

Process Ontology Specification for Enhancing the Process Compliance of a Measurement and Evaluation Strategy

Pablo Becker, Fernanda Papa and Luis Olsina

UNLPam, Facultad de Ingeniería, GIDIS_Web
General Pico, La Pampa, Argentina, 6360
{beckerp,pmfer,olsinal}@ing.unlpam.edu.ar

Abstract

In this paper, we specify a generic ontology for the process domain considering the related state-of-the-art research literature. As a result, the recently built process ontology contributes to enrich semantically the terms for the (previously developed) measurement and evaluation domain ontology by means of stereotypes. One of the underlying hypothesis in this research is that the generic ontology for process can be seen as a reusable artifact which can be used to enrich semantically not only the measurement and evaluation domain ontology but also to other domains involved in different organizational endeavors. For instance, for the measurement domain, now is explicit that the measurement term has the semantic of task, the measure term has the meaning of outcome, and the metric term has the semantic of method, from the process terminological base standpoint. The augmented conceptual framework, i.e. measurement and evaluation concepts plus process concepts, has also a positive impact on the GOCAME (*Goal-Oriented Context-Aware Measurement and Evaluation*) strategy capabilities since ensures terminological uniformity, consistency and verifiability to its process and method specifications. In order to illustrate how the augmented conceptual framework impacts on the verifiability of GOCAME process and method specifications in addition to the consistency and comparability of results in measurement and evaluation projects, an ICT (*Information and Communications Technology*) security and risk evaluation case study is used.

Keywords. Process Ontology, Generic Ontology, Measurement and Evaluation Ontology, Domain Ontology, GOCAME Strategy.

1 Introduction

From long time ago, public and private organizations produce goods and services by means of the management of work. Work can be structured as tasks, activities or processes at different abstraction levels. To perform work, the tasks should have assigned resources such as methods, tools and people, among others. An engineered way to organize the work is by performing project management. Project management is the discipline which deals with the organizing, planning, scheduling, implementing and controlling activities and resources in order to accomplish specific goals in industrial, scientific or daily problems.

Therefore, project is the entity category and basic construct to manage the work in all type of organizations. Just to quote a few definitions, the project term in the CMMI [1] glossary states “*a managed set of interrelated activities and resources, including people, that delivers one or more products or services to a customer or end user*”. Also, a note of this term indicates “*a project has an intended beginning (i.e., project startup) and end. Projects typically operate according to a plan. Such a plan is frequently documented and specifies what is to be delivered or implemented, the resources and funds to be used, the work to be done, and a schedule for doing the work*”. Another very often quoted definition of project is in PMBOK [2] that says “*a temporary endeavor undertaken to create a unique product, service, or result*”.

Considering the above definitions, we can restate the project term as “*temporary and goal-oriented endeavor with intended start and finish dates, which consider a managed set of interrelated activities, tasks and resources*”.

aimed at producing and modifying unique work products (i.e. artifacts, services, or results) for satisfying a given requester need”.

Regarding the organizational structure, projects can be conceived to achieve objectives at different organizational levels such as projects at operative, tactic and strategic levels with possible alignment of goals among them. Additionally, for instance, at operative level, specific projects can be targeted to manage the work for development, maintenance and quality assurance processes, considering different production lines across particular domains. On the other hand, with the aim to achieve a given project goal for a particular problem, specific strategies should be chosen. For example, in order to produce the project plan for a given software development project, an agile strategy or a traditional strategy should be previously selected depending on the situation and need.

Strategy also has many definitions, but commonly involves setting goals, establishing actions (activities) to achieve the goals, and mobilizing resources to execute the actions. Strategy always includes processes of formulation (design) and implementation. Basili *et al.* define strategies [3] as: “*a set of possible approaches for achieving a goal that may be refined by a set of concrete activities*”.

Considering the existing definitions, we define the strategy term as “*principles, patterns, and particular domain concepts and framework that may be specified by a set of concrete processes, in addition to a set of appropriate methods and tools as core resources for achieving a project goal*”.

We have argued [4] that aimed at formulating and implementing measurement and evaluation (M&E) projects and programs -as part of resource, process, product and service quality assurance- in a systematic and disciplined way, software organizations need to establish explicitly a set of activities and methods to specify, collect, store, and use measures and indicator values. Furthermore, in order to achieve more effective analysis, recommendation and decision-making processes, the design of metrics (for measurement) and indicators (for evaluation) and the consistent usage of their values that can be repeatable and comparable among the organization’s projects should be taken into account [5]. Consequently, software organizations have to have well-established and integrated strategies [6, 7, 8] for pursuing M&E projects and programs with these features.

Looking at our above strategy definition, we set as one principle for a strategy that be integrated. That is, an integrated strategy should support simultaneously at least the following three capabilities: i) a *domain conceptual base and framework*; ii) *specifications of process views*; and, iii) *specifications of methods*. Note that for a particular domain such as M&E, the conceptual base, and process and method specifications are in fact devoted only to the measurement and evaluation areas.

Considering the above three capabilities, we have built the GOCAME (*Goal-Oriented Context-Aware Measurement and Evaluation*) strategy for the M&E domain. GOCAME’s first capability is the C-INCAMI (*Contextual-Information Need, Concept model, Attribute, Metric and Indicator*) conceptual base and framework [4], which explicitly and formally specifies the M&E concepts, properties, relationships and constraints (axioms), in addition to their grouping into components. Particularly, we developed the domain ontology for M&E [9] in 2003. GOCAME’s second capability -the specification of M&E process views- is aimed at assuring repeatability in performing activities and consistency of results. Process specifications usually describe a set of activities, tasks, inputs and outputs, interdependencies, artifacts, roles, and so forth. Besides, process specifications can consider different process views [10] such as functional, behavioral, informational and organizational. In our opinion, process specifications should primarily state what to do rather than indicate the particular methods and tools used by specific activity descriptions. The third GOCAME capability allows assigning systematically particular ways (i.e. well-defined methods) to perform ultimately the M&E tasks.

Looking at the first capability, i.e. the C-INCAMI conceptual base and framework, we have observed a potential opportunity for improvement with regard to its above quoted M&E ontology and components. Particularly, we have analyzed that many M&E domain specific terms are related to many process generic terms. But these process terms were not explicitly modeled and linked to the C-INCAMI M&E concepts. For example, a measurement and evaluation -from the specific M&E domain standpoint- have both the semantic of task -from the process generic domain viewpoint-, likewise the metric and indicator terms have both the semantic of method, and the measure and indicator value terms have the meaning of outcome, among others. Thus, the C-INCAMI conceptual framework can be semantically enriched with process terms, particularly, the measurement and evaluation components. This concern, in the related software literature has very often been neglected as well, as we discuss in Section 2.

In recent works [11, 12], we have thoroughly specified the generic ontology for process, i.e. the definition of its key terms and their main attributes and relationships. In this paper, we extend [12] by analyzing and illustrating the main contributions of this research summarized in the following list:

- i) The development of a process generic ontology can be seen as a reusable artifact for enriching semantically a domain ontology. Moreover, we argue that the built generic ontology for process can be used to enrich semantically not only the M&E domain but also other particular domains such as software development, software maintenance, etc.

- ii) The use of stereotypes as a particular mechanism for the semantic linking between the generic ontology and domain ontologies. So we analyze why the use of stereotypes is, at least in the M&E domain, a more suitable mechanism than using inheritance relationships.
- iii) The M&E concepts enriched with the process concepts help to build better specifications for the GOCAME strategy facilitating therefore their verifiability. In order to illustrate how the augmented conceptual framework impacts on the verifiability of process and method specifications, excerpts of a previously developed ICT security and risk evaluation case study [13] are employed.
- iv) As a consequence of the semantic enhancement of the GOCAME conceptual base and its impact on method specifications, different enacted M&E projects can support more consistent, comparable and robust analysis of data and metadata. Examples of metric metadata and datasets from [13] are used to show this issue.

Lastly, this work contributes to enhance the process compliance, since the process and method specifications which are particular to the M&E domain adhere to the process generic ontology, hence, fostering their understandability, communicability and consistency.

The remainder of this paper is organized as follows. Section 2 analyzes related work in the light of process conceptual bases, particularly focusing on process ontologies and M&E ontologies, and describes our motivation. Then, Section 3 presents the developed generic ontology for process, and stresses the added value of ontologies as a richer mechanism than glossaries and taxonomies to structure conceptual bases. Section 4 shows the enhancement made to the GOCAME's conceptual framework, and also illustrates its practical impact on the process and method specification capabilities by using excerpts of an ICT security and risk evaluation case study. Finally, concluding remarks and future work are outlined.

2 Motivation and Related Work

In the previous Section, we introduced the terms project and strategy. As the reader can surmise, the process (activity, task), resource (method, tool, agent) and work product (result, artifact) terms are used also in different domains. Even more, they are used in specific domains such as measurement, development and maintenance, to mention just a few. In this sense, many measurement concepts are related to process concepts: e.g., a measurement is a task, a measure is an outcome or result, and a metric is a method. Heijst *et al.* [14] distinguish different types of ontologies regarding the subject of the conceptualization, e.g., *domain ontologies*, which express conceptualizations that are intended for particular domains; and *generic ontologies*, which include concepts that are considered to be generic across many domains. Generic ontologies can be used to enrich domain ontologies [15]. Therefore we envision that our previously developed M&E domain ontology [9] can be enriched semantically using a process generic ontology. One key motivation (and hypothesis) in this research is that the generic ontology for process can be seen as a reusable artifact which might be used to enrich semantically not only the M&E domain ontology but also other domains involved in different organizational endeavors. Next, we concentrate reviewing related work on process and M&E conceptual bases and their semantic linking.

Looking at the related work for process conceptual bases, we observe to some extent a mismatch in the used terminology and in the semantic of concepts and relationships. It seems that so far there is no broad and unanimous consensus on all key terms and their meaning for process. Among pioneers in defining process terminologies are for instance Feiler *et al.* [16], Conradi *et al.* [17] and Lonchamp [18]. In these works a set of basic terms are defined. In [16] authors recognize that many terms will suffer an evolutionary change and other terms will be added in the future. In [17], the activity term is defined but not in [16], while the task term is not taken into account in [17]. However in [18], a clear distinction between task and activity is proposed, i.e. an activity is planned, while a task is scheduled -with human, technological and monetary resources- and enacted.

Another often quoted work in the process area is the ISO 12207 standard [19]. Like in the abovementioned works, it has just a glossary of terms. However, it depicts a diagram where the relationships between process, activity and task terms are represented. According to this diagram, a process can group other processes, and can also contain at least one activity; in turn, an activity can group one or more related tasks. We adopted to a great extent this process/activity/task hierarchy for our process ontology, as we discuss in Section 3. Additionally, there are works where authors identify relationships among process terms, as in Acuña *et al.* [20] and Esteban *et al.* [21]. In [21] authors identify concepts and their relationships, as well as attributes for each term. Moreover, they include the method term not used in other related works. This term is linked to the process domain since specifies 'how' to implement or carry out the description of an activity.

In an effort to standardize terms for the process domain, OMG (*Object Management Group*) developed SPEM (*Software & Systems Process Engineering Meta-Model Specification*), a meta-model of process engineering as well as a conceptual framework, which provides the concepts necessary to model, document and manage processes [22].

SPEM focuses on defining a generic framework for process modeling. Note that in SPEM a process is defined as a special type of activity, so that the hierarchy between activity and process differs from that in ISO 12207, where a process groups activities, and an activity groups tasks.

Another related work is *OPEN Process Framework* (OPF) [23], an extensible repository with predefined process components. OPF metamodel uses the producer term, while the more accepted term in the process literature is agent. The producer definition indicates that “*produces (e.g., creates, evaluates, iterates, or maintains) versions of one or more related work products and provides one or more related **services***” (bold added to the quote). However, the service term is missing in the OPF metamodel -while is defined in its glossary-, likewise the resource and pre- and post-condition terms. On the other hand, for the tool term, OPF indicates that it “*is a **software application** that is used by one or more persons playing one or more roles to create, modify, evaluate, or manage versions of work products*” (bold added). However, a tool can be not only a software tool, but also a physical instrument, or a combination of both. Other weak aspect is that OPF represents relationships between terms, but does not include their definitions.

Finally, another worth mentioning work is [24] in which Bringuente *et al.* define a Software Process Ontology (SPO) that contains concepts such as organization and project including also the concepts of project planning and scheduling. It is based on UFO (*Unified Foundational Ontology*) [25], which provides robustness to SPO -as indicated by authors. However, we observe some semantic inconsistencies as for example: in the SPO version documented in Guizzardi *et al.* [25], authors show that hardware resource, software resource and human resource inherit from resource, while in [24] a human resource is not a resource. This happened since a resource represented in SPO is an object in UFO, and given that a human resource cannot be an object from the semantic standpoint, then they decided to remove such a link. On the other hand, SPO uses terminology which to some extent differs from recognized standards in the process area such as SPEM [22], CMMI [1] and ISO 12207 [19]. For example, instead of using the work product term authors use artifact, not making distinction with outcome and service terms. Also they do not use the task term but rather the atomic activity term.

Regarding the M&E particular domain, there are several proposals of conceptual bases as for example García *et al.* [26], Goethert *et al.* [27] and Kitchenham *et al.* [5]. However, our conceptual base and framework (C-INCAMI) has an ontological root [9] unlike [27] and [5]. This M&E ontology had considered sources as ISO standards, recognized articles and books, following also the terminology of the WebQEM methodology developed in 1999 [28]. Our M&E ontology has similarities to the one presented in [29] and then slightly refined by some co-authors in [26] (called SMO -*Software Measurement Ontology*), but in [4] we have modeled some terms (e.g., elementary indicator, derived indicator, etc.) and relationships (e.g., between measurement and measure, metric and indicator, among others) that differ semantically with those proposed in [26]. Also we have added new concepts, such as context, context properties, etc., not considered in the quoted works. On the other hand, García *et al.* define a M&E approach called FMESP (*Framework for the Modeling and Evaluation of Software Processes*) [30], which is supported by the SMO and a descriptive software process modeling ontology. Authors say that this process ontology is based on the SPEM specification and has been defined to clarify the domain for descriptive modeling of software processes. However, their process ontology has not been issued in an accessible and public way. Moreover, SMO is not semantically related or linked to the process ontology at all.

On the other hand, some authors of SPO [24] developed a software quality ontology [31] as well. This quality ontology is divided into three sub-ontologies, namely: quality models, measurement, and evaluation. They recognize that process is an important part of the universe of discourse for the quality ontology and, therefore, they developed the quality ontology integrated with SPO [32]. The semantic integration is accomplished using inheritance relationships. For example, the measurement term from the quality ontology is linked with the activity term from SPO and the measurement procedure is a procedure from the process standpoint. In turn, measurement is related to the measured value term by means of the produce relationship. However, the measured value term is not enriched with any process term as outcome, as we did (see Section 3). Moreover, the measurement term is no related with other term by means of the consumes relationship from the process standpoint. On the other hand, regarding the measurement sub-ontology [31], we observe an ambiguity in the usage of the measure term, since sometimes it refers to the value produced by a measurement, while other times to the instrument (procedure) for obtaining such a value. This semantic duality of the measure term is also observed in CMMI [1], ISO 15939 [33], and ISO 25000 [34]. Instead, we make a clear distinction between measure and metric terms linking them also to our process ontology -as we discuss in sub-section 4.2. On the other hand, Barcellos *et al.* [31] use the measurable element term to refer to measurable properties of an entity. However, the widely adopted term in the M&E literature [33, 34, 35] is attribute or property. Also, context terms are not included in the software quality ontology, as we did in [36].

In summary, the above abridged review about terminological bases in the process literature exposes that so far there was no broad and unanimous consensus on all key terms and their meaning. Besides, we have observed that many terms for the M&E particular domain are not explicitly or appropriately linked to process terms and concepts. So we propose a generic ontology for process aimed at enriching our previously developed M&E domain ontology

yielding as one result, a new enhanced version of the M&E domain ontology. A basic well-known definition for ontology is “*an explicit specification of a conceptualization*” [37] in which later authors considered also important to add that the specification should be formalized and shared [15]. Taking into account these issues, we have re-built the process conceptual base documented in [21] -which had been based on seminal works such as Feiler and Humphrey [16], Lonchamp [18], amongst others- considering now more recent and authoritative contributions in the process area such as SPEM, CMMI and ISO 12207, yielding as other result a process generic ontology (Section 3) with the most commonly used terms by the process community. This process ontology is intended to be concise but complete, as well as reusable across different particular domains.

3 Process Generic Ontology

There exist different ways to structure a terminological base such as, among others, glossaries, taxonomies and ontologies approaches. In a nutshell, a glossary is an ordered set of terms and their definitions; a taxonomy is a collection of terms -like a glossary-, but usually organized by kind-of semantic relations regarding the hierarchic structuring. Finally, an ontology includes terms, their definitions, properties and different types of relationships among terms, in addition to axiomatic restrictions. Just to write two often cited quotes: “*A body of formally represented knowledge is based on a conceptualization ... An ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an ontology is a systematic account of Existence. For knowledge-based systems, what ‘exists’ is exactly that which can be represented.*” ([37], p. 199). And, “*an ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms. An ontology is virtually always the manifestation of a shared understanding of a domain that is agreed between a number of agents. Such agreement facilitates accurate and effective communication of meaning, which in turn leads to other benefits such as inter-operability, reuse, and sharing*” ([38], p. 8). Therefore, we can state that an ontology is a richer mechanism than other approaches for structuring a conceptual base. Furthermore, an ontology can be specified by means of ontological languages such as OWL (*Ontology Web Language*), among others, which give support to machine-processable semantic inferences.

Ultimately, the main goals to build ontologies can be manifold, such as: to share a common understanding and then facilitating the communication among people; to reuse and integrate the disparate and heterogeneous representations; to formalize the representation of a domain problem or theory; and, as the basis to support semantic reasoning to full-fledged knowledge-based applications, among other aims.

For building our process conceptual base, elements from an ontology were used. These include the key generic terms, relationships and attributes, and their definitions. As per [39], an ontology with these features is called *light-weight ontology*, as opposed to *heavy-weight ontology*, which also includes axioms and constraints. On the other hand, as indicated at the beginning of Section 2, some authors like [14] distinguish among different types of ontologies regarding the subject of the conceptualization, e.g. *domain ontologies*, which express conceptualizations that are intended for particular domains, and *generic ontologies*, which include concepts that are considered to be generic across many domains. Regarding this classification, our process ontology can be considered a generic ontology, which can be reused for many different particular domains. Fig. 1 depicts the terms, properties and relationships which are grouped in the process component.

Core terms in this ontology are *Process*, *Activity* and *Task*. Specifically, a process is composed of sub-processes or activities, and in turn an activity is formed by sub-activities or tasks. A task is an atomic element that cannot be decomposed. Note that the semantic given to these three terms is compliant with the meaning given in ISO 12207. Additionally, we include the *Phase* concept, which represents a group of strongly-related processes or activities defined in a given order. While the process, activity and task terms have slightly different semantics, they do share common properties such as *name*, *objective*, and *Work Description*. Also they involve common *Roles*, *Work Products*, and *Conditions* -both *preconditions* and *postconditions* (see definitions in Table 3). The high-level *Work Definition* term -which embraces the common semantic of process, activity and task terms- is defined in Table 1 as an “*abstract entity which describes the work by means of consumed and produced Work Products, Conditions and involved Roles*”. Thus, process, activity and task terms are different specializations of it. Note that the *Work Definition* term is also used in SPEM.

Other key concept is *Work Product*. In turn, *Outcome*, *Artifact* and *Service* are kinds of work products. Outcome is defined in Table 1 as “*an intangible, storable and processable Work Product*”, while artifact “*is a tangible or intangible, versionable Work Product, which can be delivered*”. Lastly, service is defined as “*an intangible, non-storable and deliverable Work Product*”. The definition of service is based on the CMMI related term.

On the other hand, a work definition has a *Work Description*, which specifies the steps for achieving its *objective*. It represents ‘what’ should be done instead of ‘how’ should be performed. The semantic of ‘how’ is represented by the *Method* term, i.e. the specific and particular way to perform the specified steps for instance in a task. Note in Fig. 1 that the method concept has the *procedure* and *rules* attributes, which are defined in Table 2. The explicit relationship between Method (the ‘how’) and Work Description (the ‘what’) is not made as clear in other proposals as in ours.

It is important to point out that contrary to a process or activity, a task is scheduled and enacted. Hence, it has allocated *Resources* such as *Method* and *Tool* as well as *Agent* (i.e. a performer playing a *Role*). Resource is defined in Table 1 as an “*asset assigned to perform a Task*”.

Ultimately, this generic ontology for process contains the key concepts which are capable to enrich semantically different particular domains -one of the contributions indicated in the Introduction Section. In the sequel, we operationalize this capability by enriching many concepts for the M&E domain ontology.

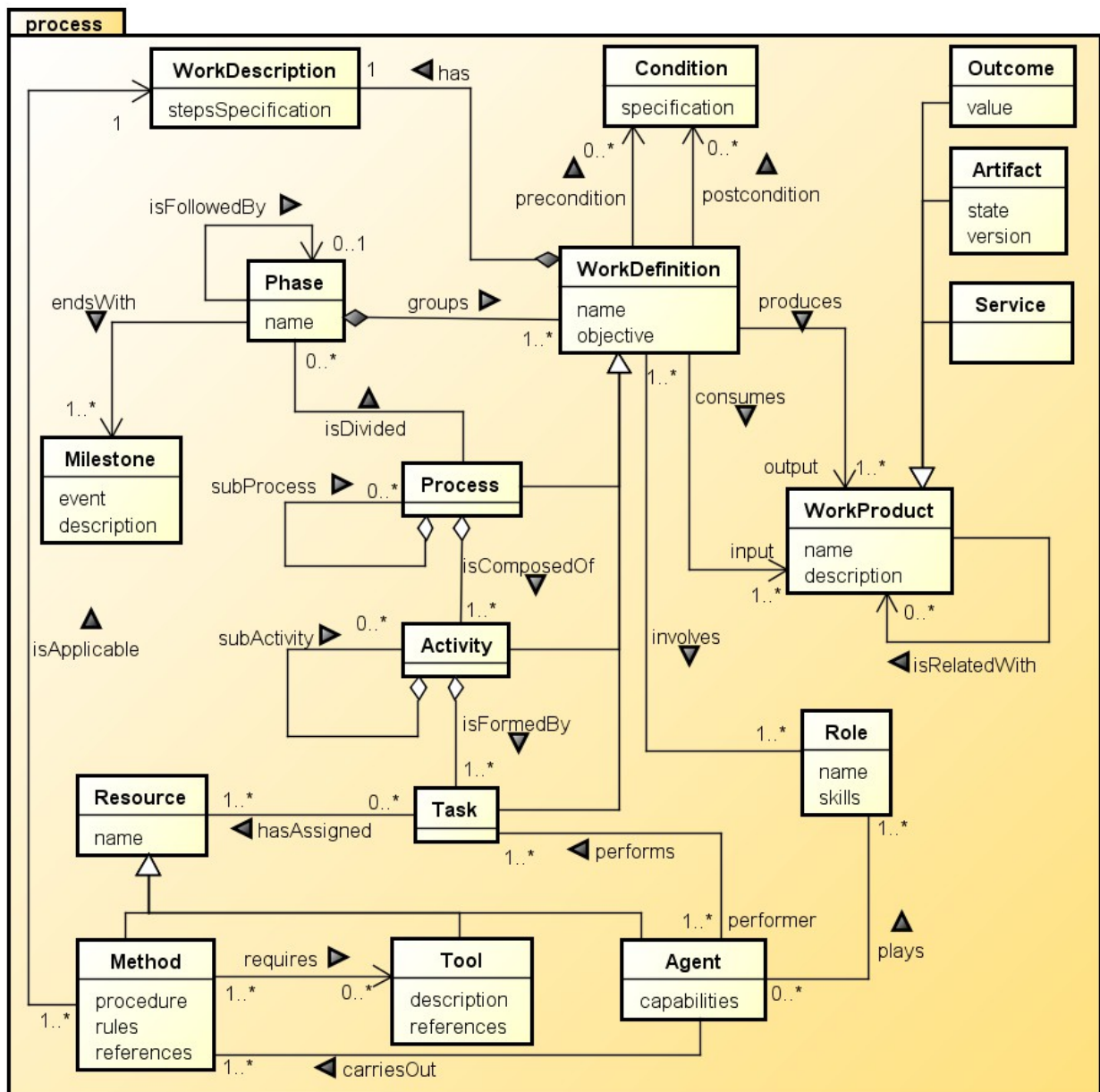


Figure 1: Generic terms, properties and relationships for the Process component

Table 1: Process generic ontology: Term definitions

Term	Definition
Activity	It is a Work Definition that is formed by an interrelated set of sub-activities and Tasks . <u>Note 1</u> : A sub-activity is an Activity at a lower granularity level. <u>Note 2</u> : In engineering projects, while Activities are planned, Tasks are scheduled and enacted.
Agent	Performer assigned to a Task in compliance with a Role . <u>Note 1</u> : An Agent can be human or automated.
Artifact	It is a tangible or intangible, versionable Work Product , which can be delivered.
Condition	Situation that must be achieved at the beginning (pre-condition) or ending (post-condition) of a Work Definition realization.
Method	Specific and particular way to perform the specified steps in the description of a Work Definition . <u>Note 1</u> : The specific and particular way of a Method –i.e. <i>how</i> the described steps in a work definition should be made- is represented by a procedure and rules.
Milestone	A meaningful event. <u>Note 1</u> : A Milestone represents for instance a Phase finalization.
Outcome	It is an intangible, storable and processable Work Product .
Phase	A group of strongly-related Work Definitions defined in a given order. <u>Note 1</u> : A Phase ends with a Milestone . <u>Note 2</u> : In a phase the Work Definitions are Processes and/or Activities .
Process	It is a Work Definition that is composed of an interrelated set of sub-processes and activities. <u>Note 1</u> : A sub-process is a Process at a lower granularity level.
Resource	Asset assigned to perform a Task . <u>Note 1</u> : An asset is an entity with added value for an organization.
Role	A set of skills that ought to own an Agent to perform a Work Definition . <u>Note 1</u> : Skills include abilities, competencies and responsibilities.
Service	It is an intangible, non-storable and deliverable Work Product .
Task	It is an atomic Work Definition , which cannot be decomposed. <u>Note 1</u> : Conversely to an Activity and Process , a Resource is assigned (scheduled) to a Task , e.g. Resources such as a Method , Agent , etc.
Tool	Instrument that facilitates the execution of a Method . <u>Note 1</u> : An instrument can be physical (hardware), computerized (software) or a mix of both types.
Work Definition	Abstract entity which describes the work by means of consumed and produced Work Products , Conditions and involved Roles . <u>Note 1</u> : Work represents a Process , an Activity or a Task .
Work Description	Specification of the steps for achieving the objective of a Work Definition . <u>Note 1</u> : The specification of the steps is a set of general actions –both Activities and Tasks - or a transformation function. It represents <i>what</i> should be done instead of <i>how</i> it should be performed. <u>Note 2</u> : The specification of the description of a Work Definition can be formal, semi-formal or informal as for example the natural language.
Work Product	A product that is consumed or produced by a Work Definition .

Table 2: Process generic ontology: Attribute definitions

Term	Attribute	Definition
Agent	capabilities	Set of abilities that the agent has.
Artifact	state	Situation or state in which the artifact is.
	version	Unique identifier which indicates the level of evolution of an artifact.
Condition	specification	Unambiguous specification of constraints or circumstances that must be achieved.
Method	procedure	Arranged set of method instructions or operations which specifies how the steps of a description of a work definition must be performed.
	rules	Set of principles, conditions, heuristics, axioms, etc. associated to the procedure.
	references	Cite to bibliographical or URL resources, where authoritative and additional information for a given method can be consulted.
Milestone	event	Name of the event which indicates a milestone.
	description	An unambiguous textual statement describing the event.

Outcome	value	Numerical or categorical result.
Phase	name	Name of a phase to be identified.
Resource	name	Name of a resource to be identified.
Role	name	Name of a role to be identified.
	skills	Set of capabilities, competencies and responsibilities of a role.
Tool	description	An unambiguous textual statement describing the tool.
	references	Cite to bibliographical or URL resources, where authoritative and additional information for a given tool can be consulted.
Work Definition	name	Name of a work definition to be identified.
	objective	Aim or end to be reached.
Work Description	stepsSpecification	Specification of steps to be followed in order to achieve the work definition objective. <u>Note</u> : Steps can be generic actions e.g., for a process, or concrete instructions e.g., for a task.
Work Product	name	Name of a work product to be identified.
	description	An unambiguous textual statement describing the work product.

Table 3: Process generic ontology: Relation definitions

Relationship	Definition
carriesOut	An agent carries out one or more methods.
consumes	In order to achieve its objective, a work definition consumes one or more work products.
endsWith	A phase ends with the occurrence of one or more milestones.
Groups	A phase groups one or more work definitions.
Has	A work definition has a work description in which specifies what to do for achieving its objective.
has Assigned	A task has assigned resources for its enactment.
involves	A work definition involves one or more roles. In turn, a role may participate in one or more work definitions.
isApplicable	A method is applicable to a description of a work definition. In turn, to a work description can be applied none or several methods.
isComposedOf	A process is composed of one or more activities.
isDivided	A process is divided in none or several phases.
isFollowedBy	A phase is followed by none or one phase.
isFormedBy	An activity is formed by one or more tasks.
isRelatedWith	A work product is related with none or several work products. <u>Note 1</u> : The relationship among work products can be of different types such as inheritance, aggregation, composition, etc.
performs	An agent performs one or more assigned tasks. In turn, a task is performed by one or more agents.
plays	An agent plays one or more roles. In turn, a role is played by none or several agents.
postcondition	A work definition may have associated conditions which must be accomplished at the end of its realization to be considered finished. <u>Note 1</u> : A post-condition defines any kind of constraint that must evaluate to true before the work described for the work definition can be declared completed or finished and which other work definition might depend upon.
precondition	A work definition may have associated conditions which must be accomplished before its initiation. <u>Note 1</u> : A pre-condition defines any kind of constraint that must evaluate to true before the work described for the work definition can start.
produces	A work definition produces (modifies, creates) one or more work products.
requires	A method requires to use none or several tools.
subActivity	An activity is formed by none or several more specific activities, named sub-activities.
subProcess	A process is formed by none or several more specific processes, named sub-processes.

4 Enriching the M&E Domain Ontology

As commented in the Introduction Section, the integrated GOCAME strategy for the M&E domain is supported by three capabilities simultaneously, viz. the *M&E conceptual base and framework*; the *specifications of M&E process*

views; and the *specifications of methods* for the M&E activities. In sub-section 4.1, we summarize the GOCAME strategy and its three capabilities, with emphasis on the first capability. Next, in sub-section 4.2, we enrich and enhance the GOCAME M&E conceptual framework with our process generic ontology. Then, in sub-section 4.3, we describe the practical impact of this improvement on the process and method specifications. In order to illustrate how the augmented conceptual framework impacts on the other capabilities, excerpts of an ICT security and risk evaluation case study [13] are employed. Finally, in sub-section 4.4, a discussion on how the enriched method specifications support more consistent, comparable and robust analysis of data and metadata is presented.

4.1. GOCAME Overview

GOCAME is a multi-purpose M&E strategy that follows a goal-oriented and context-sensitive approach in defining projects. It is an approach in which the requirements specification, M&E design, and analysis of results are designed to satisfy a specific information need in a given context, providing therefore more robust evaluation interpretations among different project results at intra- and inter-organization levels. This strategy is based on the abovementioned three capabilities, which are summarized below.

From the very beginning [4], GOCAME had its M&E terminological base defined as a domain ontology [9]. From this ontology emerges the C-INCAMI conceptual framework. This domain model allows an explicit and common vocabulary which is shared among the organization's M&E projects leading to more consistent analysis of results across projects. C-INCAMI was structured in six components, namely: i) *M&E project*, ii) *Nonfunctional requirements*, iii) *Context*, iv) *Measurement*, v) *Evaluation*, and vi) *Analysis and Recommendation*. A summarized description of each component is provided below, where M&E terms highlighted in italic are shown in Fig. 2. (Note that we also provide this description to understand and contrast better the enhancement made to the iv) and v) components in sub-section 4.2).

i) *M&E project component*

This component defines and relates a set of Project concepts needed to deal with M&E activities, resources and work products. Looking at the project definition given in the Introduction Section, we can redefine a M&E Project as a “*temporary and goal-oriented endeavor with intended start and finish dates, which consider a managed set of interrelated activities, tasks and resources aimed at defining nonfunctional requirements, measuring and evaluating entities (i.e. products, services, systems, processes, resources, etc.) for satisfying a given requester information need*”.

A clear separation of concerns among Requirements Project, Measurement Project and Evaluation Project concepts is made for reusing purposes as well as for easing the manager role. The main concept in this component is a Measurement and Evaluation Project (MEProject), which allows containing a concrete Requirements Project with the specification of the M&E information need and the rest of the nonfunctional requirements information. From this requirements project one or more Measurement Projects can be defined and associated. In turn, for each measurement project one or more Evaluation Projects could be defined. Hence, for each M&E Project we can manage associated subprojects accordingly. Each project also has information such as responsible person's name and contact information, starting and ending date, among other. Ultimately, this separation of concerns for each MEProject facilitates the reuse, traceability and consistency for ulterior intra- and inter-project analysis.

ii) *Nonfunctional requirements component*

This component includes concepts and relationships needed to define the nonfunctional requirements for measurement and evaluation. One of the main concepts is the *Information Need*, which specifies:

- the *purpose* to performing the evaluation (e.g. “understand”, “predict”, “improve”, “control”);
- the *focus concept (Calculable Concept)* to be assessed (e.g. “quality”, “quality in use”, “security”, etc.);
- the category of the entity (*Entity Category*) that will be assessed, e.g. a “Web application” (which its *super Category* is “system”), and the concrete *Entities* (such as “Facebook”, “Linkedin”). Other super categories for entities can be “resource”, “project”, “process”, “system-in-use” (e.g. as a Web application-in-use), etc.
- the *user viewpoint* (i.e. the target user as “developer”, “final user”, etc.) from which the focus concept will be evaluated;

The *focus concept* constitutes the higher-level concept of the nonfunctional requirements; in turn, a *Calculable Concept* and its *sub-concepts* are related by means of a *Concept Model*. This may be a tree-structured representation in terms of related mid-level calculable concepts and lower-level measurable *Attributes*, which are associated to the target entity. Predefined instances of metadata for information needs, entities and entity categories, calculable concepts, attributes, etc., and its corresponding data can be obtained from an organizational repository to support reusability and consistency in the requirements specification along the organizational projects.

iii) *Context component*

This component includes concepts and relationships that deal with the context information specification. The main concept is *Context*, which represents the relevant state of the situation of the target entity to be assessed with regard to the stated information need. We consider *Context* as a special kind of *Entity* in which related relevant entities are involved; consequently, the context can be quantified through its related entities. By relevant entities we mean those that could affect how the focus concept of the assessed target entity is interpreted (examples of relevant entities of the context may include resources as a network infrastructure, a working team, lifecycle types, the organization or the project itself, among many others). In order to describe the situation, attributes of the relevant entities (involved in the context) are used. These are also *Attributes* called *Context Properties*, which can be quantified to describe the relevant context of the entity under analysis. A context property inherits the metadata from the *Attribute* concept such as *name*, *definition* and *objective*, and adds others as well.

All these context properties' metadata are meant to be stored in the organizational repository, and for each MEProject the particular metadata and its values are stored as well. A detailed illustration of context and the relationship with other C-INCAMI components can be found in [36].

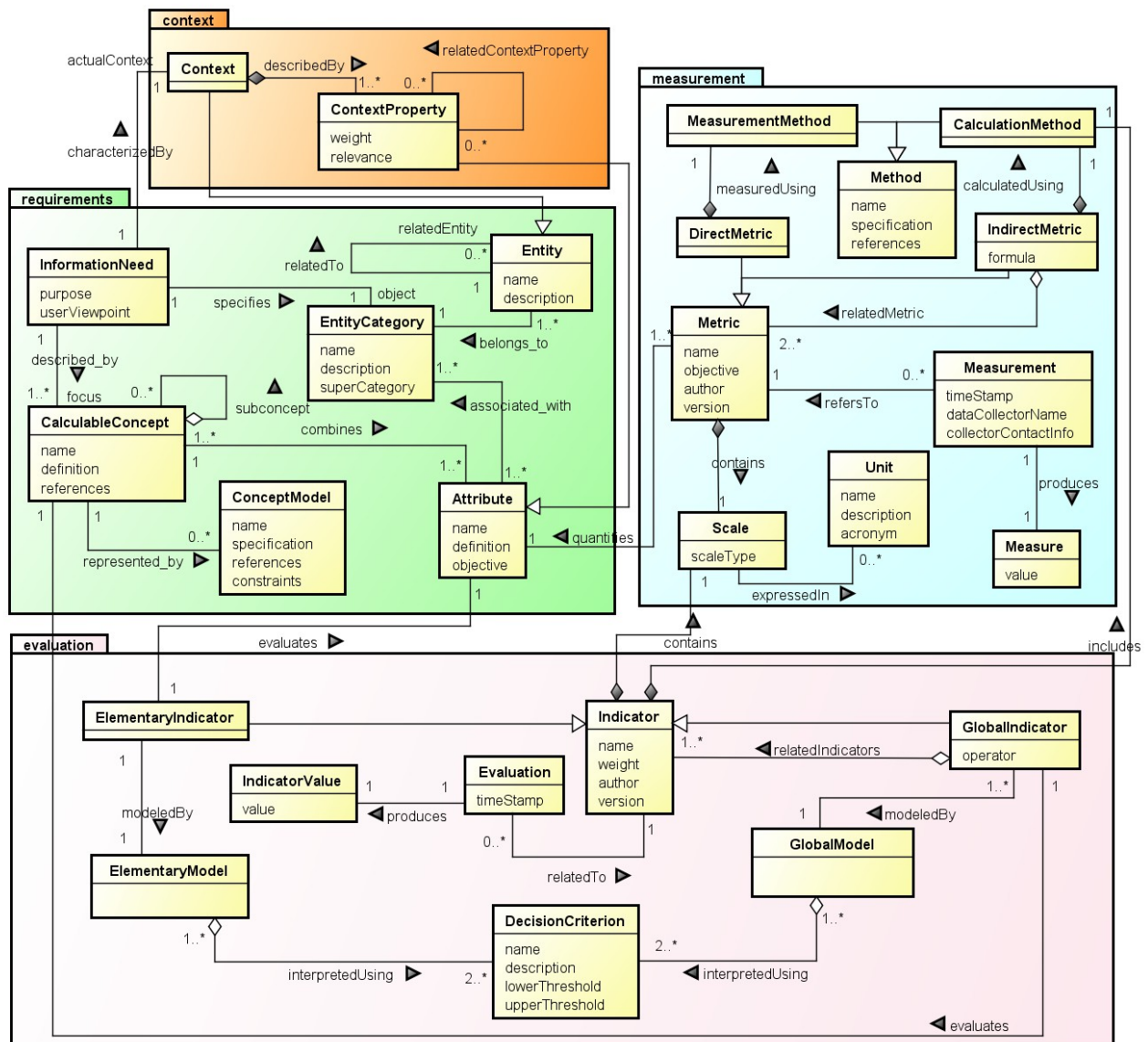


Figure 2: Main terms, properties and relationships of the former C-INCAMI conceptual framework

iv) *Measurement component*

This module includes the concepts and relationships intended to specify the measurement design and implementation, for instance, the concrete *Entities* that will be measured, the selected *Metric* for each attribute, and so on. Regarding measurement design, a *Metric* provides a *Measurement* specification of how to quantify a particular attribute of an entity, using a particular *Method*, and how to represent its values, using a particular *Scale*. The properties of the measured values in the scale with regard to the allowed mathematical and statistical operations

and analysis are given by the *scaleType*. Two types of metrics are distinguished. *Direct Metrics* are those for which values are obtained directly from measuring the corresponding entity's attribute, by using a *Measurement Method*. On the other hand, *Indirect Metrics'* values are calculated from others direct metrics' values following a *function* specification and a particular *Calculation Method*.

For measurement implementation, a *Measurement* specifies the task (note that in Fig. 2 is not explicit that *Measurement* has the semantic of task) by using a particular metric description in order to produce a *Measure* value. Other associated metadata is the *data collector* name and the *timestamp* in which the measurement was performed.

v) *Evaluation component*

This component includes the concepts and relationships intended to specify the *Evaluation* tasks through *Indicators*. *Indicator* is the main term, which allows specifying how to calculate and interpret the attributes and calculable concepts of nonfunctional requirement models.

Two types of indicators are distinguished. First, *Elementary Indicators* that evaluate lower-level requirements, namely, attributes combined in a concept model. Each elementary indicator has an *Elementary Model* that provides a mapping function from the metric's measures (the domain) to the indicator's scale (the range). The new *scale* is interpreted using a set of *Decision Criteria*, which help analyze the level of satisfaction reached by each elementary nonfunctional requirement, i.e. by each attribute. Second, *Global Indicators*, which evaluate mid-level and higher-level requirements, i.e. sub-characteristics and characteristics in a concept model. Different aggregation models (*Global Model*) can be used to perform evaluations. The global indicator's value ultimately represents the global degree of satisfaction in meeting the stated information need for a given purpose and user viewpoint.

As for the implementation, an *Evaluation* represents a task (again, it is not explicit that *Evaluation* has the semantic of task), which involves a single calculation, following a particular indicator specification –either elementary or global-, producing an *Indicator Value*.

It is worthy to mention that the selected metrics are useful for a measurement task as long as the selected indicators are useful for an evaluation activity in order to interpret the stated information need.

vi) *Analysis and Recommendation component*

This component includes concepts and relationships dealing with analysis design and implementation as well as conclusion and recommendation. Analysis and recommendation use information coming from each MEProject (which includes requirements, context, measurement and evaluation data and metadata). By storing all this information and by using different kinds of statistical techniques and visualization tools, stakeholders can analyze the assessed entities' strengths and weaknesses with regard to an established M&E information need, and justify recommendations in a consistent way.

Unlike the generic ontology described in Section 3, the M&E domain ontology is a *heavy-weight ontology* since includes concepts, their definitions, relationships, attributes and *axioms*. An axiom expresses new relationships between concepts and limits the possible interpretations adding structural or not structural restrictions [40]. In the M&E ontology, axioms are expressed in first-order logic, using unary and binary predicates to represent instances of terms and relationships, respectively. For example, the *attribute (a)* predicate means “*a* is an instance of *Attribute*” while the *quantifies (m, a)* predicate means “the *m* metric quantifies the *a* attribute”.

The second GOCAME capability refers to the specifications of M&E process models and views [7]. When specifying a process, often engineers think more about what a process must do rather than how activities should be performed. In order to foster repeatability and reproducibility, a process specifies (i.e. prescribes or informs) a set of activities, inputs and outputs, interdependencies, among other concerns. Also, to deal with the inherent complexity of processes, process views –also quoted in process modeling literature as perspectives- are used.

A view is a particular model to represent, specify and communicate different aspects of a process. For instance, according to Curtis *et al.* [10], a process can be modeled taking into account four views, namely: i) *functional* that includes the activities' structure, inputs and outputs, etc.; ii) *informational* that includes the structure and interrelationships among artifacts produced or consumed by the activities; iii) *behavioral* that models the dynamic view of processes; and, iv) *organizational* that deals with agents, roles and responsibilities.

GOCAME process embraces the following customizable activities:

- A1) *Define Non-Functional Requirements;*
- A2) *Design the Measurement;*
- A3) *Design the Evaluation;*
- A4) *Implement the Measurement;*
- A5) *Implement the Evaluation;* and,
- A6) *Analyze and Recommend.*

Fig. 3 shows the M&E process from the functional and behavioral viewpoint using the notation of the SPEM language [22]. The reader can see that concepts defined in the M&E ontology (Fig. 2) such as *Metric*, *Measure* and

Indicator, etc., are also reused in the process specification.

Summarizing, the GOCAME customizable process can be described as follows. Once the requirements project has been created, first, the *Define Non-Functional Requirements* activity has a specific goal/problem/risk as input and a nonfunctional specification document as output. Then, the *Design the Measurement* activity allows identifying the metrics from the Metrics repository to quantify attributes: the output is a metric specification document. Note that repositories are represented by <<datastore>> stereotype in Fig. 3. Once the measurement was designed, the evaluation design and the measurement implementation activities can be performed (in any order or in parallel). The *Design the Evaluation* activity allows identifying indicators in order to know the satisfaction level achieved by elementary and global requirements. The *Implement the Measurement* activity uses the specified metrics to obtain the measures, which are stored in the Measures repository. Next, the *Implement the Evaluation* activity can be carried out. Finally, *Analyze and Recommend* activity has as inputs the values (i.e., data) of measures and indicators, the requirements specification document, and the associated metrics and indicators specifications (i.e., metadata) in order to produce a Conclusion/Recommendation report. More details about the GOCAME M&E process specification using different views can be found in [7].

Note that GOCAME has a customizable process, since this strategy pattern can be instantiated to measure and evaluate the quality focus for not only service, product, system and system-in-use entity categories but also for other ones such as resource and process, by using their instantiated processes and resources accordingly. Beyond GOCAME, other strategy patterns that incorporate improvement cycles within or between quality views can be designed (as we will document in other manuscript).

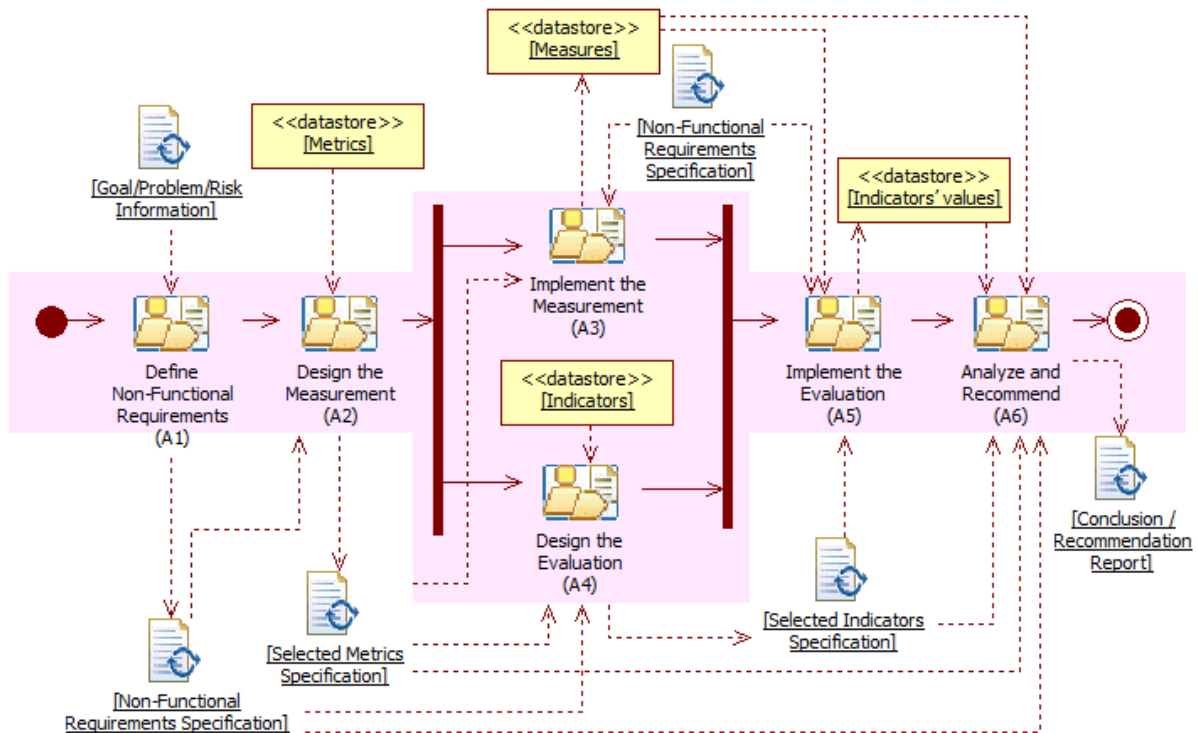


Figure 3: GOCAME's functional and behavioral process views

Regarding the third capability, GOCAME activities are supported by different method specifications. While activities state ‘what’ to do, methods describe ‘how’ to perform these activities which must be accomplished by agents playing roles. In addition, a methodology is a set of related methods. Since the above M&E process includes activities such as specify the requirements tree, identify metrics, and so on, we have envisioned a methodology that integrates all these aspects and a tool that automate it. That is, a set of well-defined and cooperative methods and tools which are applied consistently to M&E tasks for producing work products accordingly. For example, for the measurement task the metric specifications artifact is yielded, and so on. Examples of method specifications can be found in the next sub-section.

In the sequel, we demonstrate how the process generic ontology was used to enrich semantically terms, relationships and axioms of the former measurement and evaluation components (GOCAME's first capability) and then, illustrate how the enhanced M&E framework impacts on process and method specifications (GOCAME's second and third capabilities).

procedure and a *Scale* and a set of *Decision Criteria* as rules, as shown in Fig. 4.

Consequently, many of the former M&E term definitions [4, 9] have been updated to reflect this new situation. In addition, new terms such as *Direct Measurement*, *Indirect Measurement*, *Base Measure*, *Derived Measure*, etc. have emerged in order to have greater terminological completeness and detail (compare terms from Fig. 2 of the Measurement component with those in Fig. 4). All these adapted definitions and/or new terms as well as the links to process terms are shown in Table 4.

On the other hand, to increase the consistency between the M&E components and the process component, some relationships among M&E terms have been adapted accordingly, as depicted in Fig. 4. For instance, we added the *consumes* relationship between *Elementary Evaluation* and *Measure* terms. Thus an Elementary Evaluation task *consumes* a Measure of an attribute (as input) and *produces* an Indicator Value (as output). Note that the added or renamed relations are highlighted in gray and red text in Fig. 4.

Table 4: Definition of M&E terms, which are semantically enriched with process terms

M&E Term	Definition	Process Term
Measurement Term		
Base Measure	A measure that does not depend upon other measure.	Outcome
Calculation Procedure	Set of established and ordered instructions of an indirect metric or indicator that indicates how the described steps in an indirect measurement or evaluation task should be carried out.	procedure in Method
Derived Measure	A measure that is derived from other measures.	Outcome
Direct Measurement	Measurement that produces a base measure.	Task
Direct Metric	A metric of an attribute that does not depend upon a metric of any other attribute	Method
Indirect Measurement	Measurement that produces a derived measure.	Task
Indirect Metric	A metric of an attribute that depends of metrics of other attributes.	Method
Measure	The number or category assigned to an attribute of an entity by making a measurement. <u>Note 1:</u> It is the measurement output that represents an outcome as work product.	Outcome
Measurement	A task that uses a metric in order to produce a measure's value. <u>Note 1:</u> This task quantifies an attribute by producing a measure as outcome.	Task
Measurement Procedure	Set of established and ordered instructions of a direct metric that indicates how the described steps in a direct measurement task should be carried out.	procedure in Method
Metric	The defined measurement or calculation procedure and the scale. <u>Note 1:</u> A metric is a method which is applicable to the description of a measurement task.	Method
Evaluation Term		
Derived Evaluation	Evaluation that produces an indicator's value by assessing a calculable concept.	Task
Derived Indicator	An indicator that is derived from other indicators to evaluate a calculable concept.	Method
Elementary Evaluation	Evaluation that produces an indicator's value by assessing an attribute. <u>Note 1:</u> An attribute is a non-functional elementary requirement from the evaluation standpoint.	Task
Elementary Indicator	An indicator that does not depend upon other indicators to evaluate an attribute.	Method
Evaluation	A task that uses an indicator in order to produce an indicator's value.	Task
Indicator	The defined calculation procedure and scale in addition to the indicator model and decision criteria in order to provide an evaluation of a calculable concept or attribute with respect to a defined information need. <u>Note 1:</u> An indicator is a method which is applicable to the description of an evaluation task.	Method
Indicator Value	The number or category assigned to a calculable concept or attribute by making an evaluation. <u>Note 1:</u> It is the evaluation output that represents an outcome as work product.	Outcome

Regarding the used mechanism for the semantic linking between the M&E components and the process generic ontology, it is important to analyze why the use of stereotypes is, in this context, a more suitable mechanism than using inheritance relationships. Even though, the concepts in domain ontologies are often represented as specializations of concepts in generic ontologies [14], sometimes with the aim of reuse, a domain term should not be considered strictly a specialization of a generic concept. For example, in the context of a M&E process, a metric must be considered an specialization of method (i.e. method from the process generic ontology), while in the context of a metric reviewing and cataloging process, the same metric term must be considered an specialization of the work product term but not considered an specialization of the method term. Thus, in some cases, defining a term as a specialization (inheritance relationships) of generic terms can minimize the reuse of a domain ontology in addition to promoting high-coupling level between the M&E and the process components. To some extent this issue can be spotted in [24, 25] where SPO is enriched with UFO using inheritance relationships (as indicated in Section 2).

Other alternative way to deal with the above concern is using profiles, which is defined by a set of stereotypes. A profile is a mechanism provided by UML to extend its syntax and semantics and thus UML now expresses specific concepts of a particular application domain. There are several reasons why a designer may want to customize or extend a metamodel, such as: give a terminology that is adapted to a particular domain, add semantics that is left unspecified in the metamodel, or add information that can be used when transforming a model to another model or code [41]. Therefore, the use of stereotypes offered us a more suitable alternative to enrich domain terms. The C-INCAMI components can be reused in different contexts by applying stereotypes appropriately to M&E terms to add more semantic. Furthermore, a process profile from the analyzed process generic ontology supports the enrichment of other specific domains such as software development and maintenance [42], data stream processing [43], amongst many others.

As above mentioned, a set of axioms had been specified for the M&E domain ontology. Some of those axioms can be now reformulated appropriately considering the new situation. An example is the following axiom:

“an Indicator Value (*iv*) can be used to interpret an Attribute (*a*) if and only if the *a* attribute is associated to a concrete Entity (*e*) and a Measurement (*mnt*) consumes the *a* attribute and produces a Measure (*m*) by means of a Metric (*met*), and also, an Elementary Evaluation (*ee*) consumes the *m* measure and produces the *iv* indicator value using an Elementary Indicator (*ei*)”. This axiom can be expressed in a first-order logic sentence, such as:

$$(\forall iv, a) \text{ interprets}(iv, a) \leftrightarrow (\exists e, mnt, met, m, ee, ei) \text{ associated}(a, e) \wedge \text{hasAssigned}(mnt, met) \wedge \text{consumes}(mnt, a) \wedge \text{produces}(mnt, m) \wedge \text{hasAssigned}(ee, ei) \wedge \text{consumes}(ee, m) \wedge \text{produces}(ee, iv)$$

Ultimately, the augmented C-INCAMI conceptual framework has also a positive impact on the other GOCAME strategy capabilities since ensures terminological uniformity to process and method specifications while strengthening their verifiability. Next, for illustration purposes, we show excerpts of process and method specifications regarding the new situation.

4.3. Impact of Enhanced M&E Components in Process and Method Specifications: A Running Example

The GOCAME process specification capability embraces different process views. Fig. 3 depicts the M&E activity diagram stressing the functional and behavioral perspectives. While the functional view represents what activities/tasks should be performed -as well as the inputs and outputs (work products) that will be consumed and produced-, the behavioral view models the dynamics of the process, i.e., sequences, parallelisms, iterations, feedback loops, among other aspects. Note that in Fig. 3, the names of activities and work products make use of the M&E terminology such as *Measurement*, *Evaluation*, *Metric*, *Measure*, among others. As a consequence, the use of the C-INCAMI conceptual base benefits the terminological uniformity in process specification views. Moreover, the augmented C-INCAMI conceptual base with process generic terms has a positive impact on the M&E process and method specifications due to a greater semantic consistency is achieved.

In order to demonstrate how process generic terms linked to M&E specific terms allow building more consistent and verifiable process view and method specifications, following we use excerpts of an ICT security and risk evaluation case study [7].

In this case study, we customize the GOCAME M&E process for evaluating some Security sub-characteristics and attributes of a student management Web system widespread used in Argentinean public universities. *Define Non Functional Requirements for Security* is then the instantiated activity name of A1 in Fig. 3. The concrete target Entity is called “XYZ register system” (a fictitious name for confidentiality reasons), which is a Web application from the Entity Category standpoint, commonly used by students, professors and faculty members. The concrete Information Need was raised in 2012 by the ICT department responsible, in the context of the ABC organization – again a fictitious name for the real institution. The information need was related to security risks due to different

potential threats, e.g., students changing bad marks of subjects due to system vulnerabilities. Thus, the purpose of this objective was firstly to “understand” the current satisfaction level achieved for the “External Quality” of the XYZ Web application, particularly for non-vulnerabilities regarding the “Security” quality focus, from the “security administrator” user viewpoint. Once the met satisfaction levels were understood by analyzing indicators’ and measures’ values, the further purpose was to “improve” the system in those weakly performing indicators. That is to say, the ultimate purpose was to reduce security risks by improving vulnerable attributes in the web system by risk treatment.

All this information was documented in the *Non Functional Requirements Specification* artifact (Fig. 5) produced by A1 following the activity flow depicted in Fig. 6. This artifact is in turn composed of other artifacts as shown in Fig. 5. Note that the information contained in different artifacts can be verifiable regarding the M&E ontology. For example, the *Information Need Specification* includes the *Purpose Specification*, *User viewpoint Specification* and *Evaluation focus Specification*, among others, while the *Information Need* concept has in Fig. 2 a *purpose* and *user viewpoint* as attributes and it is related to a *Calculable Concept* as *focus*. This verification can be performed because the process specifications (in this case the informational view) adhere to or are in compliance with the M&E ontology.

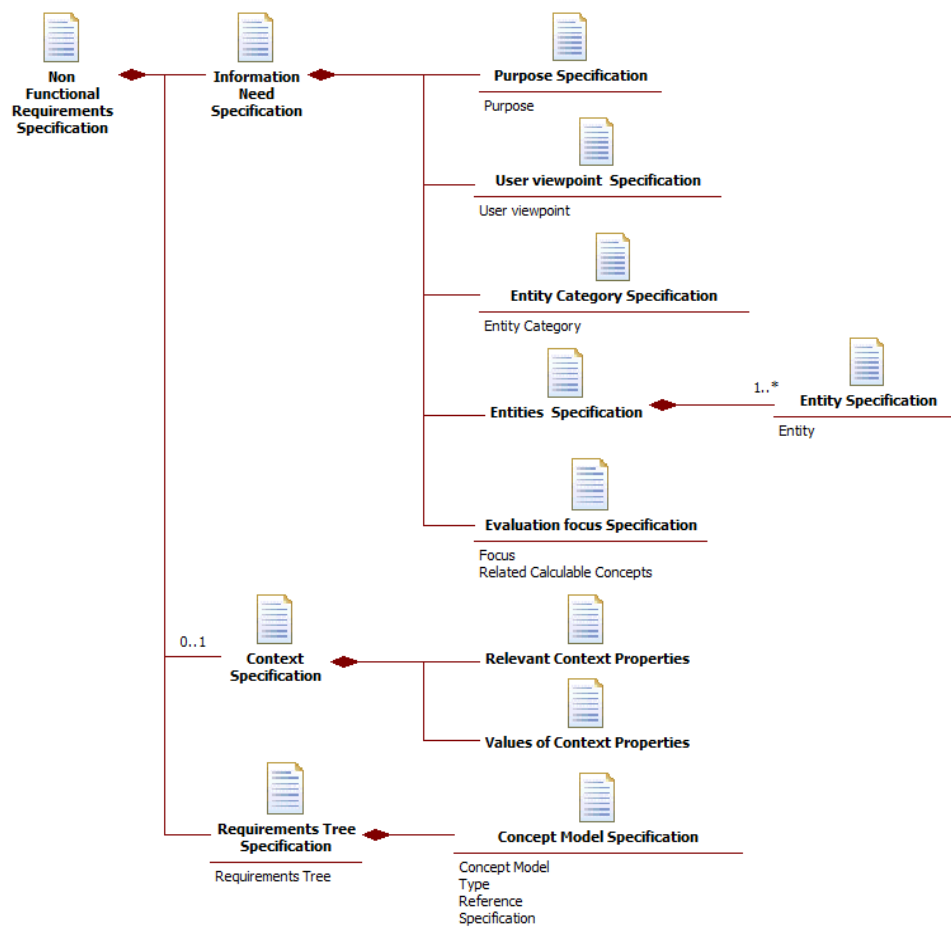


Figure 5: Informational view of the *Non Functional Requirements Specification* artifact

Moreover, the informational view for the *Non Functional Requirements Specification* artifact can be verified as well regarding the functional and behavioral views of the A1 activity. According to the informational view in Fig. 5, the *Non Functional Requirements Specification* is composed of one *Information Need Specification*, none or one *Context Specification* and one *Requirements Tree Specification*. The cardinality of these three artifacts can be verified analyzing the Fig. 6. As the reader can check it out, Establish the Information Need and Establish the Requirements Tree are mandatory A1 sub-activities, so the Information Need Specification and Requirements Tree Specification artifacts always are produced. On the other hand, the Specify the Context activity is optional (represented by a diamond decision node in Fig. 6), therefore the Context Specification is not always yielded. Consequently, the consistency of cardinalities for all artifacts can be verified checking the informational view against functional and behavioral views.

Additionally, Fig. 7 shows the resulting *Requirements Tree Specification* used in the XYZ register system case study -without sub-characteristics and attributes definitions for brevity reasons. This document is part (or a section) of the *Non Functional Requirements Specification* document as depicted in Fig. 5.

Following the process depicted in Fig. 3, the next customized activity is *Design the Measurement for Security* (A2). In this activity, the *Metrics Expert* role should select, for each attribute of the *Requirements Tree Specification*, one metric from the Metrics repository (see the object called Metrics with the datastore stereotype in Fig. 3). As result, the work product produced is the *Selected Metrics Specification* document. This artifact is composed by a set of *Metric Specification* documents (one per each attribute).

Fig. 8 represents the *Metric Specification* artifact for the “Stored Cross-Site Scripting Immunity” attribute name (coded 1.2.1.2 in Fig. 7).

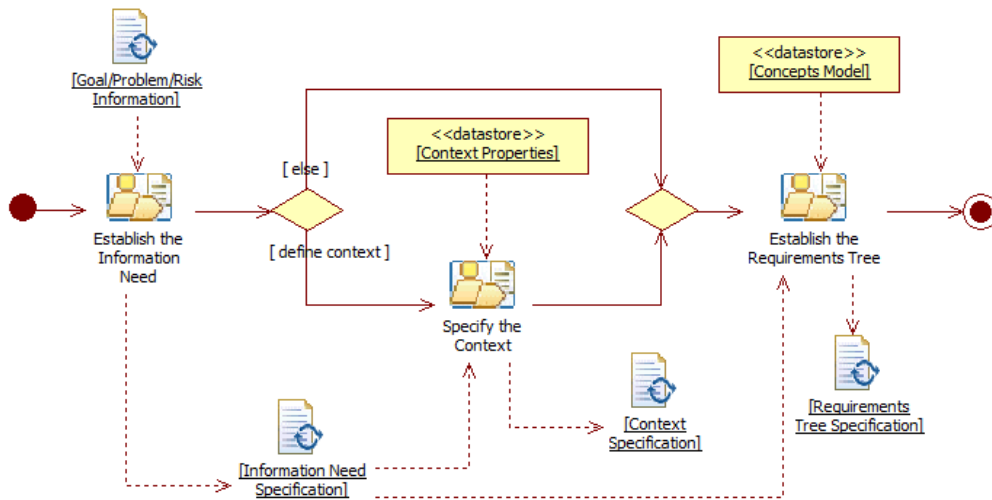


Figure 6: Functional and behavioral views for the A1 activity

<p>1. Security</p> <p>1.1. Confidentiality</p> <p>1.1.1. Authentication Schema Protectability</p> <p><i>1.1.1.1. Authentication Schema Bypass</i></p> <p>1.1.2. Session Data Protectability</p> <p><i>1.1.2.1. Session Data Theft Protectability</i></p> <p>1.2. Integrity</p> <p>1.2.1. Cross-Site Scripting Immunity</p> <p><i>1.2.1.1. Reflected Cross-Site Scripting Immunity</i></p> <p><i>1.2.1.2. Stored Cross-Site Scripting Immunity</i></p> <p><i>1.2.1.3. DOM-based Cross-Site Script. Immunity</i></p> <p><i>1.2.1.4. Cross-site Request Forgery Immunity</i></p> <p>1.2.2. Session Data Tampering Protectability</p> <p>1.3. Authenticity</p> <p>1.3.1. Session ID Protectability</p> <p><i>1.3.1.1. Session ID Theft Protectability</i></p> <p><i>1.3.1.2. Session ID Tampering Protectability</i></p> <p>1.3.2. Session Impersonation Protectability</p> <p>1.3.2.1. Session Management Protectability</p> <p><i>1.3.2.1.1. Session ID Expiration</i></p> <p><i>1.3.2.1.2. Session Expiration Due to Idle Timeout</i></p> <p><i>1.3.2.1.3. Re-authentication Mechanism Availability</i></p> <p><i>1.3.2.1.4. Session ID Regeneration Availability</i></p> <p><i>1.3.2.1.5. Keep-me-logged-in Mechanism Availability</i></p> <p><i>1.3.2.2. Session Non-Replay Protectability</i></p> <p><i>1.3.2.3. Session Fixation Protectability</i></p> <p>1.3.3. Password Protectability</p> <p><i>1.3.3.1. Password Aging Policy</i></p> <p><i>1.3.3.2. String Password Strength</i></p>

Figure 7: *Requirements Tree Specification* artifact for the “Security” characteristic. Note that attributes are highlighted in italic

<u>Metric Specification</u>		
Metric Name: <i>Ratio of Stored Cross Site Scripting (%SXSS)</i>		
Objective: Determine the ratio between the number of SXSS vulnerable persistent-data variables and the total number of persistent-data variables taken from the 'set of pages' sub-entity.		
Author: Alexander Dieser		
Version: 1.0		
Attribute: Stored Cross-Site Scripting Immunity (1.2.1.2)		
Calculation Procedure: Formula: $\%SXSS = \begin{cases} \left(\frac{\#VPDv}{\#PDv}\right) * 100, & \text{if } \#PDv \neq 0 \\ 0, & \text{if } \#PDv = 0 \end{cases}$		
Scale Type: Ratio	Representation: Continuous	Value Type: Real
Unit (Name): Percentage		Acronym: %
Related Metrics		
Metric Name: <i>Number of Vulnerable Persistent-Data variables (#VPDv)</i>		
Objective: Count the total number of SXSS vulnerable persistent-data variables within the 'set of pages' sub-entity.		
Author: Alexander Dieser		
Version: 1.0		
Attribute: Amount of Vulnerable Persistent-Data variables		
Measurement Procedure:		
#VPDv = 0;		
for each persistent-data variable within the 'set of pages':		
Set or replace the persistent-data variable value with any javascript code that triggers a response from the web browser (e.g. <script>alert(document.cookie)</script>); Perform the corresponding action to store the value into the back-end;		
if (the browser executes the injected javascript code instead of displaying the value of the persistent-data variable)		
then #VPDv = #VPDv + 1;		
Scale Type: Absolute	Representation: Discrete	Value Type: Integer
Unit (Name): vulnerable Persistent-Data var.		Acronym: PDv
Metric Name: <i>Total number of Persistent-Data variables (#PDv);</i>		
Objective: Count the total number of persistent-data variables which are in the 'set of pages';		
Author: Alexander Dieser		
Version: 1.0		
Attribute: Amount of Persistent-Data variables		
Measurement Procedure: <u>Note:</u> A persistent-data variable is a user input that is stored into the back-end (e.g. a database) and then displayed by the application.		
#PDv = 0;		
for each page within the 'set of pages' sub-entity:		
Identify all text fields or text areas longer than X _{min} characters where the user input is stored into the back-end; Fill them with Test data (i.e. any javascript code that triggers a response from the web browser) and store them into the back-end;		
for each Test data already stored:		
if (is displayed in at least one page from the 'set of pages') then #PDv = #PDv + 1;		
Scale Type: Absolute	Representation: Discrete	Value Type: Integer
Unit (Name): Persistent-Data variable		Acronym: PDv

Figure 8: *Metric Specification* artifact for the “Stored Cross-Site Scripting Immunity” (1.2.1.2) attribute

Recall that the *Metric* term has the semantic of *Method* in Fig. 4. A metric as a method represents the “*specific and particular way to perform the specified steps in the description of a Work Definition*” according to the definition given in Table 1. In other words, a metric specifies *how* should be implemented the work description (i.e. *what*) of a measurement task -where a Task is a Work Definition in Fig. 1.

The enriched M&E terms have a positive impact to the GOCAME capability of method specifications. In this way, when storing and retrieving the metric templates from the Metrics repository, all the associated metadata can be verified for consistency.

For example, the *Metric Specification* artifact in Fig. 8 can be checked regarding the enriched Measurement component. Metric term has a name, objective, author and version, and therefore these metadata are accordingly used in Fig. 8. Besides, a Metric in Fig. 4 is associated to the Attribute term by the *quantifies* relationship, therefore, in the *Metric Specification* artifact the related attribute should be indicated as is in Fig. 8. Taking into account that the Metric term is enriched with the *Method* stereotype from the process component, then a metric must specify the

procedure and rules metadata. “Ratio of Stored Cross Site Scripting” metric (see Fig. 8) is an Indirect Metric, so in the template the Calculation Procedure and the Scale (as a rule) are specified. Note that in Fig. 4 an Indirect Metric is related to other Metrics, so that the Metric Specification artifact in Fig. 8 is consistent since has the Related Metrics section -notice that related metrics are also specified in the template.

Once metrics were selected in the A2 activity, the *Data Collector/Calculator* performs A3, i.e., *Implement the Measurement for Security*. This activity implies executing, iteratively, the *Measurement* task for each attribute from the *Requirements Tree Specification* artifact, as shown in Fig. 9.

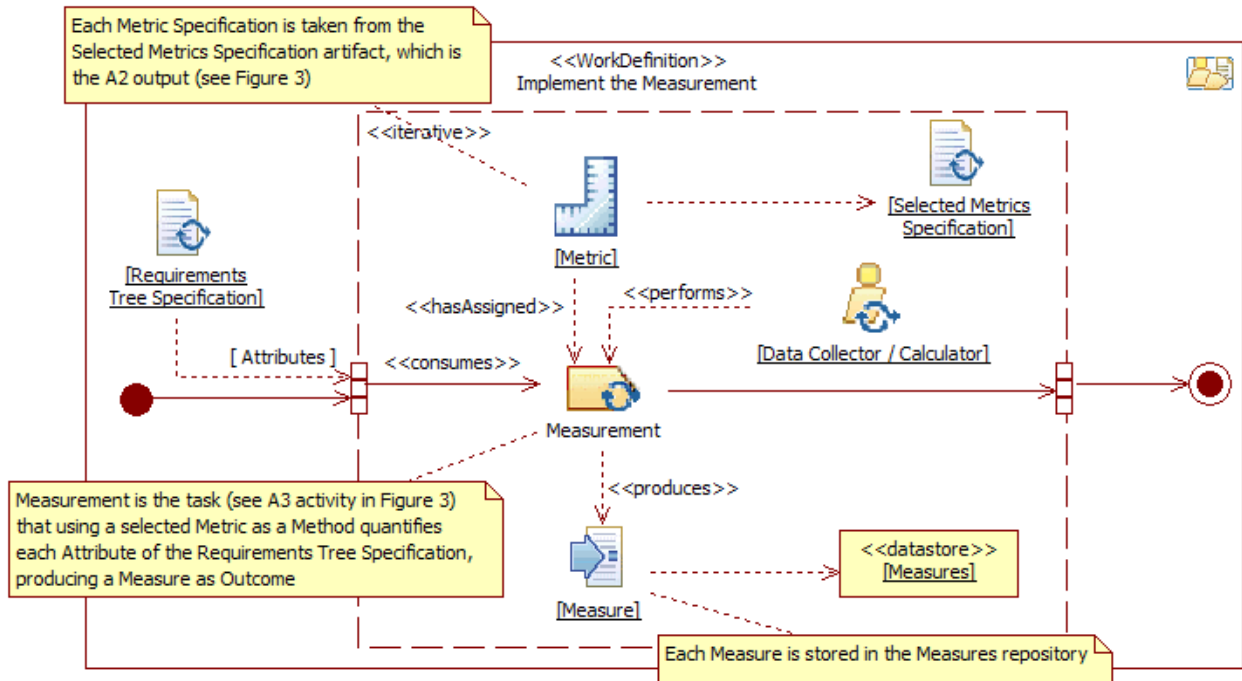


Figure 9: Functional and behavioral views for the *Implement the Measurement* activity

Looking at this figure we can see that per each iteration, the *Measurement* task execution consumes an attribute and produces a measure, which is stored in the Measures datastore. In order to perform the measurement for a given entity attribute the *Data Collector/Calculator* must follow the (measurement or calculation) procedure and the rules respectively described in the (direct or indirect) Metric template.

The process specification in Fig. 9 is semantically consistent when checked against the enriched C-INCAMI measurement component. Thus, we observe in this figure that the task named *Measurement* consumes an attribute and produces a measure. Likewise in the measurement component, the *Measurement* term -stereotyped with «Task» in Fig. 4- is associated to the *Attribute* term with the *consumes* relationship, and to the *Measure* term with the *produces* relationship. Lastly, the produced measure which is modeled as an outcome in Fig. 9 is consistent with the *Measure* term in Fig. 4, which is stereotyped in turn with «Outcome» from the process conceptual base. As a consequence, a more robust checking is achieved since the M&E process is in compliance with the enriched M&E ontology.

On the other hand, we observe in Fig. 9 that the *Measurement* task *has assigned* a metric as a resource. This is consistent with the augmented M&E ontology since the *Measurement* term (and its specializations, either direct or indirect measurement) is related to the *Metric* term (and so its specializations, either direct or indirect metric) by the *hasAssigned* relationship. Ultimately, in Fig. 4 the *Metric* term has also the semantic of the *Method* term, while a *Method* is a *Resource* for the task, regarding the process representation in Fig. 1. In the end, the *hasAssigned* relationship in Fig. 9 is semantically consistent with that in Figures 1 and 4.

After executing A3 for the case study, 25% was the outcome produced in the *Measurement* task for the “Stored Cross-Site Scripting Immunity” attribute. This value (data) was obtained using the *Ratio of Stored Cross Site Scripting* (%SXXS) indirect metric. Its Metric Specification (Fig. 8), particularly the formula field, indicates that the derived measure value is calculated from gathering the data of both related direct metrics –resulting in #VPD_v = 3 and #PD_v = 12 respectively. So, as a preliminary analysis we observe that 3 well-identified persistent-data variables for the “XYZ Web system” are vulnerable to this kind of attack. (Note that we could use the #VPD_v direct metric alone instead of the %SXXS indirect metric for quantifying 1.2.1.2, but an indirect metric –as ratio- conveys a bit more information).

Following the GOCAME workflow of Fig. 3, concurrently with A3, the *Design the Evaluation for Security* (A4) activity can be performed. An agent in the role of *Indicators Expert* selects from the Indicators repository an indicator for each element of the *Requirements Tree Specification* artifact (Fig. 7). Then, A4 produces the *Selected Indicators Specification* document, which is composed by a set of *Elementary Indicator Specifications* for attributes, and a set of *Derived Indicator Specifications* for characteristics. Fig. 10 shows the *Elementary Indicator Specification* artifact for the “Stored Cross-Site Scripting Immunity” (1.2.1.2) attribute.

Likewise a *Metric*, in Fig. 4 an *Elementary Indicator* has the semantic of *Method*. In turn, a *Method* specifies a *procedure* and *rules* as shown in Fig. 1. Hence, the *Elementary Indicator Specification* artifact in Fig. 10 includes an *Elementary Model* which should be calculated using a calculation procedure, and decision criteria in a given indicator scale, as rules.

After A4, the A5 activity named *Implement the Evaluation for Security* is performed. One A5 sub-activity is *Calculate Elementary Indicators*, which implies executing, iteratively, the *Elementary Evaluation* task for each attribute’s measure. Fig. 11 shows that each *Elementary Evaluation* task execution consumes an attribute’s measure, from *Measures* datastore, and produces an indicator’s value, which is stored in the Indicator’s values datastore. In order to perform the *Elementary Evaluation*, the *Indicator Calculator* must follow the calculation procedure and rules described in the assigned *Elementary Indicator*. Note that each *Elementary Indicator* was previously added to the *Selected Indicators Specification* artifact in the A4 activity.

Since the M&E process is in compliance with the enriched M&E ontology, the consistency of the process specification can be checked by verifying the diagram in Fig. 11 with the augmented C-INCAMI conceptual base in Fig. 4. For instance, the *Elementary Evaluation* task consumes an attribute’s measure and produces an indicator’s value. This specification is consistent semantically when verified against the C-INCAMI evaluation component, since the *Elementary Evaluation* term -enriched with the «Task» stereotype- is associated to the *Measure* term with the *consumes* relationship, and to the *Indicator Value* term with the *produces* relationship. Moreover, the produced indicator’s value which is modeled as an outcome is consistent with the *Indicator Value* term in Fig. 4, which in turn is enriched with the «Outcome» stereotype. Recall that outcome is defined as “an intangible, storable and processable Work Product” in Table 1. Therefore an indicator’s value can be stored in the *Indicator’s values* datastore and can be used as a processable item for the A6 (Analyze and Recommend) activity.

Additionally, Fig. 11 shows that the *Elementary Evaluation* task *has assigned* an elementary indicator. This is consistent with the augmented M&E conceptual base since the *Elementary Evaluation* term is related to *Elementary Indicator* by the *hasAssigned* relationship in Fig. 4. Furthermore, an *Elementary Indicator* has the semantic of a *Method*, which in turn is a *Resource* (as shown in Fig. 1).

Elementary Indicator Specification

Elementary Indicator Name: *Performance Level of the Stored Cross-Site Scripting Immunity (P_SXSS)*
Author: Alexander Dieser
Version: 1.0
Attribute: Stored Cross-Site Scripting Immunity (1.2.1.2)

Calculation Procedure: Elementary Model:

$$P_SXSS = \begin{cases} 100, & \text{if } \%SXSS = 0 \\ 90, & \text{if } \%SXSS < \%SXSS_{MAX} \\ 0, & \text{if } \%SXSS \geq \%SXSS_{MAX} \end{cases}$$

where %SXSS is the indirect metric of Fig. 4. Note that the %SXSS_{MAX} is an agreed parameter (threshold) of 2% for the XYZ case study.

Decision Criterion:

0 ≤ P_SXSS ≤ 80:	Unsatisfactory	●
80 < P_SXSS ≤ 98:	Marginal	●
98 < P_SXSS ≤ 100:	Satisfactory	●

where:
 Unsatisfactory means: indicates change actions must be taken with high priority.
 Marginal means: indicates a need for improvement actions.
 Satisfactory means: indicates no need for current actions.

Scale Type: Ratio **Representation:** Continuous **Value Type:** Real
Unit (Name): Percentage **Acronym:** %

Figure 10: *Elementary Indicator Specification* artifact for the “Stored Cross-Site Scripting Immunity” attribute

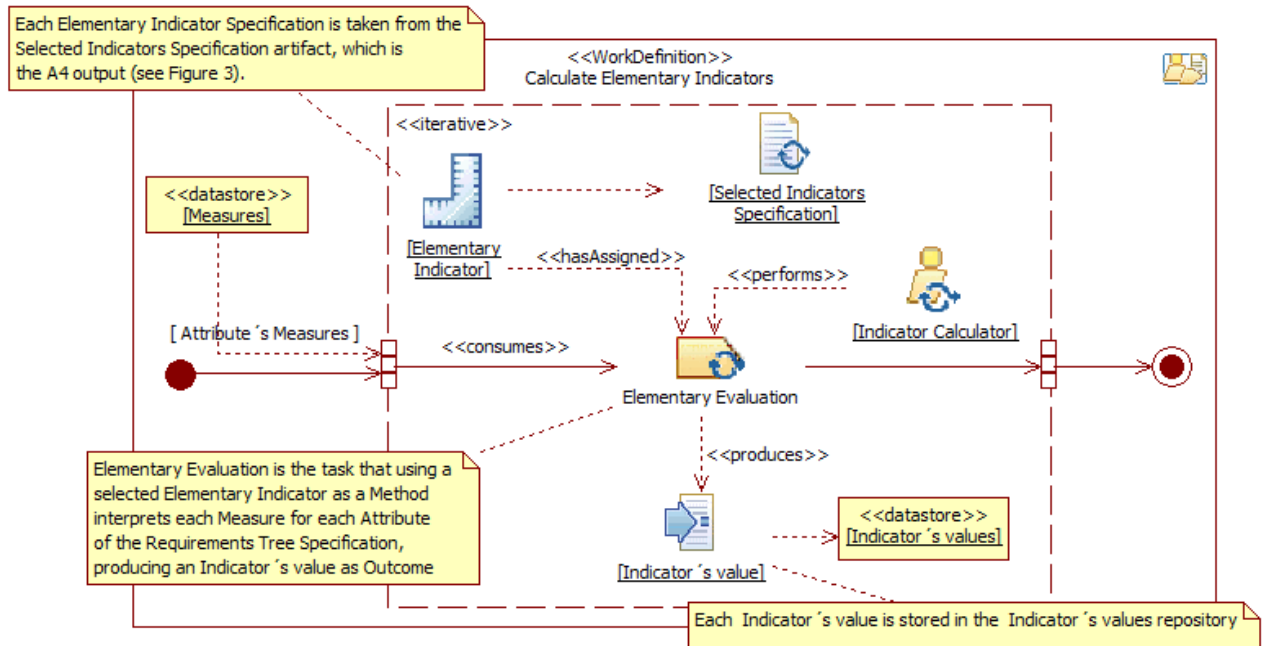


Figure 11: Functional and behavioral views for the *Calculate Elementary Indicators* activity

Reassuming the example, the derived measured value for the 1.2.1.2 attribute should be accordingly interpreted by applying the elementary indicator specified in Fig. 10. That is, using its P_SXSS elementary model, the 25% value maps to the 0% indicator value, as shown in the second column of Table 5. This outcome falls at the bottom of the red acceptability level meaning that a change action must be taken urgently –with high priority- for this attribute, because represents an actual security vulnerability.

It is worthy to underline that the enriched M&E ontology also impacts on the process specification in another positive way, since the stated axioms can be used to define pre- and post-conditions for tasks in a formal way (see the Condition term in Fig. 1).

For example, we define the following axiom for the Elementary Evaluation task:

$$(\forall iv, a, ee) \text{ produces}(ee, iv) \wedge \text{ interprets}(iv, a) \leftrightarrow (\exists e, mnt, m, ei) \text{ associated}(a, e) \wedge \text{ consumes}(mnt, a) \wedge \text{ produces}(mnt, m) \wedge \text{ hasAssigned}(ee, ei) \wedge \text{ consumes}(ee, m)$$

Descriptively, it states that “an *Elementary Evaluation (ee)* produces an *Indicator Value (iv)* that interprets an *Attribute (a)* iff the ‘a’ attribute is associated to a concrete *Entity (e)* and a *Measurement (mnt)* consumes the ‘a’ attribute and produces a *Measure (m)*, and the ‘ee’ consumes the ‘m’ measure and also has assigned an *Elementary Indicator (ei)*”.

Taking into account the above pre-condition for our running example, the elementary evaluation task that produces an indicator value that interprets the 1.2.1.2 attribute can be performed since the 1.2.1.2 attribute is associated to the “XYZ register system” entity, and a performed measurement (in the A3 activity) consuming the 1.2.1.2 attribute produced the 25% measure, so that the elementary evaluation consumes this (25%) measure and also has assigned the P_SXSS elementary indicator (in the A4 activity).

Therefore, once all elementary evaluations were performed for interpreting the 18 attributes shown in Fig. 7, derived evaluations are then performed in order to understand the satisfaction level reached for each characteristic in the requirements tree. A derived evaluation task, unlike an elementary evaluation task, consumes an indicator value and produces other indicator value. The derived evaluations yielded values shown partially in 3th column of Table 5.

Finally, the A6 (*Analyze and Recommend*) activity has as inputs the measures' and indicators' values (i.e. data), the non-functional requirements specification document and the associated metrics' and indicators' specifications (i.e., metadata) in order to produce the Conclusion/Recommendation report. Note that a thorough analysis of results and recommendations for this Security case study were made in [7]. Instead, in this section, we have emphasized the impact that the enriched M&E domain ontology has on the uniformity and verifiability on process views' and methods' specifications.

Although we do not analyze here the results for Security, in the sequel, we give an abridged explanation of how the enriched M&E ontology allows us performing more consistent and comparable analysis for inter- and intra-organizational projects.

Table 5: Indicator values in [%] for the Security sub-characteristics and attributes (a fragment is shown). EI stands for Elementary Indicator; DI stands for Derived Indicator

Characteristic and Attribute	EI	DI
1. Security		17.67 ●
1.1. Confidentiality		100.00 ●
⋮		
1.2. Integrity		10.54 ●
⋮		
<i>1.2.1.1. Reflected Cross-Site Scripting Immunity</i>	100.00 ●	
<i>1.2.1.2. Stored Cross-Site Scripting Immunity</i>	0.00 ●	
⋮		
1.3. Authenticity		12.47 ●
⋮		

4.4. Strengthening the Consistency and Comparability in Analysis of Results among M&E Projects

We have shown in Figures 8 and 10 specifications of indirect and direct metrics, and an elementary indicator for a Security attribute, which can be seen as reusable resources taken from Metrics and Indicators datastores (see Fig. 3). These metric and indicator specifications include metadata that must be kept linked –for the sake of analysis comparability and consistency- to measures (i.e. data) and indicator values (i.e. information), as shown in Table 5.

Let's suppose as proof of concept that the same "Stored Cross-Site Scripting Immunity" (1.2.1.2) attribute can be quantified by two metrics. As shown in Fig. 4, an attribute can be quantified for many metrics, but just one must be selected for each concrete M&E project. So one direct metric (DM1) in the Metrics repository is the "Number of Vulnerable Persistent-Data variables" (#VPDv) as specified in Fig. 8. The other metric (DM2) is one which has different measurement procedure and scale type. That is, DM2 considers the criticality of existing vulnerable persistent-data variables as procedure and a categorical scale, particularly, an ordinal scale type with values ranging from 0 to 3, where 3 represents the higher criticality (catastrophic category), and 0 the lower.

After executing many M&E projects using the same "Security" (sub-)characteristics and attributes, all collected/calculated data are recorded in the Measure repository. Thus, for quantifying the same 1.2.1.2 attribute in some projects were used DM1 and in others DM2.

Consequently, if metric metadata of the recorded data were not linked appropriately, e.g. to the measured value of 3 which can stem from both metrics in different projects, the A6 activity will produce inconsistent analysis if takes as inputs all these related projects. The inconsistency is due to the 3 value, depending on the used metric, has different scale properties recalling that each scale type determines the choice of suitable mathematical operations and statistics techniques that can be used to analyze data and datasets. In summary, even if the attribute is the same, both metric measures are not comparable.

On the other hand, as we commented previously, a metric can be considered a resource, i.e., a method that is assigned to a measurement task, or an artifact, which is the resulting work product after performing a reviewing and cataloging process. This versionable artifact can be stored in the Metric datastore. In this sense, each metric specification must have a unique metricID field and the version in order to keep traceability, repeatability and comparable accuracy in different analysis.

Let's suppose two metrics have the same ID since both share most of the same metadata –i.e. the same measurement procedure, scale, scale type, unit, etc., but differ each other just in the tool that automate the same measurement procedure specification. Our advice is that for assuring comparable accuracy, both metrics should share the same ID but differ likely in the metric version (e.g. v1.0 and v1.1). However, in other cases with more meaningful metadata variations, for guaranteeing repeatability and accuracy, the version number must be different. Furthermore, metrics in many cases are different, i.e. must not share the same ID, although they are intended to quantify the same attribute. This is so for the abovementioned DM1 and DM2, which are different metrics.

5 Concluding Remarks and Future Work

Regarding the strategy definition given in the Introduction Section, a M&E strategy can be seen therefore as a set of principles, patterns, and M&E domain concepts and framework that may be specified by a set of concrete M&E processes, in addition to a set of appropriate methods and tools as core resources for achieving a M&E project goal.

In a previous research, we have developed the integrated GOCAME strategy which relies on three pillars or capabilities viz. the M&E conceptual framework (so-called C-INCAMI), the specifications of M&E process views, and the specifications of M&E methods. Aimed at enriching semantically the M&E conceptual framework with process generic terms, different literature for the process and M&E area were analyzed in Section 2. This review showed that so far there is no broad and unanimous consensus on all key process terms and their meaning. Also we analyzed that the enrichment of a M&E domain ontology with a generic process ontology has very often been neglected. Considering this concern, in Section 3, a process conceptual base structured in a generic ontology was specified. While developing the definitions of its terms, attributes and relationships, state-of-the-art contributions in the process area such as SPEM, CMMI, ISO 12207, among others, have been considered.

Specifically, we have listed in the Introduction Section as first contribution that *“the development of a process generic ontology can be seen as a reusable artifact for enriching semantically a domain ontology. Moreover, we argue that the built generic ontology for process can be used to enrich semantically not only the M&E domain but also other particular domains such as software development, software maintenance, etc.”*. The former part of this statement has actually been shown in sub-section 4.2, in which the augmented C-INCAMI conceptual framework was depicted and the enriched terms for the new M&E ontology version were documented. Additionally, the relationships among M&E terms have been adapted and aligned accordingly considering the semantic of relationships for process terms. On the other hand, the latter part of the previous statement was not actually shown in this paper. But, as the reader can surmise, the process (activity, task), resource (method, tool, agent) and work product (outcome, artifact, service) terms -among others- can be reused in different particular domains. Let us think for a while about a strategy for software change and improvement to be applied in maintenance projects. The conceptual framework for this strategy could be based on maintenance domain ontologies [42, 44]. Hence, particular maintenance terms related to activities, methods and techniques for change can be enriched with the proposed process generic ontology. Note that in this paper, we have certainly emphasized the process generic ontology and its applicability to the M&E domain terminology rather than the ontology construction process itself (as we did in [9]).

A second aspect we have analyzed in sub-section 4.2 is *“the use of stereotypes as a particular mechanism for the semantic linking between the generic ontology and domain ontologies”*. Regarding the procedural way to enrich the M&E domain concepts with process generic concepts, we have argued that stereotypes are, at least in this context, a more suitable mechanism than inheritance relationships, since it generates a lower-coupling level between the M&E components and the process component. For example, in the context of a M&E process, a metric must be considered an specialization of the method term (from the process generic ontology standpoint), while in the context of a metric reviewing and cataloging process, the same metric term must be considered an specialization of the work product term but not considered an specialization of the method term. Hence, in some cases, defining a term as a specialization (inheritance relationships) of generic terms can minimize the reuse of a domain ontology in addition to promoting high-coupling level between the M&E and the process components. Additionally, stereotypes can reduce the model complexity, favoring the understandability and communicability as well.

With regard to the third contribution listed in the Introduction Section, we have stated that *“the M&E concepts enriched with the process concepts help to build better specifications for the GOCAME strategy facilitating therefore their verifiability”*. Particularly, in sub-section 4.3, we have analyzed and illustrated the practical impact on the verifiability and consistency of process and method specifications -the other two GOCAME capabilities-, by using excerpts of a previously developed ICT security and risk evaluation case study. Moreover, this research contributes to enhance the process compliance of GOCAME, since its M&E process and method specifications adhere to the process generic terminological base.

Lastly, we have indicated in the last item of the contributions that *“as a consequence of the semantic enhancement of the GOCAME conceptual base and its impact on method specifications, different enacted M&E projects can support more consistent, comparable and robust analysis of data and metadata”*. In sub-section 4.4, we have discussed about the main reasons for which the specifications of metrics and indicators need to include metadata that must be kept appropriately linked to measures' and indicators' values (i.e. data sets) among different executed M&E projects, for the sake of analysis comparability and consistency. On the other hand, since the metrics' and indicators' metadata are derived from the enriched M&E domain ontology, specific tools can be developed and used to automate semantic inferences for different analysis purposes.

In [6], a comparative study between GOCAME and GQM*Strategies [3] for evaluating the quality of the integrated strategy capabilities was developed. Following this work, as an on-going study, we are comparing the enhanced GOCAME version with its previous version, from the process compliance standpoint. This will allow us to gauge the achieved improvement gain due to the M&E terminological enrichment with the process generic terms.

Finally, as a future line of research, we envision the designing of a set of strategy patterns for different measurement, evaluation and change (ME&C) situations. M&E and ME&C strategy patterns can be seen as general reusable solutions to commonly occurring problems/goals within given measurement, evaluation and

change/improvement situations for specific projects. Strategy patterns can be represented easily as descriptions or templates for how to solve a set of problems that can be used in many different situations.

For example, the GOCAME strategy can be described as a M&E strategy pattern, which gives solutions for the recurrent problem of understanding the current situation of a quality focus/entity category. In sub-section 4.3, GOCAME was instantiated for understanding the current situation of “Security” (quality focus) of a “Web Application” (entity category). But this M&E strategy pattern could be instantiated for evaluating also the quality of a resource, the quality of a process, the quality of a service, or the cost of a product, among many others. Another recurrent problem can be not only to understand the current situation of an entity but also to understand its ulterior situation after improvement. This is therefore other situation, and requires other pattern like the ME&C strategy pattern. Moreover, M&E or ME&C strategy patterns can be applied for concrete projects to address different information needs at different organizational levels.

Acknowledgements

This paper is a substantial extension made of that published in [12] considering the kind invitation of CIbSE 2014 chairs to resubmit an extended manuscript. In addition, we thank the support given by Science and Technology Agency of Argentina, in the POIRe 2013-10 project at Universidad Nacional de La Pampa.

References

- [1] CMMI Product Team: CMMI for Development. Accessed by 6 Nov., 2014, SEI, Carnegie Mellon University: <http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=9661>, Ver. 1.3, CMU/SEI-2010-TR-033, (2010)
- [2] PMBOK: A Guide to the Project Management Body of Knowledge, Fifth Edition, Project Management Institute, USA, ISBN: 978-1-935589-67-9, (2013)
- [3] Basili, V.R., Lindvall, M., Regardie, M., Seaman, C., Heidrich, J., Jurgen, M., Rombach, D., Trendowicz, A.: Linking Software Development and Business Strategy through Measurement. *IEEE Computer*, 43(4), pp. 57–65, (2010)
- [4] Olsina, L., Papa, F., Molina, H.: How to Measure and Evaluate Web Applications in a Consistent Way. In Springer HCIS book *Web Engineering: Modeling and Implementing Web Applications*; Rossi, Pastor, Schwabe, and Olsina (Eds.), pp. 385-420, (2008)
- [5] Kitchenham, B.A., Hughes, R.T., Linkman, S.G.: Modeling Software Measurement Data. *IEEE Transactions on Software Engineering*, 27(9), pp. 788-804, (2001)
- [6] Papa, F.: Toward the Improvement of an M&E Strategy from a Comparative Study. In LNCS 7703, Springer: *Current Trends in Web Engineering, ICWE Int’l Workshops*, M. Grossniklauss and M. Wimmer (Eds.), pp. 189-203, (2012)
- [7] Becker, P., Lew, P., Olsina, L.: Specifying Process Views for a Measurement, Evaluation and Improvement Strategy. *Advances in Software Engineering, Software Quality Assurance Methodologies and Techniques*, Vol. 2012, pp. 1-27, (2012)
- [8] Olsina L., Lew P., Dieser A., Rivera B.: Using Web Quality Models and a Strategy for Purpose-Oriented Evaluations, *Journal of Web Engineering*, Rinton Press, US, 10 (4), pp. 316-352, (2011)
- [9] Olsina, L., Martin, M.: Ontology for Software Metrics and Indicators. *Journal of Web Engineering*, Rinton Press, US, 2(4), pp. 262-281, (2004)
- [10] Curtis, B., Kellner, M., Over, J.: Process Modelling. *Com. of ACM*, 35 (9), pp. 75-90, (1992)
- [11] Becker, P., Papa, F., Olsina, L.: Enhancing the Conceptual Framework Capability for a Measurement and Evaluation Strategy, LNCS 8295. Springer, Q.Z. Sheng and J. Kjeldskov (Eds.): *ICWE 2013 Workshops*, Aalborg, Denmark, pp. 104–116, (2013)
- [12] Becker, P., Papa, M.F., Olsina, L.: Process Conceptual Base for Enriching a Measurement and Evaluation Ontology. In: *CIbSE’14, 17th Iberoamerican Conference on Software Engineering*. ISBN: 978-956-236-247-4, pp. 53-66, Pucón, Chile, (2014)
- [13] Olsina, L., Covella, G., Dieser, A.: Metrics and Indicators as Key Organizational Assets for ICT Security Assessment. Chapter 2, In: *Emerging Trends in ICT Security*, Elsevier (Morgan Kaufmann), 1st Edition, Akhgar & Arabnia (Eds.), pp. 25-44, ISBN: 9780124114746, (2013)

- [14] van Heijst, G., Schreiber, A. Th., Wielinga, B. J.: Using Explicit Ontologies in KBS Development. *International Journal of Human-Computer Studies*, Vol 46, pp. 183-292, Academic Press, Inc. Duluth, MN, USA, (1997)
- [15] Ruiz, F., and Hilera, J.R.: Using Ontologies in Software Engineering and Technology. Chapter 2, In: *Ontologies in Software Engineering and Software Technology*, Calero, C., Ruiz, F., Piattini, M. (Eds). Springer Berlin Heidelberg, pp.49-102, (2006)
- [16] Feiler, P.H., Humphrey, W.S.: Software Process Development and Enactment: Concepts and Definitions. *Int'l Conference of Software Process (ICSP)*. Berlin, Germany: IEEE Computer Society, pp. 28-40, (1993)
- [17] Conradi, R., Fernström, C., Fuggetta, A.: A Conceptual Framework for Evolving Software Processes. *SIGSOFT Softw. Eng. Notes*, 18 (4), 26-35, (1993)
- [18] Lonchamp, J.: A Structured Conceptual and Terminological Framework for Software Process Engineering. *International Conference on the Software Process (ICSP)*. Berlin, Germany: IEEE Computer Society Press. pp. 41-53, (1993)
- [19] ISO/IEC 12207: Systems and software engineering - Software life cycle processes, (2008)
- [20] Acuña, S., De Antonio, A., Ferré, X., López, M., Maté, L.: The Software Process: Modeling, Evaluation and Improvement. In S. K. Chang, *Handbook of Software Engineering and Knowledge Engineering*. World Scientific Publishing Company, Vol.1, pp. 193-237, (2001)
- [21] Esteban, N., Olsina, L.: Hacia un Catálogo de Actividades para el Desarrollo de Sitios y Aplicaciones Web. En *Proceedings del VI Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software (IDEAS)*, (2003)
- [22] OMG-SPEM: Software & Systems Process Engineering Meta-Model Specification V2.0, (2008)
- [23] OPEN Process Framework. Retrieved 07/19/2014, from <http://www.opfro.org/>
- [24] Binguente, A., Falbo, R., Guizzardi, G.: Using a Foundational Ontology for Reengineering a Software Process Ontology. *Journal of Information and Data Management*, Vol. 2, pp. 511-526, (2011)
- [25] Guizzardi, G., Falbo, R., Guizzardi, R.: Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. En *Proceedings de la XI Conferencia Iberoamericana de Software Engineering (CIbSE 2008)*, pp. 127-140, (2008)
- [26] García, F., Bertoa, F., Calero, C., Vallecillo, A., Ruiz, F., Piattini, M., Genero, M.: Towards a consistent terminology for software measurement, *Information & Software Technology*, 48(8), pp. 631-644, (2006)
- [27] Goethert, W., Fisher, M.: Deriving Enterprise-Based Measures Using the Balanced Scorecard and Goal-Driven Measurement Techniques. *Software Engineering Measurement and Analysis Initiative*, CMU/SEI-2003-TN-024, (2003)
- [28] Olsina L. and Rossi G. Measuring Web Application Quality with WebQEM. *IEEE Multimedia*, 9 (4), pp. 20-29, (2002)
- [29] García, F., Ruiz, F., Bertoa, M.F., Calero, C., Genero, M., Olsina, L., Martín, M., Quer, C., Condori, N., Abrahao, S., Vallecillo, A., Piattini, M.: An Ontology for Software Measurement. University of Castilla-La Mancha, Spain, TR. UCLM DIAB-04-02-2, (2004)
- [30] García, F., Piattini, M., Ruiz, F., Canfora, G., and Visaggio, C. A.. FMESP: Framework for the modeling and evaluation of software processes. *Journal of Systems Architecture*, pp. 627-639 52(11), (2006).
- [31] Barcellos M.P., Falbo R., Dal Moro R.: A Well-Founded Software Measurement Ontology. *Proceedings of the Sixth International Conference FOIS 2010*, A. Galton and R. Mizoguchi (Eds.). IOS Press, Amsterdam, The Netherlands, pp. 213-226, (2010)
- [32] Moro, R., Falbo, R.: Uma Ontologia para o Domínio de Qualidade de Software com Foco em Produtos e Processos de Software. In *Proc. of the 3rd Workshop on Ontologies and Metamodels for Software and Data Engineering (WOMSDE'08)*, Campinas, Brazil, pp. 37-48, (2008)
- [33] ISO/IEC 15939: Software Engineering - Software Measurement Process, (2002)
- [34] ISO/IEC 25000: Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) -

Guide to SQuaRE, (2005)

- [35] ISO/IEC 14598-5: IT - Software product evaluation - Part 5: Process for evaluators, (1998)
- [36] Molina, H.; Rossi G., Olsina, L.: Context-Based Recommendation Approach for Measurement and Evaluation Projects, In: Journal of Software Engineering and Applications (JSEA), Irvine, USA, Vol. 3 N. 12, pp. 1089-1106, ISSN Print 1945-3116, (2010)
- [37] Gruber, T.R.: A Translation Approach to Portable Ontologies. Knowledge Acquisition, 5(2): pp. 199–220, 1993.
- [38] Uschold, M.: Knowledge Level Modelling: Concepts and Terminology. The Knowledge Engineering Review, 13 (1): pp. 5-29, (1998)
- [39] Corcho, O., Fernández-López, M., Gómez-Pérez, A.: Methodologies, tools and languages for building ontologies. Where is their meeting point? Data & Knowledge Engineering 46(1), 41–64, (2003)
- [40] Pisanelli, D.M., Gangemi, A., Steve, G.: Ontologies and Information Systems: the Marriage of the Century? New Trends in Software Methodologies, Tools and Techniques, pp. 125-133, (2002)
- [41] OMG-UML: Unified Modeling Language: Superstructure, v2.3, (2010)
- [42] Kitchenham B., Travassos G., Von Mayrhauser A., Niessink F., Schneidewind N., Singer J., Takada S., Vehvilainen R., and Yang H.: Towards an Ontology of Software Maintenance. Journal of Software Maintenance: Research and Practice, John Wiley & Sons, Ltd, Vol. 11, pp. 365–389, (1999)
- [43] Diván, M., Olsina, L.: Process View for a Data Stream Processing Strategy based on Measurement Metadata, In: SADIO Electronic Journal of Informatics and Operations Research (EJS), Special Issue, June 2014, Díaz Pace, J.A., and Colla P. (Eds.), 13(1), pp. 1-19, ISSN 1514-6774, (2014)
- [44] Anquetil, N., de Oliveira, K. M., and Dias, M. G.: Software Maintenance Ontology. Chapter 5, In: Ontologies in Software Engineering and Software Technology, Calero, C., Ruiz, F., Piattini, M. (Eds). Springer Berlin Heidelberg, pp. 153-173, (2006)