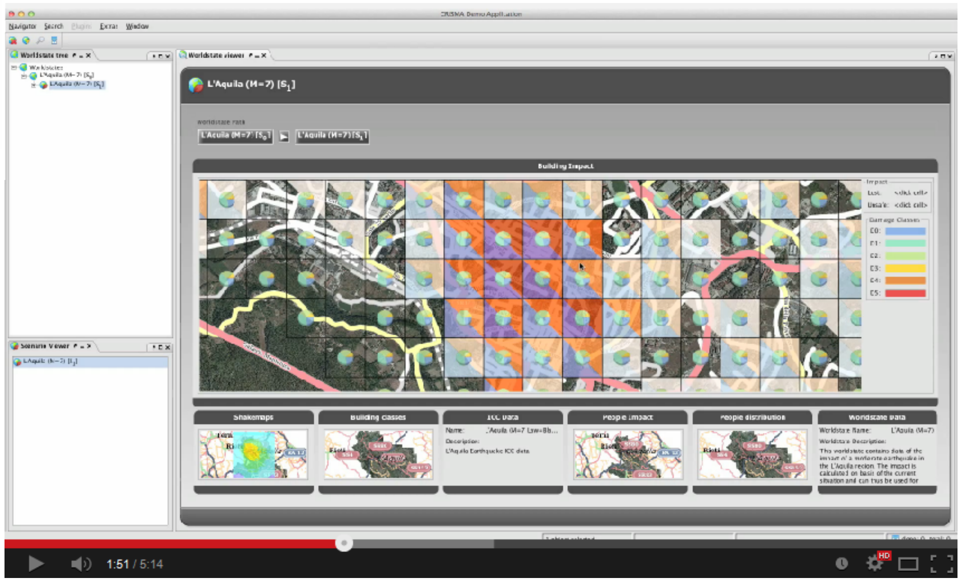


The Worldstate Tree Java Widget is implemented as cids navigator (UI Integration Platform) Renderer and Editor module.

The development of the Worldstate Tree Java Widget is finished. The Worldstate Tree Java Widget is replaced by a software component that satisfies all technological requirements of Mashable Composite UI Modules.

Documentation:

Demo



This demo video of the CRISMA Technolog Demonstrator shows, among others, the Worldstate Tree **Widget** in action.

Functional Description

- Worldstate View Building Block Description
  https://crisma-cat.ait.ac.at/bbs/worldstate-view

Downloads:

- Sourcecode on GitHub
  https://github.com/crismaproject/cids-custom-crisma

- Snaphots in the Maven Repository
  https://repo.cismet.de/simple/cismet-libs-snapshots-local/de/cismet/cids/custom/crisma/

Team:

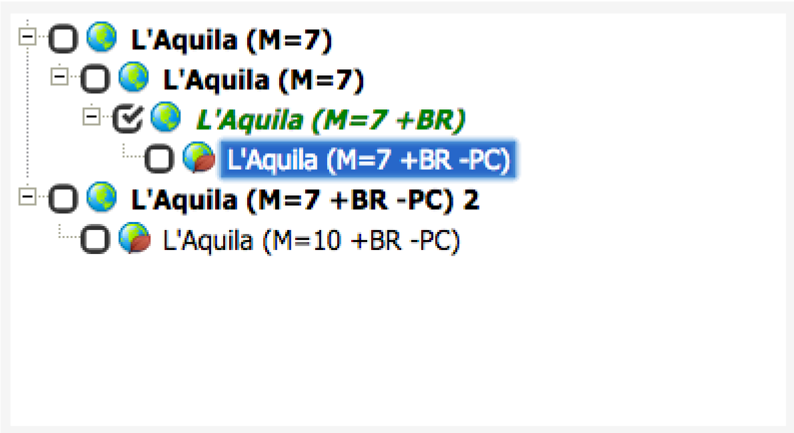
Martin Scholl

Development roles: configuration, development, installation, integration, support
Main **developer**

Worldstate Tree Widget (JavaScript)

Version: 1.0

This is a JavaScript-based implementation of the Worldstate View Functional **Building Block**. The **widget** uses the **ICMM** as backend and visualises the Worldstates in their natural tree structure. **It** lets users navigate the tree and can propagate selection and activation of worldstates so that other components may synchronise their views.



Worldstate Tree Widget (JavaScript)

The Worldstate Tree JavaScript Widget is a new software component that is developed from scratch. It satisfies the technological requirements for new software developments as it is realized as Composite UI Module (HTML and JavaScript). Thus it can be used in any UI Integration Platform that supports HTML/CSS and JavaScript such as a Browser. Moreover, there is an additional build for the WireCloud platform effectively making it a Mashable UI Component.

Documentation:

Install Guide

See the GitHub projects for info.

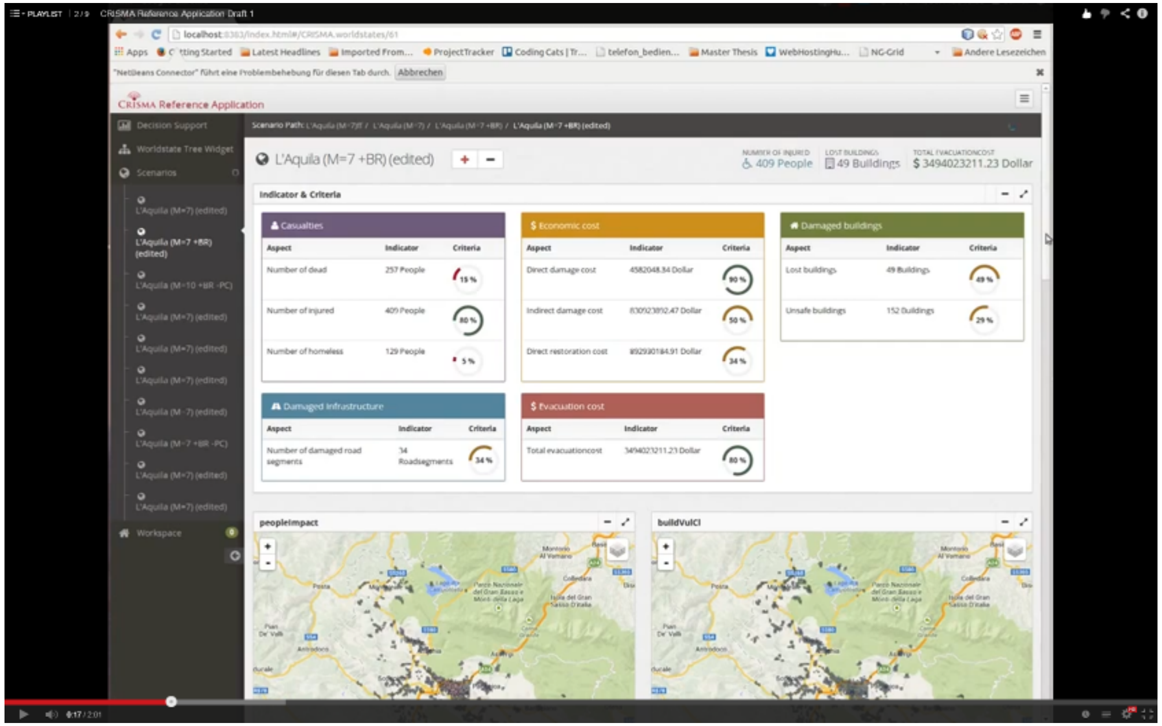
- AngularJS implementation: Readme on GitHub
  https://github.com/crismaproject/worldstate-tree-widget-angular
- WireCloud implementation: Readme on GitHub
  https://github.com/crismaproject/worldstate-tree-widget-wirecloud

Developer's Guide

See the GitHub projects for info.

- AngularJS implementation: Readme on GitHub
  https://github.com/crismaproject/worldstate-tree-widget-angular
- WireCloud implementation: Readme on GitHub
  https://github.com/crismaproject/worldstate-tree-widget-wirecloud

Demo



This demo Video of the CRISMA Reference Application for the Earthquake and Forest Fire Domains shows, among others, the Worldstate Tree **Widget** (JavaScript) in action.

Functional Description

- Worldstate View Building Block Description
  https://crisma-cat.ait.ac.at/bbs/worldstate-view

FAQ

Issue Tracking and FAQ on GitHub

- AngularJS implementation: GitHub Issue Tracker
  https://github.com/crismaproject/worldstate-tree-widget-angular/issues
- WireCloud implementation: GitHub Issue Tracker
  https://github.com/crismaproject/worldstate-tree-widget-wirecloud/issues

Tutorial

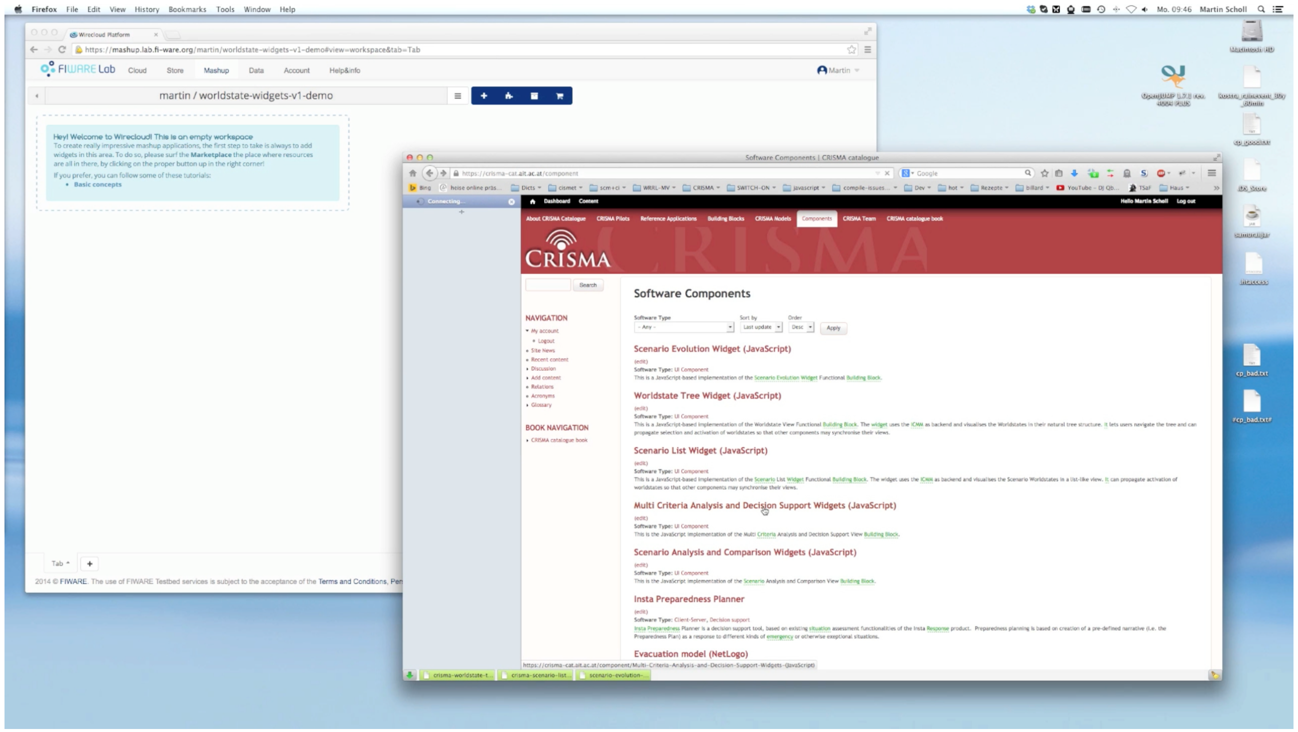
(http://youtu.be/niQC0RbEtrQ)

This is a HowTo video that showcases how the Worldstate WireCloud widgets can be wired together.

- HowTo: Wire Worldstate WireCloud Widgets
  http://youtu.be/niQC0RbEtrQ (http://youtu.be/niQC0RbEtrQ)

Downloads:

This component is provided in form of a AngularJS directive that can be easily embedded and configured as needed. The repository, however, contains a basic demo **application** that shows the capabilities of the component. Additionally, there is the repository that only contains wire code for WireCloud and thus effectively makes



the tree component available as mashable **widget**.

- AngularJS component **https://github.com/crismaproject/worldstate-tree-widget-angular**
- WireCloud component **https://github.com/crismaproject/worldstate-tree-widget-wirecloud**

Ready-to-use build of the tree WireCloud **widget**. As soon as the **CRISMA** WireCloud platform marketplace is available the newest builds can be obtained from there. A minified version of the WireCloud widget is also available.

The AngularJS components can easily obtained via the dist folder of the project or via the bower dependency management tool

shell@computer:> bower install --save worldstate-tree-widget-angular

- AngularJS component **https://github.com/crismaproject/worldstate-tree-widget-angular/tree/v1.0.8/dist**
- Mashable WireCloud widget **http://crisma.cismet.de/lib/wirecloud/crisma-worldstate-tree-widget-wirecloud.wgt (http://crisma.cismet.de/lib/wirecloud/crisma-worldstate-tree-widget-wirecloud.wgt)**
- Mashable WireCloud widget (minified) **http://crisma.cismet.de/lib/wirecloud/crisma-worldstate-tree-widget-wirecloud.min.wgt (http://crisma.cismet.de/lib/wirecloud/crisma-worldstate-tree-widget-wirecloud.min.wgt)**

Team:

Martin Scholl

Development roles: development, installation, integration, support

Lead **developer** of this **widget** and also main contact point for any **Pilot** support regarding the embedding of this widget into **CRISMA** applications.

Manuel Warum

Development roles: support

Main contact point for all questions related to Wirecloud and development of the Wirecloud widgets, connectors and mashup applicaitons within **CRISMA**. He is also the author of **AIT**'s Wirecloud widgets and mash-up applications used in "**Resource Management** Training" **technology demonstrator**.

Worldstate Widget (Java)

Version: 1.0

This is a java-based prototype of the Worldstate **Widget** Functional **Building Block**. The Worldstate Widget is a simple container with predefined layout and fixed slots for further **application** specific widgets that are capable of visualising the actual data of Worldstate, including also **ICC** data.

Worldstate Java Widget

The Worldstate Widget is a new software component that is developed from scratch. In contrast to the technological requirements for new software developments that demands the implementation of new User Interaction Building Blocks as Composite UI Module (HTML and JavaScript); the Worldstate Java Widget is developed in Java. The Worldstate Java Widget is implemented as cids navigator (UI Integration Platform) Renderer and Editor module.

Documentation:

Demo

This demo Video of the CRISMA Technolog Demonstrator shows, among others, the WorldsteteWidget in action.
Reference Application Demo Video

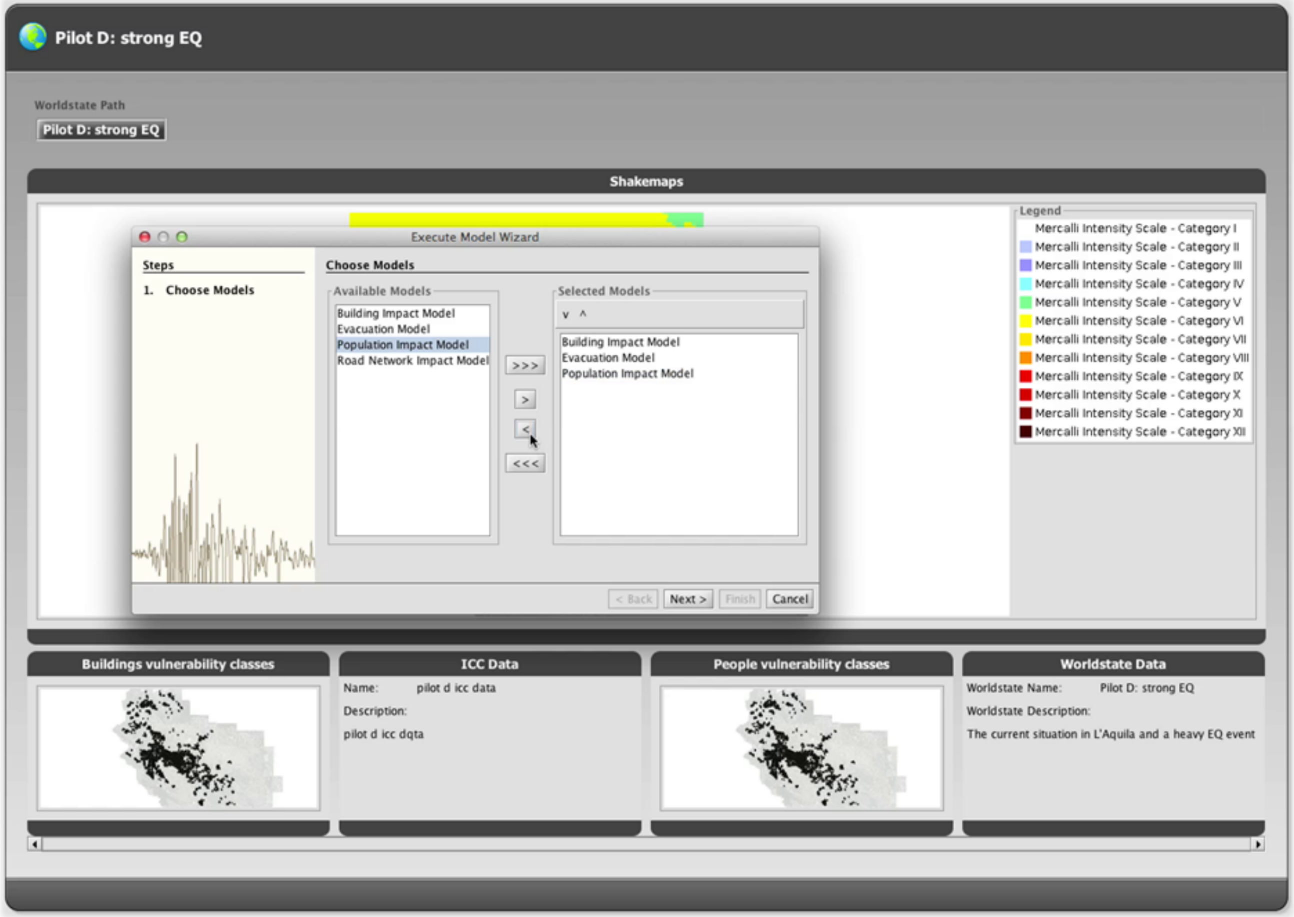
Functional Description

- Worldstate View Building Block Description
  https://crisma-cat.ait.ac.at/bbs/worldstate-view

Downloads:

- Sourcecode on GitHub
  https://github.com/crismaproject/cids-custom-crisma
- Snaphots in the Maven Repository
  https://repo.cismet.de/simple/cismet-libs-snapshots-local/de/cismet/cids/custom/crisma/

Team:

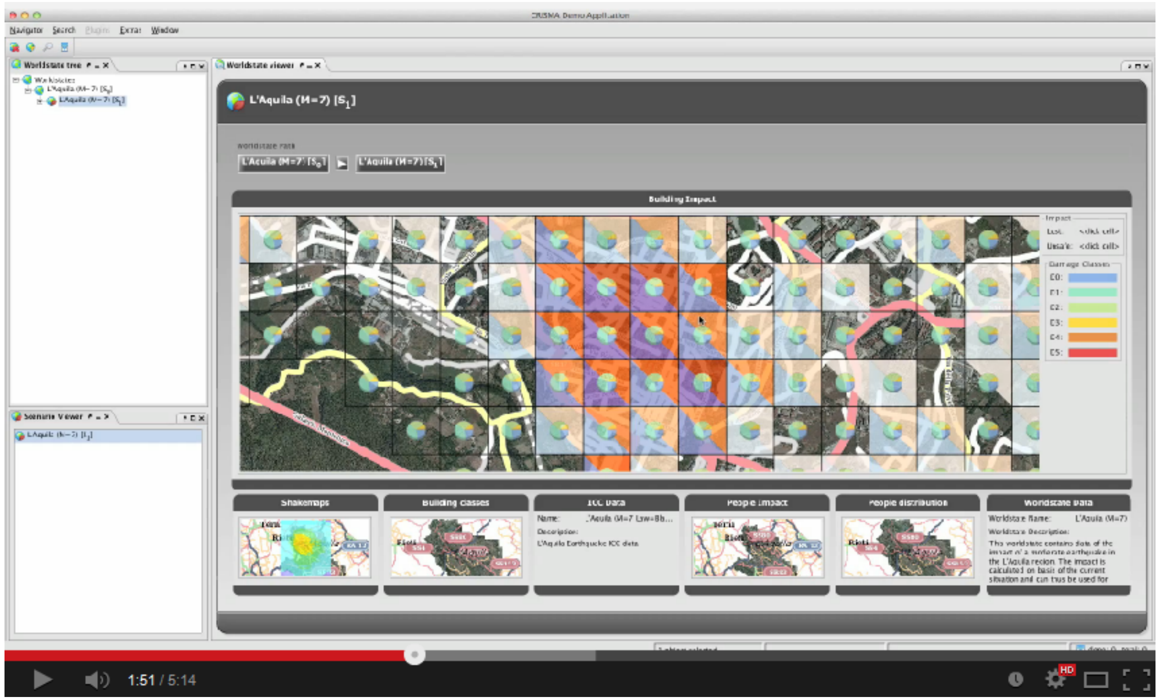


Martin Scholl

Development roles: development, installation, integration, support

Main **developer** of the **Widget**.

Worldstate Widget (JavaScript)



Version: 0.5

This is a javascript-based prototype of the Worldstate View Functional **Building Block**. The Worldstate **Widget** is a simple container with predefined layout and fixed slots for further **application**-specific widgets that are capable of visualising the actual data of Worldstate, including also **ICC** data.

The generic Worldstate Container Widget is shown in the screen shot below.

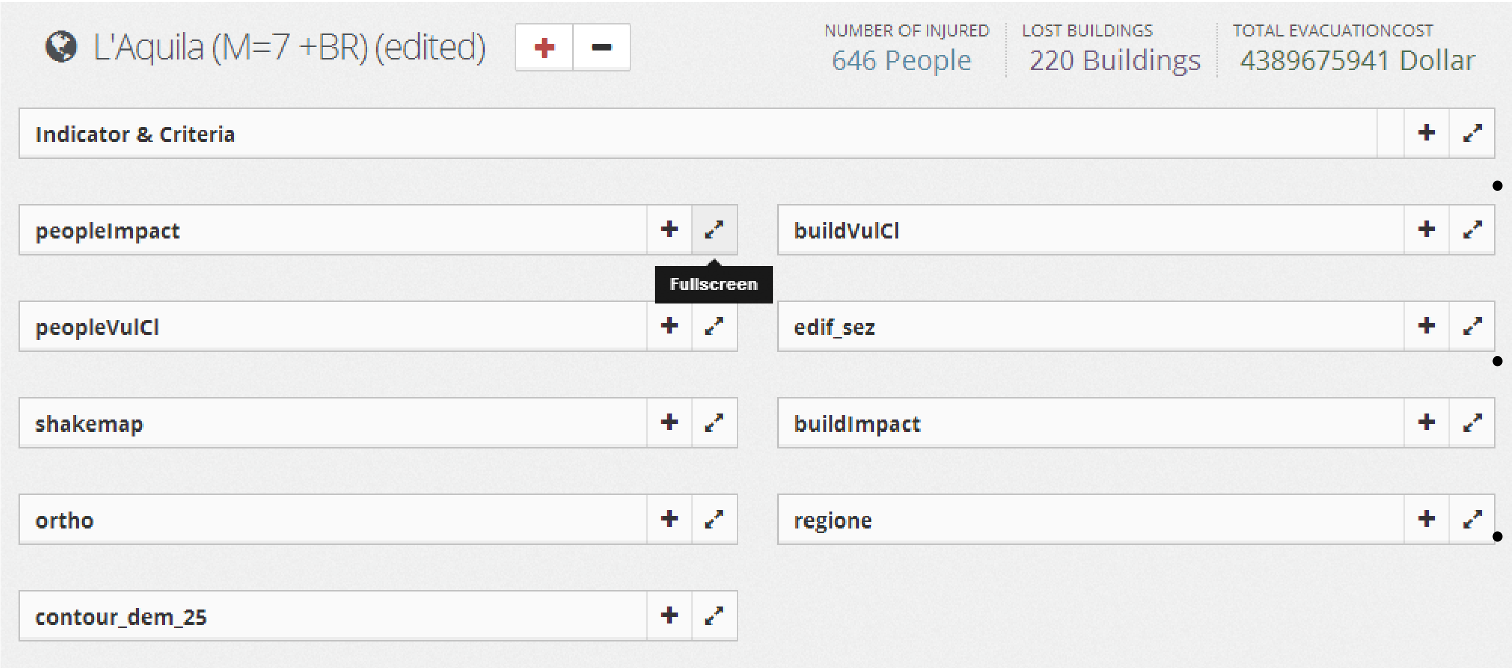
Worldstate Widget

The application-specific widgets shown in the screen shots below have been implemented for the **Reference Application** for Earthquake and Forest Fire Domains.

Indicator and Criteria in Worldstate Widget

Custom Worldstate Dataslot Visualisation

The Worldstate Widget is a new software component that is developed from scratch. **It** satisfies the technological requirements for new software developments as it is realized as Composite UI Module (HTML and JavaScript). Thus it can be used in any UI Integration Platform that supports HTML/CSS and JavaScript such as a



Browser.

Documentation:

Install Guide

- Project Wiki on GitHub **https://github.com/cismet/crisma-ref-app/wiki**

Developer's Guide

- Project Readme on GitHub **https://github.com/cismet/crisma-ref-app/blob/master/README.md**

Functional Description

- Worldstate View Building Block Description **https://crisma-cat.ait.ac.at/bbs/worldstate-view**

Demo

This demo Video of the CRISMA Reference Application for the Earthquake and Forest Fire Domains shows, among others, the Worldstate **Widget** in action.

Downloads:

- Sourcecode on GitHub **https://github.com/cismet/crisma-ref-app**
- Angular modules for common CRISMA related services **https://github.com/crismaproject/crisma-widget-shares**

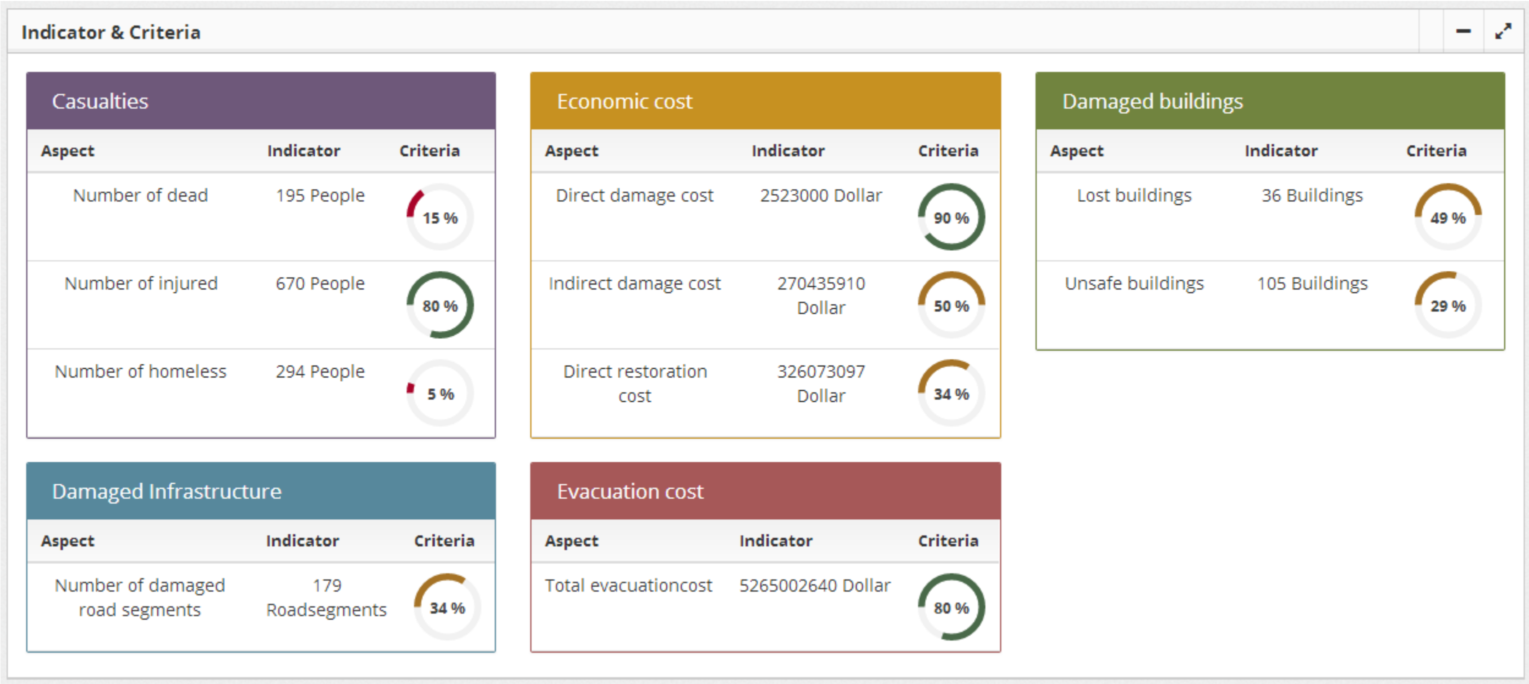
Team:

Martin Scholl

Development roles: architecture, coordinaton, support
Coordinates development activities and provides general support.

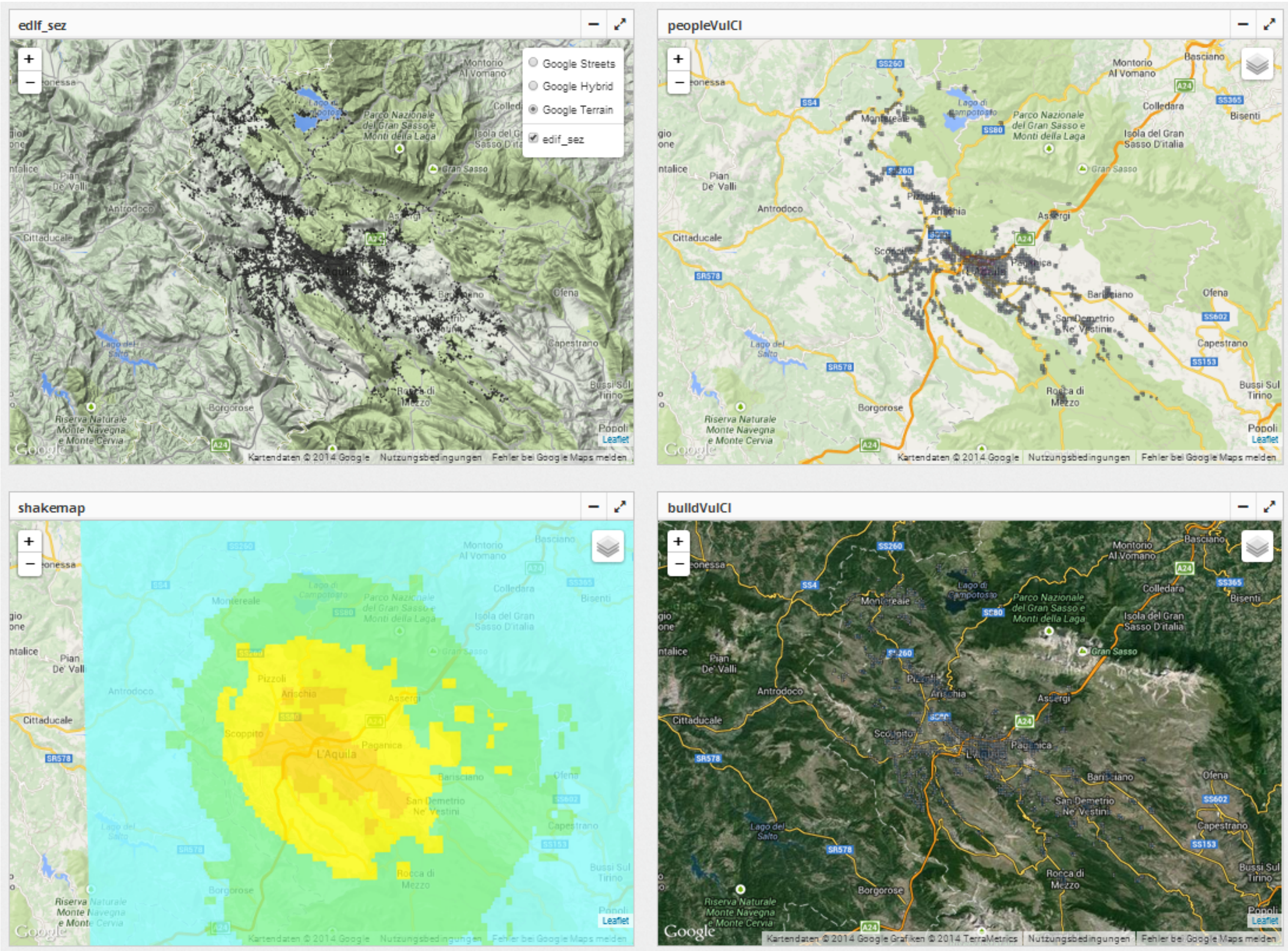
Daniel Meiers

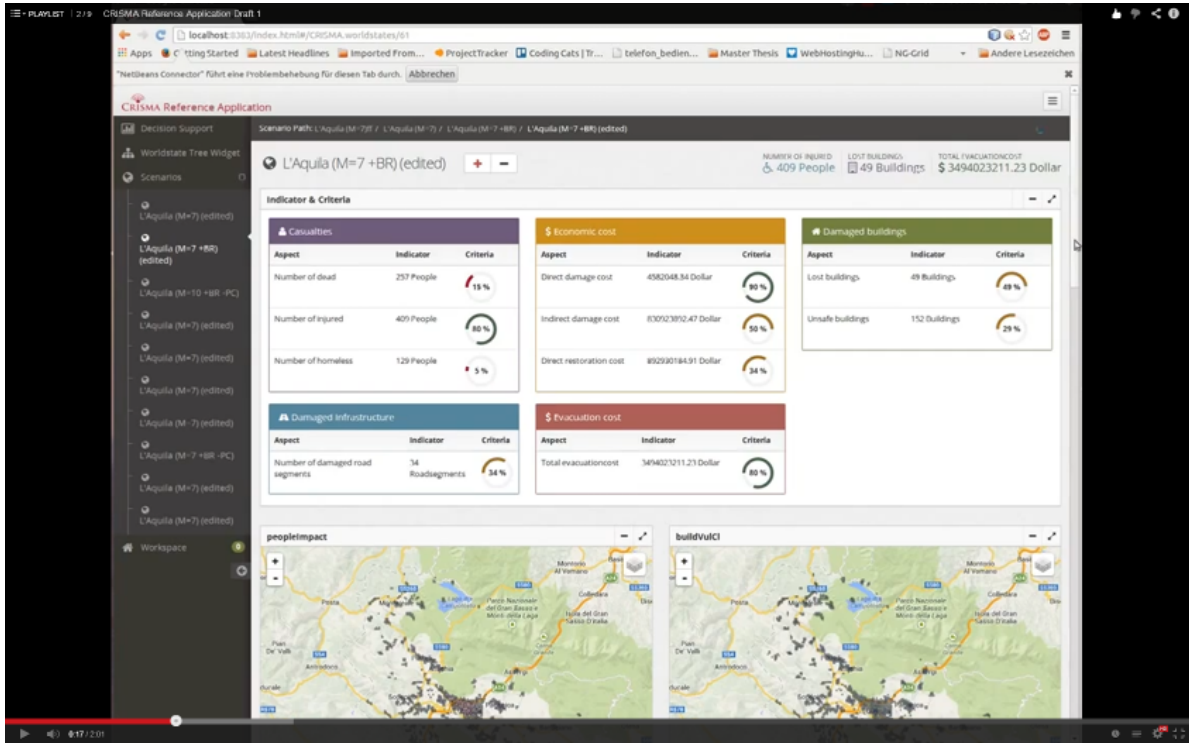
Development roles: development, installation, integration, support



Lead **developer** of this **widget** and also main contact point for any **Pilot** support regarding the embedding of this widget into **CRISMA** applications.

cids Navigator


edif_sez
Google Streets
Google Hybrid
Google Terrain
edif_sez
peopleVulCl
shakemap
buildVulCl
Kartendaten © 2014 Google
Nutzungsbedingungen
Fehler bei Google Maps melden
Kartendaten © 2014 Google Grafiken © 2014 TerraMetrics
Leaflet



Version: 2.0

Cids Navigator is one of the software components that implement the Functional **Building Block** UI Integration Platform of the **CRISMA Framework**.

cids Navigator

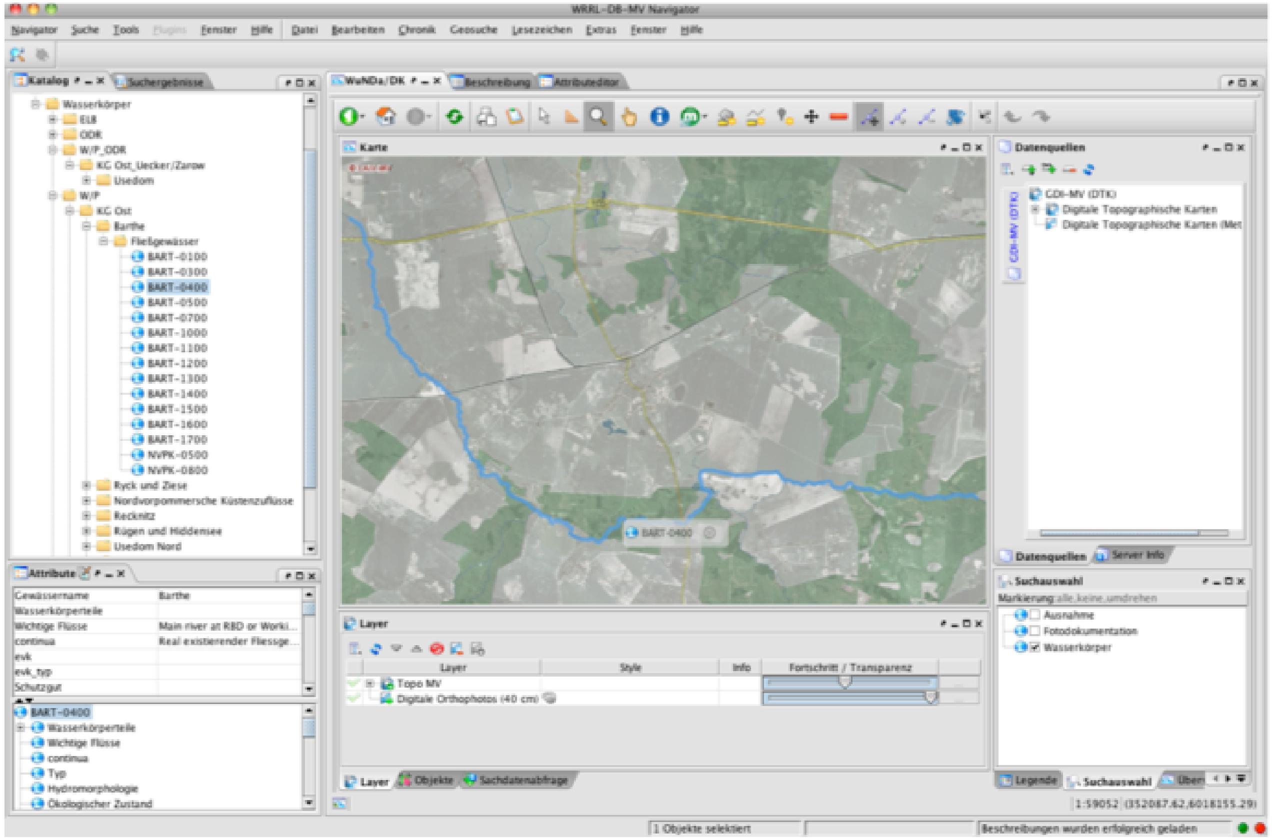
Cids Navigator is the default **GUI** client for user interactions with the cids **system** and thus in principle also compatible with the **ICMM**. The software offers a uniform, user-specific view of integrated information systems and is particularly useful for cross-system search and retrieval in space, time, and textual content. **It** can be used e.g. as the management client for Worldstates and worldstate data and thus also for executing and monitoring worldstate transitions.

cids Navigator Main Screen

Usage in CRISMA

Cids navigator is used as the basis for **Technology Demonstrator** to demonstrate the capabilities of the CRISMA Framework. As implementation of the UI Integration Platform **BB**, cids navigator is also able to host Composite UI Modules (CRISMA Building Blocks realised as HTML5 and JavaScript Widgets).

Usage as UI Integration Platform



Documentation:

Implementation Plan

Implementation Plan of cids Navigator

- cids Navigator Implementation Plan **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP34/03%20Deliverables/D34.1%20Software%20Components%20(final)/%5BX%5D%20BB%20Implementation%20Plan%20-%20cids%20navigator.doc**

Functional Description

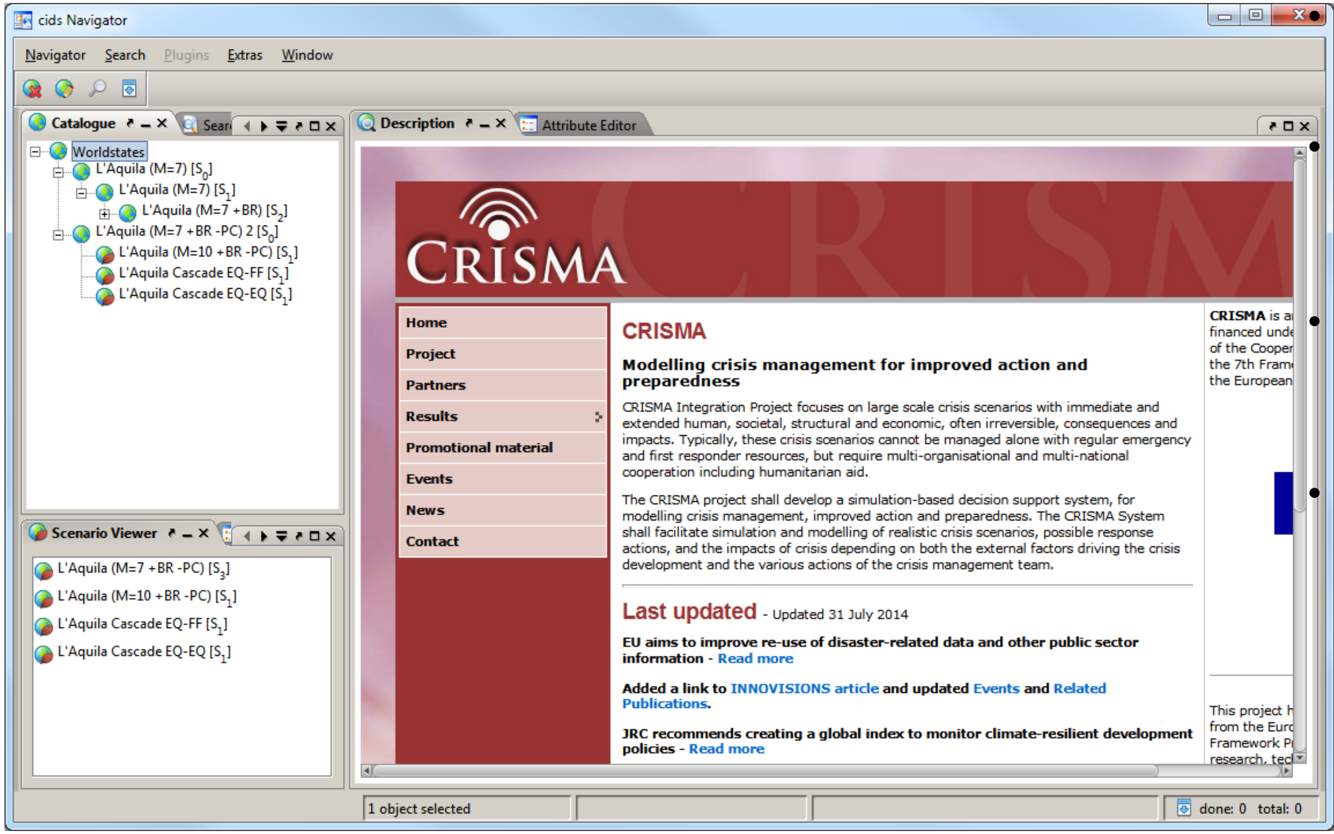
- Description UI Integration Platform BB **https://crisma-cat.ait.ac.at/bbs/ui-integration-platform-bb**

Demo

- Technology Demonstrator Video **https://www.youtube.com/watch**

Downloads:

Sourecode repository @github



- Sourcecode Technology Demonstrator **https://github.com/cismet/cids-custom-crisma-pilot-d**
- Sourceode CRISMA Customisations **https://github.com/crismaproject/cids-custom-crisma**

Binaries in the **cismet** software repository

- Custom CRISMA **https://repo.cismet.de/simple/cismet-libs-snapshots-local/de/cismet/cids/custom/crisma/cids-custom-crisma/**
- Technology Demonstrator **https://repo.cismet.de/simple/cismet-libs-snapshots-local/de/cismet/cids/custom/crisma/cids-custom-crisma-pilot-d/**

Team:

Martin Scholl

Development roles: configuration, development, integration

Develops the **CRISMA** **Reference Application** for **Pilot** D (Demo Application). Takes care about all configuration tasks, required adaptations of that are needed in a **CRISMA Application** regarding cids Navigator. Integrates cids Navigatorwith the respective BBs of a CRISMA Applications. Provides configuration and administration support.

Thorsten Hell

Development roles: development, integration

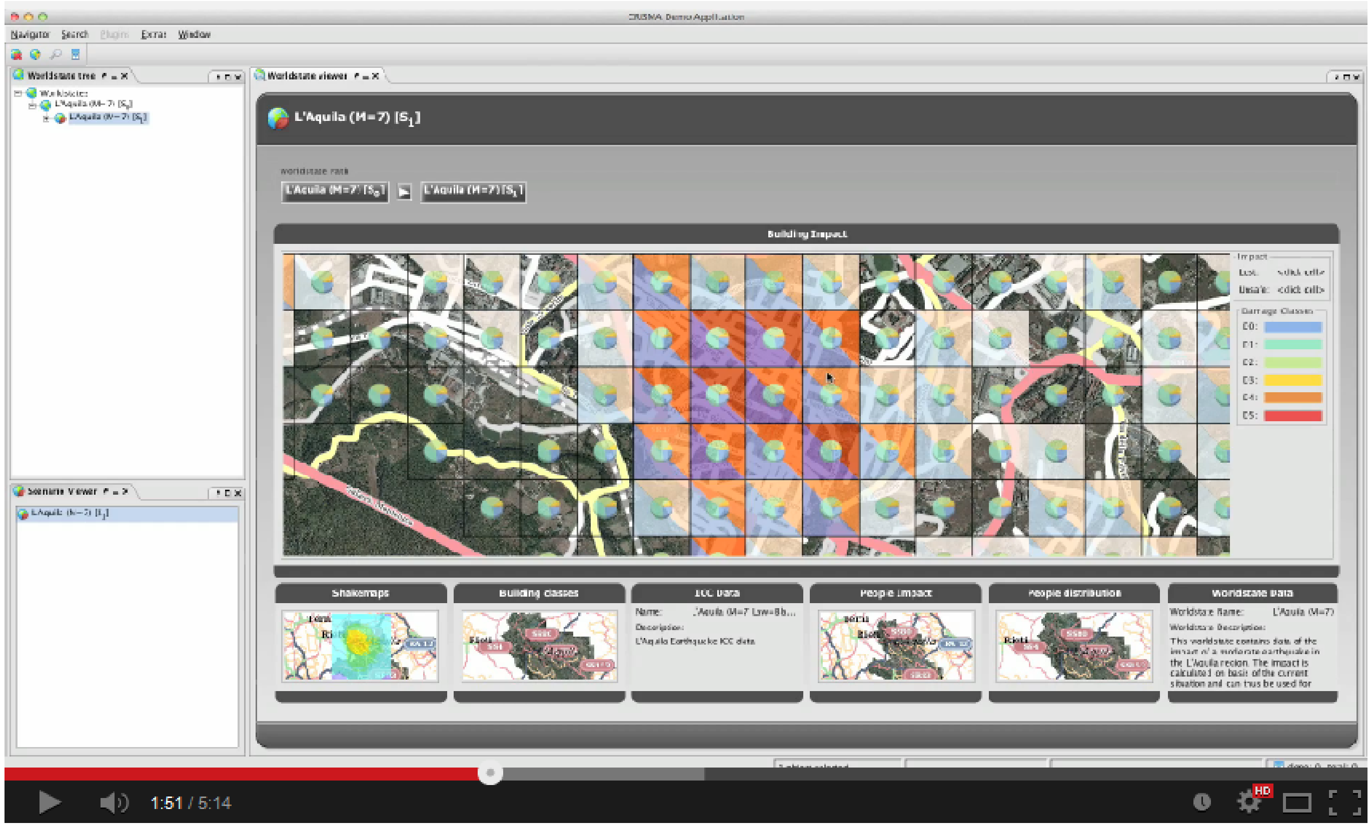
Provides general development and integration **Pilot** Support for cids Navigator.

Pascal Dihé

Development roles: development, integration

Provides general development and integration **Pilot** Support for cids Navigator.

cids System



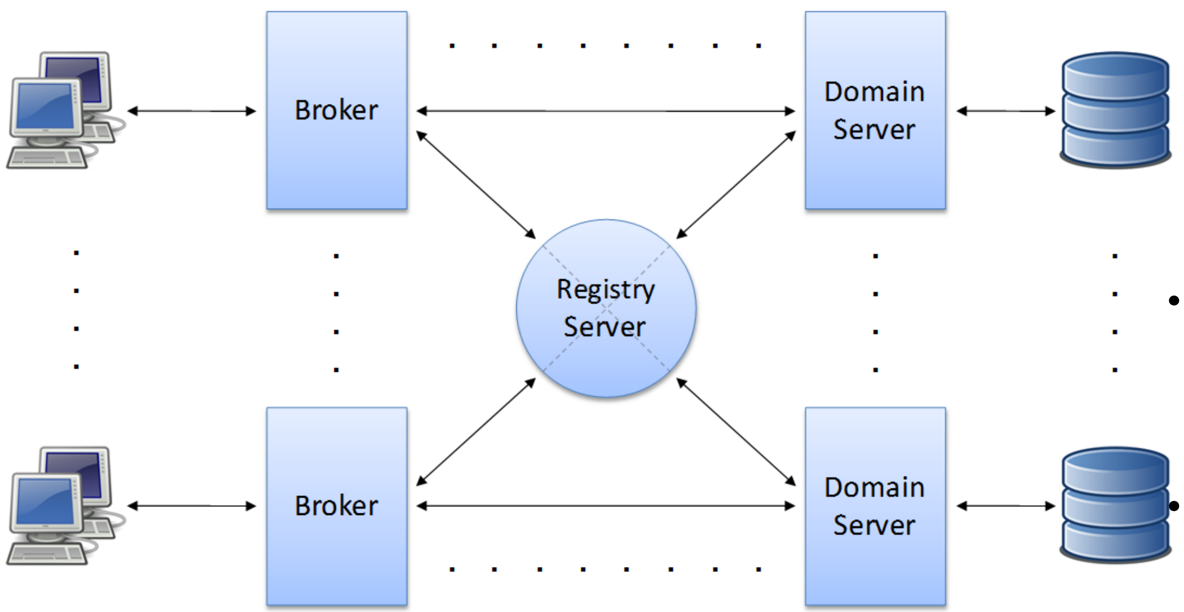
Version: 0.5

The **cismet** cids **System** implements some of the features of the **Integrated Crisis Management Middleware** **Infrastructure** **Building Block**, especially the modular **Control and Communication Information model**. **It** is **open source** and can be extended as required. Additionally, it provides powerful administration tools for Control and Communication Information management and import (Visual Information Modelling and **ETL** Tools).

The cids System

The cids Systen consists of a set of services, applications, software components, management tools, development tools, and **application** programming interfaces (APIs) for the management, integration, and development of heterogeneous information systems with a special focus on interactive geo-spatial systems. It provides a distributed integration platform, which is particularly useful for workflows that need a combination of information and processes from different source systems such as **GIS** systems, relational databases, **simulation** models and so forth.

cids system architecture



The cids **Framework** is based on a client-server architecture. The main building blocks of the cids System are the Navigator (client), the Kernel, and a set of system management tools. The part relevant for the implementation of the **ICMM** is the Kernel. The Kernel represents a network of distributed services and consists of the following four components:

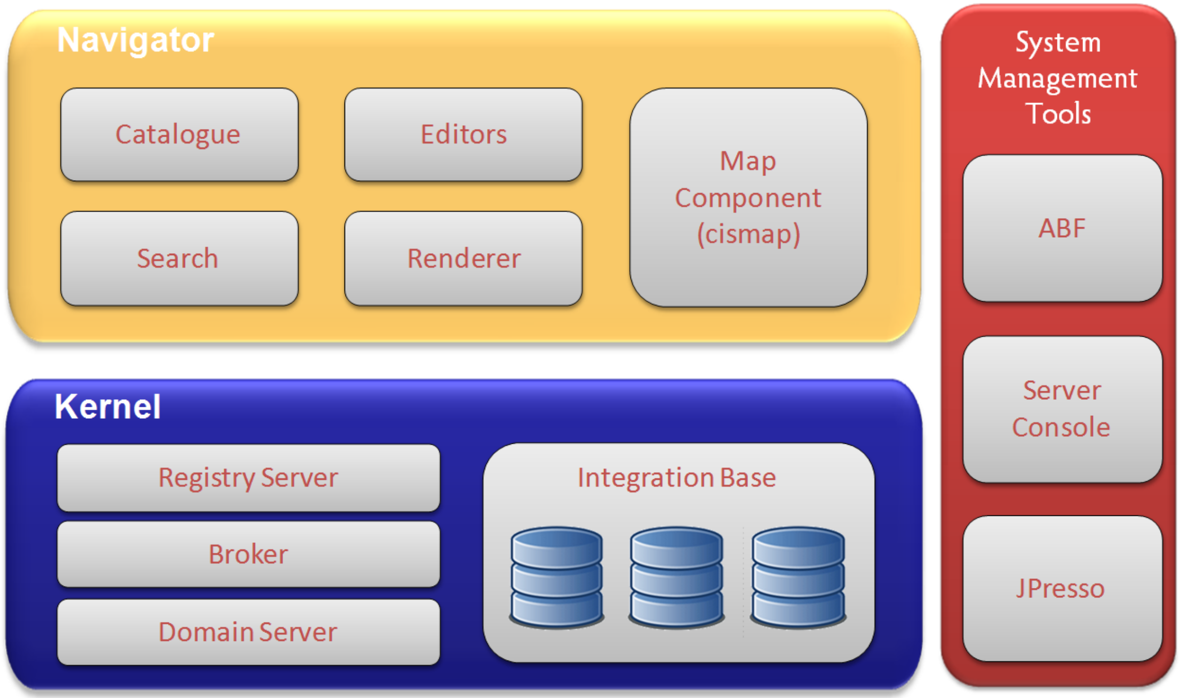
- **Integration Base**
  The Integration Base is a distributed Control and Communication Information (**CCI**) database which consists of a generic Control and Communication Information Model (**CCIM**) placed in a relational DBMS (Data Base Management System).
- **Domain Server**
  The Domain Server is the **interface** to an Integration Base and is responsible for the translation of the generic CCIM structure into concrete CCI entities and classes, thereby also supporting the creation and updating of CCI. It is also responsible for the construction of the dynamic **catalogue** structure at runtime.
- **Registry Server**
  The Registry Server is responsible for the resolution of distributed user privileges needed for the enforcement of access rights, the resolution of the distributed catalogue structure needed for the navigation, and the coordination of the distributed search. It also provides **service** infrastructure related functionalities like server name resolution, network monitoring, status information, etc.
- **Broker**
  The Broker is the interface to the clients and hides the distribution aspects of the system. It acts like a proxy and delegates client request to the appropriate Domain Servers.

cids Kernel

Core Control and Communication Information Model (Core CCIM)

The data model of the entities managed by the ICMM is defined by a Control and Communication Information Model (CCIM). Technically, such a CCIM is implemented as part of a generic and **object**-oriented relational database model of the cids Kernel (Integration Base). Thus, implementing a CCIM is a configuration **task** and does not require any changes to the cids Kernel itself.

The cids instances provided by SP3 are preconfigured with the Core CCIM of the **CRISMA Framework**. The Core CCIM defines a minimal set of entities and



relationships needed to realise the **Conceptual Business Logic** of the CRISMA Framework. The Core CCIM can be seen as the least common denominator on which all CRISMA Federations (Applications) have to agree. The figure below shows an excerpt of the Core CCIM implemented as database model of the Integration Base of the cids Kernel.

Core CCIM of the CRISMA Framework

The main purpose of the Core CCIM is to allow managing worldstates and their transitions as defined by the Conceptual Business Logic in a uniform manner across all CRISMA Federations. Thus, it leverages the development of generic Building Blocks that can be used in any **CRISMA Federation** with minimal or no configuration and development effort.

Since CRISMA **Pilot** Applications (Federations) are very heterogeneous in nature, the Core CCIM is optimised for genericity and reusability and provides only rudimentary **meta-information** about worldstates, transitions, etc. According to this, it does not define a concrete meta-information schema for data access, the description of manipulation and simulations, etc. Therefore, the CORE CCIM as such is not directly usable in a CRISMA Federation and each CRISMA Federation has to define some federation-specific extension of the Core CCIM.

ICMM REST API

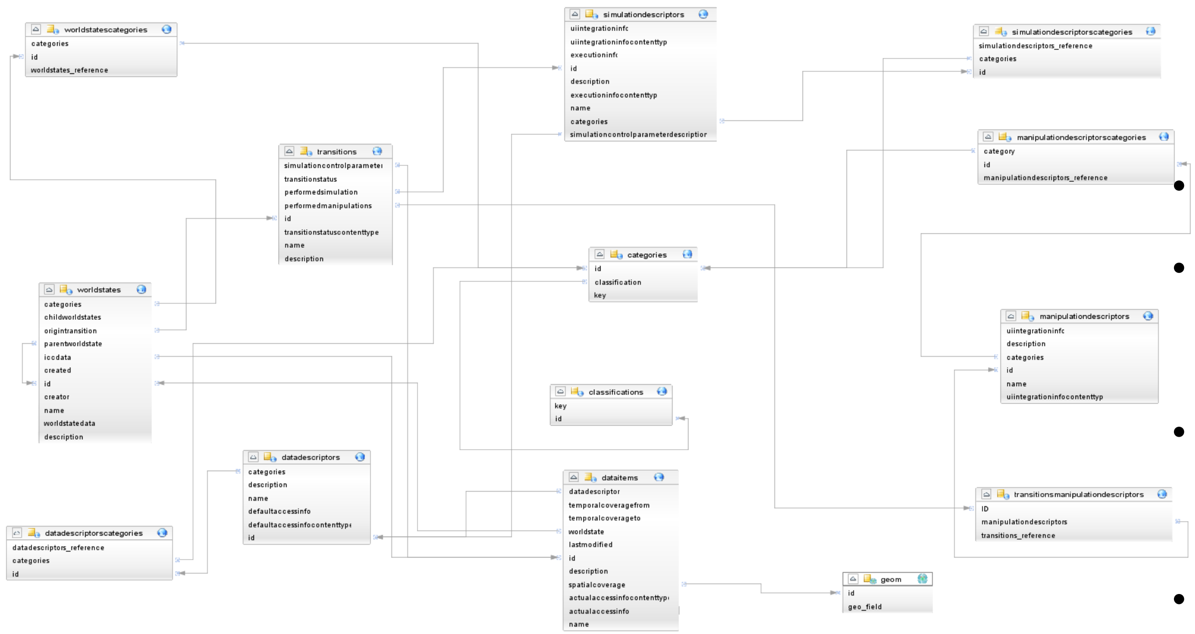
The REST API of the ICMM preview instance follows most but not all rules and recommendations for REST APIs. Conformance to the rules is being implemented successively. The Swagger documentation of the ICMM API is available under Documentation.

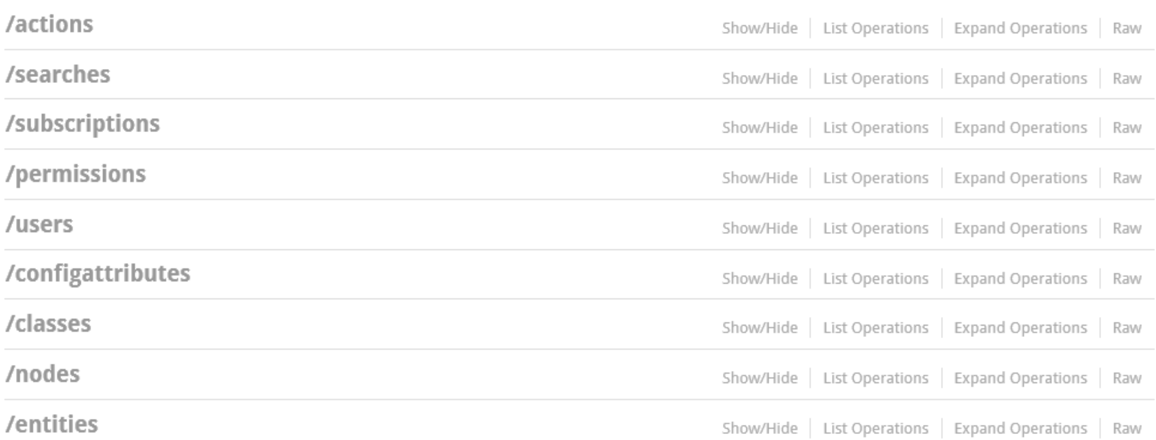
ICMM REST API

The figure above shows the different APIs currently exposed by the ICMM. The Figure below shoes the Management Console of the ICMM.

ICMM Server Console

Documentation:





Implementation Plan

This is the Implementation Plan of the cids **System**. This description consists of three main parts:

- The first optional part gives a brief overview on the cids System which is used as basis for the development of the **ICMM**;
- the second part consists of a detailed technical assessment of the cids System against technical and functional requirements of the ICMM **Building Block** and describes in detail the planned developments provides an outline of an implementation plan; and
- the third part explains how the cids System can be integrated into a **Pilot Application** and how **it** can be adapted to the needs of the pilot.

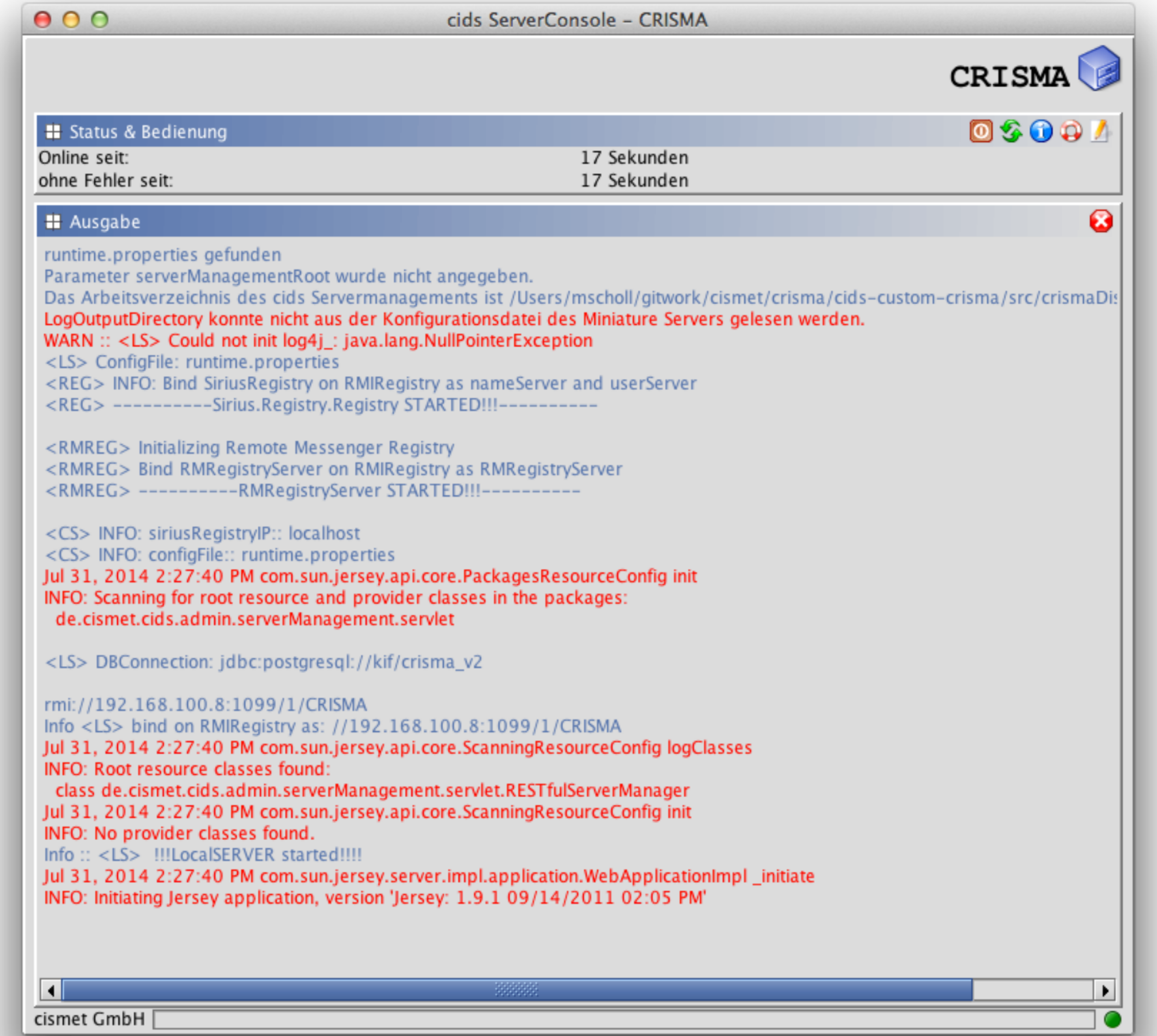
- BB Implementation Plan - cids system **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP34/03%20Deliverables/D34.1%20Software%20Components%20(final)/%5BX%5D%20BB%20Implementation%20Plan%20-%20cids%20system.doc**

Install Guide

This document is a technical usage and integration guide for the **ICMM Infrastructure Building Block**. **It** describes the current development status of the ICMM Infrastructure Building Block, its usage in a **CRISMA Application**, how to integrate other Building Block with the ICMM and how to specify application specific CCIMs.

This document is directed towards developers of CRISMA Applications and other **CRISMA Framework** Building Blocks that need to interact with the ICMM. A thorough understanding of the architectural concepts of the CRISMA Framework and knowledge of the features and limitations of the ICMM Building Block are essential for the understanding of this document.

- ICMM Integration and Usage Guide **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP35/01%20Work%20in%20Progress/Integration%20Guides**



/ICMM%20Integration%20and%20Usage%20Guide.doc

Developer's Guide

API Documentation of the **ICMM** **REST** API.

- Swagger API Documentation
  http://crisma.cismet.de/icms/ (http://crisma.cismet.de/icms/)

Specifications

This document provides a specification or the Core **Control and Communication Information Model** (Core **CCIM**) of the **CRISMA Framework**. This information model is the basis for all **application** specific information models (Application CCIMs). The Core CCIM defines the information classes of the **conceptual business logic** of CRISMA on abstract and generic level. Well defined extension points for Application CCIMs have been included. The figure below gives an informal overview on the Core CCIM.

- Core CCIM Specification
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20CCIMs/Core%20CCIM.doc
- Core CCIM Specification (XMI)
  https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP32/01%20Work%20in%20Progress/D32.2%20CCIMs/Core%20CCIM.xmi.zip

Install Guide

System Requirements

This section describes the system requirements to install and run the software component.

- **Software Architecture:** distributed system, client server architecture
- **Operating System:** every system for which aJava Runtime Environment 7 is available
- **Hardware Requirements:** server: multicore CPU, minimum 4-8 GB RAM, 2 GB disk space; client: multicore CPU, minimum 2-4 GB RAM
- **Local Services and Software:** PostgresSQL DBMS with PostGIS extensions, other types of local services, e.g. **ftp** or **webDAV** for local worldstate data storage depend on the concrete **CRISMA Application** Architecture
- **Remote Services and APIs:** public **OGC** **WMS** for background maps of the **GIS** client may needed, depends on the concrete CRISMA Application Architecture
- **CRISMA Services and APIs:** Data integration and **Simulation Model** Integration services depending on the concrete CRISMA Application Architecture

Installation and Configuration

This section describes how the software component is distributed and can be installed and configured by an administrator that sets a CRISMA Application up.

- **Software Distribution:** server:to be decided, distribution as virtual machine is most likely, client: Java Webstart (start and run from web)
- **Software Installation: IT expert** needed to install the server, client: no installation required
- **Software Configuration:** config files in user profile directory, some options configurable via **GUI**, server: config files, administration tools (user management, database management, import of data, etc.), SQL
- **Software Documentation: ICMM** APIs documented with swagger. Currently no public administration or installation documentation available

Integration and Adaption

This section describes how the software component can be adapted to the needs of a CRISMA Application by a **developer** that implements the CRISMA Application.

- **Integration with other Building Blocks:** CRUD **REST** APIS and catalogues are entirely based on CCIMs which can be created with a visual administration tool.
- **Data Integration:** the **ICMS** does not directly deal with data, however if **business logic** of certain User Interaction Building Blocks (Worldstates **Widget**) is moved to ICMS extension APIs (server actions), custom developments are needed. Server actions need to be developed (java classes), can be plugged in, no changes to core system needed
- **Simulation Model Integration:** the ICMM does not directly with simulation, however if business logic of certain User Interaction Building Blocks (Simulation Interaction) is moved to ICMS extension APIs (server actions), custom developments are needed. Server actions need to be developed (java classes), can be plugged in, no changes to core system needed

Developer's Guide

- ICMM Core API
  https://github.com/crismaproject/icmm/blob/dev/README.md

Downloads:

- cids-custom-crisma (Java ICMM Client UI)
  https://github.com/crismaproject/cids-custom-crisma
- icmm_py (Python ICMM Client Helper)
  https://github.com/crismaproject/icmm_py
- workflow demo (JavaScript Demo Workflow)
  http://jsfiddle.net/mscholl/pmQ5N/6/ (http://jsfiddle.net/mscholl/pmQ5N/6/)
- cids-custom-crisma-server (ICMM CRISMA Implementation)
  https://github.com/crismaproject/cids-custom-crisma-server
- Common library that contains Angular $resource Wrappers for the ICMM
  https://github.com/crismaproject/icmm_js

Endpoint of the the cids **System** preview instance is to demonstrate the **REST** **API** of the **ICMM**.

- ICMM REST Endpoint
  http://crisma.cismet.de/icmm_api (http://crisma.cismet.de/icmm_api)
- Swagger UI Documentation
  http://crisma.cismet.de/icmm_api (http://crisma.cismet.de/icmm_api)

Endpoints of **ICMM** Instances for Reference and **Pilot** Applications

- Pilot C ICMM Endpoint
  http://crisma.cismet.de/pilotC/icmm_api (http://crisma.cismet.de/pilotC/icmm_api)
- Pilot E ICMM Endpoint
  http://crisma.cismet.de/pilotE/icmm_api (http://crisma.cismet.de/pilotE/icmm_api)
- ICMM Endpoint for Economic Impact Model
  http://crisma.cismet.de/economic-impact/icmm_api (http://crisma.cismet.de/economic-impact/icmm_api)
- ICMM Snapshot
  https://repo.cismet.de/simple/cismet-libs-snapshots-local/de/cismet/cids/cids-custom-crisma-server/

Team:

Pascal Dihé

Is responsible for the specification of the Core **CCIM** and the development of the cids **REST** **API**. Provides general integration support and support related to the **application** and extension of the Core CCIM. Installs instances of the **ICMM** for SP5 Applications. Takes care about the hosting of ICMM instances. If required, supports the installation and hosting @**Pilot**-Site. Develops custom search functionality (Search API) and actions (Action API), if required. Take care about ICMM administration tasks. Integrates the ICMM with the Pub/Sub and possibly other **Infrastructure** BBs. Support other partners and pilots in the integration of their Building Blocks and Applications with the ICMM

Possibly definition and implementation of custom search algorithms (Search API) or implementation of simple actions (Action API)

Martin Scholl

Development roles: administration, installation

Installs instances of the **ICMM** for SP5 Applications. Takes care about the hosting of ICMM instances. If required, supports the installation and hosting @**Pilot**-Site. Develops custom search functionality (Search **API**) and actions (Action API), if required. Take care about ICMM administration tasks. Support other partners and pilots in the integration of their Building Blocks and Applications with the ICMM

Thorsten Hell

Coordinates and manages the **ICMM** **REST** **API** development, develops the ICMM Core and provides general development and integration support.

cismap

Version: 2.0

Cismap is one of the software components that implement the Functional **Building Block** **GIS** **Widget** of the **CRISMA Framework**.

Cismap is a Java-based map **application** specialised to access **OGC** compliant services like **WFS** and **WMS** for information from internal and external geospatial data sets. Thereby, **it** supports also powerful visualisation and editing functionalities for OGC compliant geospatial information (e.g. **GML**). The main features of cismap are:

- asynchronous WMS and WFS requests to load several WMS layers in parallel
- dynamic addition of services, including drag and drop of the server URL
- powerful and easy to use geometry editor for the manipulation of geospatial data, e.g. Objects of Interest served by a WFS-T
- customisable user **interface** and the ability to save the layout on a per-user basis
- the representation of geographical features on the map as complex widgets
- customisable print and report generation facilities

cismap Main Screen

Development Status

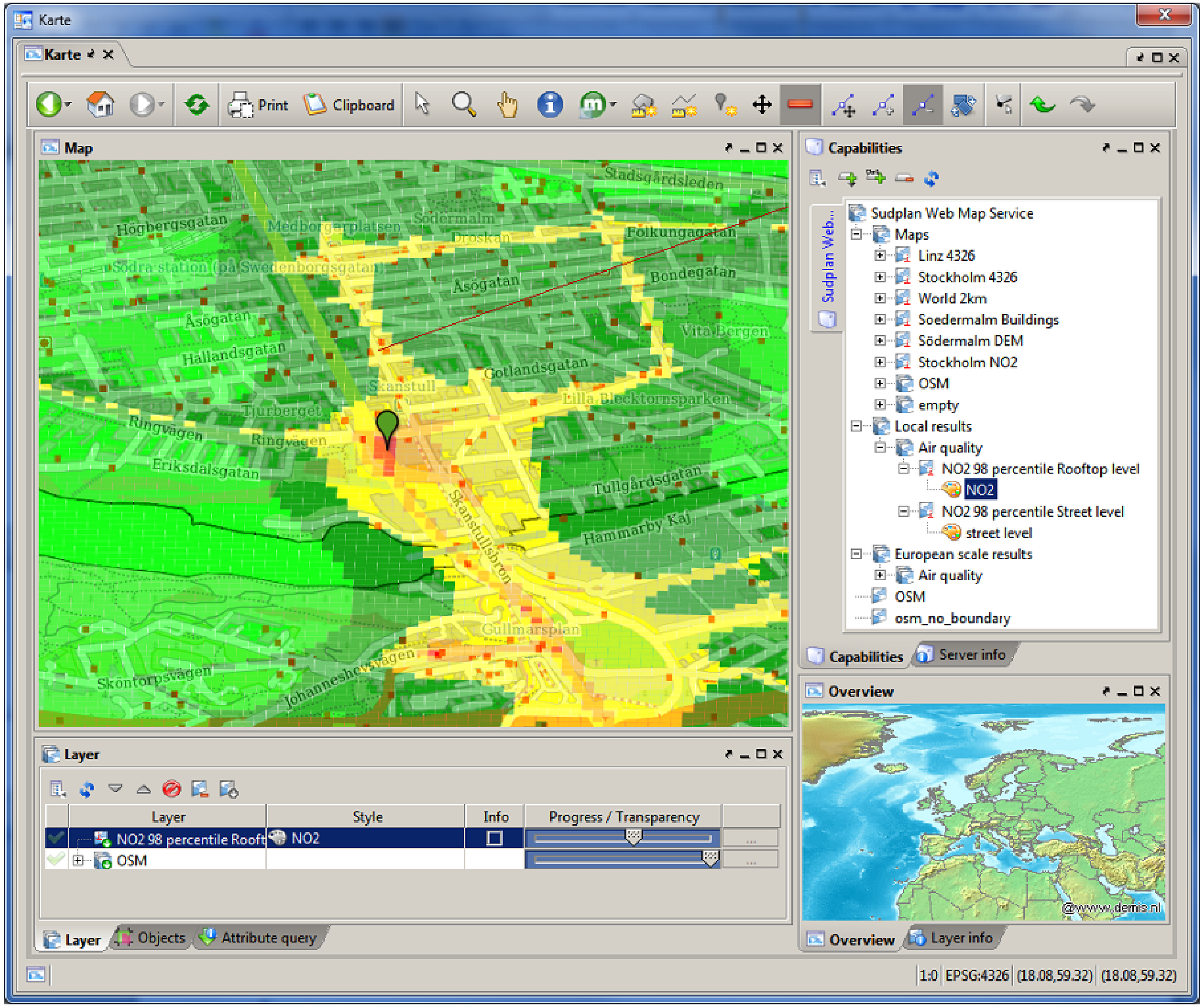
Cismap implements already many of the functionalities requested by the GIS Widget View Block. This includes for example also powerful functionalities to manipulate Worldstate (GIS) data. Additional functionalities added in the context of the CRISMA project include among others the provision of a sophisticated RESTful APIs and the support for different spatial reference systems (EPSG).

Usage in CRISMA

It is important to note, that cismap is a Java application and thus it does not fulfil the technological requirements for Composite UI Modules and Mashable Composite UI Modules (pure HTML5 and JavaScript widgets). However, cismap can be integrated into the cids navigator application which acts as UI Integration Platform. Thus the interaction between User Interaction Building Blocks implemented as Composite UI Modules and cismap can be achieved with help of the cismap **REST** APIs.

Since cismap is not suitable for the integration into a UI **Mashup Platform**, like Wirecloud, a additional Software Components which realise of the GIS Widget Building Block are provided. Cismap is currently used as GIS View **BB** and as part of the Worldstate View BB in the **Technology Demonstrator** Application as shown in the scree shot below.

Worldstate View in the Technology Demonstrator Application realised with help of cismap



Documentation:

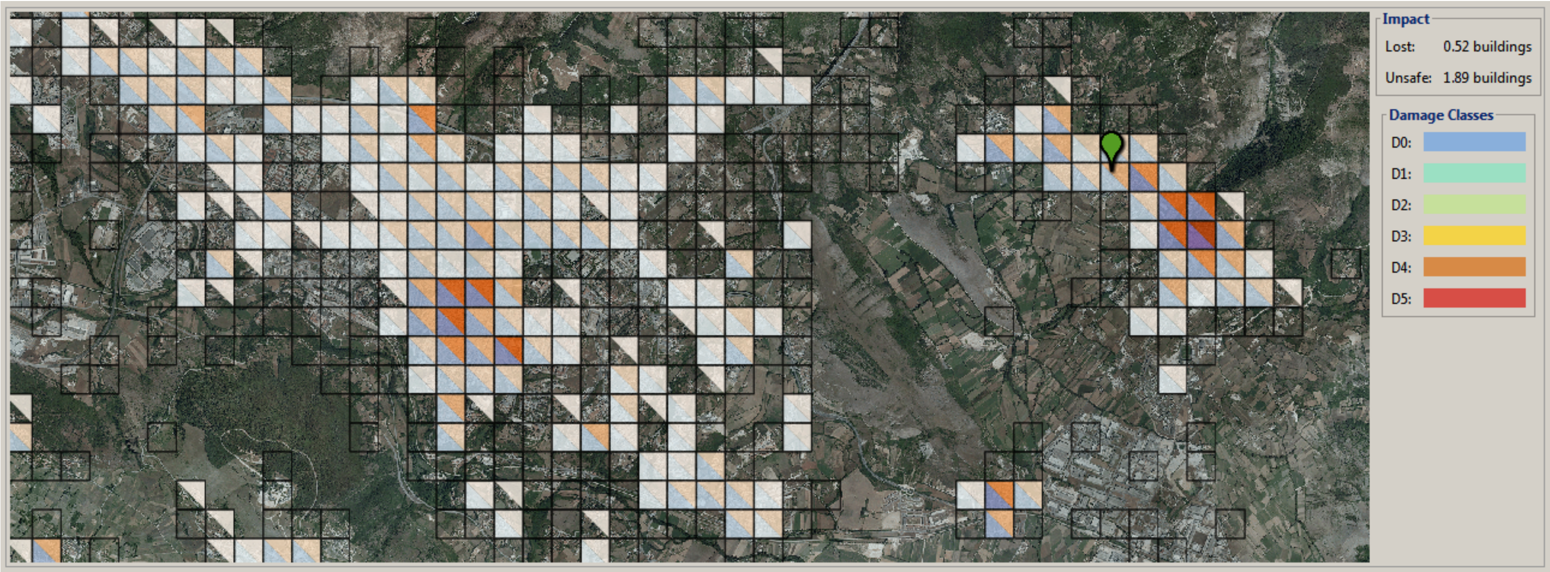
Implementation Plan

Implementation Plan of cismap

- cismap BB Implementation Plan **https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP34/03%20Deliverables/D34.1%20Software%20Components%20(final)/%5BX%5D%20BB%20Implementation%20Plan%20-%20cismap.doc**

Functional Description

- GIS View BB Description **https://crisma-cat.ait.ac.at/bbs/gis-view-building-block**



Downloads:

cismap main sourcecode repository @github

- Sourcecode **https://github.com/cismet/cismap-plugin**

  Cismap realses in the cids software repository.

- cismap realses **https://repo.cismet.de/simple/cismet-libs-releases-local/de/cismet/cismap/cismap-plugin/**

Team:

Martin Scholl

Development roles: administration, configuration, development, installation, integration, support
Provides general development and **Pilot** Integration Support for cismap.

Thorsten Hell

Development roles: configuration, support
Provides **Pilot** Support for cismap and takes care about all configuration tasks, configuration of **WMS**/**WFS** data sources, layers, etc. Takes care about all required adaptations of cismap, e.g. developing functionalities of the **GIS Widget BB** that are needed in a **CRISMA Application** but not yet not supported by cismap. Integrates cismap with the respective BBs and Applications, e.g. Data integration BBs (**OGC** Services), the **ICMM**, etc.

Wirecloud Charts

Authors: **Peter Kutschera**
Denis Havlik
Manuel Warum

Version: 0.5

This **widget** displays stacked bar charts for **indicator** values for a given worldstate.

These values are generated by indicator components and are merely rendered in this context, no calculations are done by the widget itself.

Future iterations will allow to costumize what data is displayed, and how, as well as adding the ability to perform side-by-side comparisons between different worldstates.

Documentation:

Functional Description

This view contains diagrams allowing a comparison between any number of user-picked **world** states from data provided by the **indicator** **service**, such as patient health evolution or **resource** allocation. Data is visualized as colored bar diagrams (stacked, side-by-side). **It** allows side-by-side comparison of data, so that different indicator values can be viewed on the same screen for ease of comparison.

The creation of diagrams is triggered manually through user interaction or automatically by events (eg. worldstate changes).

Downloads:

The source code for this component can be found here: **https://github.com/crismaproject/wirecloud-widgets/tree/master/chart**

Team:

Manuel Warum

Wireclound **developer**

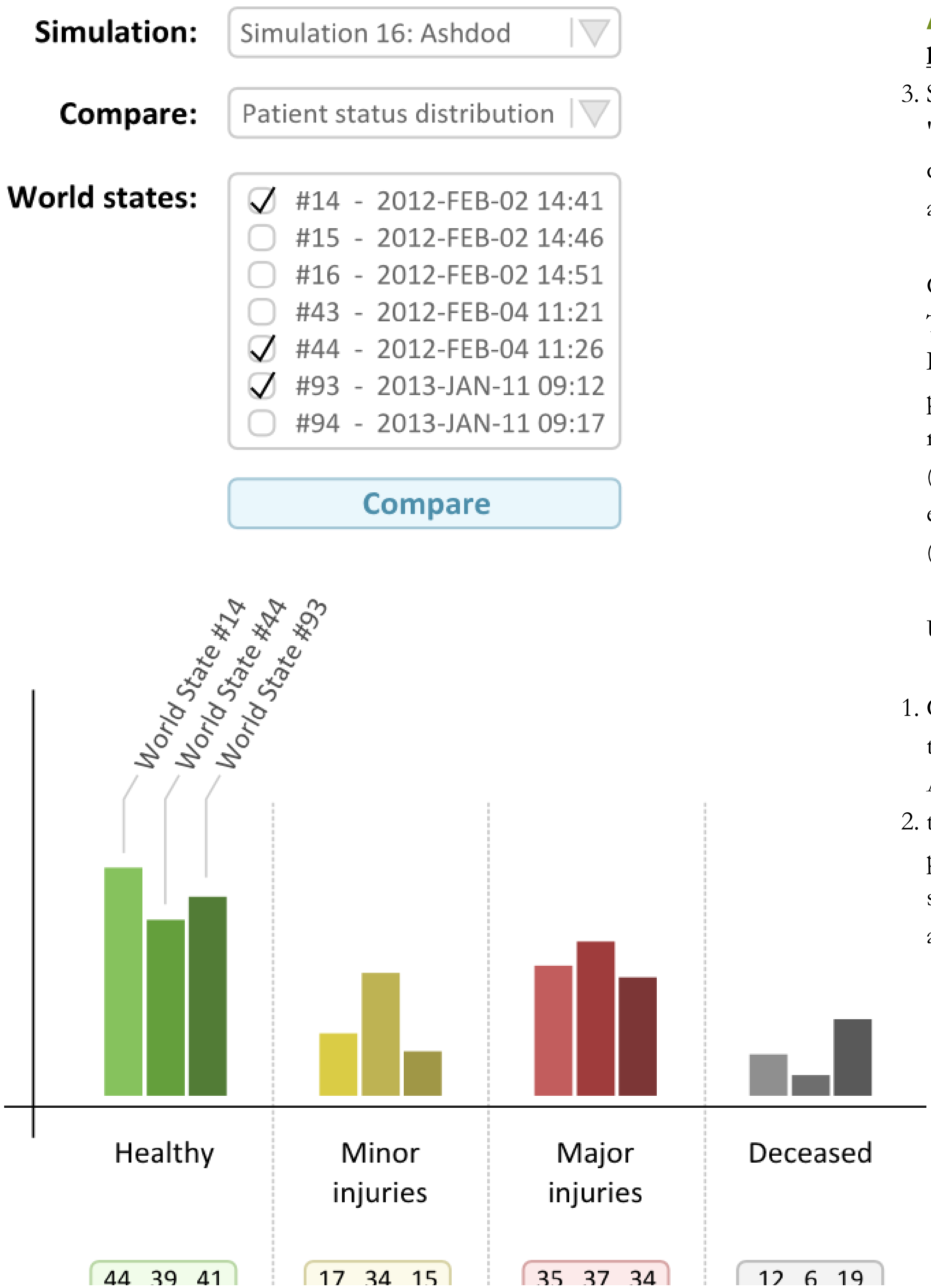
6. Auxiliary software

This sections contains the software components which are listed on the **catalogue** but do not represent reference implementations of the Building Blocks or Models.

CRISMA Catalogue

CRISMA **Catalogue** is a customized web **application** based on **Drupal** 7. Its main purpose is to simplify the **task** of finding documentation on CRISMA Applications, Building blocks, Models and on the software used to realise the CRISMA Models and the Building Blocks. Secondary functions of the catalogue are to:

1. Provide up to date overview tables on various topics. For instance, the **https://crisma-cat**.**ait**.ac.at/relations/**bb** page shows how Building Blocks are related to applications and to components, **https://crisma-cat**.**ait**.ac.at/relations/license2 provides an overview of the licenses used for various software components, and **https://crisma-cat.ait.ac.at/organisations/ait** lists the latest AIT-related contributions to the catalogue.



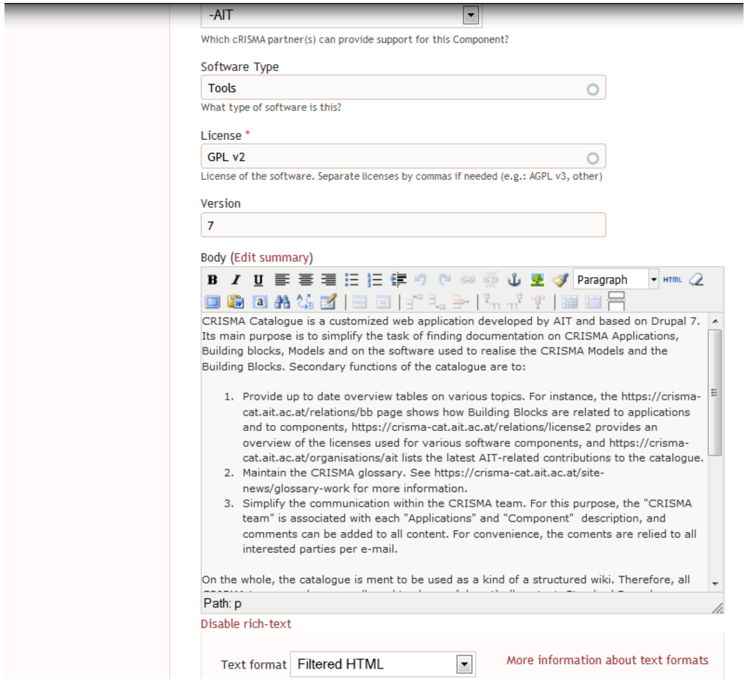
Authors: **Denis Havlik** **Version:** 27Maintain the CRISMA glossary. See **https://crisma-cat.ait.ac.at/site-news/glossary-work** for more information.

3. Simplify the communication within the CRISMA team. For this purpose, the "CRISMA team" is defined for each "Applications" and "Component", and comments can be added to all content. For convenience, the comments are relied to all interested parties per e-mail.

On the whole, the catalogue is ment to be used as a kind of a structured wiki. Therefore, all CRISMA team members are allowed to change (almost) all content. Drupal versioning **system** can be used to find out which team members have participated in document editing, compare the latest document version with previous revisions and if needed revert ack to older versions of the document (**https://crisma-cat.ait.ac.at/site-news/edits-and-revisions**). On top of this, an editorial workflow can be enforced for some content types if needed (**https://crisma-cat.ait.ac.at/site-news/revisions-workflow**).

Unlike wiki:

1. CRISMA Catalogue accepts several types of text entry. The most important ones are the "plain text" which does not allow and fancy formating and the "Filtered HTML". A basic WYSIWYG editor simplifies the formating of the "Filtered HTML" text.
2. the site enforces a content-type specific structure and relations between various pieces of content on the one, and automatically converts the information entered in a single template in a muli-tab browser friendly form. The difference between the edit and presentation views is illustrated below.



Component description appears as a single structured template in edit view (left). The same content is presented as five tabs/pages in a web view (right)

Documentation:

User Guide

The **catalogue** functionality constantly, based on the requests from the **CRISMA** team members. Since no formal specifications exist, **and** the catalogue is meant to be self-explanatory, the best way to get started is either by first taking a look at some of the exisitng content in the edit mode, and then following the **https://crisma-cat**.**ait**.ac.at/node/add link to add a new piece of content.

Descriptions and tutorials pertinent to new and less obvious features, answers to frequently asked questions and other site related news can be found on the site-news page.

- CRISMA site news page
  https://crisma-cat.ait.ac.at/site-news

Downloads:

CRISMA **Catalogue** is a web **application** and currently accessible only to the members of the CRISMA team. Public access to the site is foreseen for the future.

- CRISMA Catalogue site
  https://crisma-cat.ait.ac.at

The source code of **Drupal** and all add-on modules used on **CRISMA** **Catalogue** site is available at the Drupal.org.
The "catalogue" **application** is merely a highly customised installation thereof using a number of relatively complex modules. **AIT** does not intend to develop this site into an easy to install product package, but we are likely to deploy similar set-ups in other projects. If you need something similar, contact Denis.

- Main Drupal site
  https://drupal.org/
- Catalogue Structure on GitHub
  https://github.com/crismaproject/crisma-catalogue

Team:

Denis Havlik

Development roles: administration, configuration, development, requirements, support, specifications
Main **developer** of the **CRISMA** **Catalogue** site. In his **role** of the CRISMA technical coordinator also the main benefiter of the catalogue overview tables.

Peter Kutschera

Development roles: administration, development, installation
Responsible for the administration of the **CRISMA** **Catalogue** server.

Manuel Warum

Development roles: development, testing
Responsible for semi-automated updating of the **widget** documentation (uses RESTful site interfaces).

Docker

Authors: **Denis Havlik**
Peter Kutschera

Integrating the various software components in a prototypic **application** can be a nightmare in a co-funded research project. More often than not, each partner will develop on its own pace, using different tools, programming language and even operating systems. **Service** Oriented **Application architecture** can help in a sense that each partner can also host own services and other partners only need to know the service URI and **API**. However, this practice may lead to development of the services which are difficult to install and therefore unlikely to be used by anyone except the service owner. This is where **Docker** comes in.

Initial idea

Our integration goal in **CRISMA** was to In CRISMA integrate these heterogeneous components in a way which will faciliate re-installing the components and the whole application, while at the same time encapsulating the development environments of the various sub-teams.

We decided to do this by installing the services on virtual machines - one per sub-team and application version. Providing a couple of dedicated virtual machines with root access rights to each of the development teams was easy, and this strategy worked in the sense that:

1. Developers could do their work without jeopardizing the work of the other teams.
2. we could indeed copy the application by re-installing all VMs on a new host and re-configuring the externally accessible services

The drawback of this method is that running (at least) half a dozen of VMs for each application version is very **resource** intensive.

Docker solution

Docker is a lightweight replacement for the virtual machines, which is optimized for running of individual applications. For readers familiar with FreeBSD, docker provides a functionality similar to **jails**, with two important advantages:

- **It** runs under Linux. FreeBSD is **nice**, but a great majority of the **CRISMA partners** uses Linux and not FreeBSD.
- Developing and deploying new Docker images is very easy.

According to the **docker web page**:

*Launched in March 2013, Docker is an open platform for building, shipping and running distributed applications. Docker provides a platform that enables any application to be created and run as a collection of consistent, low-overhead Docker containers that work across virtually any **infrastructure**. This new **model** liberates developers from application and infrastructure dependencies, significantly accelerates the software development lifecycle, and enables substantial improvements in infrastructure cost and efficiency.*

Consisting of Docker Engine, a portable, lightweight runtime and packaging tool, and Docker Hub, a cloud service for sharing applications and automating workflows, Docker enables apps to be quickly assembled from components and eliminates the friction between development, QA, and production environments. As a result, IT can ship faster and run the same application, unchanged, on laptops, data center VMs, and any cloud.

With Docker, developers can build any app in any language using any toolchain. "Dockerized" apps are completely portable and can run anywhere - colleagues' OS X and Windows laptops, QA servers running Ubuntu in the cloud, and production data center VMs running Red Hat.

Sysadmins use Docker to provide standardized environments for their development, QA, and production teams, reducing "works on my machine" finger-pointing. By "Dockerizing" the app platform and its dependencies, sysadmins abstract away differences in OS distributions and underlying infrastructure.

Results

At a time of writing this article (Dec 2014), we have dockerised the **Reference Application for Exercise Support** and the **Reference Application for the Resource Management Training Support**.

As a result:

1. the number of virtual machines which need to be installed in order to run these applications went down from "far too many" to three: one Linux docker host and two virtual images for the windows services of **TTU** and **NICE**).
2. Resource consumption, as well as the administrative overhead are drastically lower than they used to be.
3. Encapsulation is just as good as it used to be, the teams can install any libraries they need on the own docker instances.
4. Installation of the dockerised services is much easier than installing the virtual machines.

In short, we consider the use of docker as "best practice" for CRISMA-like projects.

Documentation:

Tutorial

Docker provides a 10-minute online tutorial shows how to:

- search for and find an image another user built and shared in the Docker Hub Registry
- download and run **it**
- install the 'ping' utility into the container
- commit all your changes, and run a **test** of your updated image.
- push your image to the Docker Hub Registry so that other developers can find and use it.

- Docker Tutorial
  https://www.docker.com/tryit/

FAQ

Docker FAQ is a "must read" for new docker users.

- Docker FAQ
  https://docs.docker.com/faq/

Downloads:

Docker is a native linux **application**. Installing **it** on contemporary linux **system** is as easy as typing "sudo apt-get install docker". In order to use docker on a Windows or OS X machine, you will need to install a virtual linux machine first.

Alternatively, you can install one of the pre-configured images from docker.com site

- Extended installation instructions for Ubuntu linux
  https://docs.docker.com/installation/ubuntulinux/
- Mac OSX installer
  https://docs.docker.com/installation/windows/
- Windows Installer
  https://docs.docker.com/installation/windows/

Team:

Peter Kutschera

Development roles: configuration, integration, support
Main **AIT** integrator and author of all **CRISMA** **docker** images so far (status December 2014)

Denis Havlik

Development roles: coordinaton

Docker proselytizing within **CRISMA**.

GeoServer

Authors: **Peter Kutschera**

Version: 2.4.1

GeoServer is an **open source** software server written in Java that allows users to share and edit geospatial data. Designed for **interoperability**, **it** publishes data from any major spatial data source using open standards.

Being a **community-driven project**, GeoServer is developed, tested, and supported by a diverse group of individuals and organizations from around the **world**.GeoServer is the reference implementation of the **Open Geospatial Consortium** (**OGC**) **Web Feature Service** (**WFS**) and **Web Coverage Service** (**WCS**) standards, as well as a high performance certified compliant **Web Map Service** (**WMS**). GeoServer forms a core component of the Geospatial Web.

Usage in CRISMA

GeoServer is used in several CRISMA Applications to serve geospatial content. The screen shot of the **GIS** View **BB** (cismap) shows a map consisting of static and dynamic map layers GeoServer instance.

Map Layers provided by GeoServer

Documentation:

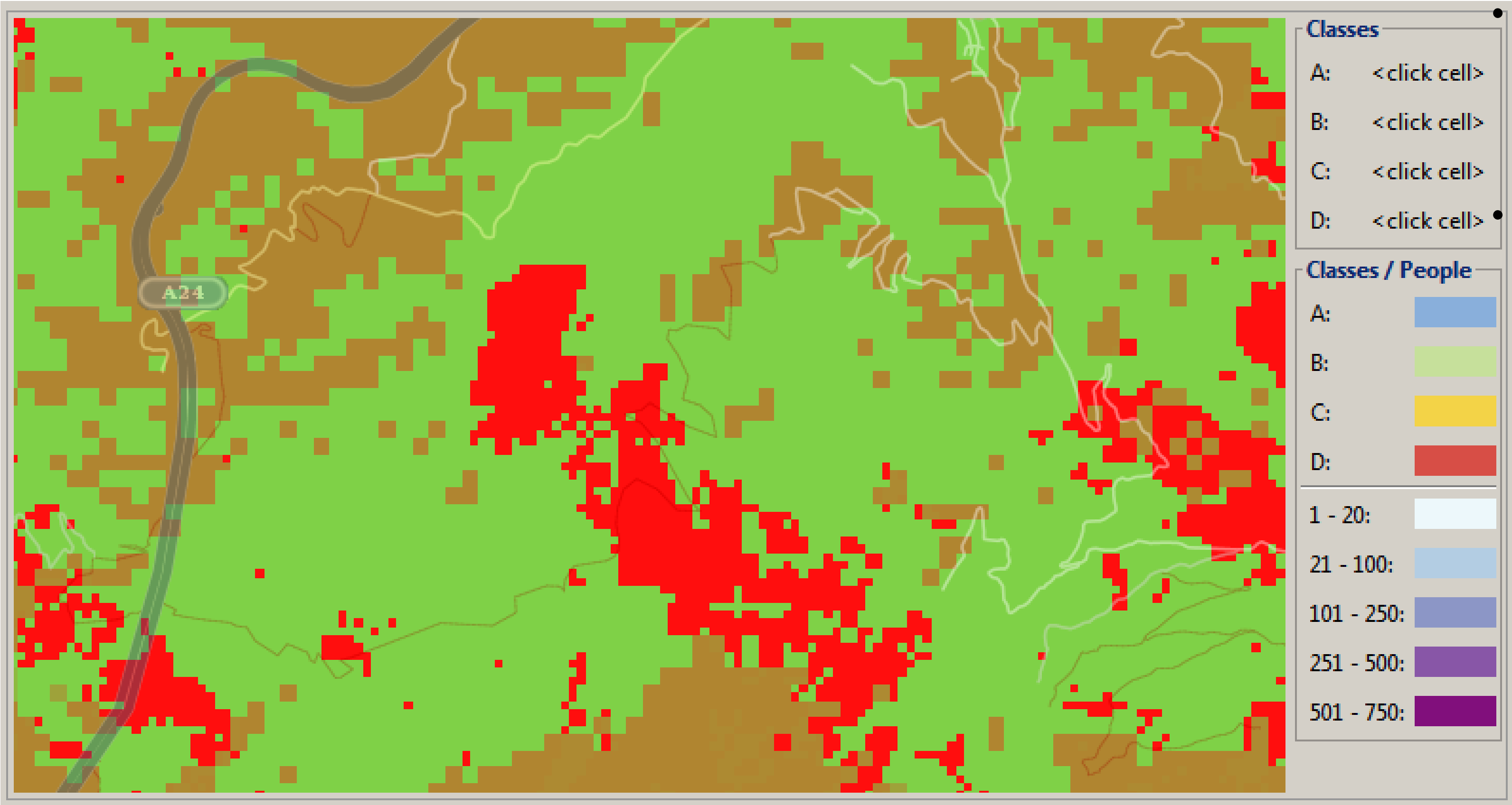
User Guide

- User Manual **http://docs.geoserver.org/stable/en/user/ (http://docs.geoserver.org/stable/en/user/)**
- Project page **http://geoserver.org/display/GEOS/Welcome (http://geoserver.org/display/GEOS/Welcome)**

Downloads:

Downloads are available in different formats.

My preferred method is to install java and tomcat and drop the **GeoServer** WAR file into tomcat.



- Downloads **http://geoserver.org/display/GEOS/Stable (http://geoserver.org/display/GEOS/Stable)**
- Downloads **http://geoserver.org/display/GEOS/Stable (http://geoserver.org/display/GEOS/Stable)**

Team:

Peter Kutschera

Development roles: installation

Installation of Java, Tomcat and **GeoServer**.

Sorry, I can't help with configuration of actual data sources.

MapServer

Version: 6.4.1

MapServer is an **Open Source** geographic data rendering engine written in C. Beyond browsing **GIS** data, MapServer allows you create "geographic image maps", that is, maps that can direct users to content. For example, the Minnesota DNR Recreation Compass provides users with more than 10,000 web pages, reports and maps via a single **application**. The same application serves as a "map engine" for other portions of the site, providing spatial context where needed.

MapServer was originally developed by the University of Minnesota (UMN) ForNet project in cooperation with NASA, and the Minnesota Department of Natural Resources (MNDNR). Later **it** was hosted by the TerraSIP project, a NASA sponsored project between the UMN and a consortium of land management interests.

MapServer is now a project of OSGeo, and is maintained by a growing number of developers (nearing 20) from around the **world**. It is supported by a diverse group of organizations that fund enhancements and maintenance, and administered within OSGeo by the MapServer Project Steering Committee made up of developers and other contributors.

- Advanced cartographic output
  - Scale dependent **feature** drawing and application execution
  - Feature labeling including label collision mediation
  - Fully customizable, template driven output
  - TrueType fonts
  - Map element automation (scalebar, reference map, and legend)
  - Thematic mapping using logical- or regular expression-based classes
- Support for popular scripting and development environments
  - PHP, Python, Perl, Ruby, Java, and .NET
- Cross-platform support
  - Linux, Windows, Mac OS X, Solaris, and more
- Support of numerous **Open Geospatial Consortium** (**OGC**) standards
  - **WMS** (client/server), non-transactional **WFS** (client/server), WMC, **WCS**, Filter Encoding, SLD, **GML**, **SOS**, OM
- A multitude of raster and vector data formats
  - TIFF/GeoTIFF, EPPL7, and many others via GDAL
  - ESRI shapfiles, PostGIS, ESRI ArcSDE, Oracle Spatial, MySQL and many others via OGR
- Map projection support
  - On-the-fly map projection with 1000s of projections through the Proj.4 library

(Source: **http**://mapserver.org/about.html#about)

Usage in CRISMA

Although MapServer is foreseen as a realisation of the Data Integration **BB**, it is currently not used in any **CRISMA Application** in favour of **GeoServer**.

Documentation:

Publications

Full documentation can be found at **http://mapserver.org/documentation.html (http://mapserver.org/documentation.html)** .

Downloads:

Mapserver source code can be downloaded from github as well as from the osgeo repository. The development is hosted on the github,

- Mapserver - all downloads
  http://mapserver.org/de/download.html (http://mapserver.org/de/download.html)

- Mapserver Github repository and license
  https://github.com/mapserver/mapserver

Mapserver binaries can be downloaded from several sources. The pre-compiled binaries are available for Windows, several Linux flavours (Ubuntu, Debian, RH) and Mac OS. MapServer is also included in OSgeo and FWtools packages.

- Mapserver - all downloads
  http://mapserver.org/de/download.html (http://mapserver.org/de/download.html)
- Mapserver nightly builds
  http://www.gisinternals.com/sdk/ (http://www.gisinternals.com/sdk/)

Team:

Peter Kutschera

Development roles: mediation
Can give some general help.

PyShp

Version: 1.2.0

The Python Shapefile Library (pyshp) provides read and write support for the Esri Shapefile format. The Shapefile format is a popular Geographic Information **System** vector data format created by Esri.
Pyshp is compatible with Python 2.4-3.x.

Documentation:

User Guide

Information on Pyshp (utilization, example, downloads)

- Pyshp documentation
  https://code.google.com/p/pyshp/

Downloads:

- Source code
  http://pyshp.googlecode.com/svn/trunk/shapefile.py (http://pyshp.googlecode.com/svn/trunk/shapefile.py)

Team:

Mehdi Pierre Daou

Development roles: modelling
use to **model** the **vulnerability** of dikes

PyWPS

Authors: **Arnaud De Groof**
Peter Kutschera
Version: 3.2.1

PyWPS (Python **Web Processing Service**) is an implementation of the **Web processing Service (http://www.opengeospatial.org/standards/wps) standard** from **Open Geospatial Consortium (http://opengeospatial.org)**. Within **CRISMA**, the PyWPS is (can be) used to make legacy models and other software available within the **CRISMA Framework**. In order to do so, one has to extend the basic PyWPS classes and introduce invocations of the own software. By nature of GPL, this extension is automatically licensed under GPL as well, but the invoked software (e.g. models) is not linked to PyWPS and can be published under a different license.

What is PyWPS not

- PyWPS is no analytical tool or engine. **It** does not perform any type of geospatial calculation.
- PyWPS is not special **XML** parser or generator. It does not validate your GMLs against given schemas (yet), it does not build **GML** from Python objects.
- It is not complicated. Or, it should not be.

Work on PyWPS has been started in Mai 2006 as a project supported by **DBU (http://dbu.de)** and is currently (2009) mainly sponsored by **HS-RS (http://www.bnhelp.cz)**. It offers an environment for programming own processes (geofunctions or models) which can be accessed from the public. The main advantage of PyWPS is, that it has been written with native support for **GRASS GIS (http://grass.itc.it)**. Access to GRASS modules via web **interface** should be as easy as possible.

Documentation:

Developer's Guide

PyWPS provides two pieces of documentation: a "tutorial/course" and a "documentation".

Both of them combine several documentation aspects, includign the instalation and administration, but the main target of these documents are the developers which wish to **WPS**-enable their own software using PyWPS.

- PyWPS tutorial
  http://pywps.wald.intevation.org/documentation/course/ (http://pywps.wald.intevation.org/documentation/course/)

- PyWPS 3.2.0 Documentation¶
  http://pywps.wald.intevation.org/documentation/pywps-3.2/ (http://pywps.wald.intevation.org/documentation/pywps-3.2/)

Specifications

Web Processing Service is a well-established **standard** of the **Open Geospatial Consortium** (**OGC**). The specifications of the service interfaces and data models are available **free of charge** from the OGC web site.

- OGC WPS specifications
  http://www.opengeospatial.org/standards/wps (http://www.opengeospatial.org/standards/wps)

Downloads:

PyWPS is available for download as a tarball, debian or RPM package from the wald **intevation** download server.

- PyWPS downloads
  http://pywps.wald.intevation.org/download/index.html (http://pywps.wald.intevation.org/download/index.html)

Docker image providing **WPS** to start PilotC and **Pilot** E models.

The same image is used for both pilots since the URL of **model** **service** behind WPS is set as **parameter** when starting the docker container.

There is no need to download the docker image, this wil be done automatically when needed.

Usage:
docker run -P -d --name c_resourcemodel --env MODEL_ENDPOINT='**http**://192.168.120.40/Startup/startup.aspx' peterkutschera/**crisma**-**resource**_model_wps

- Docker image source code on github
  https://github.com/crismaproject/crisma-resource_model_wps
- Docker image on hub.docker.com
  https://registry.hub.docker.com/u/peterkutschera/crisma-resource-model-wps/

Team:

Peter Kutschera

Development roles: development, support
Help installing the **service** and implementing processes, e.g. connecting to a **simulation model**

Arnaud De Groof

Development roles: support
Provides support for the use of **PyWPS** (installation, configuration, models integration)

Situator Training System

Version: 7.3.0.0

Situation Management is a solution approach that comprises tools and methods for coordinating the interaction between people, technologies, and responses. **NICE** Systems is a major player in the Situation Management market with over 20 years in experience in the industry and expertise in a wide variety of industries such as seaports, airports, railways, banking, government, telecom, utilities, military, manufacturing, etc.

Designed for the unique needs of Tactical Training, **NICE** Security Solutions for Training and **Simulation** combines a powerful set of tools and technology that merge all cameras, sensors, communication systems, data sources and operating procedures into a single unified platform to secure the entire transportation **infrastructure**. In the process, public safety and security agencies not only gain valuable insight into everyday operational issues, they can substantially optimize them for measurable and valuable improvements and cost savings.

Technical Information:

Situator is a mission critical open Client/Server system that runs on high-performance Windows servers, capable of managing even the largest multi-site implementations.

NICE Situator is designed to support a wide range of workstations and display monitor layout options. Each client machine runs the Control Room software and can be connected to several different display monitors. Depending on local hardware setup, every Control Room workstation can be manipulated to display various **application** screens, or “views”. The most commonly displayed views are:

- Incidents view – interactive log of events and notes with procedural check lists.
- Maps view – dynamic presentation of objects on maps and images of the facilities with drill-down functionality.
- Video view – multiple displays of video sources with click & drag functionality.
- **Access Control** view – log of access control events with advanced filtering functionality.
- Messages view – lets users send, receive, and manage e-mail messages when responding to situations.

A typical Control Room workstation will be configured to display three or four NICE Situator views which will usually present at least the Maps view, Incidents view, and the Video view. This layout displays to users a comprehensive view answering the "what happened", "where **it** happened" and "how to respond" of situation management.

Situator distributes functionality between its servers and the various user clients. By using the computing power that is distributed over the network, server overhead is greatly reduced as well as hardware requirements. In addition, **LAN**/**WAN** overhead is also greatly reduced by this **architecture** since the client can perform processing activities on data that already resides on it. Consequently, no additional traffic is added to the network, freeing bandwidth for other applications.

The system is built for growth and is scalable for all operation sizes. The system can be installed on a single server without any redundancy or in a full production clustered environment.

A high level block diagram of the software architecture follows.

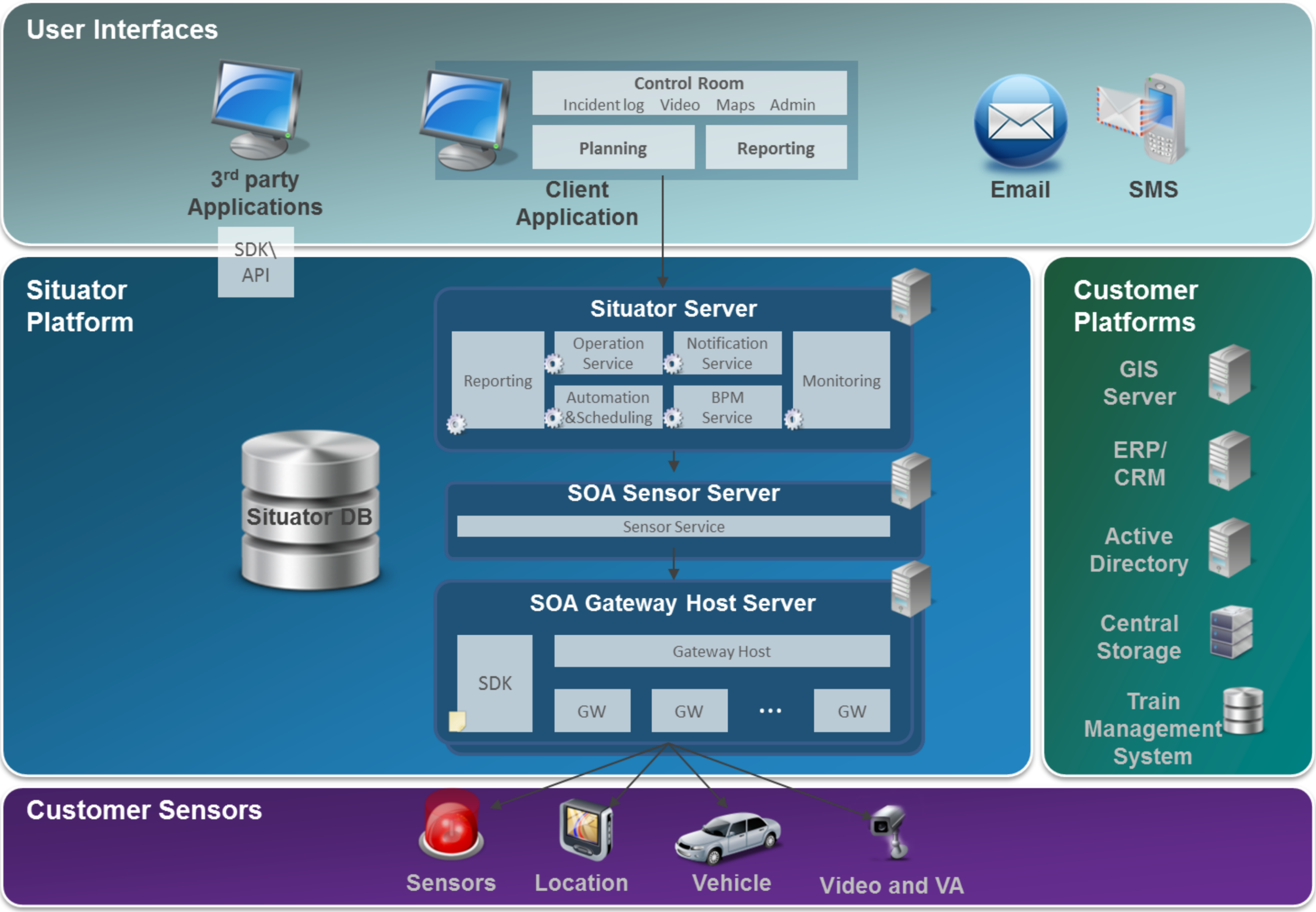
High Level Software Architecture Diagram

Situator components shown in the high level software architecture diagram are described in the following list:

- **Situator Supervision server:** manages the internal communication, routing, and processing of all information handled by the Situator system, including edge device alerts and commands, notifications, escalation requests, **incident** logic, and **event** triggers. The Situator server also handles all database related operations throughout the Situator system.
- **Control Room: a**cts as the client side in the Situator system. It provides unified monitoring and interactive control of all connected technologies with a real-time **GIS** (Geographic Information System) **interface**, a matrixed display for viewing live and recorded video, quick launch buttons for urgent and frequently used procedures and a full-featured, multimedia messaging module.
- **SOA Sensor server:** responsible for all sensor-related information and events in the Situator system. As such, it interfaces between the Situator gateways and the Situator server by giving the gateways access to Situator core functionalities and giving the Situator server access to specific sensor functionalities.
- **SOA Situator Gateway: a** software component that interacts with a 3rd party external device and interfaces between the functionalities provided by the external device and Situator core components. The Gateways are vendor agnostic, supporting any third party system using APIs and exposing a well defined **service** based interface.
- **Notification server:** the publisher responsible for distributing notifications originating from the Situator server to the various Situator clients, acting as subscribers. It facilitates and coordinates on-demand and automated communication between the clients and the server.
- **Orchestration engine:**
  - **BPM Rules engine** is a powerful rule engine for defining activation rules. If conditions defined in the rules are met, alarms are triggered activating adaptive **response** plans automatically or recommendations are sent to operators to activate plans on-demand
  - **The BPM Workflow engine** is a highly flexible and robust engine that correlates different events according to a number of parameters enabling complex and adaptive response planning. The workflow consists of a sequence of activities or steps used to implement a continuous and dynamic business process and provide automatic response.

Documentation:

Implementation Plan

- 
User Interfaces
3rd party Applications
SDK\ API
Control Room
Incident log Video Maps Admin
Planning
Reporting
Client Application
Email
SMS
Situator Platform
Situator Server
Reporting
Operation Service
Notification Service
Automation &Scheduling
BPM Service
Monitoring
Situator DB
SOA Sensor Server
Sensor Service
SOA Gateway Host Server
SDK
Gateway Host
GW
GW
...
GW
Customer Platforms
GIS Server
ERP/ CRM
Active Directory
Central Storage
Train Management System
Customer Sensors
Sensors
Location
Vehicle
Video and VA

Implementation Plan
https://workspace.vtt.fi/sites/eu_crisma/Shared%20Documents/Sub-Projects/SP3/WP34/01%20Work%20in%20Progress/Software%20Components/Resource%20Management%20Tactical%20Training/BB%20Implementation%20Plan%20-%20Resource%20Management%20Tactical%20Training.doc

User Guide

- Situator on NICE website
  http://www.nice.com/situation_management (http://www.nice.com/situation_management)

Downloads:

- NICE Contact Information
  http://www.nice.com/contact-us (http://www.nice.com/contact-us)

Team:

Oren Deri

Development roles: architecture, configuration, development, integration

Annex Documents

This section contains various materials which may be helpful for the **CRISMA** team or for the visitors but do not fit in the main story flow of the CRISMA **catalogue** book. Some of them are summary documents which are generated as needed. Generation can take a long time, so please be patient. Once the page is generated, **it** will remain in the cache for a while and the subsequent page loads will be fast.

Article Authors

This page summaries the relations between articles and article authors. "Article editor" is the person who "owns" the article on the site.
This is usually also the principal author of the articles.

The table provides a convenient way for discovering the articles whish haven't been assigned to authors and to contact the person responsible for fixing this issue. It can also be convenient for other purposes, e.g. finding out which names to include in citations for multiple articles on one place.

Type
- Any -

Title

search by title word(s). Order of words does not matter.

Corresponding Author

Enter a comma separated list of user names.
Filter by article editor

Sort by
Title

Order
Asc

Apply

Corresponding Aothor	Authors	Title	Last update
Oren Deri	Oren Deri, Denis Havlik, Chaim Rafalowski	Accidental spillage from a container at large city port (Israel)	2015-02-18
Peter Kutschera	Peter Kutschera	BasicIndicators	2014-12-16
Denis Havlik	Denis Havlik	BB to be defined	2014-12-12
Pascal Dihé		cids Navigator	2014-08-01
Pascal Dihé		cids System	2014-11-04
Pascal Dihé		cismap	2014-07-31
Arnaud De Groof	Mehdi Pierre Daou, Marianne Grisel, Armonie Cossalter	Coastal submersion defense for the Charente Maritime region (France)	2015-03-13
Denis Havlik	Denis Havlik	Component to be defined	2014-12-10
Johannes Sautter		Configuration Component	2014-11-14
Markus Jähi	Jussi Yliaho, Markus Jähi	CRISECON GUI	2014-12-15
Jussi Yliaho	Jussi Yliaho, Markus Jähi	CRISECON Service	2014-12-15
Denis Havlik	Denis Havlik	CRISMA Catalogue	2014-12-17
Martin Scholl		CRISMA Technology Demonstrator	2014-07-30
Hanna Honkavuo	Hanna Honkavuo, Ari Kosonen, Markus Jähi, Kalev Rannat	Cross Border Emergency Crisis (Finland)	2015-02-16
Denis Havlik	Denis Havlik, Peter Kutschera	Docker	2014-12-16
Kalev Rannat		Dynamic Map Agents	2014-11-05
Martin Scholl		Earthquake and forest fire application (Italy)	2014-08-01
Hermann Huber	Hermann Huber, Denis Havlik	Emikat	2014-12-15
Frank Jonat		Exercise Worldstate Data Chart Widgets (Wirecloud)	2014-11-05
Peter Kutschera	Peter Kutschera	GeoServer	2014-12-10

Corresponding Aothor	Authors	Title	Last update
Ari Kosonen		Insta EvacSim	2014-11-28
Ari Kosonen		Insta Response Preparedness Planner	2015-02-23
Manuel Warum	Manuel Warum	Integration of Wirecloud into native applications	2014-12-12
Pascal Dihé		Leaflet.js	2014-08-01
Armonie Cossalter	Armonie Cossalter	Life Safety Model 2D	2015-03-13
Pascal Dihé		MapServer	2014-07-31
Martin Scholl		Multi Criteria Analysis and Decision Support Widget (Java)	2014-08-01
Pascal Dihé		Multi Criteria Analysis and Decision Support Widgets (JavaScript)	2014-11-07
Martin Scholl	Martin Scholl, Denis Havlik, Johannes Sautter, Sandra Frings	Multi hazard site (Germany)	2015-02-10
Oren Deri		OOI World State Repository	2014-05-02

1 2 3 4 next › last »

BB to be defined

Authors: Denis Havlik

BB Description:

This is just a place-holder to be used in **application** descriptions. The idea is to refer to this **BB** and explain what is missing as a TODO reminder. We should occasionally check for applications still declaring to use this BB as a part of QA.

Documentation:

User Guide

The "Site news" doubles as a kind of FAQ for all questions related to use of this site.

- Catalogue "site news"

https://crisma-cat.ait.ac.at/site-news

Realization:

Component to be defined

This is a placeholder which allows us to define building blocks and **CRISMA** models for which no components have been defined yet.

Building block relations

This page summaries the relations between Building Blocks, Applications and components. It provides a convenient way to find out which applications use a certain BB, as well as to find out which components are used to realize a certain BB.
"UNDEFINED" in the "Used in Applications" column is a bad news: it either means that no application is using a BB or that our documentation is outdated.
Empty "components" column is also a bad news: it either means that no realization of the BB exists or that our documentation is outdated.

Title	Use in applications	Used components

Title	Use in applications	Used components
Agent Oriented Simulation Models The **Agent**-Oriented **Simulation** Models **Building Block** serves for the development of dynamic maps – specific (individual-based) simulation models composed of interacting software agents situated in some environment. This Building Block comprises a collection of generic agents and interaction templates for dynamic map construction, for describing, defining and specifying points, areas and layers of interest. **It** provides furthermore a dynamic-map-based user **interface** for interaction and visualization. Thus, it can be considered both an Integration and User Interaction Building Block.	**Reference Application for the Nordic Winter Storm Domain** (**Integrated**): **Agent** Oriented **Simulation** Models **BB** provides an agent-based interaction and visualisation platform for the models used in the **Reference Application**. **Cross Border Emergency Crisis (Finland)** (**Integrated**): See related **reference application** **Reference Application for Resource Planning** (**Integrated**): This **Reference Application** integrates **simulation** with the help of this view. **Reference Application for the Resource Management Training Support** (**Integrated**): This **building block** is the platform on which all **OOI**-related models used in this applications are realized. **It** governs the development of all objects of interest (OOIs), as well as the interactions between them, taking into account the initial **world state** and the subsequent decisions.	**Dynamic Map Agents** (**Proprietary**, **using Google Maps API for non-commercial purposes**): This is a description of a software candidate that offers tools for **agent**-based simulations for the implementation of a functional **Building Block** (Agent-Oriented **Simulation** Models Building Block), together with **OOI** **World State** Storage **Service** (**WFS**) **BB** (**NICE**), **Resource management** integration and deployment (**WPS**) BB (CASS), Indicators BB (**AIT**), Tactical RM Training BB (Situator integration) (**NICE**) and RM Training Dispatch and Monitor View (**AIT**), RM Training Simulation **Scenario** Setup View (AIT), RM Training Indicators and Statistic View (AIT).

Title	Use in applications	Used components
BB to be defined This is just a place-holder to be used in **application** descriptions. The idea is to refer to this **BB** and explain what is missing as a TODO reminder. We should occasionally check for applications still declaring to use this BB as a part of QA.	UNDEFINED (**Todo for V2**): A Component that supports the manipulation of **world state** data is required. Given the different types of data types available **it** is questionable whether such a component can be generic and reusable and thus be provided as **Building Block** of the **CRISMA Framework**. Thus, the development of such a manipulation component will most likely be performed during the development of the respective **Pilot** **Application**. UNDEFINED (**Todo for V1**): This **application** requires several bits and pieces which aren't available at the **catalogue** yet. This includes for instance: 1) "patients view" 2) some **model** which will execute the preparednes plans so that we can **test** them There may be more..	**Component to be defined** (**To be defined**): This is a placeholder which allows us to define building blocks and **CRISMA** models for which no components have been defined yet.

Title	Use in applications	Used components
Cascade Events Configuration and Interaction View The **Cascade Effects** View is a User interaction Building that allows a user to configure and run a Cascade Effects **Scenario**. The user can select a triggering **event** (for example, an earthquake) and provide may either specify the characterisation of the event (**Simulation Control Parameter**) and thus initiate a new **Simulation Model** Run for this particular event, or select (if available) the output of a past event or an event already simulated.	**Reference Application for the Earthquake and Forest Fire Domains** (**Integrated**): This **Building Block** allows the user to choose a **path of analysis** from a predefined **transition matrix**. This **Widget** is similar to the **Simulation Model** interaction Widget. A respective Functional Description of **Cascade Events** Building Block is described in V2 of the **CRISMA Framework** **Architecture** (D32.2). **CRISMA Technology Demonstrator** (**Integrated**): The java-based Implementation of the **Cascade Events** Configuration View is fully integrated in the **Technology Demonstrator**.	**Cascade Events Configuration Widget** (**LGPL v3 (permissive OS)**): This is a java-based prototype of the **Cascade Events** Configuration and Interaction View. **It** is mainly used in the **CRISMA** **Technology Demonstrator**.

Title	**Use in applications**	**Used components**
Data Integration The Data Integration **Building Block** provides components that can be used to easily serve data in a **CRISMA**-compliant (**OGC** open **standard** compatible) way so that other Building Blocks may use them for further processing like viewing or editing. That way data can be made accessible for CRISMA components.	**Reference Application for Exercise Support** (**Integrated**): The **exercise** capture data is served by a **REST**-enabled **service** backend. Map data is served by Google (TM). **Reference Application for the Coastal Submersion Domain** (**Integrated**): This **Building Block** is used to access geospatial data of the various Worldstates of the **Reference Application**. The geospatial results of the Coastal Submersion **Model** and the **Evacuation** Model are published thanks to Data Integration Building Block. **CRISMA Technology Demonstrator** (**Integrated**): **WMS**/**WFS** Services are used to provide the relevant data. **Reference Application for the Earthquake and Forest Fire Domains** (**Integrated**): This **Building Block** is required for accessing respective Data Slots of the Worldstate of the **Application**. **It** is a very generic **BB** which is realised entirely by existing software (**GeoServer**, MapServer). Therefore its configuration is in general performed during the development of the respective **Pilot** Applications. Installation and Configuration of the Building Block Software Component (GeoServer) for the **Reference Application** and provision of representative **test** data is performed by **CIS** and **AMRA**. UNDEFINED (**Optional**): This **Building Block** is required for accessing respective Data Slots of the Worldstate and to integrate external data sources, such as legacy systems to the **Application** environment. **Preparedness** Plan **BB** integrates to the **World State** Repository via the Data Integration BB.	**MapServer** (**MIT (permissive OS)**): MapServer is an **Open Source** geographic data rendering engine written in C. Beyond browsing **GIS** data, MapServer allows you create “geographic image maps”, that is, maps that can direct users to content. For example, the Minnesota DNR Recreation Compass provides users with more than 10,000 web pages, reports and maps via a single **application**. **GeoServer** (**GPL v2 (controlled OS)**): **GeoServer** is an **open source** software server written in Java that allows users to share and edit geospatial data. Designed for **interoperability**, **it** publishes data from any major spatial data source using open standards.

Title	Use in applications	Used components
Economic impacts analysis view This **Building Block** is an **economic evaluation** tool to support **crisis management** and to be used in the **preparedness** phase for planning and training purposes. The main objective of an economic evaluation in **CRISMA** is: - to present the economic impacts arising from crises (ex post performance) and - to assess different **mitigation** proposals and their costs/benefits (ex ante planning).	UNDEFINED (**Todo for V2**): The **Economic Impact** Calculation View is used to help decision makers and experts to perform cost/benefits analysis. The economic **losses** are calculated with **ICC** functions. **Reference Application for the Nordic Winter Storm Domain** (**Integrated**): The **Economic Impact** Calculation View is used to input the data used in calculations, to initiate the calculation processa and to present the calculated results to the **end user**, to be used to support decisions related to **response** tasks and their priorities. **Cross Border Emergency Crisis (Finland)** (**Integrated**): **TBD**	**CRISECON Service** (**Proprietary**): CRISECON **Service** implements, together with the related UI component CRISECON **GUI**, the CRISECON **model** developed in SP4. The key functionalities CRISECON Service are: **CRISECON GUI** (**Proprietary**): CRISECON **GUI** (see figures below) implements, together with the related web-**service** component CRISECON Service, the CRISECON **model** developed in SP4.

Title	**Use in applications**	**Used components**
GIS View Building Block The **GIS** View is a User Interaction **Building Block** that enables the visualisation and manipulation of geospatial data. Geospatial data plays a predominant **role** in all **crisis management** related applications, because most if not all information playing a role in crisis management has a geospatial component.	**Reference Application for the Coastal Submersion Domain** (**Integrated**): The **GIS** View is used in the **Reference Application** to display results of **Simulation** Models and for manipulating Worldstates. **SPB** has developed a GIS View Software Component for the Reference Application in compliance with requirements for the Coastal Submersion **Scenario**: - **WMS**/**WFS** with different type of geographic resources - Popup to display dikes **vulnerability** - Time slider - Modify the area of interest (AOI) - Modify the opacity of the layers - Create new features aiming to Modify dikes' vulnerability **Reference Application for the Earthquake and Forest Fire Domains** (**Integrated**): The **GIS** **Widget** **Building Block** is integrated in the **Reference Application** and is responsible for visualising data items that have a spatial context. **CRISMA Technology Demonstrator** (**Integrated**): The Java-based implementation of the **GIS** **Widget** **BB** (cismap) is integrated into the **demonstrator**. UNDEFINED (**Optional**): **GIS** Widged **BB** may be used in the **Reference Application** to visualise the **simulation** result. Alternatively, the native GIS platform of **Agent** Oriented Simulation Models BB is used.	**cismap** (**LGPL v3 (permissive OS)**): Cismap is one of the software components that implement the Functional **Building Block** **GIS** **Widget** of the **CRISMA Framework**. **Leaflet.js** (**BSD (permissive OS)**): Leaflet.**js** is one of the software components that implement the Functional **Building Block** **GIS** **Widget** of the **CRISMA Framework**. **It** is used "as is" and thus not extended or modified for CRISMA. **OpenLayers** (**BSD (permissive OS)**): OpenLayers is one of the software components that implement the Functional **Building Block** **GIS** **Widget** of the **CRISMA Framework**. **It** is used "as is" and thus not extended or modified for CRISMA. **Wirecloud OOI GIS Map** (**AGPL v3 (controlled OS)**, **Negociable**): A map **widget** showing **Object Of Interest** (**OOI**) on a background map.

Title	Use in applications	Used components
Indicator Building Block The Indicators **Building Block** (formerly Algebraic **Evaluation**) is a component that allows definition, storage and evaluation of "simple" algebraic models in order to evaluate consequences of decisions made by **CRISMA** users during a training session. In this context "simple" means that there is no large number crunching needed. Usage is for calculation of key performance indicators, heuristic models and to some extend the implementation of models actually running in interactive **GIS** environments in order to make them usable as services in the **CRISMA framework**.	UNDEFINED (**Optional**): The **Indicator** **BB** is used to calculate non-complex Key Performance Indcators based on the information s UNDEFINED (**Todo for V2**): **Indicator** **BB** is used to calculate KPIs, based on information stored in the **World State** Repository. **Reference Application for Exercise Support** (**Integrated**): Used to calculate **indicator**/**criteria** data **Reference Application for Resource Planning** (**Integrated**): The **Indicator** **Building Block** is used in this **application** to calculate all indicators needed for the ressource planning **simulation** analysis. Indicators are e.g. the duration of the triage or of the **evacuation** of all patients or the number of vehicles or responders per patient. The results of the indicators calculation are stored in the **ICMM** worldstates generated during a **simulation run**. They are visualized using the Multi **Criteria** Analysis and Decision Support View and the **Scenario** Analysis and Comparison View. UNDEFINED (**Todo for V2**): The **Indicator** **Building Block** is used to calculate indicators for each worldstate of the **Reference Application**. The indicators for the **pilot** B must be define. **Reference Application for the Earthquake and Forest Fire Domains** (**Integrated**): The Indicators **Building Block** is required for calculating Indicators, including simple Economic Indicators. For this reason, no dedicated Economic Impacts **Model** is used in this **Reference Application**. **CRISMA Technology Demonstrator** (**Integrated**): Calculation of Indicators is realised within the **Technology Demonstrator**.	**BasicIndicators** (**GPL v2 (controlled OS)**): BasicIndicators is a reference implementation of the **Indicator** **BB**. **It**'s **role** is to calculate the some of the indicators needed in **CRISMA** applications and store the results in the **World State**. **Emikat** (**Proprietary**): Emikat is a **data management** and modelling **system** for spatially resolved data. The system allows the integration of emission-relevant activity data (e.g. from surveys, traffic models or statistics) as well as the definition and analysis of **model** scenarios. Scenarios allow a comparative examination of model results corresponding to different versions of data inventories – for example the overall effects of trends in emission generating activities and pollution control measures or the influence of different calculation models on estimated results.

Title	Use in applications	Used components
	Reference Application for the Resource Management Training Support (**Integrated**): BasicIndicators implementation of the Indicators **Building Block** is used in this **demonstrator** to calculate some simple indicators such as the number of patients in different conditions. These results are attached to **World** states, stored in the **World State** repository **BB** as and visualized using the Indicators and Statistics View.	

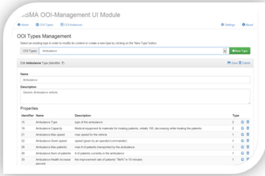
Title	Use in applications	Used components
Integrated Crisis Management Middleware BB The **ICMM** is a central **Building Block** in every **CRISMA Application**. **It** connects **Crisis Management** Simulations with the Analysis and Decision Support functionality of CRISMA by providing a central repository for harmonized **world state** and **indicator** information. The ICMM is fed by simulations providing the basic information to be used for world state analysis and decision support Building Blocks. On an ICT conceptual level the ICMM is a generic distributed **resource**-oriented Control and Communication Information Management **System**. Thereby it is important to note, that the term 'resource-oriented' in the ICMM refers to the **concept** of generic resources as used in the	**Reference Application for Exercise Support** (**Integrated**): The **ICMM** is the central worldstate/**exercise** management component. **Reference Application for the Coastal Submersion Domain** (**Integrated**): The Integrated **Crisis Management** Middelware is mandatory for each **CRISMA Federation** as specified by the **CRISMA Framework** **Architecture**. The **ICMM** collects information regarding to data, **Simulation** Models, Worldstate Transitions, Indicators, etc. of the **Reference Application**. **Reference Application for the Earthquake and Forest Fire Domains** (**Integrated**): This **Building Block** is mandatory for each **CRISMA Federation** as specified by the **CRISMA Framework** **Architecture**. The main challenge is defining an appropriate extension of the Core **CCIM** to be able to describe the domain specific aspects of the **Application**. Furthermore detailed Information on Data Slots, **Simulation** Models and **Transition** Points are required to be able to populate the **ICMM** with the respective control and communication information. This information has been collected in the **Pilot** Architecture of Pilot D (Earthquake and Forest Fire). Integration with the **REST** **API** of the ICMM is straightforward and supported by the on-line documentation available in the **catalogue**. Installation, configuration and hosting of the ICMM is provided by **CIS**. **CRISMA Technology Demonstrator** (**Integrated**): The Java-based implementation of the **ICMM** (cids-server) is integrated into the **demonstrator**. **Reference Application for the Resource Management Training Support** (**Todo for V1**): **ICMM** provides the **CRISMA** applications with information on available (computational) resources, thus simplifying the **task** of designing other parts of the **system**. **Reference Application for the Nordic Winter Storm Domain** (**Todo for V2**): **ICMM** **BB** provides the platform for integration and management of the information from different sources, e.g. simulations, indicators and the **world state** **model**.	**cids System** (**LGPL v3 (permissive OS)**): The **cismet** cids **System** implements some of the features of the **Integrated Crisis Management Middleware Infrastructure** **Building Block**, especially the modular **Control and Communication Information model**. **It** is **open source** and can be extended as required. Additionally, it provides powerful administration tools for Control and Communication Information management and import (Visual Information Modelling and **ETL** Tools).

Title	**Use in applications**	**Used components**
context of the Resource Oriented **Architecture** (ROA).	**Reference Application for Resource Planning** (**Integrated**): Core component of every **Reference Application** **Cross Border Emergency Crisis (Finland)** (**Todo for V2**): **TBD**	

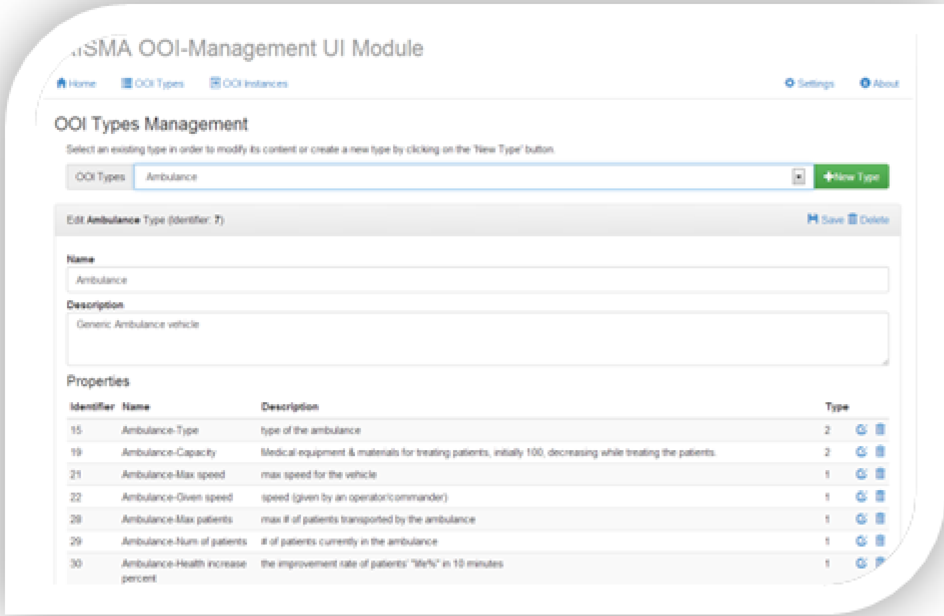
Title	**Use in applications**	**Used components**
Integrated Planning View The Integrated Planning View **BB** is a generic integrated view for the configuration and inspection of arbitrary **crisis management** scenarios in planning situations. The **CRISMA** planning use cases (most use cases of CRISMA) intend to realize functionalities that are related to opening **world** states, changing them and storing them. Further running simulations, inspecting them and comparing results is addressed in all of them and the user work for planning purposes in a calm office environment.	**Reference Application for Exercise Support** (**Integrated**): The "**CRISMA** **PIlot** E Debriefing **Widget**" (Name in Wirecloud) aka. Configuration Component (name in **Catalogue**) realizes the core component of the Integrated planning view and is used in the **Exercise**-Support **Reference Application**. **Reference Application for the Resource Management Training Support** (**Todo for V2**): In the v2, this **reference application** will be extended to incorporate the planing aspect which is required by the **pilot** C. Most likely result of this action is that: (1) the German and Israeli pilots will end up sharing almost all of the code, with the exception of the actual ressoruce models; and (2) we should be able to **test** training and planing functionality on both of these pilot sites.	**Wirecloud OOI GIS Map** (**AGPL v3 (controlled OS)**, **Negociable**): A map **widget** showing **Object Of Interest** (**OOI**) on a background map. **Exercise Worldstate Data Chart Widgets (Wirecloud)** (**To be defined**): Several widgets that display worldstate data from **ICMM** in graphical form (as pie charts and stacked bar charts) and in table form. Developed with AngularJS. **Configuration Component** (**Open source**): The configuration component is the core component of the Integrated Planning View. **It** has been exemplary implemented visualizing the Charts in the Debriefing View of the **Reference Application** for **Exercise**-Support in **Pilot** E. **Wirecloud Application Mashup Platform** (**AGPL v3 (controlled OS)**): Wirecloud is a reference implementation of the **FI-Ware (http://www.fi-ware.eu/)** (Web) **Application** Mashup Generic Enabler. **It** allows users with no programmers

Title	Use in applications	Used components
		experience to rapidly build web applications by "wiring" the available widgets and operators and positioning them on a screen. In **CRISMA**, the Wirecloud is used as a reference implementation of the **UI Mashup Platform** **Building Block** and as a platform for development of the CRISMA web widgets and Views (Mashup applicaitons) based on these widgets. Currently available CRISMA-specific widgets and operators for this platform: • **Charts** • **OOI** Commands • **OOI GIS Map** • **OOI Table** • **Simulation** Picker • **Worldstate Loader** (to be replaced by an interactive **Worldstate Picker** in future iterations) • Worldstate Saver **Wirecloud Charts** (**AGPL v3 (controlled OS)**, **Negociable**): This **widget** displays stacked bar charts for **indicator** values for a given

Title	**Use in applications**	**Used components**
		worldstate.
Multi Criteria Analysis and Decision Support View The Multi **Criteria** Analysis View and Decision Support View is a User Interaction **Building Block** that allows performing a ranking of different **Crisis Management** Scenarios with respect to specific Criteria.	**Reference Application for the Earthquake and Forest Fire Domains** (**Integrated**): The **Reference Application** integrates provides advanced decision support capabilities on the basis of Multi **Criteria** Analysis and Decision Support View. **CRISMA Technology Demonstrator** (**Integrated**): The Java implementation of the **OWA** based MCA and **DSS** is available. **Reference Application for Resource Planning** (**Integrated**): This **Reference Application** integrates **DSS** with the help of this view. Further the **indicator** table from the MC & DSS View is used with in the **simulation results** view. **Reference Application for the Coastal Submersion Domain** (**Integrated**): The Multi **Criteria** Analysis and Decision Support View is used to compare specific Indicators and Criteria for different scenarios of Coastal Submersion. The Indicators that have been defined relate to the number of flooded houses (depending on the water depth) , economic **losses**, flooded area, ...	**Multi Criteria Analysis and Decision Support Widget (Java)** (**LGPL v3 (permissive OS)**): This is the Java implementation of the Multi **Criteria** Analysis and Decision Support View. **Multi Criteria Analysis and Decision Support Widgets (JavaScript)** (**LGPL v3 (permissive OS)**): This is the JavaScript implementation of the Multi **Criteria** Analysis and Decision Support View **Building Block**.

Title	Use in applications	Used components
OOI Management View This **Building Block** enables to view and edit the actual data available for specific **scenario** or **simulation**. **It** enables the **system** administrator to detect the **OOI** data type's properties and based on the OOI Information Models definitions. It also enables to monitor and edit a specific snapshot context (time, user, and workflow).	**Reference Application for the Resource Management Training Support** (**Todo for V1**):  This view is used to define the types of OOIs that are used in the training **scenario**. **It** is de-facto the management **interface** for the **OOI** **World State** Repository **BB** and normally used only by programmers and/or technically savvy administrators as a pre-requisite to set-up the training. In addition, it allows this view can be used to clean up the database from old results which aren't needed any more, as well as to set up the World State details which can not be manipulated using the more user-friendly Scenario Setup View **GUI**. **Reference Application for the Nordic Winter Storm Domain** (**Integrated**): This view is used by administrators to define the types of OOIs that are used within the **Reference Application**. **Reference Application for Resource Planning** (**Todo for V1**): This view is used to define the types of OOIs that are used in the training **scenario**. **It** is de-facto the management **interface** for the **OOI** **World State** Repository **BB** and normally used only by programmers and/or technically savvy administrators as a pre-requisite to set-up the training. In addition, it allows this view can be used to clean up the database from old results which aren't needed any more, as well as to set up the World State details which can not be manipulated using the more user-friendly Scenario Setup View **GUI**.	**OOI Management UI Component** (**LGPL v3 (permissive OS)**): This **Building Block** enables to view and edit the actual data available for specific **scenario** or **simulation**. **It** enables the **system** administrator to detect the **OOI** data type's properties and based on the OOI Information Models definitions. It also enables to monitor and edit a specific snapshot context (time, user, and workflow).

Title	Use in applications	Used components



Title	Use in applications	Used components
	UNDEFINED (**Todo for V1**): **API** documentation: **http**://87.139.99.236/swaggerui/ **OOI**-Management UI : **http://87.139.99.236:3030/app-ooi-management/#/management/types (http://87.139.99.236:3030/app-ooi-management/#/management/types)** TODO: describe how and why this **BB** will be used.	

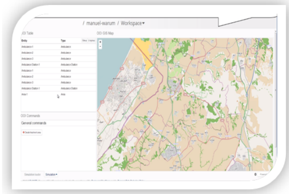
OOI World State Repository BB
OOI-WSR is a **Resource Management** related **Building Block** that enables archiving, querying and manipulation of Objects of Interest (OOI) **world state** data. This **BB** serves as a Repository **service** for OOI data that can be consumed or manipulated by other interaction or functional building blocks and resource management models.
	Reference Application for the Resource Management Training Support (**Todo for V1**):
	Central data repository for all OOIs used in this **application**. Currently: ambulances, citizens/patients and CDM areas. In order to simplify the **API** and decouple the **GUI** from the **model**(s), the **OOI** state is also used to encode the decisions. for instance, the "destination" **parameter** is interpreted by the model depending on the OOI type and context:
	
	• empty ambulances drive to their destination
	• ambulances with patients drive to patients destinations
	• an area destination is assigned to patients after treatment.
	
	Reference Application for the Nordic Winter Storm Domain (**Integrated**):
	OOI **World State** Repository acts as the centralized data repository in this **application**. The following OOI types and their attibutes are used to represent the world in each state:
	
	• Geocell (1 km x 1 km grid):
	    ◦ weather (temperature, wind)
	    ◦ electricity network status
	    ◦ living condition index
	    ◦ building condition index
	    ◦ population
	    ◦ **infrastructure** (residental buildings of different types)
	    ◦ needs (e.g. need for **evacuation**)
	    ◦ **emergency** level
	• Buildings
		OOI World State Repository (**LGPL v3 (permissive OS)**):
		OOI-WSR is a **Resource Management** related module that enables archiving, querying and manipulation of OOI **world state** data. This module serves as a Repository **service** for OOI data that can be consumed or manipulated by other interaction or functional building blocks and resource management models.

Title	Use in applications	Used components
	- location - buiding type - in-house temperature - eligibility status - Resources - location **resource** type (e.g. bus, ambulance, hotel) - capabilities (e.g. people transportation, accomodation) - availability state (e.g. available, incoming to target, outgoing from target) - usage (e.g. amount of kilometers driven in this world state) - travel time to target (in case of tranportation resource) - average speed (in case of transportation resource) - Costs - transportation costs - accomodation costs - damage costs The attributes represent the state of each OOI within the given world state. **Cross Border Emergency Crisis (Finland)** (**Integrated**): **OOI** **World State** repository is used as the centralized storage for the **Object of Interest** in this **pilot** **application**. **It** it also acts as a communication channel between the other building blocks. The OOIs stored to the OOI-WSR are: geocells, buildings, population **Reference Application for Resource Planning** (**Todo for V1**): <p>Central data repository for all OOIs used in this **application**. Currently: ambulances, citizens/patients and CDM areas. In order to simplify the **API** and decouple the **GUI** from the **model**(s), the **OOI** state is also used to encode the decisions. for instance, the "destination" **parameter** is interpreted by the model depending on the OOI type and context: empty ambulances drive to their destination ambulances with patients drive to patients	

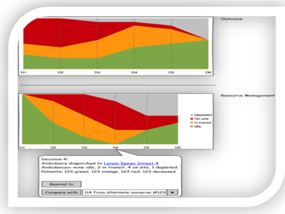
Title	Use in applications	Used components
	destinations an area destination is assigned to patients after treatment.</p>	
Preparedness Plan BB The **Preparedness** Plan **Building Block** is a decision support mechanism, which helps the decision maker to take the needed actions in case of **emergency** according to plans based on analysis of threats, vulnerabilities and possible emergency scenarios.	**Reference Application for the Nordic Winter Storm Domain** (**Integrated**): The **Preparedness** Plan **BB** enables the end-user to create Preparedness Plans based on the analysis of risks, threats and vulnerabilities. Preparedness Plans can be created and maintained continuously as part of the strategic planning phase but also during the operational use. In the **emergency** **situation** the Preparedness Plan BB acts as an decision support tool, proposing actions to be taken to mitigate the situation, as described by the end-user following the pre-defined narrative. Preparedness Plan BB utilizes the **resource** information stored into the **World State** Repository. Preparedness Planning Views and Preparedness Plan Excecution View are the web-based UIs of the Preparedness Plan BB **Cross Border Emergency Crisis (Finland)** (**Integrated**): **Preparedness** Plan **BB** is a generic preparedness planning tool which can be applied to multiple scenarios with user-defined plans. **It** maintains the **resource** information internally, or alternatively it can use resource information from external sources. When the user excecutes the chosen preparedness plan, the tool provides set of resource allocation proposals, based on the rules defined in resource models.	**Insta Response Preparedness Planner** (**Proprietary**): **Insta** **Response** **Preparedness** Planner is a decision support tool, based on existing **situation** assessment functionalities of the Insta Response product family. Preparedness planning is based on creation of a pre-defined narrative (i.e. the Preparedness Plan) as a response to different kinds of **emergency** or otherwise exeptional situations.

Title	Use in applications	Used components
Publish Subscribe Context Broker BB The Publish Subscribe Context Broker **Building Block** is a cross-over between an **event** broker which accepts events and dispatches them to subscribers and an access **service** providing the information on current state of the “world”. Thereby, the Publish Subscribe Context Broker Building Block can be used to realize the event subscription and event delegation functionality of the **ICMM**. Thus, the Building Block is both suitable for dispatching events related to Control and Communication Information (**CCI**) managed by the ICMM and events related to (Worldstate) Data managed by arbitrary data access services, e.g. the **OOI World State** Repository **BB** or implementations of the Data Integration BB	**Reference Application for the Coastal Submersion Domain** (**Todo for V2**): The Publish Subscribe Context Broker is used to inform users when specific events are recorded in other components of the **Reference Application**. Depending on the interest of the users, **it** send messages when, for example, a new Worldstate is available. **Reference Application for the Earthquake and Forest Fire Domains** (**Integrated**): The Pub/Sub Context Broker is used in the communication of the Indicators **Building Block** with the **ICMM**. **Reference Application for the Resource Management Training Support** (**Todo for V1**): The Pub/Sub context broker receives messages (events) from other constituents and dispatches **it** to all constituents which registered their interest. This is used to simplify the coordination of the work between the different BBs, in particular to send out the note that new **World** states are available. This type of events is for instance used by the **Indicator** **BB** to trigger the indicator calculation, and by the **simulation** models platform in order to trigger new world **transition** calculation. Pub/sub events are also used by the **ICMM** to populate its **catalogue** of World States and simulation runs (training sessions). UNDEFINED (**Todo for V2**): Publish-Subscribe Context Broker receives subscription of data stored in to **World State** repository from the data consumers as well as publishes the updates from the data providers to the subscribers. The other BBs are acting in both roles, i.e. the **Indicator** **BB** consumes (subscribes) the data needed in the **KPI** calculations and provides (publishes) the calculated KPI values. **Reference Application for Exercise Support** (**Integrated**): Used to propagate events in order to be able to calculate **indicator**/**criteria** data. **Cross Border Emergency Crisis (Finland)** (**Optional**):	**Orion Context Broker** (**AGPL v3 (controlled OS)**): The Orion Broker is a reference implementation of the FI-Ware Publish/Subscribe Context Broker GE, providing the NGSI9 and NGSI10 interfaces. In **CRISMA**, **it** is used as a reference implementation of the CRISMA Publish Subscribe Context Broker **BB**. The Context Broker is used by other building blocks to exchange events and process updates.

Title	**Use in applications**	**Used components**
(**WMS**, **WFS**, …).	**TBD** **Reference Application for Resource Planning** (**Integrated**): The Pub/Sub context broker is used by the **ICMM** to notify the **Indicator** **BB** that a new worldstate was created and thus triggering the indicator calculation. Furthermore, this **building block** is used by the **simulation** models platform in order to trigger new **world transition** calculation.	
Resource Management Tactical Training BB **Resource Management** Tactical Training **Building Block** (RMTT **BB**) simplifies the **task** of designing the Tactical Training applications for control room operator and on-scene commanders. RMTT enables a Trainee to learn **emergency management** by assigning tasks to various resources and analysing the results in a virtual environment.	**Reference Application for the Resource Management Training Support** (**Todo for V1**): This **BB** represents a desktop **application** that ncorporate the **Resource Management** "views" realized as web mashups for improved user experience. **Simulation results** would be presented on the desktop application **GIS** map in order to support a realistic tactical training session. The following integration tasks are planned: • Develop **HTTP** **REST** **API** that support manipulating OOIs (Situator Assets). • Develop **CRISMA** Situator Gateway that enable Real-Time Sync of **OOI** data with Situator's Assets. • Display Models Results via **WFS**/**GML** using ESRI ArcGIS • Develop CRISMA UI Widgets host to enable using CRISMA web application inside Situator desktop application. • Develop **World State** Geographic viewer using the OOI-WSR WFS API that support display World State as Situator Map Layer and overlay of multiple World States.	**Situator Training System** (**Proprietary**): **Situation** Management is a solution approach that comprises tools and methods for coordinating the interaction between people, technologies, and responses. **NICE** Systems is a major player in the Situation Management market with over 20 years in experience in the industry and expertise in a wide variety of industries such as seaports, airports, railways, banking, government, telecom, utilities, military, manufacturing, etc.

Title	**Use in applications**	**Used components**
Resource Management Training Dispatch and Monitor View The Dispatch and Monitor view provides a high-level overview over the **resource management** **simulation**'s **world state**. Its purpose is to display one world state at a time and allow the user to distribute resources (ambulances, etc.) among different areas where the **crisis** plays out.	**Reference Application for the Resource Management Training Support** (**Todo for V1**):  This is the main view used by the decision maker (trainee) to indicate his or her decisions during the **exercise**.	**Wirecloud WorldState Picker** (**AGPL v3 (controlled OS)**, **Negociable**): This **widget** lets the user pick and load a **world state** from the Objects of Interest World State Repository (**OOI**-WSR). **It** displays all world states and their hierarchy to the **end user**, allowing him/her to select the one that should be continued. **Wirecloud OOI Table** (**AGPL v3 (controlled OS)**, **Negociable**): This **widget** shows all OOIs associated with a worldstate in a tabular form. **It** is possible to filter which elements should be visible (eg. only OOIs of certain types such as Ambulances and Hospitals). In addition, it also allows the user to group any OOIs for convenience; groups are persisted on the client-side using HTML5 storage mechanics and are automatically restored inbetween sessions. **Wirecloud Simulation Picker** (**AGPL v3 (controlled OS)**, **Negociable**): A **widget** showing all simulations available on the **Object of Interest**

Title	Use in applications	Used components
		World State Repository (**OOI**-WSR). **It** allows the user to select one of these simulations and load it, which will initialize other connected widgets and gadgets of the Wirecloud **application** with relevant data. **Wirecloud OOI Commands** (**AGPL v3 (controlled OS)**, **Negociable**): **Widget** that allows to issue commands to one or more OOIs. The commands available depend on the OOIs' types; for instance, Ambulances can treat in a specific area. In addition, there is also a support for "global" commands, ie. commands that do not require an **OOI** (such as "create area"). **Wirecloud OOI GIS Map** (**AGPL v3 (controlled OS)**, **Negociable**): A map **widget** showing **Object Of Interest** (**OOI**) on a background map.

Title	**Use in applications**	**Used components**
		Wirecloud Charts (**AGPL v3 (controlled OS)**, **Negociable**): This **widget** displays stacked bar charts for **indicator** values for a given worldstate.
Resource Management Training Indicators and Statistics View The **Resource Management** Training Indicators and Statistics View **Building Block** is a User Interaction Building Block that will be realised as Mashable Composite UI Module. **It** focuses on the visual presentation of statistics and key indicators of a given Worldstate in order to provide a quick overview of the **situation** and to allow for comparison between any two given Worldstates.	UNDEFINED (**Todo for V1**): This view, which was initially developed for Ressource manageement training **application** appears to be quite similar to the one that's required here. TODO: explain what will need to be changed and such. **Reference Application for the Resource Management Training Support** (**Todo for V1**):  This view is used by the trainee, trainer and/or by the board of examiners to analyse the results of the training. **It** allows the users to visualize various indicators relevant for the **crisis management** (mainly the Key Performance Indicators, KPIs) and compare the results of different decisions.	**Wirecloud WorldState Picker** (**AGPL v3 (controlled OS)**, **Negociable**): This **widget** lets the user pick and load a **world state** from the Objects of Interest World State Repository (**OOI**-WSR). **It** displays all world states and their hierarchy to the **end user**, allowing him/her to select the one that should be continued. **Wirecloud Simulation Picker** (**AGPL v3 (controlled OS)**, **Negociable**): A **widget** showing all simulations available on the **Object of Interest World State** Repository (**OOI**-WSR). **It** allows the user to select one of these simulations and load it, which will initialize other connected widgets and gadgets of the Wirecloud **application** with relevant data.

Title	Use in applications	Used components
Resource Management Training Simulation Scenario Setup View The **Resource Management** Training **Simulation** **Scenario** Setup View is a User Interaction **Building Block** that will be realised as Mashable Composite UI Module. **It** allows the creation of new resource management simulations or modification of existing ones. It allows the user - most likely a trainer - to create incidents and scenes as well as the creation and management of objects of interest (**OOI**) instances as shown in the figure below.	**Reference Application for the Resource Management Training Support** (**Todo for V1**): This View is used by the trainer and/or administrator to set up the training **scenario**, e.g. by choosing the training area indicating the whereabouts of the citizens, availability of the ambulances, state of the roads etc.	**Wirecloud OOI Table** (**AGPL v3 (controlled OS)**, **Negociable**): This **widget** shows all OOIs associated with a worldstate in a tabular form. **It** is possible to filter which elements should be visible (eg. only OOIs of certain types such as Ambulances and Hospitals). In addition, it also allows the user to group any OOIs for convenience; groups are persisted on the client-side using HTML5 storage mechanics and are automatically restored inbetween sessions. **Wirecloud Simulation Picker** (**AGPL v3 (controlled OS)**, **Negociable**): A **widget** showing all simulations available on the **Object of Interest World State** Repository (**OOI**-WSR). **It** allows the user to select one of these simulations and load it, which will initialize other connected widgets and gadgets of the Wirecloud **application** with relevant data.

Title	Use in applications	Used components
Scenario Analysis and Comparison View The **Scenario** Analysis and Comparison View is able to visualise **Indicator** and **Criteria** data in a way that users are able to analyse and compare different Simulated **Crisis Management** Scenarios and ultimately come to a decision which fits the **simulation objective** best for a specific **Simulation Case**.	**Reference Application for Resource Planning** (**Integrated**): This **Reference Application** compares and analyses worldstates with the help of this view. **Reference Application for the Nordic Winter Storm Domain** (**Todo for V2**): The **Scenario** Analyisis and Comparison Support View is used to visualise the **KPI** data to the **end user** in order to support the decisions related to **response** tasks and their priorities. UNDEFINED (**Todo for V2**): For basic comparability between exercises **Reference Application for the Earthquake and Forest Fire Domains** (**Integrated**): This view enables the analysis and comparison of Worldstates on the basis of the Indicators for this **Reference Application**.	**Scenario Analysis and Comparison Widgets (JavaScript)** (**LGPL v3 (permissive OS)**): This is the JavaScript implementation of the **Scenario** Analysis and Comparison View **Building Block**. **Scenario Analysis and Comparison Widgets (Java)** (**LGPL v3 (permissive OS)**): This is the Java implementation of the **Scenario** Analysis and Comparison View. **It** provides a Table visualisation for both the indicators and the **criteria**. Moreover the **indicator** values are visualised via bar charts and the single indicators can be related to each other so that a graph visualises the correlation between the different criteria.

Title	Use in applications	Used components
Simulation Model Integration BB The **Simulation Model** Integration **Building Block** provides components that can be used to easily enable simulation models to participate in a **CRISMA Application**. Due to the heterogeneity of existing simulation models CRISMA has to ensure that they can be integrated in a standardized way and ultimately become a **CRISMA Federate**. That way they are made CRISMA-aware. The envisioned technique to be used is that of so called wrapping so that this Building Block provides a piece of software that can be used to make (existing) simulation models CRISMA-aware with as little effort as possible with respect to the individualities of the different simulation models.	**Reference Application for the Coastal Submersion Domain** (**Integrated**): The **Simulation Model** Integration **Building Block** is used to wrap the different models used in the **Reference Application**. **Reference Application for the Earthquake and Forest Fire Domains** (**Integrated**): The **Simulation Model** Integration **Building Block** is used to integrate various domain-specific Simulation Models of this **Reference Application**, like the Building **Impact Model**. UNDEFINED (**Optional**): **Simulation Model** Integration **BB** may be utilized if other models are seen feasible to be integrated in to the **Application**	**PyWPS** (**GPL v2 (controlled OS)**): **PyWPS** (Python **Web Processing Service**) is an implementation of the **Web processing Service (http://www.opengeospatial.org/standards/wps) standard** from **Open Geospatial Consortium (http://opengeospatial.org)** . Within **CRISMA**, the PyWPS is (can be) used to make legacy models and other software available within the **CRISMA Framework**. In order to do so, one has to extend the basic PyWPS classes and introduce invocations of the own software. By nature of GPL, this extension is automatically licensed under GPL as well, but the invoked software (e.g. models) is not linked to PyWPS and can be published under a different license.

Component to be defined

Authors: **Denis Havlik**

This is a placeholder which allows us to define building blocks and **CRISMA** models for which no components have been defined yet.

Documentation:

User Guide

The "Site news" doubles as a kind of FAQ for all questions relaetd to this site.

- Catalogue "site news"
  https://crisma-cat.ait.ac.at/site-news

Downloads:

Team:

Sascha Schlobinski

Development roles: coordinaton, mediation

Leader of the SP3 (**CRISMA framework**). Can help in the case mismatches between BBs and components are discovered.

Denis Havlik

Development roles: coordinaton, mediation

CRISMA Technical manager. Can help in case of misunderstandings and deadlocks caused by newly discovered needs which no-one feels responsible for.

Klaus Steinnocher

Development roles: coordinaton, mediation

Main contact for SP4 related questions (component/**model** relations)

How to edit this content?

Here is how to edit various aspects of the book:

1. Editable view of the book is **available here** (other views exist which will not allow editing). This view allows you to add "children pages" to any of the book pages, as well as to edit the existing pages.
2. In order to add the existign content to a book, invoke the edit mode for that conent and set the "Book" and "Parent item" properties in the "Book outline" tab

on the bottom of the edit form.
3. Re-aranging the content within a book is best done at **book administration page**.

It's likely that some, of these tasks will be restricted to team members with higher privileges such as Editors. If you need to perform a **task** and can't, please contact one of the site administrators. Curently these are: **Denis**, **Peter**, and **Pascal**

Licensing overview

This page shows the overview of the licenses which are used in CRISMA. It provides a convenient way to find out which of the CRISMA components are licensed under a particular license.

Terms listed under "To be defined" (if any) are a bad news. They either mean that the component owner has not made up it's mind or that our documentation is outdated. Either way, this is not good for CRISMA exploitation.
Terms with dual licenses are usually a good news from the point of view of exploitation, especially the ones with "License 1, Other". They indicate that the owner is ready (and able) to negotiate alternative licensing terms with partners whose business model would suffer from the use of the primary license.

Licenses	Components
GPL v2 (controlled OS)	**BasicIndicators** (**AIT**) **cids Navigator** (**cismet**) **CRISMA Catalogue** (**Drupal**) **GeoServer** (**community-driven project**) **PyWPS** (**Intevation**)

Licenses	Components
LGPL v3 (permissive OS)	**Cascade Events Configuration Widget** (**cismet**) **cids System** (**cismet**) **cismap** (**cismet**) **Multi Criteria Analysis and Decision Support Widget (Java)** (**cismet**) **Multi Criteria Analysis and Decision Support Widgets (JavaScript)** (**cismet**) **OOI Management UI Component** (**NICE**) **OOI World State Repository** (**NICE**) **Scenario Analysis and Comparison Widgets (Java)** (**cismet**) **Scenario Analysis and Comparison Widgets (JavaScript)** (**cismet**) **Scenario Evolution Widget (Java)** (**cismet**) **Scenario Evolution Widget (JavaScript)** (**cismet**) **Scenario List Widget (Java)** (**cismet**) **Scenario List Widget (JavaScript)** (**cismet**) **Worldstate Tree Widget (Java)** (**cismet**) **Worldstate Tree Widget (JavaScript)** (**cismet**) **Worldstate Widget (Java)** (**cismet**)

Licenses	Components
	Worldstate Widget (JavaScript) (cismet)
To be defined	Component to be defined (AIT) Exercise Worldstate Data Chart Widgets (Wirecloud) (Cassidian) TDV Python package (AMRA)
Open source	Configuration Component (Fraunhofer IAO)
Proprietary	CRISECON GUI (VTT) CRISECON Service (VTT) Emikat (AIT) Insta EvacSim (Insta) Insta Response Preparedness Planner (Insta) Life Safety Model 2D (HR Wallingford and BC Hydro) PostgreSQL stored procedure (PLINIVS) Situator Training System (NICE) VTT House service (VTT)
Apache v2.0	Docker (Docker)
Proprietary, using Google Maps API for non-commercial purposes	Dynamic Map Agents (TTU)
GNU	Evacuation model (NetLogo) (TTU)

Licenses	Components
BSD (permissive OS)	**Leaflet.js** (**Vladimir Agafonkin**) **OpenLayers** (**OpenLayers Community**)
MIT (permissive OS)	**MapServer** (**Open Source Geospatial Foundation**)
AGPL v3 (controlled OS)	**Orion Context Broker** (**Telefonica**) **Wirecloud Application Mashup Platform** (**UPM**)
MIT License	**PyShp** (**Geospatialpython**)
LGPL V3	**Simulation Model Interaction Widget** (**Spacebel**)
GPL (controlled OS), **LGPL (permissive OS)**	**TELEMAC MASCARET System** (**Eléctricité De France**)
AGPL v3 (controlled OS), **Negociable**	**Wirecloud Charts** (**AIT**) **Wirecloud OOI Commands** (**AIT**) **Wirecloud OOI GIS Map** (**AIT**) **Wirecloud OOI Table** (**AIT**) **Wirecloud Simulation Picker** (**AIT**) **Wirecloud WorldState Picker** (**AIT**)

Major issues

Articles in this section have multiple unresolved issues.

PostgreSQL stored procedure

PostgreSQL stored procedures allows users to extend a database with user-defined functions by using various procedural languages.

Documentation:

Presentation

PostgreSQL stored procedures allows users to extend a database with user-defined functions by using various procedural languages.

The stored procedures define functions for creating custom reusable **application** procedures, for data processing internally stored in the database elaboration engine. The language used is PL/pgSQL.

PostgreSQL, often simply "Postgres", is an **object**-relational database management **system** (ORDBMS), **it** is free and **open source** software, released under the terms of the PostgreSQL License, a liberal Open Source license, similar to the BSD or MIT licenses.

Downloads:

Team:

Stefano Nardone

Development roles: development, integration, support

Login

Source URL (retrieved on *03/16/2015 - 01:17*): https://crisma-cat.ait.ac.at/node/109