Situation Awareness Unified Process

Vikram Sorathia  
Dhirubhai Ambani Institute of  
Information and Communication technology  
Gandhinagar, Gujarat, India  
vikram_sorathia@daiict.ac.in

Anutosh Maitra  
Dhirubhai Ambani Institute of  
Information and Communication technology  
Gandhinagar, Gujarat, India  
anutosh_maitra@daiict.ac.in

Abstract

This paper identifies the process for developing information system for Situation Awareness application domain and identifies novel process artifacts that need to be introduced along with existing approaches. Appropriately engineered method is an important requirement for successful implementation of any software system targeted at situation awareness. When it comes to Information System Development (ISD) for dynamic organizations, the method engineering plays even more critical role. The existing approaches for architectural description, method composition and process guidance are derived based on experiences from successful past implementations. Yet, how existing team utilizes the experience is the determining factor for its efficient use. Today’s organization, where one information system is the result of continuous efforts of multiple teams forming a virtual organization, calls for extending the approaches to get proper benefit from Method Engineering.

1 Introduction

With increased amount of understanding in engineering software systems, the domain is now equipped with many standards, best practices, conventions and procedures. The standards like ISO Reference Model-Open Distributed Processing (RM-ODP) defines a view-point based approach. Regarding enterprise architecture, the Open Group has proposed[1] TOGAF - an architecture framework with important concepts like gap analysis for developing robust enterprise systems. Department of Defense Architectural Framework (DoDAF) specifies creation of a comprehensive IS blue-print with 28 well defined architectural products. Among other important approaches Zachman framework, Federal Enterprise Architecture Framework and many others are well evaluated[2] for their suitability to address the enterprise modeling issues. The method engineering domain[3] provides host of unique method proposals towards building Information Systems. Till date, Rational Unified Process[4] and its extensions are widely accepted for its rich source of guidance in form of concepts, guidelines, templates, tool mentors, checklists, white papers and supporting materials to address the software development related issues.

With ever expanding business enterprises, consortiums, trade groups, governmental alliances, and international organizations, building of information system has become quite challenging even with the modeling techniques mentioned above. For example global policies for reducing disaster risks require information collection, processing, sharing and dissemination at various geographical scales[5]. Wide geographical coverage, multiple disciplines and dynamic scenario make it difficult to capture all the aspects of the Universe of Discourse (UoD) a priori.

Considering a kind of a scenario given in Figure 1 for an information system, the universe of discourse can be represented from multiple domain point-of-views for the given organization. The interpretation of situation is therefore based on rules defined in respective domains[6]. To realize the information flow as indicated, various specialized tasks are to be carried out to obtain appropriate domain representations. Once representation is available, the information processing and determination of required set of actions can be done, result of this activity should be sent to real-world actors who can alter or maintain the situation in their desired status. Hence actors in real-world, as well as the actors playing different roles in creating representations, domain world view, inferring required actions, and communicating to others in the given UoD- should be equipped with appropriate Situation Awareness.

1.1 Situation Awareness

The concept of Situation Awareness was initially introduced[7] in reference to flight automation domain. With required modifications, the same concepts can be adopted in building reactive systems for any domain in
Information Flow in Universe of Discourse

Figure 1. Information Flow in Universe of Discourse

Information systems in dynamic enterprises provide a challenging case for designing a reactive system. All team members responsible for information system must be able to react to the changing needs of the organization. The following definition of situation awareness for information systems domain can be offered.

Situation Awareness is a state achieved in which a role is provided with information at specific space, time and conceptual granularity; determined in the prevailing context and the underlying information communication configuration. It not only provides world-view of domains relevant to the role, but also provides actions required.

Information systems with situation awareness needs domain knowledge, knowledge about the users and their association and must be able to handle information sources beyond sensors with proper spatial, temporal and conceptual granularity.

1.2 SME for SA Application

Situational Method Engineering (SME) is proposed as a solution to the issues of dynamically changing environments. SME is naturally suitable for Situation Aware application as such applications are based on the concept of determining instantaneous needs. The information processing is done as a response to the changing situation. In dynamic environments where Situation Awareness applications are employed, tasks of the instantaneous set of team members are identified allocated, and monitored using SME methodology. Information System Development (ISD) for such SA applications requires proper adaptation in method engineering approach.

2 Orthogonal Concerns

The following representation depicts how a specific instance in enterprise information system can be traced back to real world processes.

This mapping reveals that at each of the defined stage, various roles carry out certain activity to produce specific artifacts. An important observation is that though each of these activities are having specific sequence and input output dependency among them, the actors performing the activity may have completely orthogonal interests. For example service developers, service providers, data providers, configurators and users: all are associated to a specific service, but a service developer may not show any concern about how data will be provided, or how service will be deployed. In service oriented paradigm every role may not know all other roles active in the system, but when the need be, they must be able to use the system to do exact kind of work that they are supposed to do.

Software Process Engineering discipline provides solution of this problem by providing rich source of guidance for carrying out large scale projects in diversified team environments. Rational Unified Process (RUP) is one such widely accepted source. Conventionally, RUP content is referred explicitly by the team members by setting process guidance preferences appropriate for their assumed roles in their Integrated Development Environment (IDE) workspace.

For SA systems, the team configuration is not fixed, hence the task assignment itself is not easy for the software project manager. As a potential solution, available members must be able to provide information of their skill-set for which specific tasks can be assigned along with the required process guidance for the same. Hence, the task assignment and the access to process guidance should be carried out in an event-driven manner. This can be achieved without much alterations in existing practice as RUP content already have basic information required for this approach. Apart from rich process guidance, RUP allows representation of information like required skill set, task assignment approach, if an activity is event triggered and other details that can be
utilized for the required event-driven approach. The resulting approach that extends the capabilities of existing practice in architectural framework and process engineering domain to suit the needs of SA Systems is discussed below as SA Unified Process.

3 SA Unified Process

During the requirement gathering stage of an enterprise information system many non-functional requirements that are essential for collaboration in dynamic environment can be missed. Efforts toward team integration later becomes difficult task. The basic principle of the unified process is that, organizations will define their commitment toward methods instead, and then the developers will develop and test systems according to the given reference model. Thus rapid adoption of missing components in the system could be made possible.

3.1 Life cycle

As described in representation depicting the traceability from Real World Processes to Instances in Section 2, the information system requirement is continuously evolved for a given organization. SA Process that allows creation, monitoring and upgradation of such information system becomes important. As the process itself must incorporate changing needs of organization, it must support transition from present configuration to new configuration and this cycle continues to exist. Different phases in a life cycle of a SA Process are depicted in Figure 2. The phases are defined as

- **Policy:** The life-cycle starts with the policy of the organization to collaborate for situational awareness needs. For example, any governmental organization planning to confirm to various actions priorities identified in [5], must create and demonstrate provisions for sustaining information flow among local, national and international organizations and the target community at large.

- **Analysis:** This phase identifies how situational needs can be fulfilled with the available infrastructure. By committing to comply with such demanding information sharing policy, organization must carry out analysis to realize the newly identified needs.

- **Design:** Designing allows the development of required services and components to suite the identified needs. This phase includes development of architectural products that can be used by the software development teams in constructing the designed system. Comprehensive list of architectural products that reveal the architecture of system are created as prescribed in DoDAF.

- **Mapping:** Mapping is done at the semantics level. Once the application is defined, the organizational rules and other Ontologies can be mapped appropriately.

- **Configuration:** Configuration is the process in which actual instances are configured.

- **Management:** Once the configuration is up, the intermediate tasks for data management, fine tuning and resource management for load balancing are required.

- **Review:** The behavior of the configuration is evaluated according to the needs. A number of traceability matrices are studied to determine the capability of the system. Coverage analysis from Traceability matrix provides basic information required for comprehensive Gap Analysis.

- **Archive:** The archival phase purges existing instances, stores the traces, and other logs for future references.

3.2 Available Architectural Products

The proposed SA unified process is founded using DoDAF, TOGAF, RUP and some other existing proposals and then appending novel artifacts in it. Many of the architectural products in the SA Unified process are same as identified by the source products. For example, SA Unified Process imbibes all the architectural products of DoDAF[11]. SA Process is planned to be service centric and service related products are same as defined in SOMA[12].
3.3 Novel Architectural Products

Architectural Products mentioned in the previous section are based on the conventional approach and are not sufficiently equipped to support situational method engineering. For the task of building situation aware process, a few novel products are added. These products are defined to provide appropriate situation awareness (SA) to actors playing specific role(s), hence the products are defined under category of Situation Awareness Views (SAV). A brief description of each is provided below:

3.3.1 Organizational Knowledge base URL View (SAV-1)
The situation awareness is achieved with proper representation of knowledge base regarding organizations, their goals, and domain knowledge from relevant domain, technology, standards and heuristics that affect the decisions during the organizational process. Once the mapping phase is over, the organization can publish a knowledge URL where all concepts and rules are made available consistently to the users.

3.3.2 SA Role Product Matrix (SAV-2)
This is an important matrix that determines various SA Roles that are being introduced in succeeding sections. The details regarding the role can be retrieved from the knowledge base URL product. The prime purpose of this matrix is to define association of every SA architecture product with one or more users defined in the SA namespace.

3.3.3 SA Information Need-Component Matrix (SAV-3)
The proposed unified process employs the method of determining the instantaneous information need based on the mapping. Thus for each inferred information need, a component must be identified that can achieve the required task.

3.3.4 SA Information Need-Service Matrix (SAV-4)
The information need can be fulfilled with each functionality exposed as a service. This Matrix reveals how various services are traced to identified information need. This architectural product is also helpful in determining the coverage of the services. It may reveal the information need for which services are not defined. This product is mainly useful for SA Service Provider Role.

3.3.5 SA Information Need-Data Matrix (SAV-5)
The service provides mechanism for accessing, processing, handling and displaying the data, but availability of data is an important matter. This architectural product allows identification of need vs. availability of sources that provide data at given spatial and temporal granularity.

3.3.6 SA Information Need-ETL Matrix (SAV-6)
The raw data at the most granular level is collected on field. But users at different level of an execution center requires different level of granularity in data. The aggregation of miniature data can be achieved with specific function. For example some values can be directly added for achieving aggregates whereas to aggregate properties like climatic conditions; a kind of average can be used. Thus depending upon the property being observed, the rule will define specification for aggregation at next level. This architectural product defines information need for a given role at specific granularity to an ETL specification.

3.3.7 SA Information Need-MoM Pattern Matrix (SAV-7)
The data is collected in the forms of attributes of an entity. The Message Oriented Middleware (MoM) used for collaborative environment is capable of defining data type channel according to specific type of attributes inferred in information need document. Creation and destruction of messaging endpoints will be based on inferred time period. Hence, this architectural product defines mapping among information need and the instances of relevant MoM Patterns.

3.3.8 SA Information Need-VO Matrix (SAV-8)
The messaging related patterns considered are not limited only to MoM, but they must also be mapped to actual nodes where a particular type of user will produce and consume the message. These nodes are in effect members of the VO. They are constantly polled for availability. They are provided appropriate security certificates and they all use required services of the grid. This matrix maps the nodes onto the type of data they consume or generate.

3.3.9 SA Information Need-Coverage Matrix (SAV-9)
This architectural product provides coverage analysis to the SA Configurator Role, stating the traceability among Information Need vs. Service and Component instances.

3.3.10 SA Need-Response Matrix (SAV-10)
The need is determined by the knowledge base in the form of a set of actions, resources, environment that should be carried out or facilitates in response to the current situation. This architectural product provide traceability amongst a need to the actual response in answer to the identified need.
This product is important in avoiding duplication of effort, idling of resources and other allocation related issues which are generally faced in dynamic environments.

3.3.11 Artifact-Standard Matrix (SAV-11)

The Technical Standards View Products of DoDAF provides snapshot of standards adhered to while developing a system architecture. Yet this is not made available in real time to the standardization related stakeholders. Artifact to standard metric reveals the specific purpose for which the standard is utilized. The standardization authorities can consistently update and review the standards in the knowledge base and monitor the performance of the standards. This architectural product is useful in a scenario if during a configuration phase; use of certain standards create additional issues that needs to be communicated to the standardization authorities. Since the participation in standardization process is voluntary, the organizations identifying issues in this product will be encouraged to participate in standardization process.

3.3.12 Information Need-Research Matrix (SAV-12)

The recent incidents like avian flu, bio-terrorism and other such uncommon situations reveal that organization as well as research community may have very little understanding or formal knowledge about handling such events. This architectural product defines the gap in scientific understanding. And as soon as the gap is filled; it can be instantaneously reflected in the knowledge base.

3.3.13 Policy/Resource-Action/Utilization Matrix (SAV-13)

Each organization has some set policies for allocation and utilization or resources. This matrix reveals if for a given policy, the organization has enough resources available. Thus the resources gaps, or updates required in the policy itself can be determined.

3.3.14 Needline-System Matrix (SAV-14)

The concept of a needline as defined in DoDAF as a static specification of information exchange among entities participating in information system. For SA systems, where information need dynamically changes among the members of the VO, static representation is not sufficient. Determination of needlines on a given instance in dynamic environment can be made possible with inference in knowledge base. The matrix provides traceability and visualization of available Grid Nodes, role being played by them and corresponding the needline supported by them.

3.3.15 Organization Goals-SA Artifact Matrix (SAV-15)

The SA Manager is the interface between organization and the SA Process engineer. The dynamically changing organization structures, policies, and decisions for response are communicated to SA Process engineer through the SA Manager. Hence, this architectural product is used for the manager to track how the changes are reflected in the present configuration.

3.3.16 Organization Need-SA Artifact (SAV-16)

This architectural product is for a reviewer who can do coverage analysis regarding the implied organization needs and how that is matched by the SA Artifacts.

3.4 Process Plug-in

Process engineering is currently supported with method composition tools both in open-source and commercial domain. The Eclipse Process framework (EPF) have provided tooling with open method library like OpenUP. A well established commercial counterpart is Rational Method Composer (RMC). RMC is packaged with Rational Unified Process plug-in with various extensions for business and enterprise development with a number of software engineering paradigms. RMC also provides additional method plug-ins like SOMA, DoDAF. As of now the first version of SA Unified Process is authored as RMC Plug-in extending these available required method parts. Figure 3 and Figure 4 depicts the snapshots of the developed Method Plug-in.

4 Future Work

The proposed Situation Awareness Unified Process introduces novel artifacts on top of existing process framework. These artifacts are generated and updated automatically once the configuration is defined. Hence, the Process
should be coupled with the tool that supports generation of these artifacts. The tool that support the required execution environment is currently under development process. Future work therefore is targeted towards development and testing of a meta CASE tool that support creation, management and sharing of architectural products developed under the proposed unified process. The eclipse community have recently released various technology projects to support team communication, collaboration, task management and other important SE aspects. The future work also include incorporation of such contributions to enhance the Situation Awareness capability of involved team members.

5 Conclusion

This paper presented some process engineering ideas for building information systems that are targeted at Situation Awareness Systems in dynamically changing environments. With brief overview of the present status in Unified Processes, Enterprise Architecture Frameworks and other aspects in building information systems, a Unified SA process is proposed that extends the current capability of available approaches. Many novel architectural products that are required for the purpose have been presented. By means of traceability artifacts, the approach also demonstrated separation of concern amongst various types of users in collaborative virtual environments.

References


