Systematic Support for Measurement and Evaluation in Software Projects

Hernán Molina and Luis Olsina

GIDIS Web, Engineering School, National University of La Pampa,
General Pico, La Pampa, Argentina
{hmolina,olsinal}@ing.unlpam.edu.ar

Abstract. It is widely agreed that successful Measurement and Evaluation (M&E) programs and projects require consistent specifications of not only data, but also metadata associated to M&E design and implementation. Metadata and data should include –in addition to non-functional requirements, methods, scales, etc.- context descriptions in which projects are performed. It is also recognized that the incorporation of M&E capabilities requires considerable effort and cost to an organization; however, current proposals lack sound specifications of M&E metadata. In this article we present a solution which includes an M&E information model (IM) and a software architecture to support the effective incorporation of an M&E capability to software projects. To illustrate the proposed solution a proof of concept is used.

1 Introduction

Nowadays, many organizations develop and/or maintain software systems, in many cases as a critical asset to their operation. For such reason software projects must be efficiently managed following a set of processes, making use of a number of limited resources, under particular constraints. In this scenario, the success of organizations depends largely on the achieved quality (of their processes, products, resources, etc.) while keeping balance with performance and cost.

Quality is influenced by the complexity of the product requirements, the characteristics of methods and tools used to develop them, the stakeholders involved in the process and a set of environment factors that relate and affect all these elements. An effective and accurate control over these factors requires a significant effort and should provide quantitative information of products, resources, projects, processes, etc., thoroughly based on relevant facts. In this way, the estimation of project outcomes could be improved, the fulfilment of goals could be managed and verified objectively and the necessary corrective actions could be planned in a more accurate way. Decisions taken in the absence of these conditions may increase the chances of unexpected results and lead the organization away from its settled goal.

To this aim, an M&E capability serves as a key organizational asset to provide quantitative insight on the key attributes or properties of these entities, as well as to keep track on the properties characterizing the set of factors of the environment that affect the overall quality. However, in order to this M&E capability delivers consistent and coherent results as input for decision making, it should be built on clearly specified and structured IMs providing a way to explicitly represent meta-data and data regarding: (a) information needs and goals specifications; (b) entities being measured and evaluated; (c) specifications of entities' attributes; (d) descriptions of the relevant contexts in which M&E are performed; (e) quality models representing the non-functional requirements established by the information need; (f) specifications of how to obtain attribute values (measurement methods) and how to represent them (scales and units); (g) the measured values and related descriptions; (h) specifications of how to interpret characteristics and attributes values to satisfy the stated information need; and (i) resulting values corresponding to non-functional requirements' interpretations either in a quantitative or categorical way. The IM should also allow these meta-data and data be recorded, accessed and used in decision making to enable the organization to effectively achieve its goals.

In this article we emphasize on some of the M&E elements of a conceptual framework, called C-INCAMI [1, 2], that addresses the above concerns. C-INCAMI defines in a clearly and structured form the concepts, properties and relations describing the meta-data and data necessary to specify M&E activities. All its elements are semantically defined in an ontology and supported by a process definition and methodology [3, 4].

Particularly, we highlight the following issues of C-INCAMI with regard to the previous INCAMI version [4, 5]:

1. the incorporation of the context-aware approach to allow the structured specification of the relevant context in which M&E is performed –as recommended in [6, 7]; and
2. the incorporation of the meta-data necessary to integrate the information (concepts and instances) from the application domain of the organization to the specifications made with C-INCAMI. We argue this integration mechanism eases the inclusion of an M&E capability to the organization's projects.

The rest of this article is outlined as follows: In Section 2 related works are reviewed emphasizing the problems found. Then, in Section 3 the C-INCAMI IM is described, providing also a proof of concept. Additionally the mechanism designed to integrate information from the organization's domain space to M&E meta-data and data is presented. Then in Section 4 architectural issues to support C-INCAMI are addressed. Finally, in Section 5 conclusions are drawn and future work is outlined.

2 Related Work

Regarding M&E models, most of the authors agree that M&E meta-data help to ensure trustworthiness, repeatability and comparability of the corresponding information allowing to validate gathered data, to secure the consistency of conclusions during data analysis, and to properly guide updates to data repositories from meta-data changes, among other benefits. This statement is true if M&E meta-data and data includes also the information of the relevant context affecting how these activities are designed and performed, and results are lastly interpreted for decision making, as asserted in [6–11]. Also, to increase the benefits from using an IM to represent M&E activities, the model should support a mechanism to integrate and make use of the organization information space, thus taking advantage of its existing information assets.

Card presents in [12] an IM based on the general structure of the GQM approach [6] and the standard ISO/IEC 99:93 [13]. In this model, attributes of software entities are related to information needs by three levels of measures –base measures, derived measures and indicators. This model was adapted and included in the standard ISO/IEC 15939 [8]. However, all these works lack detailed specifications of how each concept is structured, i.e. which properties define them. In [11] Kitchenham *et al.* present an IM as an *Entity-Relationship* diagram to support databases of measurement data and its meta-data. This model is organized in three domains: an independent *generic domain* including attributes, units and scales; a *development model domain* translating attributes and units to measures linked to software entity types; and a *project domain* including actual entities linked to measured values obtained from the defined measures. As before, this model lacks a detailed specification of the included concepts. In [7] Briand *et al.* present an IM to support their proposed process, called GQM/MEDEA. This model includes also terms and relations related to measurement programs and their resources. However it provides neither detailed specifications of proposed concepts nor distinction between measurement procedures and corresponding measured values.

On the other hand, in [14] García *et al.* present a measurement meta-model based on a software measurement ontology. This meta-model allows to represent meta-data of measurement results and it can be used by organizations as a reference in the definition of its own measurement models and repositories, allowing all of their data be represented in an homogeneous way. As in cited works, the proposal does not provide a complete and structured specification for concepts and properties of the M&E meta-model. Olsina *et al.* present a conceptual framework for M&E, called INCAMI [4, 5], defining in a structured form the concepts, properties and relations allowing to represent the necessary meta-data and data. This model is built around the *Information Need*, *Concept model*, *Attribute*, *Metric*, *Indicator* concepts. INCAMI allows for measured values and corresponding interpretations to be repeatable and comparable among different projects. It was used to develop an M&E repository [15] and also in different WebApps quality evaluation cases [16, 17]. The underlying ontology is based on concepts taken from the ISO standards [8, 18, 19] and other authors [11, 7]. Lastly, Barcellos *et al.* [20] propose a *Software Measurement Ontology* as a reference ontology to clearly and precisely describe the entities used in software measurement. Authors define a meta-model as a foundational ontology arguing the existing proposals in the field rather rely on low expressivity models. This ontology is structured in five sub-ontologies which, in turn, are associated with domain ontologies for software process and the organization. Nevertheless, the proposed model does not include the properties to characterize its concepts.

So far, regarding the context information issue, only some of the cited works include the term *context* or *environment* in their M&E models, although none of them provide a structured and complete specification. Regarding the domain integration issue, some of these works provide hints to that matter: Lawler & Kitchenham [21] argue their model can be integrated to support tools of the organization but without explaining the mechanisms to do so. On the other side, by using ontologies, the works in [4, 14, 20] offer the possibility of domain integration provided by the linking mechanisms of semantic web technologies.

3 M&E Data and Metadata

C-INCAMI (*Contextual INCAMI*) defines in a clearly and structured form the concepts, properties and relations describing the meta-data and data –listed in Section 1- necessary to specify the design and implementation of M&E [2, 4]. C-INCAMI extends its predecessor [5] with *(i)* concepts, properties and relations to represent context descriptions associated to particular context-sensitive M&E elements, and corresponding mechanisms to use context descriptions in the design and interpretation of M&E activities, and *(ii)* a set of properties supporting a mechanism to integrate information from the organization's domain space to M&E specifications, and thus to ease the incorporation of C-INCAMI to the organization's conceptual and technological infrastructure.

To represent M&E information, C-INCAMI follows two modelling approaches: *(i)* an *object oriented approach* to describe the conceptual design of the terms and relations –particularly, the UML standard was used; and *(ii)* an *ontology approach* was used to represent and integrate definitions of concepts and relations taken from specific application domains, from which the entities and properties subject to M&E activities come from. The ontology approach was supported by *RDF Schema* due to its standardisation level and the availability of great deal of support tools for storage, management and validation of RDF data.

C-INCAMI is composed of four interrelated modules namely: *requirements*, *context*, *measurement* and *evaluation*, plus the main module *c-incami* containing a minimal set of concepts allowing to specify basic information of M&E projects, both as a way to organize the information related to each activity and also as a starting point for project management. We describe these modules below, illustrating its concepts and relations by means of a proof of concepts based on an actual case study performed with INCAMI [16].

3.1 Non-Functional Requirements

The *requirements* module (`c-incami::requirements` package in Fig. 1) includes key concepts allowing to specify non-functional requirements for concrete entities, e.g. products, resources, processes, etc. First, an *information need* is specified stating the *purpose*, *point of view*, *focus* (a *Calculable Concept*), actual relevant *context* (see relation `characterizedBy`) and the *category of the entities* to be assessed. Actual entities, belonging to these categories can also be specified. Note that a number of related entity categories, and concrete entities, can be specified as well. Non-functional requirements to represent the stated focus concept are designed in *concept models* by selecting and combining sub-concepts and, lastly, *attributes* associated with entities.

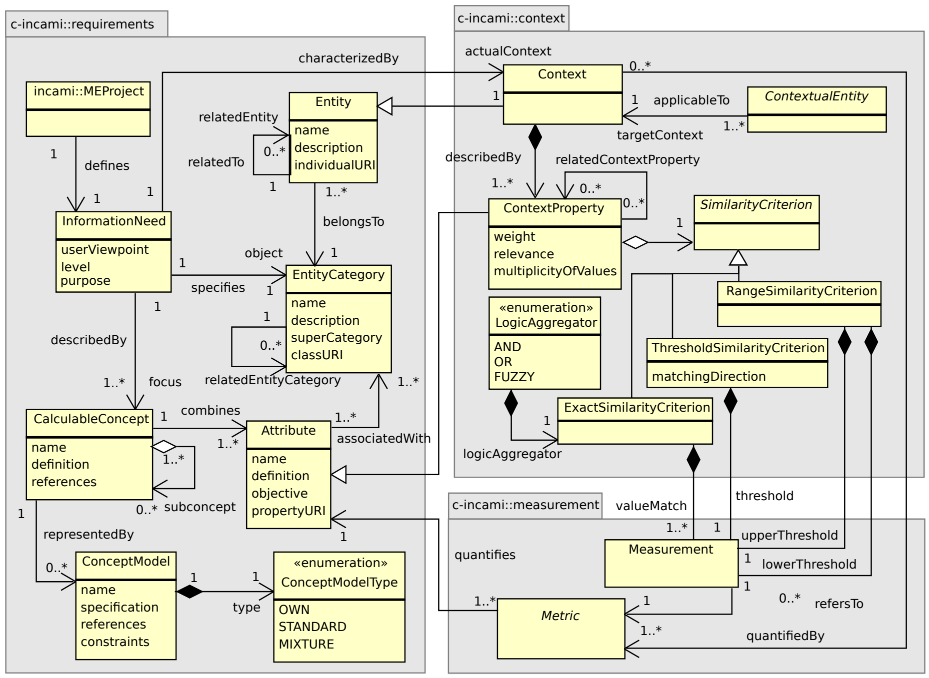


Fig. 1. UML Class Diagram for the *requirements* and *context* modules of C-INCAMI.

For example, the *information need* to evaluate *External Quality* (EQ) for the *Cuspide.com* WebApp can be specified as in Table 1. Then, for the *EQ* focus a *Concept Model* is selected from the M&E catalogue (Fig. 2(a)) and then populated with attributes also retrieved from the catalogue (Fig. 2(b)).

Table 1. *Information Need* for the EQ evaluation of *Cuspide.com*

Purpose	***Focus***	***User Viewpoint***	***Entity Category***	***Entity***
Improve	External Quality	Customer user	Web Application	Cuspide.com

3.2 Context Description

The context module (`c-incami::context` package in Fig. 1) includes concepts and relations to specify context descriptions. *Context* is defined as *a special entity*

External Quality
1. Usability
   1.1. Understandability
   1.2. Learnability
   1.3. Operability
   1.4. Accesibility
2. Functional suitability
   2.1. Functional completeness
   2.2. Functional correctness
3. Information Quality
   3.1. Suitability
      3.1.1. ...
4. ...

(a) Selected Concept Model

External Quality
1. Usability
   1.1. Understandability
      1.1.1. Shopping cart control ease to be recognized
      1.1.2. Add-to-cart control ease to be recognized
   1.2. Learnability
      1.2.1. How-to-buy help availability
      1.2.2. Readability of text
   1.3. Operability
      1.3.1. Shopping cart control permanence
      1.3.2. Add-to-cart control permanence
      1.3.3. ...
2. ...

(b) Extract of the edited Concept Model

Fig. 2. Concept Model used to represent EQ of *Cuspide.com*

(from `requirements`)*, described by a set of context properties which characterize the relevant state of the situation of an entity of interest. The situation is determined by the task and purpose towards that entity and its relations with other entities for that task and purpose.* A *Context Property* is *an attribute that describe the context of an entity and is associated with the category of some of the entities involved in that context*. It is characterized by name, definition, objective and, particularly, *relevance* (the rationale of using the property to describe the context), *weight* (relative importance among the sibling properties in the context) and the *multiplicity of values* the property may hold. A context property may also specify related relevant properties. Thus, a context is described by a set of entities involved in the situation (`Entity::relatedEntity` relation in Fig. 1) and by a set of context properties belonging to the involved entities.

In C-INCAMI, contexts can be described for two purposes or scenarios [2]: first, to describe the *actual context* in which M&E activities are performed – as defined by an *information need*- and second, to describe the *target context* to which a given *contextual entity* is coherently applicable (see `applicableTo` relation). A *Contextual Entity* is defined as *an entity whose correct usage or interpretation is sensitive to the context in which it will be applied and/or analysed*. Here, a contextual entity is an instance of a given concept from the C-INCAMI model. Concepts are declared as *contextual entities* by establishing an *is-a* relation with the latter. In the `requirements` module, the concepts *Attribute* and *Concept Model* are considered contextual entities.

The context module also allows to specify and associate a *similarity criterion* to each context property to be used when comparing two given context descriptions. A similarity criterion is *an assessment pattern that allows to determine the semantic similarity between values specified for a given context property in different context descriptions*. The similarity is determined taking one of the context descriptions as a reference and comparing it against a second context description, so the similarity criterion of the first context property is used. To this aim, three types of criterion were considered: an *exact match* comparison (a one-to-one equality for a set of values using a logic aggregator), a threshold match comparison and a range match comparison.This comparison is the building block of a context sensitive recommendation mechanism [2] in which target context descriptions of a given contextual entity are compared against the actual context of an M&E project to determine the applicability of such contextual entity to the project (for example an attribute to be included in a concept model).

Following its definition, context properties can be quantified in the same way attributes are quantified for measurement purposes. Details of this mechanism are given in subsection 3.3; here we just say that a *metric* is assigned to a context property and *measurements* are performed on the corresponding entity involved in the context. However, for the two scenarios presented before, contexts are described in different ways. For *actual contexts*, relevant *entities* are associated to it, as well as relevant *(context) properties* belonging to these entities. The selected context properties are then specified (remember Fig. 1); particularly, an exact match *similarity criterion* is set as default. Then each context property is quantified following the specification of the associated *metric*. Each recorded measurement is associated to the similarity criterion and to the entity onto which the measurement was taken. *Target contexts* are described similar to actual contexts, with the difference that no entities are identified. Also context properties are quantified by measurements (following a selected metric) but in this case any similarity criterion type can be specified, adjusting its parameters, to cover the different contexts to which the contextual entity can be applied.

In the example, to describe the *actual context*, a number of relevant entities and properties are selected and quantified, as presented in Table 2.

Table 2. Some properties describing the *actual context* of the EQ evaluation

Name	**Entity category**	**Entity Category**	M&E Proyect
Definition	Category to which the evaluated entity belongs.	**Related Context Property**	Evaluation target
Relevance	The entity category may determine other r properties to be used in the context descrip may also help in selecting information elem during M&E design .	**Categorical Scale (Categorical Values)**	[Web application | Desktop application | Embeeded component | Web page | Test case | Application server]
		Multiplicity Of Values	0
Prop. URI	`http://myorg.com/context/EntityCategory`	**Measures (Value)**	Web Application

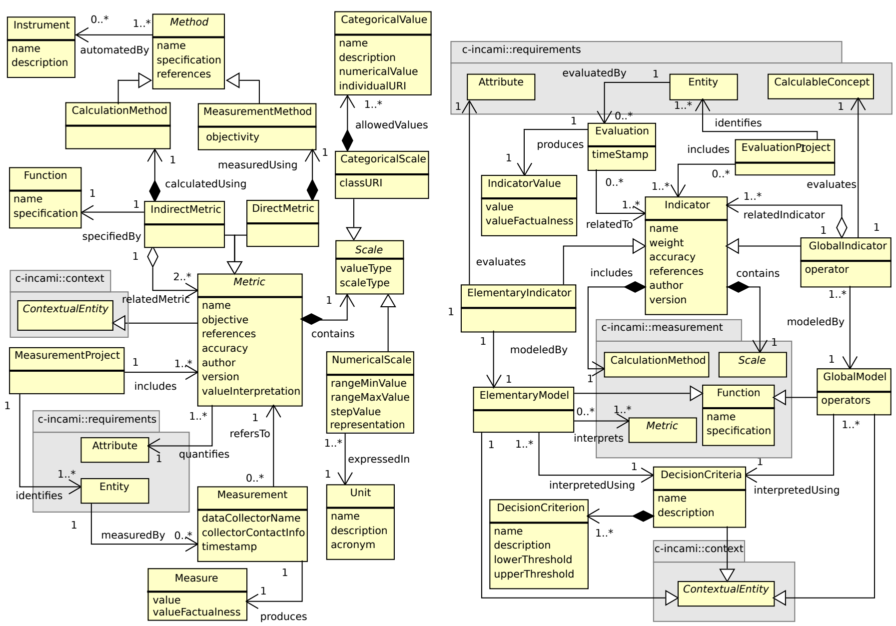
Name	**Evaluation target**	**Entity Category**	M&E Proyect
Definition	The scope or range within which evaluation results are interpreted for decision making.	**Related Context Property**	–
Relevance	It may help in selecting information elemen evaluation design.	**Categorical Scale (Categorical Values)**	[internal evaluation | external certification]
		Multiplicity Of Values	1
Prop. URI	`http://myorg.com/context/EvaluationTarget`	**Measures (Value)**	internal evaluation

Name	**Automated support**	**Entity Category**	M&E Proyect
Definition	Support for the automation of different methods and procedures to be used in the implementation of M&E activities.	**Related Context Property**	–
Relevance	It may help in selecting information elemen M&E design that rely on automated metho	**Categorical Scale (Categorical Values)**	[Additive Model | LSP Method | Lexile Analyzer® | Web crawler]
		Multiplicity Of Values	0
Prop. URI	`http://myorg.com/context/AutomatedSupport`	**Measures (Value)**	LSP method, Lexile Analyzer®, Web crawler

Name	**Supported natural language**	**Entity Category**	Web Application
Definition	Natural languages in which the content of the Web site is written.	**Related Context Property**	Target market scope, Target geographic region, Main content type
Relevance	It may help in measurement design since th site's text is part of the content whose qua measured and evaluated	**Categorical Scale (Categorical Values)**	An enumeration of natural languages' names.
		Multiplicity Of Values	0
Prop. URI	`http://myorg.com/context/SupportedNaturalLanguage`	**Measures (Value)**	Spanish

3.3 Measurement Design and Execution

The *measurement* module (see Fig. 3(a)) includes key concepts allowing to specify the design and execution of such activity by means of *Metrics* and related concepts. To each attribute in a concept model (recall Fig. 1) a metric is assigned which specifies a *method* (how to obtain a value), and a *scale* (how to represent it). The module allows to define *numerical and categorical scales*, each of one defined with their particular meta-data. All metrics are grouped under a *Measurement Project*, which also maintains references to entities being measured. Results of applying metrics can also be specified in *measurements* meta-data, associated to the entity from which they were taken, including the resulting *measure* (i.e. the value). The module allows to specify *Direct Metrics*, which include a *measurement method* to produce a value directly, and *Indirect Metrics*, which include a *calculation method* to produce a value by performing a calculation based on results obtained from related metrics.



(a) Measurement module (b) Evaluation module

Fig. 3. Class Diagram for the *measurement* and *evaluation* modules of C-INCAMI.

Here, *metrics* are considered *contextual entities* since their definition may entail a particular characteristic of the target context, mainly associated to the nature of attributes being measured or to the available resources needed to perform the measurement as defined in the metric.

Following the example, we illustrate the measurement design for the *Readability of text* attribute (coded 1.2.2 in the Fig. 2(b)). Here a metric must be selected from the M&E catalogue, this time using the *target context* of the metrics that quantify such attribute to determine their applicability to the *actual context*. One of the metrics found is *Lexile Text*, whose specification, together with its target context, are presented in Table 3. It can be observed that this metric can be applied both to English an Spanish texts and also requires a specific software tool. Also the metric *Fog Index* quantifying the same attribute is found. By definition, this metric is applicable to English texts and the obtained values are interpreted in terms of U.S. education grades needed to comprehend the text (reflected in the context property: *Supported Natural Language* = `English`). Reviewing the actual context properties described in Table 2, it can be seen that the *Lexile Text* metric is the only one applicable to the current M&E project.

Table 3. Specification of "Lexile Text" metric to quantify the "Readability of Text" attribute.

(a) Specification of the "Lexile Text" metric.

DIRECT METRIC			
Name	Lexile Text	**Objetive**	To assess a text's readability.
References	`http://www.lexile.com`	**Accuracy**	100%
Author	MetaMetrics, Inc.	**Version**	1.0
Value interpretation	The numeric representation of a text's readability (or difficulty). The bigger the value, the higher the level of difficulty of the text.		
SCALE (NUMERICAL SCALE)			
Value type	INTEGER	**Scale type**	ABSOLUTE
Representation	DISCRETE	**UNIT**	
Range Min Value	0	**Name**	Lexile
Range Max Value	2000	**Description**	The numeric representation of an individual's reading ability or a text's readability (or difficulty)
Step Value	10	**Acronym**	L
MEASUREMENT METHOD			
Name	Lexile Text Measure		
Specification	Follow the Lexile Analyzer® Guidelines for analyzing texts.		
References	`http://www.lexile.com`		

(b) Target context for the "Lexile Text" metric.

CONTEXT PROPERTY			
Name	**Supported Natural Language**	**Weight**	0.5
Relevance	This metric can be applied to a number of natural languages.	**Multiplicity of Values**	0
		Measures (value)	english, spanish
Exact Similarity Criterion		**Logic Aggregator**	OR
		Value Match	english
		Value Match	spanish
CONTEXT PROPERTY			
Name	**Automated support**	**Weight**	0.5
Relevance	This metric requires the use of a particular tool to perform the	**Multiplicity of Values**	1
		Measures (value)	Lexile Analyzer®
Exact Similarity Criterion		**Logic Aggregator**	OR
		Value Match	Lexile Analyzer®

Once all attributes in the concept model were assigned a metric, measurements are performed for each attribute, associating the result to the corresponding entity. In the example, the metric *Lexile Text* produces a measurement whose measure (value) is $750L$.

3.4 Evaluation Design and Execution

The *evaluation* module (see Fig. 3(b)) includes key concepts allowing to specify the design and execution of the evaluation. Evaluation is guided by *Indicators* that interpret non-functional requirements specified in *concept models* (from `requirements`). Like metrics, indicators produce values to be used for later interpretation, by using a *calculation method* and *scale* to represent these values. All indicators are grouped under an *Evaluation Project*, also maintaining references to actual entities being evaluated.

Two kinds of indicators can be specified in C-INCAMI. *Elementary indicators* evaluate or interpret *attributes* in a concept model by applying an *elementary model* that maps domain specific values obtained by means of a *metric* to values represented in a normalized scale that allows to uniformly interpret non-functional requirements. *Global indicators* evaluate or interpret *calculable concepts* in a concept model by applying a *global model* that aggregates values obtained from related indicators. Results of executing the evaluation can also be specified in *evaluations*, each of them with the associated *indicator value* and corresponding meta-data. Both global and elementary models allow to specify *decision criteria* to represent different acceptability levels for the resulting indicator's values.

Here, both *elementary and global models* are considered *contextual entities*. In the first case, elementary models are closely related to *metrics* –also contextual entities. In the second case, different *global models* may be applied due to particular assessment requirements, considering that global models may provide different degrees of flexibility and precision. Likewise for *decision criteria*.

Following with the example, for the evaluation design, *elementary and global indicators* –and their corresponding models- must be decided. First, a common *scale* is selected for all indicators to unify the interpretation. In this case it is established a *numeric scale* of type RATIO, using a REAL value type with CONTINUOUS representation, ranging between 0.0 and 100.0. Then, an elementary indicator and model is assigned to each attribute in the concept model. For the *Readability of Text* attribute (coded 1.2.2 in Fig. 2(b)) only one elementary model is found in the M&E catalogue, which converts values in the scale of the *Lexile Text* metric to three possible indicator values (see Equation 1). This elementary model is targeted to contexts where the main entity category is a *Web Page* or a *WebApp*, satisfying the applicability to the actual context.

$$EM_{LT} = \begin{cases} 0 & si\ 1500 < M_{LT} \\ 50 & si\ 800 < M_{LT} \leq 1500 \\ 100 & si\ 0 \leq M_{LT} \leq 800 \end{cases} \tag{1}$$

For the global model to interpret calculable concepts in the concept model, the LSP (Logic Scoring of Preference) Model [22] is selected. This model takes a number of *preference values* and corresponding *weights* as input, and uses a *logic operator* (from a set of operators ranging from full disjunction to full conjunction), to produce a preference value for a given calculable concept, also using the scale selected before (see Equation 2). Because of this model, weights must be set for all indicators and also a logic operator for each global indicator (as shown in Fig. 4).

$$e_0 = (W_1 e_1^r + ... + W_k e_k^r)^{1/r}, where (W_1 + ... + W_k) = 1, W_i > 0, i = 1, ..., k. \tag{2}$$

Then, to unify indicator interpretations a single set of decision criteria is selected: the *"internal evaluation criteria"* (see Table 4) specifies four accept-

(GI) External Quality /Operator: CA

1. **(GI)** Usability **/Operator**: $C-$ **/Weight**: 0.2
   1.1. ...
2. **(GI)** Functional Suitability **/Operator**: $C-$ **/Weight**: 0.2
   2.1. ...
3. **(GI)** Information Quality **/Operator**: $C-$ **/Weight**: 0.3
   3.1. **(GI)** Suitability **/Operator**: $C-$ **/Weight**: 0.7
      3.1.1. **(GI)** Coverage **/Operator**: $C--$ **/Weight**: 0.45
         3.1.1.1. **(EI)** Foreign language support **/Weight**: 0.25
         3.1.1.2. **(EI)** Line item information completeness **/Weight**: 0.2
         3.1.1.3. **(EI)** Product description completeness **/Weight**: 0.25
         3.1.1.4. **(EI)** Shipping and handling information completeness **/Weight**: 0.15
         3.1.1.5. **(EI)** Return policy information completeness **/Weight**: 0.15
      3.1.2. ...

Fig. 4. Excerpt of the designed indicators to interpret *EQ* for *Cuspide.com*

ability levels to interpret evaluation results and to make decisions accordingly. Also in this case, the *target context* of this criteria, which includes only the context property *"evaluation target"* with the value `internal evaluation`, fully satisfies the applicability to the *actual context*.

Table 4. Specification of the *"Internal evaluation criteria"* Decision Criteria

Decision Criterion	**Name**	Satisfies completely	**Upper Threshold**	100
	Description	Requirements are fully satisfied.	**Lower Threshold**	90
Decision Criterion	**Name**	Satisfies with improvements	**Upper Threshold**	90
	Description	Improvement actions should be implemented in the medium-term.	**Lower Threshold**	80
Decision Criterion	**Name**	Satisfies marginally	**Upper Threshold**	80
	Description	Improvement actions should be implemented in the short-term.	**Lower Threshold**	60
Decision Criterion	**Name**	Does not satisfies	**Upper Threshold**	60
	Description	Improvement actions should be implemented urgently.	**Lower Threshold**	0

3.5 Domain Integration

As introduced in Section 1, an improvement made to INCAMI was the capability to integrate application domain specifications to M&E specifications. This capability relies on the *URI* mechanism used to identify resources both in the traditional Web and lately in the Semantic Web. Particularly, the integration mechanism designed for C-INCAMI requires that the information from the application domain of an organization be represented in *RDF*, following the *ontology modelling approach* mentioned at the beginning of Section 3.

So, the integration mechanism consists of basically an *URI attribute* in the corresponding C-INCAMI concept specification linking to a particular description in the RDF domain information space of the organization, which contains the definitions used in its activity. To better understand this mechanism, follow the example in Fig. 5. In the *requirements* module (remember Fig. 1), an *Entity Category* links to an RDF class definition (`classUri`) corresponding to some concept in the organization's application domain; an *Entity* links to an RDF resource or individual description (`individualURI`), instance of the RDF class linked by entity category to which the entity belongs; and an *Attribute* links to a RDF property definition (`propertyURI`) whose domain is the RDF class linked by the entity category with which the attribute is associated.

In the *measurement* module (Fig. 3(a)), a *Categorical Scale* can be linked to an RDF class definition (`classURI`), analogously to Entity Categories, and *Categorical Values* included in such scale can be linked to resources or individuals descriptions (`propertyURI`) whose type is an RDF class linked to the corresponding categorical scale.

This integration mechanism strengthens the consistency of M&E specifications with regard to information in the application domain. Also further mechanisms can be devised to automatically gather specifications of entities, attributes, context properties, and so forth, from organization's repositories.

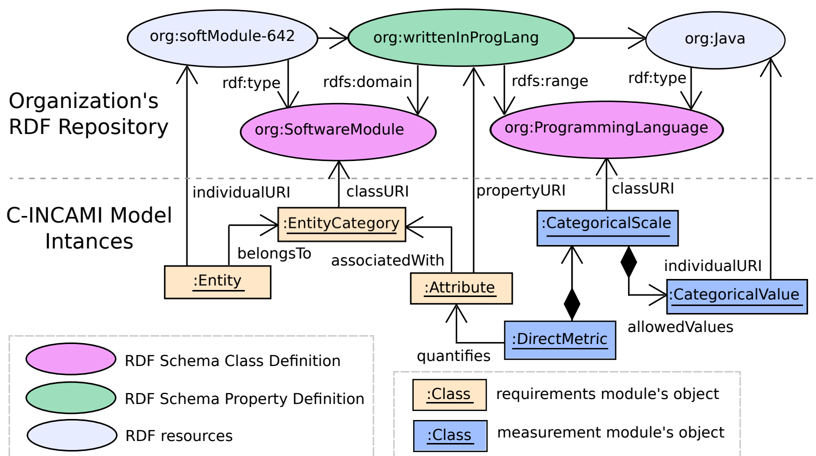


Fig. 5. C-INCAMI specifications integrated to application domain RDF descriptions.

4 Software Architecture

To support the C-INCAMI modelling proposal, a software architecture and tool was designed and developed. In the designed architecture (see Fig. 6) a WebApp includes, at the core, the *C-INCAMI Subsystem* which implements the C-INCAMI modules, as presented in Section 3, plus a set of utility modules to coordinate their operation. On top of it lies a software layer enforcing the workflow of the M&E activities –as defined in the corresponding process [3]-, using the core C-INCAMI modules to represent the information. A presentation layer is responsible to provide a web interface to evaluators. In turn, the C-INCAMI subsystem uses a set of data access modules to provide persistence capabilities for *M&E projects*, to access the *organization's repositories* containing application domain information, and to manage the *C-INCAMI Catalogue*.

The *C-INCAMI catalogue* [15] stores M&E specifications following the structure of the C-INCAMI concepts definitions. The stored specifications include reusable information elements, such as *attributes*, *entity categories* and *entities*, *context properties*, *concepts* and *concept models*, *metrics*, *elementary* and *global models* and *decision criteria*. Also, each *contextual entity* (as presented

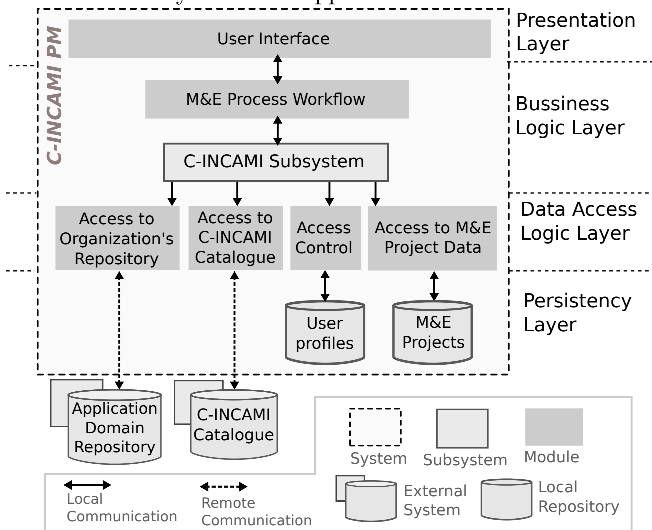


Fig. 6. Software Architecture supporting the C-INCAMI information model.

along Section 3) is stored with its corresponding *target context* description. These are represented in RDF and made available through *Sesame* (`http://www.openrdf.org/`), a framework for RDF/S storage, querying and reasoning.

The *Application Domain Repository* is typically a relational database. Although the module to retrieve M&E specifications from DB's was not included in the implementation supporting C-INCAMI, an important number of tools exists to accomplish this (`http://www.w3.org/2001/sw/rdb2rdf/`).

5 Concluding Remarks and Future Work

In this article we have highlighted two aspects of C-INCAMI, an IM defining in an explicit and structured way the concepts, properties and relations necessary to represent consistent and coherent information (data and meta-data) in the design and execution of M&E activities. Particularly, we have stressed on the capability of C-INCAMI to represent context information associated to M&E specifications, used also to provide context sensitive recommendations when designing M&E and interpreting results. We have also discussed the integration mechanism designed for C-INCAMI allowing to reuse information elements (data and meta-data) from the organization's application domain in the specification of M&E activities. Finally, we have described a software architecture to support the C-INCAMI model, including the highlighted aspects, to ease the implementation of the proposal. As reviewed in Section 2, both aspects represent a contribution with respect to existing proposals. These aspects were also illustrated by excerpts of examples as a proof of concept.

Future works will strive in provide C-INCAMI the capability of specifying information needs and objetives at different organizational levels to track low level M&E results to high level goals of the organization.

References

1. Molina, H.: Soporte organizacional de medición y evaluación orientada a objetivos y sensible al contexto. PhD thesis, Universidad Nacional de La Plata (2012)
2. Molina, H., Olsina, L., Rossi, G.: Context-based recommendation approach for measurement and evaluation projects. Journal of Software Engineering and Applications **3**(12) (2010) 1089–1106
3. Becker, P., Lew, P., Olsina, L.: Specifying process views for a measurement, evaluation, and improvement strategy. Advances in Soft. Eng. (2012) 28
4. Olsina, L., Molina, H., Papa, F.: How to Measure and Evaluate Web Applications in a Consistent Way. In: Web Engineering: Modelling and Implementing Web Applications. Springer London (2008) 385–420
5. Olsina, L., Papa, F., Molina, H.: Organization-oriented measurement and evaluation framework for software and web engineering projects. In: Proceedings of ICWE 2005. Volume 3579 of LNCS., Springer-Verlag (2005) 42–52
6. Basili, V.R., Caldiera, G., Rombach, H.D.: The goal question metric approach. In: Encyclopedia of Software Engineering. Wiley (1994) 528–532
7. Briand, L.C., Morasca, S., Basili, V.R.: An operational process for goal-driven definition of measures. IEEE Transactions on SE **28**(12) (2002) 1106–1125
8. ISO/IEC 15939: Software engineering - software measurement process (2001)
9. Gresse, C., Hoisl, B., Wüst, J.: A process model for GQM-based measurement. Technical Report STTI-95-04-E, STTI-KL (1995)
10. Florac, W.A., Park, R.E., Carleton, A.D.: Practical software measurement: Measuring for process management and improvement. Guidebook CMU/SEI-97-HB-003, SEI, Carnegie Mellon University (1997)
11. Kitchenham, B., Hughes, R.T., Linkman., S.G.: Modeling software measurement data. IEEE Transactions On Software Engineering **27**(9) (2001) 788–804
12. Card, D.N.: A practical framework for software measurement and analysis. Technical report, Auerbach Software Management Strategies (2000)
13. ISO/IEC 99:1993: Int'l vocabulary of basic and general terms in metrology (1993)
14. García, F., Serrano, M., Cruz-Lemus, J., Ruíz, F., Piattini, M.: Managing software process measurement: A metamodel-based approach. IS **177**(12) (2007) 2570–2586
15. Molina, H., Papa, F., de los Angeles Martín, M., Olsina, L.: Semantic capabilities for the metrics and indicators cataloging web system. In Matera, M., Comai, S., eds.: Engineering Advanced Web Applications, Rinton Press Inc. (2004) 97–109
16. Olsina, L., Garrido, A., Rossi, G., Distante, D., Canfora, G.: Web application evaluation and refactoring: a quality oriented improvement approach. Journal of Web Engineering **7**(4) (December 2008) 258–280
17. Lew, P., Olsina, L., Becker, P., Zhang, L.: An integrated strategy to systematically understand and manage quality in use for web applications. Requirements Engineering **16** (2011) 1–32
18. ISO/IEC 9126-1: Software engineering- product quality: Quality model. (2001)
19. ISO/IEC 14598-1: ISO/IEC 14598-1:1999 – information technology – software product evaluation: General overview (1999)
20. Barcellos, M.P., Falbo, R.d.A., Dal Moro, R.: A well-founded software measurement ontology. In: Proc. Formal Ontology in Information Systems, IOS Press (2010) 213–226
21. Lawler, J., Kitchenham, B.: Measurement modeling technology. IEEE Software **20**(3) (2003) 68–75
22. Dujmovic, J.: A method for evaluation and selection of complex hardware and software systems. In: CMG 96 Proceedings. (1996) 368–378