J. Software Engineering & Applications, **2010, 3, 1089-1106**
doi:10.4236/jsea.2010.312128 Published Online December 2010 (http://www.SciRP.org/journal/jsea)

Context-Based Recommendation Approach for Measurement and Evaluation Projects

Hernan Molina[1], Luis Olsina[1], Gustavo Rossi[2,3]

[1]GIDIS Web, School of Engineering, National University of La Pampa, General Pico, Argentina; [2]LIFIA, School of Informatics, National University of La Plata , La Plata, Argentina; [3]National Council of Scientific and Technical Research, La Plata, Argentina
Email:{hmolina,olsinal}@ing.unlpam.edu.ar, gustavo@lifia.info.unlp.edu.ar

Received November 19th; 2010, revised November 30th, 2010; accepted December 10th, 2010.

ABSTRACT

Data, information and knowledge are recognized as useful assets for analysis, recommendation and decision making at any business level of an organization. Providing the right information for decision making considering different user-requirements, projects and situations is, however, a difficult issue. A frequently-neglected challenge is to cope with the influence of contextual issues affecting the success of outcomes and decisions. Particularly, when conducting quality evaluations in software organizations, it is of paramount importance to identify beforehand the contextual issues affecting outcomes and interpretations for measurement and evaluation projects. Therefore, the relevant context information should be clearly identified, specified and recorded for performing more robust analysis and recommendations. In this work, a domain-independent context model and a mechanism to integrate it to any application domain is presented. The context model is built upon a measurement and evaluation framework enabling quantification and semantic capabilities. The context model is then integrated in the mentioned framework itself to enable recommendations in measurement and evaluation projects.

Keywords: *Context-Aware, Context Model, Recommender System, Measurement, Evaluation, C-INCAMI Framework*

1. Introduction

It is well recognized that information and knowledge play a central role in the success of organizations involved in the development or use of software and web applications. This is evidenced by the increasing amount of research and events in areas related to information and knowledge management, spanning a wide range of different fields such as e-business, e-learning, web engineering and Web 2.0, organizational memory management, just to mention a few. In any case information cannot be interpreted in isolation due to the influence of contextual factors such as temporal, functional, structural, and user viewpoint aspects. Indeed, recent research has started paying attention to the important role context plays on how information is handled and interpreted and how context can be represented and gathered to that end. These are the concerns of the so called context-aware approach, firstly applied in mobile and ubiquitous applications [1] where the need of context is evident as a key driver for delivering content and services. However context-awareness is also useful in other fields such as software and web engineering. Regardless of the field of application, a context model is a key issue and a starting point in successfully applying a context-aware approach. To that aim, we believe that in order to achieve an effective context-aware application, the designed context model should have support for context property types, both at a structural and semantic level, and for the contextualization of domain concepts. Nevertheless we observe in existing related works, regardless of the chosen modeling approach, the lack of a sound approach towards the structuring and quantification of context properties: context models are built from scratch disregarding measurement theory concepts, i.e. no meta-data about scales (and scale types), units and quantification methods are specified. In this sense, we argue that the quantification of context properties is just the quantification of some entity's attributes; thus, context models could benefit from existing measurement models and frameworks to build upon them. In this work we present a context model built upon the measurement and evaluation framework so called INCAMI (*Information Need, Concept Model, Attribute, Metric and Indicator*) [2]. This framework provides concepts such as *Entity*, *Attribute*, *Metric*, *Scale*, *Unit*, among others, which can be used as a foundation to build applications capable of specifying data and meta-data for measurement and evaluation design and implementation. Another issue to be addressed by a context-aware application is the establishment of mechanisms to gather relevant context information from the actual and relevant situation of the entity of interest, and also to use that information in providing context-based recommendations. Lastly, architecttural support to ease the integration of context-related functionality with existing domain applications must be provided in order to realize the benefits of having context descriptions of the situation at hand. In this work a mechanism based on semantic catalogues was designed and implemented together with a prototype of a context-based recommender system intended to be used by organizations (details of this implementation are out of the scope of this article).

In this article the context-aware approach is applied to web engineering; particularly, we are interested in context information affecting the design, implementation and outcomes interpretations for measurement and evaluation (M&E) projects in software organizations. To this aim, it is of paramount importance to clearly identify, specify and record from the start context information relevant to a particular M&E project [3-6] to 1) assist in design decisions—e.g. selecting a metric, whose correct applicability depends on the target context, to quantify some entity attribute-, and 2) to make explicit the underlying issues that characterize a given entity or the project itself for analysis and comparison purposes. In the end more robust analysis and recommendations could be achieved among the whole organization's projects. Several proposals exist in the literature covering different views for the M&E domain—i.e. processes, conceptual models, methods [3-5, 7-10] although none of them cover the context issue in such a comprehensive way to achieve the above mentioned capabilities.

Therefore we extend and improve our previous proposal [11] in which a context model was included in the INCAMI M&E framework allowing to specify in a clearly structured way context-aware data and meta-data for M&E projects as a foundation to support context sensitive recommendations to M&E activities. The result was the C-INCAMI (as for Contextual INCAMI) framework. The extension presented here includes a mechanism to integrate existing domain concepts in the context model and to compare context descriptions as a foundation for a recommender system.

Summarizing, the main contributions of this article are the following:

- A domain-independent context model built upon a M&E framework—an enhancement of [11] and a mechanism to integrate existing domain concepts with the context model;
- A context comparison mechanism—based on similarity and applicability metrics—allowing the realization of the above two context-aware recommendation cases for M&E;

The result is an improved context-aware M&E framework with recommendation capabilities that can be integrated to existing application domains.

The organization for the reminder of this article is as follows: Section 2 presents an overview of the applied context-aware approach, followed by an explanation of the proposed context model and the domain integration mechanism. In Section 3 the application of the context-aware approach to measurement and evaluation is presented. First, in subsection 3.1, we give details of how context is specified and used; then, the design of metrics for the context comparison mechanisms are presented; lastly, a proof of concepts shows some details of the use of the C-INCAMI framework and the supporting architecttture. Related work and discussion are addressed in Section 4; and, finally, Section 5 discusses, conclusions and future work.

2. A Context-Aware Approach

The rationale of the context-aware approach is based on the widely accepted premise that information cannot be interpreted in isolation but rather considering its context. Applying this approach implies knowing and processing quantitatively the context attributes or properties that influence how domain information is interpreted and processed, for which a properly designed context model is necessary. A context-aware application is then one which depends on or uses context information to offer its services [12].

Applying the context-aware approach implies decisions on a set of issues, namely, a context model and mechanisms to capture and use context information. Decisions on these issues should be made according to the intended purpose of the approach. These will decide on the generality of the context-aware application, which would be desirable to some extent, for the sake of reusability, but without disregarding the trade-off with practical applicability.

But before engaging in any design or technical endeavor, it is necessary to understand what context is and how it relates to specific-domain information. In [12-15] authors provide slightly different definitions of context with different purposes and target domains. For this matter, we agree with Dey's general definition of context [12]: "any information that can be used to characterize the situation of an entity", which may be "a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves". However, rather than adopting a definition of context as is, we analyzed the definitions found in the cited literature and observed some common issues about context information:

- It is relative to a specific object or entity of interest;
- It should be relevant with regard to:
  - The specific task related to that entity,
  - The specific intention towards the entity related to that task,
  - The interaction of the entity with other entities of the context regarding the task and purpose;
- It is composed of a set of relevant properties with an associated conceptualization (i.e. structure) that makes their interpretation explicit. These properties may belong to the entity of interest or relevant entities related to it.

Summarizing the above listing (see **Figure 1**), we consider context information as the set of relevant properties (and its values) of the relevant entities involved in the situation. By *situation* we understand the summing of the entity of interest, the task and intention towards that entity and the interaction between this and other entities; and, by *relevant*, we mean that it may affect on how information is interpreted regarding the situation at hand.

Regarding the context model a decision should be made about the modeling approach following alignment with [16] taking into account the corresponding pros and cons that should be analyzed for the particular needs of the application. In any case, the model should enable representation of context information and its integration to the target domain based on the understanding of the characteristics of context information, as listed before.

Regarding the mechanisms to capture or specify context information two approaches arise [17], namely, context can be specified and attached at some point in time with the pertinent information to be used later as a key for selection, or it can be updated dynamically to adjust the system behavior actively. Since both approaches satisfy different needs the choice must be taken considering the intended purpose of the application.

Regarding the use of context information, mechanisms to process data instances of the context model should be designed. Basically, mechanisms are necessary to compare two given context descriptions, one from the actual situation or environment and the other from the relevant entity in the application domain, or both context descriptions from two separate application domain entities. These mechanisms should produce a value that expresses in some objective way the degree of similarity of the compared contexts. Again, the concrete mechanism depends on the purpose of the application.

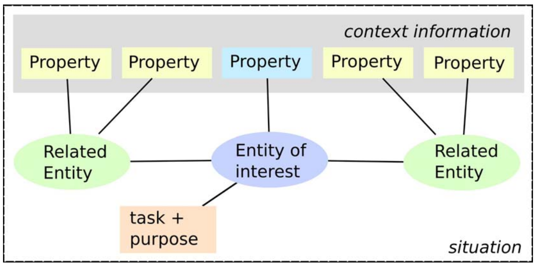


Figure 1. Overview of the elements in the characterization of context information.

The above considerations remain valid regardless of the application domain. For instance, in the case of the ubiquitous computing field, where a mobile user is the focus entity, context should be modeled and mechanisms should be devised for instance to discover and capture relevant information from the surrounding environment of users in order to offer them better services (relevant context aspects in this case are *time, location, device, activity, role, etc*.). In our proposal, like in [14,18,19], context is considered from the information and/or knowledge management standpoint, particularly in a software engineering domain with the aim of improving interpretations and analysis results. In this case, the entity of interest is any information element involved in some task, and the relevant context information comes from related entities in the domain considered relevant with regard to the task and purpose, e.g. related to processes, projects, resources, stakeholders, etc.

In this work we focus on the application of the context-aware approach to the measurement and evaluation domain aimed at making more coherent measurement designs, and more sound interpretations from evaluation results among projects. Context descriptions will be specified both for actual situations of M&E projects and for reusable information elements—for example metrics, quality models. etc. —which may be applicable to specific target contexts. These context specifications will provide ultimately the foundation for a M&E recommender system.

2.1. Context Model

Based on the general characterization of context presented earlier a model to represent context information was designed. To this aim, different representation approaches were analyzed [16], taking into account a set of requirements of interest to our proposal—some of which are common to the ubiquitous computing domain:

- context information should be able to be validated against the context model both at structure and instance level;
- contextual information should be precise and unambiguous allowing the consistent reuse of it;
- the context model should be able to be integrated to current systems;
- the model should be simple enough to be specified and processed with the aim of maintaining the sys-

tem cost-performance trade off;

- the model should be domain independent and allow tailoring to cover any application domain;

An object-oriented modeling approach was chosen for the high-level definition of the context model because of its expressiveness and, at the same time, its understandability for communication purposes. Moreover, because of the requirement of domain independence and integration to existing domains, an ontology approach will be used for the representation at the implementation level to take advantage of its facilities to interconnect different application domains. Both approaches are complementary because of their similarities.

In the above characterization we said that context is composed of properties belonging to the entity of interest or relevant entities associated to it in a given situation. We also argued that these context properties can be seen as ordinary attributes of some entity, thus the context model could benefit from existing measurement frameworks providing these concepts and those related with attributes quantification. In this work we use the INCAMI M&E framework [2] as a foundation to build our context model. For that purpose we briefly present next only the necessary framework's concepts; further details of the framework will be shown in Section 3.

The INCAMI framework specifies the concepts and relationships necessary to specify the design and implementation of measurement and evaluation activities. However, in order to build the context model only measurement related concepts are used as presented in **Figure 2**.

The gray boxes group related concepts as indicated by their labels. The framework provides for the specification of concrete *Entities* belonging to some *Entity Category*. In turn each category has a number of associated *Attributes* that can be used to describe entities belonging to that category. Each attribute can be quantified by a number of-*Metrics*—although only one is used in a particular case. Broadly speaking, each metric is defined by a *Scale* de scribing the values the attribute may assume and a *Method* indicating how that value can be obtained. Notice from the related concepts (and their attributes) in the diagram of **Figure 2** that different scales and methods can be clearly specified. To quantify a given attribute of some entity a

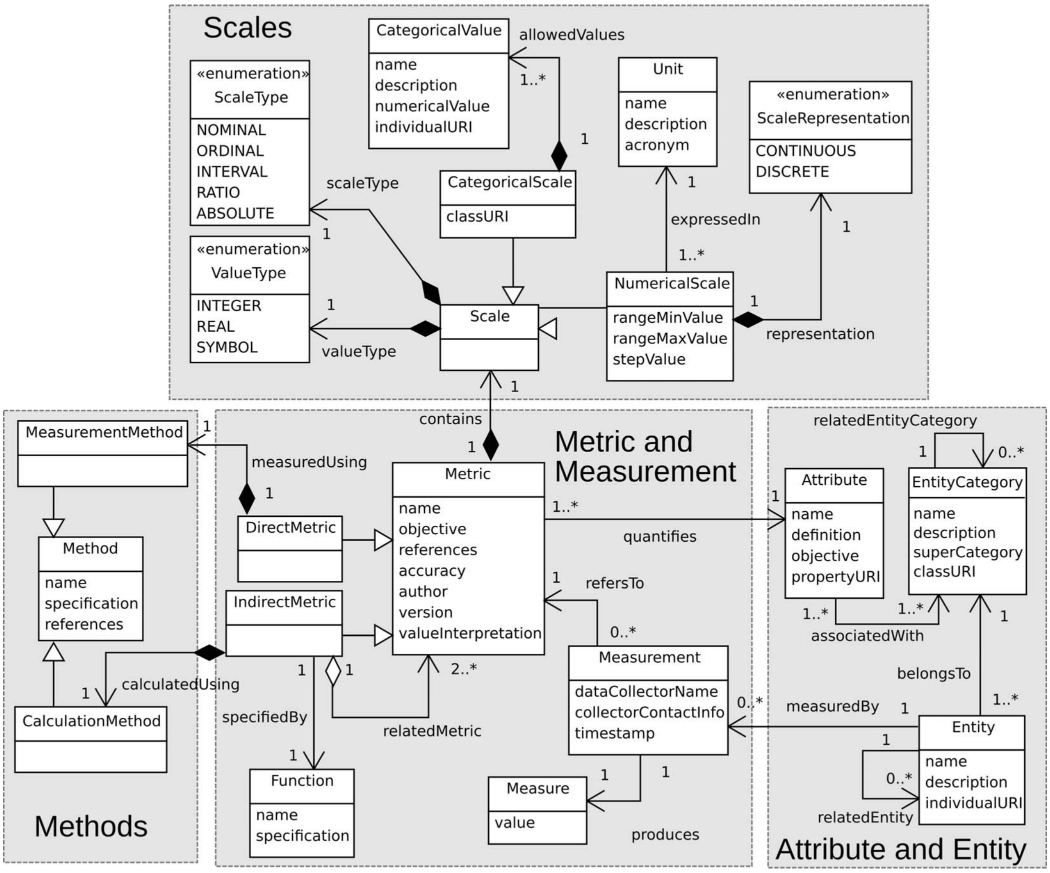


Figure 2. An UML class diagram showing some concepts and relationships of the INCAMI framework used by the context model.

Measurement is performed using the metric assigned to the attribute. These measurements, and their corresponding measured values, are then associated to the entity on which they were performed.

In order to reuse the quantification capabilities of the INCAMI framework and based on the characterization of context information, the context model is obtained by extending the concepts *Entity* and *Attribute*. The resulting model (see **Figure 3**) includes the definitions of the concepts and relationships necessary to specify context information and to contextualize (i.e. associate context information to) domain-specific information. The model was designed as a separate module to ease its integration with existing systems.

The context model is composed of three concepts. The main concept is *Context* which represents the relevant state of the situation of an entity:

***Definition* 1 (*Context*)** A special entity, described by a set of context properties that characterizes the relevant state of the situation of a given entity of interest. The situation of an entity involves the task and purpose towards that entity and its relationships with other entities regarding that task and purpose.

We consider *Context* as a special kind of *Entity* in which related relevant entities are involved (as depicted in **Figure 3** by the *related Entity* relationship). Relevant entities are those whose attributes are relevant in the interpretation of data relative to the task and purpose towards the entity of interest.

The situation is described by attributes of the relevant entities involved in the context of the entity of interest. These attributes are *Context Properties* that, properly specified and quantified, can be used to describe the relevant context of the entity of interest regarding the task and purpose:

***Definition* 2 (*Context Property*)** A attribute that describes the Context of a given entity; this attribute belongs to (the definition of) one of the entities that participates in the described Context.

A context property is described by the meta-data inherited from the Attribute class—i.e. name, definition and objective—and also by the following meta-data, particular for its purpose:

- *weight*: the relative value of the context property within a given context description; values range between 0.0 and 1.0.
- *relevance*: the pertinence of using the context property to describe the context for a particular information need or contextual entity.
- *multiValueCriterion*: logic criterion to determine if multiple values will be allowed for the context property and, if so, how they will be treated. Possible values are AND, OR, XOR—the first two values allow more than one value to be assigned to the property, while the last allows only one (the use of the logic properties of each operator are explained in Section 3).

The first two class attributes are meant to be assigned a value when the context property is being used to describe a particular context instance because of its definition: *weight* is relative to all the properties used to describe the context, and its relevance depends on the case and rationale of why the property is relevant. On the other hand, the multi Value Criterion is intrinsic to the property so its value is stored in its own definition.

As defined so far, this extension allows specifying the context, for a given entity of interest, by identifying, and associating to it, related relevant entities and then adding their attributes as context properties to the context. Then, metrics can be assigned to each context property and measurements be performed on the corresponding entity; these are then associated to the context object.

For the contextualization of domain-specific information, the model also allows specifying the ideal context in which a given context-sensitive entity could be applied or used, namely the target context, based on the nature and definition of that entity. These entities are called Contextual Entities:

D*efinition* 3 (*Contextual Entity*) An entity whose correct usage and/or interpretation is sensitive to the context in which it is applied and/or analyzed**.**

As it is designed, the context model is pluggable and reusable in existing applications, since the concepts in the context model do not specify which entity situation it characterizes or which are the contextual entities; rather, the domain implementing the context-aware approach is responsible for deciding which concepts are to be contextualized by establishing relationships with the corres ponding elements in the context model. Thus, this model can be used to describe context information for any application domain, such as ubiquitous computing, even though it was first conceived to apply to a web engineering environment.

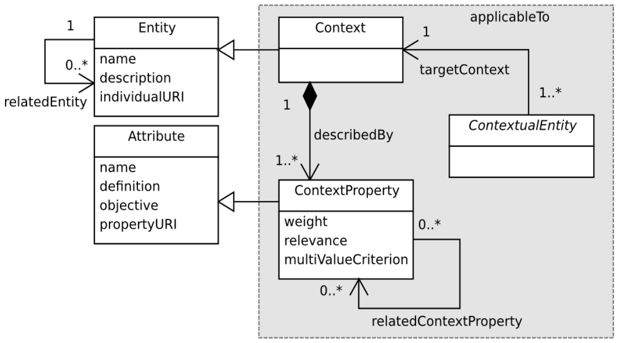


Figure 3. The context information module.

Notice that the simplicity of the context model defined by specializing measurement related concepts from existing frameworks. In this way details of how properties are quantified and associated to the entities from where they were taken remain in the extended measurement framework, thus hidden from the context model.

2.2. Representation and Integration of Context Information

As mentioned above an ontology approach will be used in the representation of context information, taking advantage of the characteristics of the approach and its similarities with the object model used for the context model. Particularly, we decided to use RDF (Resource Description Framework) [20] considering its standard status and since there is already a comprehensive set of supporting tools for storage, management and validation. This way, semantic catalogues could be implemented to store both domain-specific as well as context data and meta-data. Of course, this decision may cause a restriction to the implementation of the approach in the sense that application domains be encoded in RDF. Nevertheless we believe that currently RDF has become an industry standard for information interchange. In the worse case, a set of tools for relational to RDF mapping does exist to fill the gap[1].

Now, for the integration of the context model with application domains two aspects need to be addressed. First, concepts in the application domain can be contextualized by simply linking to *Context* descriptions or sub-classing *ContextualEntity*. Second, the context model can be linked to the application domain, as shown in **Figure 4**, by using the URI (Uniform Resource Identifiers) mechanism of RDF. Thus, the attributes *classURI*, *individualURI* and *propertyURI* in some of the concepts of the M&E model (**Figure 4(a)**), such as *Attribute*, *Entity* and *Entity Category*, are used to store the URI locator of the corresponding definition in the application domain (**Figure 4(b)**). This way, as shown in **Figure 4**, each entity involved in a context can be linked to the corresponding entity description in the organization's databases and the associated context properties can be linked to the corresponding properties of those entities as well. Moreover, the elements involved in the context model, entities and context properties, could be automatically gathered and specified from its source in the organization's databases.

3. Context-Aware Measurement and Evaluation

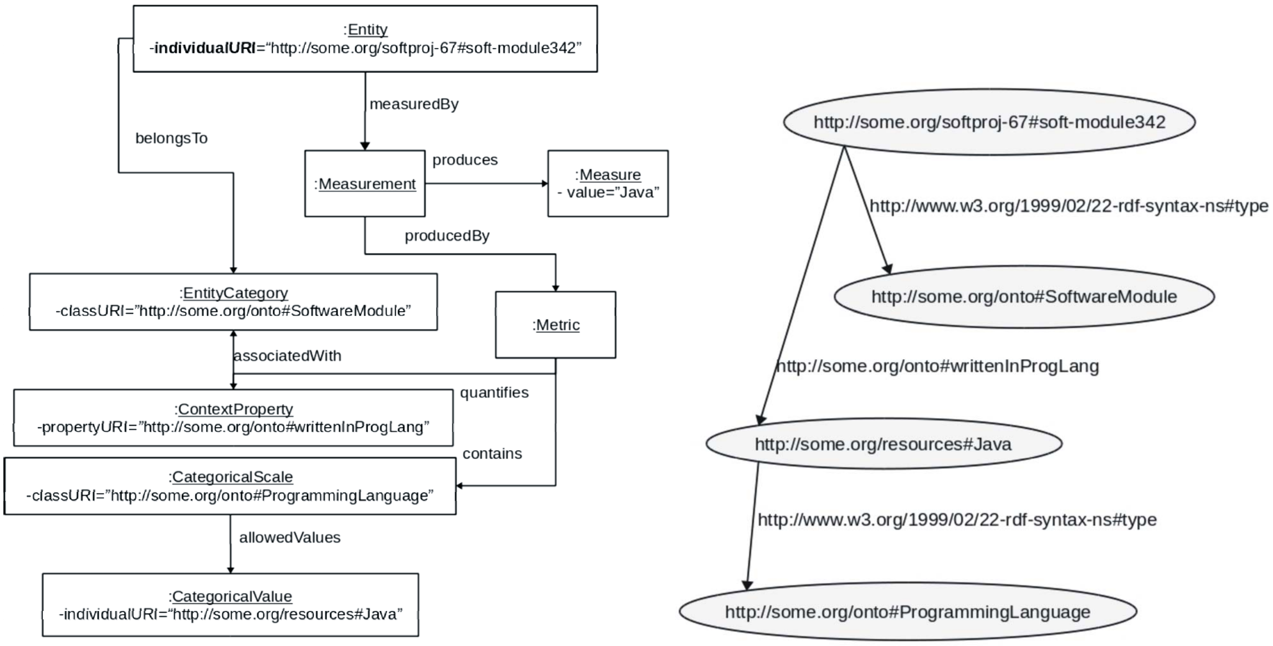
Measurement and evaluation are key processes in achieving quality for products and services offered by organizations, particularly in software and web engineering, where quality is a must for the success of any endeavor. This is evidenced by the amount of researches and standards [3-6,8-10,21] that document the activities and artifacts needed to that aim. A common denominator among these works is the assertion that the explicit specification and storing of context descriptions is needed as a parameter for selecting metrics and interpreting evaluation outcomes. However none of these works provide a thorough proposal to perform these context-related activities.

In order to illustrate the above statement, consider the following scenario in which we show how context information, if clearly structured, may help in a M&E design process. The scenario is based on an M&E case study [22] we performed using INCAMI. Initially, the *information need* is specified: the purpose is to *improve* the *external quality* of an *e-Commerce shopping cart*, as perceived by *customer users*. The shopping cart belongs to the *Cuspide.com WebApp*, an Argentinian company which sells books online. Then, non-functional requirements for the external quality are designed, including a set of high-level characteristics and associated measurable attributes. One of the attributes to measure is the *Readability of text* in help and support pages, necessary to determine the achieved satisfaction level for the *Learnability* concept. During measurement design two metrics are found in the organization’s catalogue to quantify this attribute: the *Gunning fog index*[2] and the *Lexile text*[3] metrics, so one of them should be selected. According to the available meta-data of each metric, that includes for instance its objective, value interpretation, author, references and scale-, the Lexile text metric’s values are interpreted as a numeric representation of an individual’s reading ability or a text’s readability (or difficulty), while Gunning fog index’s values are interpreted as an indicator of the number of years (of USA formal education) that a person requires in order to easily understand the text on the first reading. From these descriptions, either metric could be successfully applied. Nevertheless, some information is missing in these descriptions since the *Gunning fog index* metric is designed for English texts, while the *Lexile text* metric supports both English and Spanish. This information becomes important if we consider that the Cuspide’s shopping cart content is written in Spanish. Although this fact could have been included in the metric’s interpretation, it would still need a manual inspection to determine its applicability. Otherwise, by having a clear and structured specification of the supported *natural language* context property associated to each metric, this choice can be made automatically. Thus, we argue that measurement and evaluation is contextual, so explicit context descriptions should be provided to support more sound analysis and interpretations of projects outcomes.

[1] See developments of the W3C Workshop on RDF Access to Relational Databases at http://www.w3.org/2007/03/RdfRDB

[2] http://www.idph.state.ia.us/health_literacy/common/pdf/tools/gunning pdf

[3] http://nces.ed.gov/pubs2001/200108.pdf


:Entity
-individualURI="http://some.org/softproj-67#soft-module342"
measuredBy
belongsTo
:Measurement
produces
:Measure
- value="Java"
producedBy
:EntityCategory
-classURI="http://some.org/onto#SoftwareModule"
:Metric
associatedWith
quantifies
:ContextProperty
-propertyURI="http://some.org/onto#writtenInProgLang"
contains
:CategoricalScale
-classURI="http://some.org/onto#ProgrammingLanguage"
allowedValues
:CategoricalValue
-individualURI="http://some.org/resources#Java"
http://some.org/softproj-67#soft-module342
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://some.org/onto#SoftwareModule
http://some.org/onto#writtenInProgLang
http://some.org/resources#Java
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://some.org/onto#ProgrammingLanguage

(a) C-IMCAMI Object model

(b) RDF graph from the organization's data base

Figure 4. Example of the integration of context information with the application domain.

Next, we present an overview of the INCAMI framework which will be extended to support this capability.

In Section 2.1, we presented an initial overview of INCAMI, focusing only on measurement related concepts. Now we go further by giving more details about the INCAMI framework. As mentioned earlier, INCAMI provides a domain model defining all the concepts and relationships needed to support M&E processes in software and web projects, as part of an organization's quality assurance program. Thus, data and meta-data about *Information Needs*, *Concept models*, *Attributes*, *Metrics*, *Indicators*, and related concepts (see **Figure 5**) can be specified to document artifacts involved in the process.

INCAMI is based on an ontology [23] and also give support to the WebQEM (Web Quality Evaluation Method) methodology [2]. In addition, a prototype tool and an organizational catalogue were built [24] following the INCAMI framework model to maintain the instances of all defined concepts. So the catalogue can be used in different M&E projects, providing reuse mechanisms and making the check for meta-data consistency in analysis processes easier. In turn, the INCAMI framework and the defined M&E process can be used to instantiate methods and tools.

The framework is structured in modules or packages, namely: *project management*, *non-functional requirements specification*, *measurement design and implementtation*, and *evaluation design and implementation* (see [2] for a thorough discussion of all modules):

- The *project management* module deals with the specification of managerial information about M&E projects;
- The *non-functional requirements specification* module deals with the definition of the information need and the specification of requirements by means of one or more concept models (e.g. external quality models, quality in use models, etc.) and their attributes;
- The *measurement design and implementation module* deals with the specification of concrete entities to be measured, the metrics selection (from the catalogue) to quantify the attributes, and the recording of the gathered measures; this module is centered in the metric concept;
- The *evaluation design and implementation module* deals with the definition of indicators, decision criteria and aggregation models that will help to enact and interpret the selected concept model.

Briefly outlined, INCAMI follows a goal-oriented approach in which all the activities of the M&E process (e.g. metrics and indicators selection, measurement and evaluation execution, etc.) are guided by a stated *Information Need* which specifies particular non-functional requirements of some *Entity* for a particular *purpose* and *user viewpoint*. Non-functional requirements are repre sented by *Concept Models* including high-level concepts

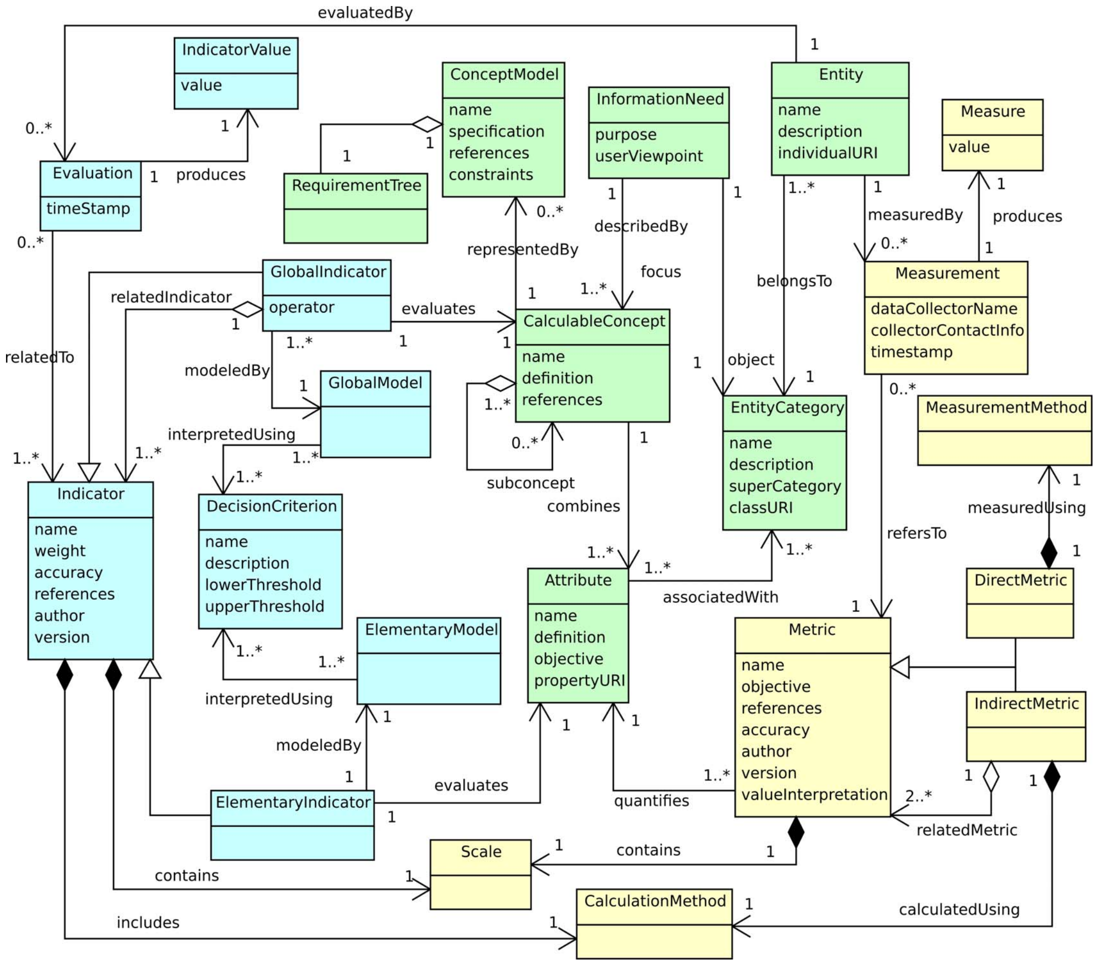


Figure 4. Diagram showing the main concepts of the INCAMI framework.

or characteristics, as in ISO 25010's quality models [21], as well as quantifiable *Attributes* associated with the entity under analysis. Concept models are the backbone for measurement and evaluation. *Measurement* is specified and implemented by using *Metrics*, defining how to obtain and represent attributes' values; basically, a metric represents meta-data about the *measurement or calculation method* and the measurement *scale*. *Evaluation* is defined and implemented by using *Indicators*, which specify how to interpret attributes' values and high-level concepts of the quality model. Also the framework is organization centered in the sense that the definition of the involved concepts allows its integration and implementation along software and web projects within an organization, carrying out measurement and evaluation activities as part of its quality assurance program.

In order to support the contextualization of these concepts, the M&E framework must be extended using a context-aware approach. In the following sub-sections we present the application of the context model and related mechanisms presented in the previous sections to the INCAMI framework to obtain support for context-aware M&E in a clear and structured way in software and web projects. The improved framework is called C-INCAMI.

From the definitions presented in Section 2.1, we now perform a thorough analysis of their elements in order to apply them to the M&E domain based on the terms defined in the INCAMI framework. It turns out that the entity of interest, for which context is specified, is the entity to be measured and evaluated to satisfy a specific information need. This could be any instance of a product, resource, process, or service being assessed. For example, in the above motivating scenario the entity of interest is the *Cuspide.com's shopping cart*. The situation from which relevance is considered is the M&E of the entity of interest to satisfy the stated information need purpose, focus concept and associated non-functional requirements, and viewpoint, and also the entities related to the entity of interest, as far as they are relevant to the assessment meaning those that could affect how the focus concept of the evaluation is interpreted on the entity to be assessed. The entities to assess are typically involved in a software project performed by an organization. Therefore, potential relevant related entities from which information may be extracted as context properties are:

- The project in which the entity is involved, including the resources, artifacts, assets or other products used in the project affecting the entity;
- The processes applied to the entity;
- The organization that runs the project itself;
- The stakeholders interested in the outcomes of the evaluation;
- The environment external to that organization.

So context properties are obtained from properties associated to the definition of these concepts in the data base schemata of the organization's domain model. In turn, values for these properties may be obtained from projects and organizational memories. For example, in the situation of the "*Cuspide.com shopping cart*" evaluation in the motivating scenario, the entity "*Cuspide.com WebApp*" where the shopping cart is included—is relevant since its attribute "*supported natural language*" may affect the measurement and interpretation of *readability of text* in the shopping cart help page a quality attribute included in the non-functional requirements for external quality; thus this attribute is considered as a context property. From the previous analysis, we could summarize a tailored definition of context as *a special entity, described by a set of context properties that characterize the relevant state of the situation of an entity subject to M&E. The situation of the entity is defined by the information need and its interaction with other entities from which context properties are taken- regarding that need, as long as they affect the implementation and interpretation of M&E for the subject entity.*

Once the context model has been understood and interpreted for the M&E domain it should be included in the INCAMI framework as a new module. Then the necessary relationships should be established to contextualize M&E related concepts (see **Figure 6**). Two scenarios may be distinguished at this point. First, to describe the *actual context* characterizing the current situation of the entity being assessed in a given M&E project, the corresponding *Information Need* is associated to a *Context* object (see the *characterizedBy* association in the cited Figure). Second, concepts in the INCAMI framework, whose definition could determine they are more or less applicable to a given *target context* are therefore considered a kind of *Contextual Entity*. The identified contextual entities in INCAMI are: *Concept Model*, *Metric*, and *Indicator* (see the generalization relationships in the same Figure).

To complete the integration, all the context related data and meta-data need to be stored in an organizational catalogue together with M&E data and meta-data to provide all the projects with a single and consistent set of properties.

3.1. Specifying and Using Context

Now we discuss the specification and use of context information for the two scenarios of contextualization presented in the previous section.

In order to specify the *actual context*, which characterizes the current situation defined by the information need in a particular M&E project, the relevant entities involved in the context are identified. This is done by inspecting in the organization's catalogues the relationships of the en--tity to be assessed with entities belonging to different categories such as product, process, project, resource, stakeholder and the hosting organization itself (or related ones). Once they are identified, the relevant properties associated to each of them should be selected together with the corresponding value, again from the organization's catalogues. The choice of including each property as relevant to describe the actual context is a decision of the evaluator and its rationale should be specified for each context property. Actual context descriptions can be used in the interpretation and analysis of M&E results. For instance, when comparing evaluations results from two M&E projects assuming both of them are comparable on the basis of the entity category, non-functional require-

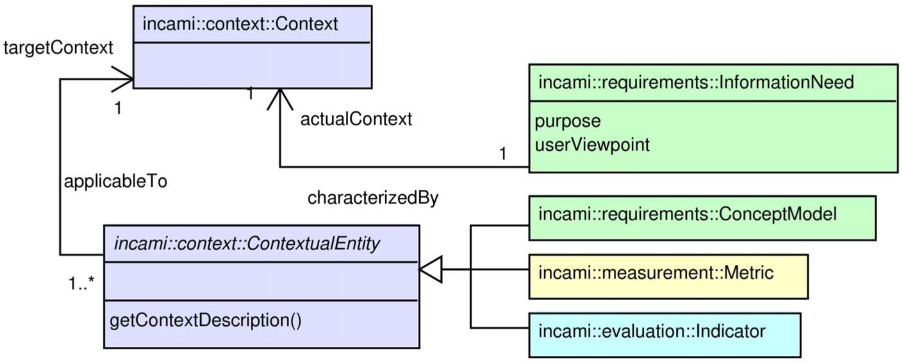

incami::context::Context
targetContext
1
1
actualContext
incami::requirements::InformationNeed
purpose
userViewpoint
1
applicableTo
characterizedBy
incami::context::ContextualEntity
1..*
getContextDescription()
incami::requirements::ConceptModel
incami::measurement::Metric
incami::evaluation::Indicator

Figure 5. Contextualization in C-INCAMI.

ments, metrics and so forth the actual context can be compared to assure both projects are actually comparable; otherwise, analysis results can be unreliable.

In the case of a *target context*, which characterizes the ideal situation in which a contextual entity of the INCAMI framework is applicable, the specification should be done *a priori* and independently of any particular M&E project and should be stored together with the contextual entity in the organization's catalogue. As before, the rational of including each context property should be recorded for each of them. In this case, no related entity is associated with the target context. Target contexts can be used in M&E design, when decisions may be required to select context-sensitive artifacts for example, a metric to quantify a given attribute to determine whether they can be coherently applied to the M&E project by comparing it to its actual context.

Now, whether describing *actual* or *target* context, the relevant context properties need to be quantified in order to characterize a given situation. When describing the *actual context*, a *Context* object is associated with a number of relevant entities and a set of *Context Properties*. Since context properties are defined as an *Attribute* specialization (recall **Figure 3**) a *Metric* can be associated to each property in the catalogue. This way, *Measurements* can be performed on the relevant entity, involved in the actual context, following the associated *Metric* specification to quantify each context property. Conversely to attribute quantification for evaluation purposes which allows only one measurement per attribute, more than one measurement is allowed for each context property whose *multiValueCriterion* value is either *AND* or *OR*. For instance, in the introduced motivating scenario, for the *actual context* description, the "*supported natural language*" roperty is relevant and more than one value, i.e. "*panish*" and "*English*", can be recorded simultaneously (see **Figure 7**)

If the *multiValueCriterion* is XOR then only one measurement is allowed for the context property; this would be the case of, for example, the "development *team size*" property.

When describing the *target context* of a contextual entity, since the target context describes the ideal context in which a given contextual entity is suitable—and since we sustain that the contextual entity can be applied successfully in more than one context—the multiple measurements assigned to each property represent the diffe- rent values that these different contexts may present for that property. For instance, the "*Lexile text*" metric[3] to quantify the help page *Readability of Text* attribute in the

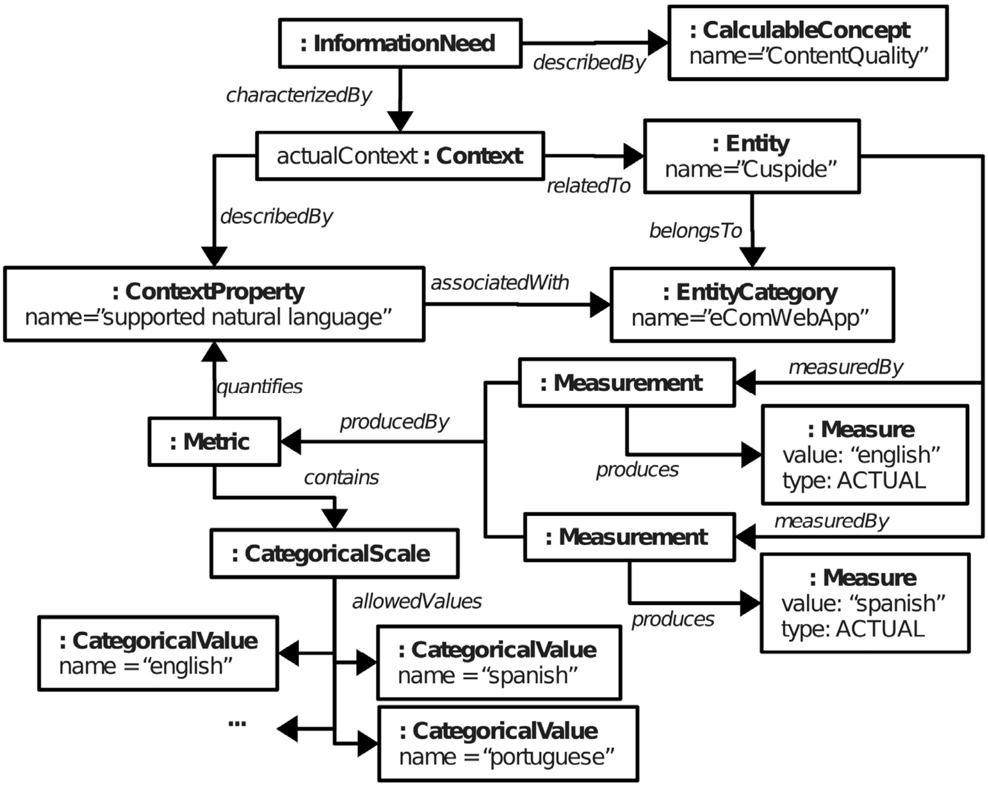


Figure 6. Example of an actual context specification.

[3] http://www.lexile.com/about-lexile/lexile-overview/

Cuspide's shopping cart is a contextual entity since its applicability depends on the natural language in which the text is written, being useful just for *English* OR *Spanish* languages. Note that our context model uses to a great extent the same INCAMI conceptual base for specification and quantification of properties (see **Figure 8**).

The two above scenarios are the base for the core functionalities of the context-aware recommender system which are detailed in the following sub-sections.

3.2. Context Based Recommendation System

So far the two scenarios of how context information is used in C-INCAMI were addressed, stressing its role in a context-aware M&E recommender system. Next we present the calculations designed to compare contexts descriptions to provide automated recommendations.

As commented before, contexts may be compared to determine whether results from two M&E projects can actually be compared or a contextual entity can be coherently applied in a given M&E project, e.g. when selecting a metric at measurement design time. The comparison mechanism is based on metrics designed for the two above scenarios: 1) the *Contextual Similarity* metric intend to compare two actual contexts from two separate M&E projects, and 2) the *Contextual Entity Applicability* metric, intended to compare the target context of a contextual entity against an actual context. In both cases, the metric values should be interpreted as the more similar/applicable when closer to 1. These metrics are discussed below.

3.2.1 Contextual Similarity Metric

This metric applies when comparing two actual contexts from two different projects. In this comparison, all the properties specified in both contexts are considered altogether regardless of whether or not they are present in both contexts at the same time. Before going into details, we present the notation we use:

By *AC* we denote an *actual context* defined by (*CP*,*CV*,*CW*) where:

- $CP = \{cp_1, \cdots, cp_n\}$ s the set of *context properties* specified for a given context; they represent only their meta-data (the properties definitions) as stored in the organization's catalogue;
- $CVcp_1 = \{CVcp_1, \cdots, CVcp_n\}$ are the whole sets of values describing some context, one set for each context property, being $CVcp_1 = \{CV_1, \cdots, CV_m\}$ the set of values for the context property cp$_i$;
- $CWcp_1 = \{CWcp_1, \cdots, CWcp_n\}$ is the set of relative weights for each context property cp$_i$; where each CWcp$_i$ $\in \Re$ and $\sum_{i=1}^{n} CWcp_i = 1$.

So, given the actual contexts *AC*(*a*) = (*CP*(*a*), *CV*(*a*), *CW*(*a*)), and *AC*(*b*) = (*CP*(*b*), *CV*(*b*), *CW*(*b*)) specified for projects *a* and *b* respectively, we define:

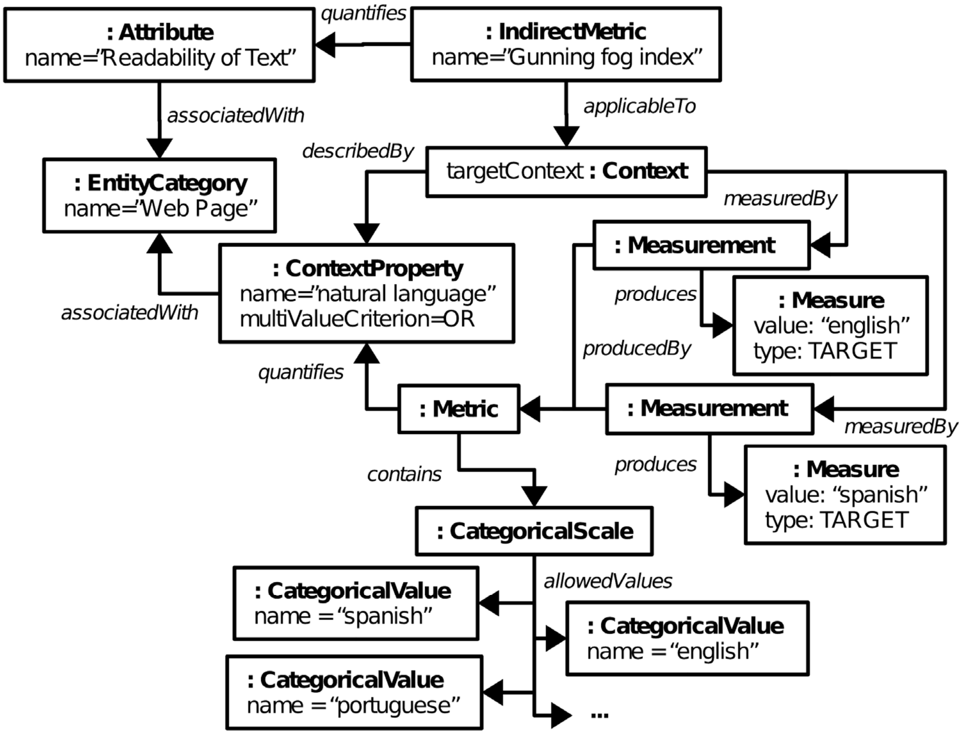


Figure 7. Example of a target context specification.

CP(*a*,*b*) = *CP*(*a*)∪*CP*(*b*) as the definitions of all the distinct context properties specified by both *AC*(*a*) and *AC*(*b*), and # *CP*(*a*,*b*) as the number of properties in *CP*(*a*,*b*)

Then, contextual similarity *CSim* is calculated as the weighted sum of the similarity between the equivalent properties of *AC*(*a*) and *AC*(*b*) for all the properties included in *CP*(*a*,*b*), normalized by #*CP*(*a*,*b*):

$$CSim\left(AC\left(a\right),AC\left(b\right)\right)=\sum_{i=1}^{\#CP(a,b)}\frac{CPW\left(cp_i,AC\left(a\right)\right)CPsim\left(cp_i,AC\left(a\right),AC\left(b\right)\right)}{\#CP\left(a,b\right)},\forall cp_i\in CP\left(a,b\right) \quad (1)$$

Where *CPSim* is a metric used to obtain the similarity between two equivalent properties cp_i from both contexts *AC*(*a*) and *AC*(*b*) (presented below), and *CPW* is a function that returns the weight of a given context property cp_k in the actual context of some entity *e*; if the property is not present, it returns 0. In Equation 1 the weight for each cp_i are taken from the actual context *AC*(a).

$$CPW\left(cp_k,AC\left(e\right)\right)=\begin{cases}CWcp_k\left(e\right) & \text{if } cp_k\in CP\left(e\right)\\ 0 & \text{otherwise}\end{cases}$$

3.2.2 Context Property Similarity (CPSim) Metric

Remember that, from the definition of *CP*(*a*,*b*), cp_i may appear in one or both contexts, so the following is held:

$$\forall cp_i\in CP\left(a,b\right),$$
$$\exists cp_j\in CP\left(a\right)\mid cp_i\equiv cp_j\vee\exists cp_k\in CP\left(b\right)\mid cp_i\equiv cp_k$$

As before, let $CVcp_k\left(\alpha\right)=\left\{CV_1,\cdots,CV_m\right\}$ be the values specified for context property cp_k in the context associated to the entity *a* and $CVcp_k\left(b\right)$ the values specified for the same context property in entity *b*, so we define:

- $CVcp_k(a,b)=CVcp_k(a)\cup CVcp_k(b)$ as all the distinct values specified for context property cp_k from both entities *a* and *b* altogether. Note that values in each property comes from the same property definition, as explained in Section 3.2;
- *#CVcp*$\#CVcp_k(a,b)$ as the number of values in $CVcp_k(a,b)$;
- $ECVcp_k(a,b)=CVcp_k(a)\cap CVcp_k(b)$ as the set of equal values specified for context property cp_k for both entities; and,
- $\#ECVcp_k\left(a,b\right)$ as the number of values in $ECVcp_k\left(a,b\right)$.

Then, the similarity between two equivalent context properties in the contexts of entities *a* and *b* is defined as:

$$CPSim\left(cp_k,AC\left(a\right),AC\left(b\right)\right)=\begin{cases}\dfrac{\#ECVcp_k\left(a,b\right)}{\#CVcp_k\left(a,b\right)} & \text{if } cp_k\in CP\left(a\right)\wedge cp_k\in CP\left(b\right)\\ 0 & \text{otherwise}\end{cases} \quad (2)$$

3.2.3. Contextual Entity Applicability Metric

The metric presented next can be used to determine whether a given contextual entity can be applied coherently to a given M&E project. The metric compares a contextual entity's target context with a project's actual context to determine the applicability degree of the contextual entity to the project. In this comparison, only the properties specified for the target context are taken as parameters, different from what occurs in the *CSim* metric (Equation (1)) where all the properties specified for both contexts where included. Now, the comparison is made on the basis of the target context instead of the actual context, which may specify properties not relevant for the contextual entity.

Therefore, given the actual context *AC*(*p*)=(*CP*(*p*), *CV*(*p*), *CW*(*p*)) specified for project *p* and the target context *TC*(*e*)=(*CP*(*e*), *CV*(*e*), *CW*(*p*)) specified for the contextual entity *e*, the contextual applicability *CApp* is calculated as the weighted sum of the applicability between the equivalent properties of *TC*(*e*) and *AC*(*p*) for all the properties included in *CP*(*e*), normalized by #*CP*(e):

$$CApp=\sum_{i=1}^{\#CP(e)}\frac{CPW\left(cp_i,TC\left(e\right)\right)CPApp\left(cp_i,TC\left(e\right),AC\left(p\right)\right)}{\#CP\left(e\right)},\forall cp_i\in CP\left(e\right) \quad (3)$$

Here, the weight returned by CPW for each cp_i are taken from the target context *TC*(e).

3.2.4. Context Property Applicability (CPApp) Metric

Unlike the *CPSim* metric (Equation 2), in order to perform the calculation, the *CPApp* metric takes into account the *multiValueCriterion* of the context property in the target context. When in a target context's property the OR criterion is specified, it should be checked that at least one value of the target context's property be present in its actual context's counterpart. If the AND criterion is specified, it should be checked that all values in the target context's property are present in its actual context's counterpart. So, *CPApp* is calculated as follows:

If $cp_k \in CP(a) \wedge cp_k \in CP(b)$ does not hold, then

$$CPApp(cp_k, TC(e), AC(p)) = 0$$

Otherwise, if the *multiValueCriterion* is OR or XOR then

$$CPApp(cp_k, TC(e), AC(p)) = \text{Equation (2)}$$

if the *multiValueCriterion* of cp_k is AND then

$$CPApp\left(cp_k, TC(e), AC(p)\right) = \begin{cases} 1 & \text{if } \#ECVcp_k(e,p) = \#CVcp_k(e,p) \\ 0 & \text{otherwise} \end{cases}$$

It is worth mentioning that for the above discussed metrics, the amount of context properties available in each compared context is an important issue. If relevant properties are omitted when specifying either contexts involved in the comparison, the final results for the degree of similarity/applicability will be undervalued. This may lead to unreliable results and consequently to wrong interpretations and decisions. That is why special attention in the selection of the proper relevant context properties in both actual and target contexts should be paid.

3.3 Proof of Concept

By the following proof of concept, which is based on a case study shown in [22], we try to illustrate how the context-aware approach in the C-INCAMI framework works. The starting point is to define non-functional requirements, and particularly the *Information Need* as shown in **Table 1**.

Next, the actual context characterizing the information need of this project should be specified. To this aim, the relevant entities related to the "*Cuspide*'*s shopping cart*" are identified by inspecting the organization's domain catalogues for related products, processes, projects, resources and stakeholders. As a result, we found that the only related relevant entity is the "*Cuspide e-Commerce WebApp*" product, which contains the entity to assess, because some of its properties may affect the way measurement and/or interpretation is performed on the external quality of the shopping cart.

The next step is to select from the catalogue the relevant properties associated with the above entity and to quantify them properly. **Table 2** presents the properties available for the actual context. Each context property contains all meta-data related to its definition (recall **Figure 3**) of which only name, definition and relevance are shown, and the assigned value describing the actual state of the relevant context for that property.

After the context specification is completed a concept model is selected and/or designed according to the particular evaluation needs. For this case an external quality model was taken from the M&E catalogue and modified to meet the project's needs as shown in **Figure 9**. Then the measurement design activity should be performed so a metric is selected and assigned for each attribute in the external quality model. In the example, to quantify the attribute *Readability of text* (coded 1.2.1.2 in **Figure 9**), two metrics are found in the catalogue (see **Table 3**) together with the context where they are applicable, i.e. their target contexts.

As discussed before, the applicability of both metrics depends on the natural language in which the content is written, as specified by the "*Supported Natural Language*" property in their target contexts. From these specifications we can see the "*Gunning fog index*" can only be used for English texts while the "*Lexile text*" metric can be used either for English or Spanish texts. Therefore, their applicability to quantify the "*Readability of text*"

External Quality
1 Usability
1.1 Understandability
1.1.1 Icon/label ease to be recognized
1.1.2 Information grouping cohesiveness
1.2 Learnability
1.2.1 Help and Support
1.2.1.1 Help page
1.2.1.2 Readability of text
1.2.2 Predictive information for link/icon ...
1.3 Operability
1.3.1 Control permanence
1.3.2 Expected behavior of controls
2 Information Quality
2.1 Content Suitability
…

Figure 9. Excerpt of the external quality model for the cuspide's shopping cart [19].

Table 1. The information need for the cuspide. com shopping cart case study.

Purpose	**Focus**	**Point of view**	**Entity Category**	***Entity***
Improve	*External Quality*	*Customer user*	*WebApp's shopping cart*	*Cuspide WebApp's shopping cart*

Table 2. The properties of the actual context characterizing the information need.

Name	Market Scope
Definition	The scope of countries to which the marketing of the e-Commerce Web Site's products are constraint.
Relevance	May affect outcomes interpretation regarding Content Suitability, particularly Coverage. Related to the Website owner's country property
Scale/Unit	[domestic, regional, global] (nominal categorical scale)
Value	global
Name	**Website owner's country**
Definition	The country in which the Web site's owner company resides.
Relevance	May affect outcomes interpretation regarding Content Suitability, particularly Coverage. Related to Market scope property
Scale/Unit	An enumeration of the countries' names (nominal categorical scale).
Value	Argentina
Name	**Supported natural language**
Definition	Natural languages in which the content of the Web site is written.
Relevance	May help in measurement design since the Web site's text is part of the content whose quality will be measured and evaluated
Scale/Unit	An enumeration of the natural languages' names (nominal categorical scale).
Value	Spanish
Name	**Software release life-cycle stage**
Definition	Stage that describes the stability of a released piece of software as the development process proceeds.
Relevance	May affect outcomes interpretation since different quality levels may be observed at different stages.
Scale/Unit	[Pre-alpha, Alpha, Beta, Release Candidate, Release to Marketing, General Availability, Live Release] (ordinal categorical scale).
Value	Live Release
Name	**Content provider role type**
Definition	Role or User type that provides the main content of the Web site.
Relevance	May affect outcomes interpretation since different quality levels may be observed on content provided by different types of providers.
Scale/Unit	[owner, registered user] (nominal categorical scale).
Value	owner

Table 3. Metrics to quantify the Readability of Text attribute and their target contexts.

Metric Name	Lexile Text	Numerical Scale	
Objective	To obtain a numeric representation of a text's readability.	**Value type**	INTEGER
Value Interpretation	Number representing an individual's reading ability or a text's readability. The lower the value, the easier to read.	**Scale type**	ABSOLUTE
References	http://nces.ed.gov/pubs2001/200108.pdf	**Representation**	DISCRETE
Author	MetaMetrics, Inc.	**Unit**	Lexile

Target Context

Property name	Entity	Value	Logic Operator
Supported Natural Language	Cuspide.com WebApp	Spanish, English	OR

Metric Name	Gunning Fog Index	Numerical Scale	
Objective	To obtain the understandability of a piece of text.	**Value type**	INTEGER
Value Interpretation	Number of years of U.S. formal education that a person requires to easily understand the text on the first reading. The lower the value, the easier to understand.	**Scale type**	ORDINAL
References	http://www.idph.state.ia.us/health_literacy/common/pdf/tools/gunning.pdf	**Representation**	DISCRETE
Author	Robert Gunning	**Unit**	Years

Target Context

Property name	Entity	Value	Logic Operator
Supported Natural Language	Cuspide.com WebApp	English	OR

attribute in the situation of the M&E project can be decided by comparing their corresponding target contexts against the actual context characterizing the M&E project's information need.

This comparison can be achieved automatically by computing the applicability degree of each metric's target context to the actual context, using the *CApp* metric (see Equation (3)). Then, evaluators will be able to select the more applicable one, by observing the applicability value of each metric. These calculations are illustrated next.

We first calculate the *CApp* metric on the "*Lexile text"* metric. We begin by calculating *the CPApp* metric for its only context property: "*Supported natural language*" So let *p* be the M&E project on which the "*Cuspide's shopping cart*" is assessed and *lexile* be the contextual entity;

then:

- $CV_{\text{Supported natural language}}\left(lexile\right) = \{'english','spanish'\}$
- $CV_{\text{Supported natural language}}\left(p\right) = \{'spanish'\}$
- $CV_{\text{Supported natural language}}\left(lexile, p\right) = \{'english','spanish'\}$
- $\#CV_{\text{Supported natural language}}\left(lexile, p\right) = 2$
- $ECV_{\text{Supported natural language}}\left(lexile, p\right) = \{'spanish'\}$
- $\#ECV_{\text{Supported natural language}}\left(lexile, p\right) = 1$

So, according to the specification of the *CPApp* metric and applying Equation (2), results:

$$CPApp\left(Supportednaturallanguage, TC\left(lexile\right), AC\left(p\right)\right) = \frac{\#ECV_{Supportednaturallanguage}\left(lexile, p\right)}{\#CV_{Supportednaturallanguage}\left(lexile, p\right)} = 0.5$$

And, since "*Supported natural language*" is the only property specified in the context of the *lexile* metric, by calculating Equation (3) we obtain: *CApp*(*lexile*, *p*) = 0.5. Besides, by performing the above calculi on the *Gunning fog index* metric it results that: *CApp*(*gunning*,*p*) = 0. As we can see from these outcomes, the "*Lexile text*" metric got the highest degree of applicability, so it is chosen to quantify the "*Readability of text*" attribute.

Note that the above calculations could appear a bit complex to make just a simple comparison. Nevertheless if more context properties would be specified in the target context of the contextual entity, it would require only to calculate the CPApp metric for each additional context property and then compute *CApp* as defined in Equation. 3. When a large amount of attributes and their potential metrics are involved, the advantage of using this procedure is paramount, even more if the procedure can be automated in a recommender system.

4. Discussion and Related Work

In this article we argue that being able to specify in an explicit and formal way the relevant context of M&E projects, is of paramount importance. Thus we have applied the context-aware approach to a M&E framework enabling the usage of context information to recommend, at design time, the better contextual entity in a given project or to perform, at later stages, more robust comparative analysis among projects. Disregarding this information may lead to less reliable analysis and recommendations [25].

Regarding the applied context-aware approach, it is worth mentioning that although first designed for ubiquitous computing, it can be successfully applied to any domain if a thorough analysis of the relevant entities types and their properties is performed. To this aim, a model should be designed to represent both context properties data (values) and metadata, defining in an unambiguous way their meaning and representation rules (e.g. measurement methods, scales and units). The chosen model and representation should match the intended purpose for the context information taking into account the trade-off between acceptable complexity and needed expressiveness.

Several works can be found in the literature applying a context-aware approach. For instance, in [18] a process to identify context descriptions from software engineering research papers and a method to compare them, using a hierarchical clustering algorithm, are presented. The comparison outcome is then used as an affinity parameter to compare the results presented in the papers. In that work, context information is recorded in a single-entry preset template with no other definition than the entry name making it more difficult to use it in intepretation activities. In [19], a context-aware approach to represent and integrate static enterprise information and dynamic process information in knowledge management is presented. Conversely to the previous work, this approach includes a generic context knowledge structure model based on a set of meta-information elements at different conceptual levels for context description that can be interconnected with ontologies in the business environment. The approach proposes to use the foundations of the Semantic Web to store and interconnect context descriptions and enterprise knowledge although nothing is said about how to use or process context information. In [14] the context-aware approach is applied in a web engineering process for the adaptation of applications to deliver context-relevant services and/or information to end users on a discovery basis. The proposal includes a conceptual model based on context parameters that are connected with a certain weight, i.e. a degree of relevance, to elements of an existing domain ontology. A preset categorization of context parameters (User&Role, Process&Task, Location, Time, Device) to satisfy a broad variety of web application scenarios is provided, each one requiring specific values relevant to the application domain. Authors argue that such a categorization allows for a large degree of reuse when using context catalogues. Nevertheless this approach may not be the most appropriated for domains other than ubiquitous and pervasive computing (such as software engineering as in our work). For the representational aspect authors chose OWL (*Ontology Web Language*). The proposal also describes a scenario of use of how context is used in the engineering process of developing an application. Authors in [26] present a version model that supports context-dependent alternatives and a matching algorithm for context-aware queries to provide context-aware data management. These elements are implemented in an object-oriented database system with which a mobile tourist information system was implemented. The model, that supports typing of context properties, was built using an object-oriented data model and the implementation is based on a generic context engine both of their own. The proposed model allows linking any context property to any application domain concept. A common issue of these works, and others, is the lack of a sound conceptual base when it comes to context properties quantification, i.e. few or no metadata is specified about the scales (and scales types) and units in which values are expressed nor the methods used to obtain them. Consequently, interpretations and comparisons may lead to invalid conclusions.

In this work, having analyzed previously different representation approaches [16] based on a set of requirements particular to our purpose, we designed a context information model using a measurement and evaluation framework as a foundation. This way, capabilities related to properties quantification are reused resulting in a clearer and simpler definition of the concepts related to context specification. The designed context model can be implemented as a separate module to ease its integration to existing systems. Also, its representation in RDF allows connecting its elements with existing application domains and to implement context catalogues to enable a more consistent reuse of context properties definitions. Also a mechanism to compare context descriptions was designed, although keeping a simple matching approach. The resulting context-aware approach has the following characteristics:

- *Simple*: a context property is defined as a specialization of an attribute which is already defined along with measurement capabilities in a well structured framework;
- *Flexible*: the context model can be plugged to any application domain and reuse its entities and attributes definitions to specify context descriptions;
- *Semantically defined and able to be validated*: the semantic and structure of each context property is explicitly defined in the organization's data base schemata from which the corresponding instances were gathered;

Regarding the application of the context-aware approach to the M&E, the need of being aware of the relevant context in this domain has been previously expressed in several works although none of them have provided a solution to explicitly specify and use context information as we propose. For example, the ISO-15939 [5] measurement process states that “Characteristics of the organizational unit that are relevant to selecting measures and interpreting the information products shall be explicitly described” providing “the context for measurement” in order to make clear “the assumptions that it embodies and constraints that it imposes”. In addition it recommends that context should be stored when collecting data to verify it, understand it, and evaluate it. However, no formal specification is represented to deal with such a characterization. In [6], authors remark the importance of having explicit representations of context to be effective in interpreting and acting on measurement results. In this sense they argue that “every process operates in an environment that contributes to or detracts from its prospects for success”, and enumerate a number of factors which, if quantified, may help to interpret data gathered from the entities under examination. Guidelines are provided to specify context information and to determine its boundaries in terms of the entities (and their relevant attributes) affecting the entity being measured and analyzed. Examples of environment descriptions are given although using a simple mechanism, i.e. a template form. In [4] authors present a GQM-based (Goal Question Metric) process model for measurement programs, which starts by characterizing "the current project and its environment with respect to existing models and metrics". One of the main principles in this work is that measurement goals should be explicit, including the environment of the analysis involving “the organization, the project, process model used”, among others. Also a “process prestudy” is specified in which existing measurement documentation from “programs performed in the same or a comparable context” is used, yet no specification of how to perform this comparison is presented. Other authors as Kitchenham *et al.* [10], and Briand *et al.* [3] advocate for the explicitness and use of context information but a structured proposal to achieve such a goal is missing.

Ultimately, the application of the context-aware approach to the INCAMI framework, as presented in this article represents a contribution in the sense that a clear and structured support to the specification of context information for M&E projects in software organizations has often been neglected.

5. Conclusion and Future Work

In this article we have discussed the application of the context-aware approach for recommendation capabilities in the INCAMI M&E framework result in the C-INCAMI framework. We have noted, however, that the designed context information representation can be used in any domain so in this sense it can be considered a domain-independent approach.

In addition, we have shown, as an important contribution, that the application of the context-aware approach to the M&E domain clearly benefits two scenarios: 1) by assisting in the recommendation of design solutions, and 2) by making the interpretation of outcomes among different evaluation projects more consistent and comparable. Particularly, we have illustrated as proof of concept the case where the more suitable metric (the contextual entity) was recommended at measurement design time for the Cuspide's shopping cart evaluation project (the actual context).

In both cases the foundation for the M&E recommender system is the proposed context specifications. As another contribution we have provided the rationale of how to quantify the context properties -reusing the concept of *Attribute* and *Metric* from the INCAMI framework- as well as the specification of the contextual similarity and applicability metrics that constitute the core of the recommender system.

As result, C-INCAMI currently offers not only a simple, flexible, goal-oriented organization approach but also a consistent, semantically defined and able to be validated mechanism for representing context information, therefore, giving support in the design and selection of contextual entities as well as in the consistent interpretation and analysis of results coming from similar M&E projects.

Future work will focus on improving the matching approach to context comparison. Lastly, an in-progress line of research is the implementation of an ontology-based organizational memory application and its integration with the contextual C-INCAMI framework and tool. That is, the knowledge acquired and stored in a case knowledge base from lessons learned in M&E projects may serve as the basis for a recommendation system to support more robust decision making.

6. Acknowledgements

This work and line of research is supported by the PAE-PICT 2188 project at UNLPam, from the Science and Technology Agency, Argentina.

REFERENCES

[1] M. Baldauf, S. Dustdar and F. Rosenberg, "A Survey on Context-Aware Systems," *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 2, No. 4, 2007, pp. 263-277. doi:10.1504/IJAHUC.2007.014070

[2] L. Olsina, H. Molina and F. Papa, "How to Measure and Evaluate Web Applications in a Consistent Way," *Web Engineering: Modelling and Implementing Web Applicati*ons, G. Rossi, O. Pastor, D. Schwabe and L. Olsina, Editors, Springer, London, 2008, pp. 385-420. doi: 10.1007/978-1-84628-923-1_13.

[3] L. C. Briand, S. Morasca and V. R. Basili, "An Operational Process for Goal-Driven Definition of Measures," *IEEE Transactions on So*ftware Engineering, Vol. 28, No. 12, 2002, pp. 1106-1125. doi:10.1109/TSE.2002.1158285

[4] C. Gresse, B. Hoisl and J. Wüst, "A Process Model for QGM-Based Measurement," Gresse *et al.*, 1995.

[5] ISO/IEC: ISO/IEC 15939: 2002—Software Engineering—Software Measurement Process, 2002.

[6] R. E. Park, W. B. Goethert and W. A. Florac, "Goal-Driven Software Measurement: A Guidebook," Software Engineering Institute, Carnegie Mellon University Pittsburgh, Pittsburgh, 15213, 1996.

[7] A. Abran, A. Sellami and W. Suryn, "Metrology, Measurement and Metrics," *9th IEEE International Software Metrics Symposium* (*METRICS 2003*), Sydney, September 2003, 2003, pp. 2-11.

[8] CMMI Product Team, "CMMI for Development, Version 1.2," Software Engineering Institute, Carnegie Mellon University Pittsburgh, Pittsburgh, 15213, 2006.

[9] ISO/IEC: ISO/IEC 14598-1: 1999—Information Technology—Software Product Evaluation—Part 1: General Overview, 1999.

[10] B. A. Kitchenham, R. T. Hughes and S. G. Linkman, "Modeling Software Measurement Data," *IEEE Transactions on Software Engineering*, Vol. 27, No. 9, 2001, pp. 788-804. doi:10.1109/32.950316

[11] H. Molina and L. Olsina, "Assessing Web Applications Consistently: A Context Information Approach Web Engineering," *2008 ICWE'08. Eighth International Conference on Web Engineering*, 2008, pp. 224-230. doi: 10.1109/ICWE.2008.21

[12] A. K. Dey, "Understanding and Using Context," *Personal and Ubiquitous Computing*, Vol. 5, No. 1, 2001, pp. 4-7. doi:10.1007/s007790170019

[13] L. Gong, "Contextual Modeling and Applications," *IEEE International Conference on Systems, Man and Cyberne*tics, 2005, pp. 381-386. doi:10.1109/ICSMC.2005. 1571176

[14] J. W. Kaltz, J. Ziegler and S. Lohmann, "Context-Aware Web Engineering: Modeling and Applications," *RIA Revue d'Intelligence Artificielle, Special Issue on Applying Context Management*, Vol. 19, No. 3, 2005, pp. 439-458.

[15] T. Strang, C. Linnhoff-Popien and K. Frank, "COOL: A Context Ontology Language to enable Contextual Interoperability," *Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems* (*DAIS2003*), 2003, pp. 236-247.

[16] T. Strang and C. Linnhoff-Popien, "A Context Modeling Survey," *Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004—6th In-*

ternational Conference on Ubiquitous Computing, Nottingham, 2004, pp. 34-41.

[17] P. Dourish, "What We Talk about When We Talk about Context," *Personal and Ubiquitous Computing*, Vol. 8, No. 1, 2004, pp. 19-30. doi:10.1007/s00779-003-0253-8

[18] D. Cruzes, M. G. Mendonça, V. Basili, F. Shull and M. Jino, "Using Context Distance Measurement to Analyze Results across Studies," *ESEM '07: Proceedings of the First International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society*, Washington DC, 2007, pp. 235-244.

[19] W. Huang and T. Tao, "Adding Context-Awareness to Knowledge Management in Modern Enterprises," *Second IEEE International Conference On Intelligent Systems*, 2004, pp. 393-398

[20] "Resource Description Framework Standards," The World Wide Web Consortium, http://www.w3.org/ standards/ techs/ rdf#stds

[21] ISO/IEC FCD 25010.3 Software engineering—Software product Quality Requirements and Evaluation (SQuaRE) —Quality Model and Guide, 2009.

[22] L. Olsina, G. Rossi, A. Garrido, D. Distante and G. Canfora, "Web Applications Refactoring and Evaluation: A Quality-Oriented Improvement Approach," *Journal of Web Engineering*, Vol. 7, No. 4, 2008, pp. 258-280.

[23] M. Martin and L. Olsina, "Towards an Ontology for Software Metrics and Indicators as the Foundation for a Cataloging Web System," *Proceedings of IEEE Computer Society* (*1st Latin American Web Congress*), Santiago de Chile, 2003, pp. 103-113.

[24] H. Molina, F. Papa, M. Martin and L. Olsina, "Semantic Capabilities for the Metrics and Indicators Cataloging Web System," *Engineering Advanced Web Applications*, Rinton Press Inc., 2004, pp. 97-109.

[25] B. Kitchenham, D. R. Jeffery and C. Connaughton, "Misleading Metrics and Unsound Analyses," *IEEE Software*, Vol. 24, No. 2, 2007, pp. 73-78. doi:10.1109/ MS.2007.49

[26] M. Grossniklaus, M. C Norrie, "An Object-Oriented Version Model for Context-Aware Data Management," *Proceedings of WISE 2007, 8th International Conference on Web Information Systems Engineering*, Nancy, France, December 2007, pp. 398-409.