Using a Reference Domain Ontology for Developing a Software Measurement Strategy for High Maturity Organizations

Monalessa P. Barcellos[1,2], Ricardo de A. Falbo[1]
[1]Department of Computer Science,
Federal University of Espírito Santo
Vitoria, Brazil
{monalessa, falbo}@inf.ufes.br

Ana Regina C. da Rocha[2]
[2]COPPE
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
darocha@cos.ufrj.br

Abstract — **Software measurement is a key process for software process improvement (SPI). Measurement provides useful information for organizations making decisions that impact their business performance. At high maturity levels, such as CMMI levels 4 and 5, SPI involves carrying out statistical process control (SPC), which requires measures and data suitable for this context. However, measurement problems are pointed as one of the main obstacles for a successful implementation of SPC in SPI efforts. With this scenario in mind, we proposed a strategy to help software organizations prepare themselves regarding measurement aspects in order to implement SPC. The strategy is made of three components. One of them is a reference software measurement ontology. In this paper, we discuss how this ontology helped us to develop the other components and how the use of these components aided to evaluate the conceptualization defined by the ontology.**

Keywords – Software Measurement; Software Measurement Ontology; High Maturity; Domain Ontology Application.

I. Introduction

Nowadays, software measurement is recognized as a crucial process for software project management and software process improvement. Measurement provides useful data for organizations to analyze their performance, guiding both decision-making and daily activities. Successful organizations use measurement as part of their day-to-day activities [1]. Depending on its maturity level, a software organization can perform measurement in different ways. At initial maturity levels, such as CMMI (Capability Maturity Model Integration) levels 2 and 3, the focus is on supporting management information needs [2]. At high maturity levels, such as CMMI levels 4 and 5, measurement should be performed for the purpose of statistical process control (SPC), in order to understand the process behavior and to support process improvement efforts [2, 3].

Taking into account this scenario, we started to work on a strategy to support software organizations to obtain and maintain measurement repositories suitable for SPC, as well as to perform measurements appropriately in this context. Initially, the strategy would be composed by two components: (i) an *Instrument for Evaluating the Suitability of a Measurement Repository for SPC* (IESMR), which goal is to evaluate existing measurement repositories and to determine their suitability for SPC, identifying corrective actions, when necessary; and (ii) a *Body of Recommendations for Software Measurement* (BRSM), which aims to provide guidelines regarding how to perform measurement suitable for SPC.

When we started to work on the strategy, we noticed that we needed a common vocabulary about software measurement to be used. Unfortunately, in spite of the fact that there are some standards and proposals devoted specifically to measurement, such as ISO/IEC 15939 [4] and PSM [1], the vocabulary used by them, and as a consequence by software organizations, is diverse [5]. Very often, the same concept is designated by different terms in different proposals. Other times, the same term refers to different concepts. Besides, the proposals do not address measurement high maturity aspects satisfactorily. As a consequence, some concepts we needed could not be found in these proposals.

In order to deal with these problems, we added a third component to the strategy: the *Reference Software Measurement Ontology* (RSMO), which aims to capture the conceptualization involved in this domain, including traditional and high maturity aspects of software measurement. RSMO is a domain reference ontology. As a domain ontology, it can be used for human communication, providing knowledge and promoting common understanding. As a reference ontology, it is developed with the sole objective of making the best possible description of the domain in reality, with regard to a certain level of granularity and viewpoint [6].

By adding the RSMO to the strategy, we established a synergy between its components. In one hand, RSMO provided the common vocabulary and knowledge used in the other two components and improved some aspects of their definition. On the other hand, the use of the IESMR and the BRSM in real situations served as an additional means of evaluation of the RSMO.

The purpose of this paper is to discuss the synergy between the components. Our focus here is to show how the use of the RSMO helped us developing the strategy, and to discuss how applying the strategy in real cases served as an additional evaluation of the RSMO.

This paper is organized as follows. Section II talks briefly about software measurement. Section III presents an overview of the strategy. Section IV discusses the use of RSMO for defining the other components of the strategy and the synergy between them. Section V discusses the evaluation of the components. Section VI discusses some related work. Finally, Section VII presents our conclusions and future work.

II. Software Measurement

Software measurement is a primary tool for managing software life cycle activities, assessing the feasibility of project plans, and monitoring the adherence of project activities to those plans. It is also a key discipline for evaluating the quality of software products and the capability of organizational software processes. The software measurement process includes, among others, the following activities: planning the measurement process, performing the measurement process, and evaluating the measurement process [4].

For performing software measurement, initially, an organization must plan it. Based on its goals, the organization has to define which entities (processes, products and so on) are to be considered for software measurement and which of their properties (size, cost, time, etc.) are to be measured. The organization has also to define which measures are to be used to quantify those elements. For each measure, an operational definition should be specified, indicating, among others, how the measure must be collected and analyzed. Once planned, measurement can start. Measurement execution involves collecting data for the defined measures, according to their operational definitions. Once data are collected, they should be analyzed, also following the guidelines established by the corresponding operational definitions. Finally, the measurement process and its products should be evaluated in order to identify potential improvements.

As said before, software measurement is performed in different ways according to the organization's maturity level. At initial levels traditional measurement occurs, consisting basically in collecting data from projects and comparing them with their planned values. At high maturity levels, traditional measurement is not enough. It is necessary to carry out statistical process control in order to know the processes behavior, determine their performance in previous executions, and predict their performance in current and future projects, verifying if they are able to achieve the established goals [2].

SPC uses a set of statistical techniques to determine if a process is under control, considering the statistical point of view. A process is under control if its behavior is stable, i.e., if their variations are within the expected limits, calculated from historical data. The behavior of a process is described by data collected for performance measures defined to the process.

A process under control is a stable process and, as such, has repeatable behavior. Consequently, it is possible to predict its performance in future executions and, thus, to prepare achievable plans and to improve the process continuously. On the other hand, a process that varies beyond the expected limits is an unstable process and the causes of these variations (said special causes) must be investigated and addressed by improvement actions, in order to stabilize the process. Once the processes are stable, their levels of variation can be established and sustained, being possible to predict their results. Thus, it is also possible to identify the processes that are capable of achieving the established goals and the processes that are failing in meeting the goals. In this case, actions to change the process in order to make it capable should be carried out [3].

III. A Strategy For Software Measurement Aiming at Statiscal Process Control

The strategy is made up of three components: the *Reference Software Measurement Ontology (RSMO),* the *Instrument for Evaluating the Suitability of a Measurement Repository for SPC (IESMR)*, and the *Body of Recommendations for Software Measurement (BRSM).* The strategy target are software organizations that are interested in implementing SPC. These organizations are generally in one of the following scenarios: (i) organizations that have already achieved an initial maturity level or (ii) organizations that are starting a SPI program. In the first case, organizations want to use measures and data previously collected along the initial levels in SPC. In the second case, there are organizations that, although are starting a SPI program, intend to build a measurement repository and perform measurement suitable for SPC since the initial levels.

Organizations that have already a measurement repository can use, as a reactive approach, the IESMR in order to evaluate and adapt, when possible, their measurement repositories for SPC. On the other hand, as a proactive approach, organizations that are starting SPI programs can use the knowledge provided by RSMO and the recommendations provided by BRSM for building a measurement repository, elaborating a measurement plan and carrying out measurements suitable for SPC.

Since an organization starts doing SPC, new data will be collected and, sometimes, new measures will be defined. Therefore, the strategy can be used continuously, aiming to maintain the suitability of the measurement repository for SPC. In other words, organizations can continuously use RSMO and BRSM as a source of knowledge for defining new measures and carrying out measurements. They can also use IESMR as a support for evaluating the measurement repository, when it changes.

Following, the components of the strategy are presented. Since the purpose of this paper is discuss the use of RSMO and the synergy between the strategy's components, details about each component are not discussed here.

A. The Reference Software Measurement Ontology

RSMO was developed based on the vocabulary used in several standards (such as CMMI [2], ISO/IEC 15939 [4] PSM [1], and IEEE Std. 1061 [7]) and on specific requirements of software measurement at high maturity levels, which were identified in a study based on systematic review of the literature. Besides, since ideally reference domain ontologies should be built based on foundational ontologies [6], RSMO was built based on the Unified Foundational Ontology (UFO) [8, 9]. In this paper we do not discuss the use of UFO for building RSMO, so the concepts from UFO used in the RSMO are not shown. Discussions regarding the use of UFO in the RSMO development can be found in [10, 11, 12].

For developing RSMO, we used SABiO (Systematic Approach for Building Ontologies) [13]. This method has been used for the last ten years in the development of a number of domain ontologies in areas ranging from Harbor Management to Software Process to Electrocardiogram domain. SABiO prescribes an iterative process comprising the following activities: (i) *purpose identification and requirement specification* that concerns to clearly identify the ontology purpose and its intended uses, i.e., the competence of the ontology by means of competency questions; (ii) *ontology capture*, when relevant concepts, relations, properties and constraints should be identified and organized; (iii) *ontology formalization*, which comprises the definition of formal axioms in First-Order Logic; (iv) *integration of existing ontologies*, which involves searching for existing ontologies that can be reused; (v) *ontology evaluation*, for identifying inconsistencies as well as verifying the truthfulness with the ontology purpose; and (vi) *ontology documentation*.

RSMO is divided into six sub-ontologies. The *Measurable Entities & Measures* sub-ontology is the core of the RSMO. It treats the entities that can be submitted to measurement, their properties that can be measured, and the measures used to measure them. The *Measurement Goals* sub-ontology deals with the alignment of measurement to organizational goals. The *Operational Definition of Measures* sub-ontology addresses the detailed definition of operational aspects of measures, including data collection and analysis. The *Software Measurement* sub-ontology refers to the measurement per se, i.e., collecting and storing data for measures. The *Measurement Results* sub-ontology deals with the analysis of the collected data for getting information to support decision making. Finally, the *Software Process Behavior* sub-ontology refers to applying the measurement results in the analysis of the behavior of the organizational software processes. Figures 1 and 2 show fragments of RSMO regarding basic concepts. Figure 3 shows a fragment of RSMO including some concepts regarding measurement at high maturity levels. In order not to visually pollute the models, some relations were omitted.

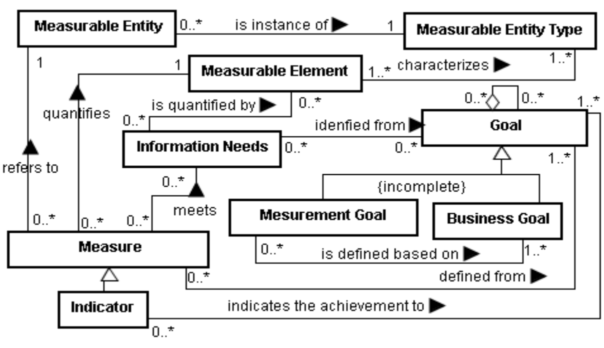

Figure 1. A RSMO Fragment (1).

In the Figure 1, a *Measurable Entity* is anything that can be measured, such as processes, artifacts, projects and resources. Measurable entities can be classified according to types (*Measurable Entity Type*). For instance, process is a type of measurable entity. A measurable entity type indicates which measurable elements can be used to measure entities of this type. *Measurable Element* is a property of a measurable entity type through which measurable entities of this type can be described (e.g., size, cost). Measurable elements are quantified by measures. A *Measure* is an instrument that is used to associate a value to a measurable element, regarding to a measurable entity. Organizations carry out measurement to achieve their goals and/or to meet their information needs. A *Goal* expresses the intention for which actions are planned and performed. In the context of software measurement, a goal can be, among others, a *Business Goal* (e.g., "increase 10% the number of clients") or a *Measurement Goal* (e. g., "monitor the critical processes behavior"). *Information Needs* are identified from goals and they are met by measures. An organization with the goal of "improving the adherence to projects plans" can, for example, take as information needs "get to know the requirements stability after their approval by the client". This information need could be attended by the measure "requirements changing rate". Measures can be used in order to indicate the achievement of goals. In this case, measure plays the role of an *indicator.* Considering the example cited above, if the measure "requirements changing rate" is used for monitoring the achievement of the goal "to improve the adherence to projects plans", then, in this context, it is an *indicator*.

During the development of the RSMO, we identified several constraints which could not be captured by the diagrams. Therefore, we defined axioms to make them explicit. For instance, we defined an axiom to make explicit that if a measurable entity *men* is an instance of the measurable entity type *t* and *t* is characterized by the measurable element *mel*, then *mel* also characterizes *men.*

$$(\forall men \in MeasurableEntity, t \in MeasurableEntityType, mel \in MeasurableElement)\ (isInstanceOf(men, t) \wedge characterizes(mel, t) \rightarrow characterizes(mel, men)) \quad \text{(A1)}$$

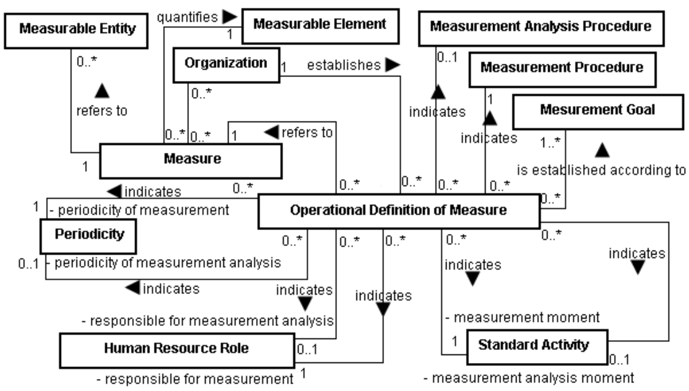

Figure 2. A RSMO Fragment (2).

In Figure 2, an *Operational Definition of Measure* (ODM) details some aspects related to the collection and analysis of a *Measure* in an *Organization*. An organization establishes ODMs taking into account *Measurement Goals*. An ODM should indicate: *(i)* the procedures to be followed in order to guide data collection and analysis (*Measurement Procedure* and *Measurement Analysis Procedure*, respectively); *(ii)* the moment when measurement should occur (*measurement moment*) and the moment when collected data for a measure should be analyzed (*analysis moment*). These moments are established in terms of the activities (*Standard Activity*) of the software process during which measurement and analysis should occur (e.g., Requirements Specification Approval); *(iii)* the, frequency with which measurement (*measurement periodicity*) and measurement analysis (*analysis periodicity*) should be performed (e.g. monthly, weekly, in each occurrence of the activity designated as measurement moment); *(iv)* the organizational role (*Human Resource Role*) responsible for performing the measurement (*responsible for measurement*) (e.g., requirements engineer), and the responsible for analyzing the collected data for the measure (*responsible for measurement analysis*).

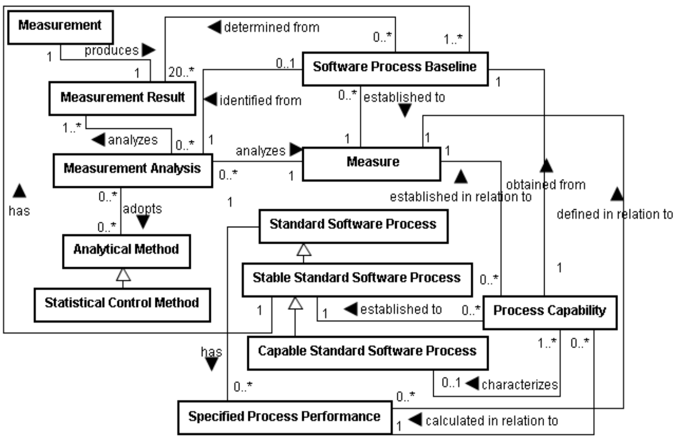

Figure 3. A RSMO Fragment (3).

In Figure 3, *Measurement* is an action performed to measure a Measurable Element of a Measurable Entity by applying a Measure, obtaining a *Measurement Result*. Measurement results are analyzed during *Measurement Analysis,* which adopts *Analytical Methods*. In a Measurement Analysis that adopts a *Statistical Control Method*, it is possible to identify a *Process Performance Baseline*, established in relation to a Measure for a *Stable Standard Software Process*. A Stable Standard Software Process is a Standard Software Process with stable behavior, i.e., it is a Standard Software Process that has at least one Process Performance Baseline.

A *Process Performance Baseline* is identified from twenty or more Measurement Results. It is the range of results achieved by a Stable Standard Software Process, obtained from measured values of a particular Measure. A *Specified Process Performance,* in turn, is the range of values that describes the desired results of a Standard Software Process, considering a particular Measure.

Process Capability characterizes the ability of a Stable Standard Software Process to achieve the Process Performance Specified for it, considering a particular Measure. Process Capability is obtained from a Process Performance Baseline and it is calculated in relation to a Specified Process Performance. When the Process Capability reveals that the process is capable of achieving the expected performance, we have a *Capable Standard Software Process*.

B. The Instrument for Evaluating the Suitability of a Measurement Repository for SPC

The IESMR is composed by four checklists that are used for evaluating the following items: measurement plan, measurement repository structure, measures and collected data. In each checklist, there is a set of requirements that must be satisfied in order to use the corresponding item in SPC. These requirements were identified from a study based on systematic review of the literature and they were refined considering the results of three practical experiences with previous versions of the IESMR. These practical experiences allowed us to evolve the IESMR until its current version. Information regarding the development and evolution of IESMR is presented in [14].

The evaluation of an item against each requirement present on the checklists can produce one of the following results: *(i) Satisfied (S),* which means that the item satisfies totally the requirement and no corrective action is necessary; *(ii) Largely Satisfied (LS), Reasonably Satisfied (RS) or Precariously Satisfied (PS),* which means that the item does not completely satisfy the requirement, but it is possible to take actions to adapt it in order to satisfy the requirement and, consequently, to allow the use of the evaluated item in SPC. The level of satisfaction (*largely, reasonably or precariously*) is related to the effort required to perform the actions (the more effort is necessary, the lower the satisfaction level will be); and *(iii) Dissatisfied (D)*, meaning that the item does not satisfy the requirement and there are no possible actions to adapt it for being used in SPC.

When the result of the evaluation of a requirement is *Largely Satisfied*, *Reasonably Satisfied* or *Precariously Satisfied*, *Actions for Suitability* are suggested. These actions are guidelines for correcting the item so that it can be used in SPC. When the result is *Dissatisfied,* recommendations of the Body of Recommendations for Software Measurement can be used to rebuild the item. Figure 4 shows an overview of the IESMR.

The IESMR content includes, in addition to the checklists, detailed requirements descriptions, guidelines for evaluating each requirement according to the possible results of evaluation, and actions for suitability. The results of an evaluation are recorded in a document called Evaluation Diagnosis. It includes the detailed evaluation of each item, the actions for suitability suggested, and the degree of suitability of the measurement repository for SPC. The degree of suitability is informed as a percentage and calculated using Fuzzy Logic (see [14]).

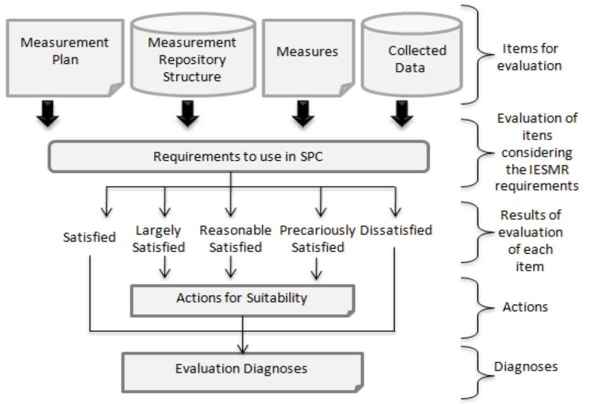

Figure 4. Overview of the IESMR.

A fragment of the checklist used to evaluate a measure is shown in the Table 1. Each requirement has a description. For instance, the description of the requirement "*The measure has appropriate granularity level*" is: "*The granularity level of a measure must allow daily monitoring of projects. For this, the measure must be related to short activities or processes, which usually are performed several times during a project*".

Besides, in order to guide the evaluation, for each requirement, we provided a description of what a possible answer means. For instance, concerning the requirement cited above, some of the guidelines for evaluation provided are: (i) *Satisfied*: The granularity level of the measure allows analyzing its collected data and identifying problems along the execution of a macro-activity or a sub-process; (ii) *Largely Satisfied*: The operational definition of the measure establishes an inappropriate granularity level (e.g., the measure is associated to a long process), but there are data stored with appropriate granularity (e.g., data collected from activities of that process). Little effort is required to make satisfactory the granularity level of this measure.

As discussed earlier, if the result of an evaluation of a requirement is *Largely Satisfied, Reasonably Satisfied or Precariously Satisfied,* actions for suitability are suggested, aiming to support the organization to change the evaluated item to fulfill the requirement. So, for each requirement, we identified potential problems and actions for suitability. For instance, for the requirement "*The measure has appropriate granularity level*", one of the potential problems and actions for suitability identified are: *Problem I*: The operational definition of the measure establishes an inappropriate granularity level, but there are data stored with satisfactory granularity level. *Actions for Suitability:* (a) Correct the operational definition of the measure, setting an appropriate granularity level. (b) Record the new operational definition of the measure in the Measurement Plan. If necessary, a new measure should be created. (c) Record the new operational definition of the measure (and the new measure, if it is the case) in the measurement repository and associate it to the measure and the recorded data.

TABLE 1. FRAGMENT OF THE CHECKLIST FOR EVALUATING MEASURES

Evaluated Item: Measure **Measure Evaluated:**________________					
Requirements	**Evaluation**				
RM1. The measure is well defined and it is possible to know its name, measurable entity type, measurable element, measure unity and scale.	S	LS	RS	PS	D
RM2. The operational definition of the measure is correct and satisfactory. The operational definition of the measure includes: measurement procedure, measurement moment, measurement periodicity, responsible for measurement, measurement analysis procedure, measurement analysis moment, measurement analysis periodicity, and responsible for measurement analysis.	S	LS	RS	PS	D
RM3. The measure is aligned to business goals.	S	LS	RS	PS	D
The measure is associated to:					
RM3.1 Organization goals.	S	LS	RS	PS	D
RM3.2 Project goals.	S	LS	RS	PS	D
RM4. The measure is related to the performance of a process.	S				D
RM5. The measure is related to a critical process.	S				D
RM6. The related measures are defined.	S	LS	RS	PS	D
RM7. The related measures are valid.	S	LS	RS	PS	D
RM8. The measure has appropriate granularity level.	S	LS	RS	PS	D
RM9. It is possible to normalize the measure (if applicable).	S	LS	RS	PS	D
RM10. The measure is correctly normalized (if applicable).	S	LS	RS	PS	D
RM11. The data grouping criteria for measurement analysis are defined.	S	LS	RS	PS	D
RM12. The measure does not consider aggregated data.	S	LS	RS	PS	D

C. The Body of Recommendations for Software Measurement Suitable for SPC

Although there are models and standards devoted specifically to address measurement, they do not satisfactorily address how to carry out measurement for SPC. Thus, aiming to complement our strategy with a practical guide to organizations carry out software measurement suitable for SPC, we defined the BRSM.

BRSM is composed by recommendations related to eighteen aspects organized in five groups. For each aspect, we defined a set of recommendations. The BRSM groups are (Table 2): *(i) Software Measurement Preparation*, which contains recommendations related to aspects that should be treated before starting the measurement; *(ii) Alignment between Software Measurement and Organizational Goals*, which contains recommendations for carrying out measurement aligned with organizational business goals and projects goals; *(iii) Software Measures Definition,* which contains recommendations for correctly elaborating operational definitions of measures; *(iv) Software Measurement Execution,* which contains recommendations for appropriately collecting and storing data for the measures defined; and *(v) Software Measurement Analysis*, which contains recommendations for analyzing the data collected, aiming to meet the information needs previously identified.

TABLE 2. BRSM GROUPS AND ASPECTS

Groups	Aspects
Software Measurement Preparation	Measurement Repository Creation Project Characterization Identification of Similar Projects Identification of Processes Versions
Alignment between Software Measurement and Organizational Goals	Identification of Measurement Goals Identification of Information Needs according to Measurement Goals Identification of Measures to attend Information Needs and according to its use
Software Measures Definition	Operational Definition of a Measure Identification of Granularity Level of a Measure Normalization of a Measure Data Grouping Criteria for a Measure
Software Measurement Execution	Execution of Consistent Measurements Data Collected Validation Recording of Measurement Context
Software Measurement Analysis	Measurement Analysis Periodicity Data Grouping for Analysis Identification of Process Performance Baseline Determination of Processes Capability

Although BRSM has recommendations that must be used aiming at SPC implementation, there are recommendations that deal with measurement at initial maturity levels. This occurs because, as discussed earlier and pointed out by results of the study based on systematic review, most problems related to measurement at high maturity levels usually had origin at initial levels. Frequently, these problems have less impact on measurement results at initial maturity levels so, only when organizations start SPC practices, they realize these problems.

Table 3 shows, as an example, some recommendations of the BRSM regarding the aspect "*Granularity Level of a Measure*".

TABLE 3. FRAGMENT OF THE BRSM

Group: Software Measure Definition	**Aspect:** Granularity Level of a Measure

Purpose: Guide about the granularity level required to a measure according its use.

A Brief Theoretical Foundation: The granularity level of a measure influences mainly on the amount of data collected during a project. It is determined by two aspects of the operational definition of a measure: the entity associated with the measure (measurable entity type) and the measurement periodicity. If a measure is collected once in each occurrence of the entity, measures related to smaller entities, such as components of project or product (e.g., modules, artifacts, activities or tasks) have finer granularity than measures associated with larger entities, such as project. However, a measure is not necessarily collected once in each occurrence of the entity. The measurement periodicity determines the collection frequency, influencing directly the number of values collected. It is possible that a measure associated with an entity that normally would have ticker granularity has its granularity level reduced because of the measurement periodicity. For example, the measure number of errors reported by the client can be associated with the measurable entity type "Project" and have periodicity "once a week". This leads to collecting and recording many values throughout project, rather than a single value at the end of the project.

Recommendations:

R1. Define the granularity level according to the measure use. Measures used in traditional monitoring and control usually require coarser granularity level than measures used in process performance analysis, because the last one uses SPC techniques and so requires granularity level for monitoring processes and projects daily.

Example: the measure "rate of defects detected in a project", used to support traditional monitoring and controlling and whose analysis consists of comparing rates of defect of projects, may have ticker granularity. Thus:

Measure	*Rate of defects detected in a project*
Use	*Traditional Monitoring and Controlling*
Measurable Entity Type	*Project*
Measurement Periodicity	*Once a project*

On the other hand, the measure "rate of defects detected in an inspection", used to support performance analysis of the Inspection process and whose analysis includes statistical control techniques, should have finer granularity. Thus:

Measure	*Rate of defects detected in an Inspection*
Use	*Process Performance Analysis*
Measurable Entity Type	*Inspection Process*
Measurement Periodicity	*Once by Inspection (considering that the Inspection process is performed several times in a same project)*

R2. Try to select for submitting to statistical process control critical processes that are carried out several times throughout projects. This way, measures associated with them will be collected several times during a project.

IV. DEVELOPING THE COMPONENTS OF THE STRATEGY: EXPLORING THE SYNERGY BETWEEN THEM

The components of the strategy are related to each other in both the context of the strategy development and its application. Regarding the strategy application, RSMO provides organizations with the knowledge needed for constructing measurement repositories and performing measurement appropriately at initial and high maturity levels. In addition, the knowledge provided by the RSMO can also be useful for better understanding the requirements, guidelines and actions for suitability present in the IESMR, as well as the BRSM recommendations. Concerning the strategy development, the IESMR and the BRSM were defined using the conceptualization (and consequently the vocabulary) provided by the RSMO. Figure 5 shows an integrated view of the strategy components. Black lines represent the relations between components in the context of the strategy development. Gray lines represent how the components are used when the strategy is applied.

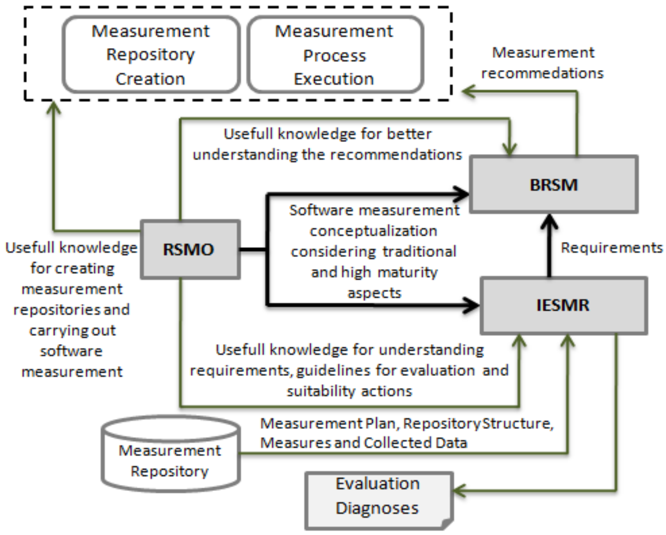

Figure 5 - Integrated view of the strategy components.

The use of RSMO mainly helped us to: *(a)* refine the set of requirements used as basis to the IESMR; *(b)* harmonize terms used in different sources; *(c)* integrate the strategy components; and *(d)* identify recommendations that are not explicit in the measurement proposals. Next, discussions regarding each use of RMSO are presented.

A. Using the RSMO to refine the IESMR requirements

The IRSM requirements were identified preliminary from a systematic review of the literature. However, many times, they were not described satisfactorily. For instance, some of them contained some tacit knowledge. Therefore, for each requirement, we performed an analysis taking the conceptualization provided by RSMO into account, in order to solve ambiguities and to adjust the IESMR to the RSMO conceptualization.

As an example, from the systematic review of the literature, we identified the following requirement: *"Measurement is aligned with the organization's goals and provides information needed to monitor them"*. Using the RSMO conceptualization, we analyzed this requirement and we noticed that its first part (*measurement is aligned with the organization's goals)* is addressed in the RSMO by the concepts of *Measurement Goal* and *Business Goal,* and the relation *defined based on* between them. The second part of the requirement (*measurement provides information needed to monitor the organization's goals*) is addressed in the RSMO by the concepts of *Measurement Goal* and *Information Need,* and the relation *identifies* between them. In fact, in the RSMO the relation *identifies* occurs between *Goal* and *Information Need*, but since *Measurement Goal* is a subtype of *Goal,* that relation also exists between *Measurement Goal* and *Information Need*. The concepts of *Measure* and *Information Need,* and the relation *meets* between them also address the second part of this requirement. Thus, the initial requirement was refined into three requirements, which are in the IESMR checklist for evaluating the measurement plan: *(RMP1) The measurement goals are appropriately defined based on business goals*; *(RMP2) The information needs for monitoring the measurement goals are appropriately identified and related to them*; and *(RMP3) The measures that are able to meet the information needs are identified and appropriately related to them*.

In addition to the use of the RSMO conceptualization to refine requirements, the RSMO axioms were particularly useful to establish guidelines for evaluating some requirements. An example is the requirement "*The measure is well defined and it is possible to know its name, measurable entity type, measurable element, measure unity and scale*" (RM1 in Table 1). According to RSMO, *Measures quantify Measurable Elements*, *Measurable Elements characterize Measurable Entity Types,* and *Measurable Entities are instances of Measurable Entity Types*. These relations are constrained by axiom A1, which relates measurable entities to measurable elements. A1 was useful to establish the following guideline: "*Check the consistency between the measurable entity type and the measurable element (can the measurable element be used to characterize the measurable entity type?)*".

B. Using the RSMO to harmonize terms used in different sources

As discussed earlier, software measurement proposals, models and standards found in the literature were important sources to define the IESMR requirements and BSMR recommendations. However, many times different standards and models use different terms to designate the same concept. In this context, RSMO was used to harmonize terms used in the components of the strategy.

With respect to BSRM recommendations, guidelines established in quality models and standards were grouped according to the BRSM aspects. Next, the RSMO terms were used to harmonize the diverse vocabulary used in these guidelines, in order for us to write the recommendations. For instance, with respect to goals identification, the following guidelines were found: *(a) Establish and maintain measurement objectives that are derived from objectives* (CMMI) [2]; *(b) Quantitative objectives in support of relevant business goals are established* (ISO/IEC 15504) [15]. Analyzing these guidelines, we noticed that the terms *measurement objectives (a)* and *quantitative objectives (b)*, although different, had the same meaning. In the RSMO, they correspond to *Measurement Goal.* The terms *objectives (a)* and *business goals (b*), in turn, correspond to *Business Goal* in the RSMO. In both guidelines *(a)* and *(b)* there is a relation between these terms. In *(a)*, measurement objectives *are derived from* objectives. In *(b)*, quantitative objectives *are established in support of* business goals. As the terms *measurement objectives* and *quantitative objectives* are equivalent to *Measurement Goal* in the RSMO, and the terms *objectives* and *business goals objectives* are equivalent to *Business Goal* in the RSMO, the relations *are derived from (a)* and *are established in support of (b)* are equivalent to *is defined based on* in the RSMO. Thus, the recommendation resulting from *(a)* and *(b),* using the RSMO vocabulary is: *(B1) Identify measurement goals based on the business goals recorded in the organization's strategic planning*.

C. Using the RSMO to integrate the other two strategy components

Although it is possible to use each component of the strategy separately, the use of the strategy as a whole can provide more advantages than the use of each component singly. The IESMR and the BRSM could be used in an integrated way. IESMR can be used to evaluate a measurement repository and BSMR can be used to help organizations perform measurement appropriate to SPC. In other words, IERSM guides the correction of items (measurement plan, measurement repository structure, measures and data) that are not suitable for SPC (i.e., what can be done to fix an item), and BRSM guides the development of new items suitable for SPC. Thus, BRSM can be used when a requirement is not satisfied, making the item to be discarded. The integration of the IERMS and BRSM is mainly made by the RSMO. RSMO allows that requirements and recommendations addressing the same concepts be identified and used in an integrated way.

During the strategy development, in order to ensure homogeneity and integration between the components, we created tables associating RSMO concepts to BSRM aspects/recommendations and IESMR requirements. Table 4 shows an example concerning the granularity level of measures.

TABLE 4. EXAMPLE OF COMPONENTS INTEGRATION

BRSM ASPECT	IERSM REQUIREMENT	RSMO CONCEPTS
Identification of Granularity Level of a Measure	RM8. The measure has appropriate granularity level.	All concepts from the Measurable Entities &Measures and Operational Definition of Measures sub-ontologies.

As another example of component integration, in the IESMR checklist for evaluating measures, there is a requirement that deals with operational definition of a measure (RM2 in Table 1): "*The operational definition of the measure includes: measurement procedure, measurement moment, measurement periodicity, responsible for measurement, measurement analysis procedure, measurement analysis moment, measurement analysis periodicity, and responsible for measurement analysis*". In the BRSM, in turn, there is a specific recommendation that deals with the same aspects addressed by requirements RM1 and RM2. Since this recommendation is quite long, here we described only a small part of it: *(B2) The operational definition of a measure must include: (a) the Responsible for Measurement, which is the role fulfilled by the human resource responsible for collecting the measure. The responsible for measurement should be the one that collects the data in a measurement. Examples: systems analyst, programmer, project manager, etc; (b) the Measurement Moment, that is, the moment in which measurement must be performed. The measurement moment must be an activity of the process defined for the project or an activity of an organizational process. Examples: at the "Agreement of the Project Requirements Specification" activity; at the "Perform Unity Tests" activity.*

RM1 concerns aspects inherent to the measure itself. RM2 deals with aspects of the measure collection and analysis. At high maturity levels, as aid before, data are used for SPC. In this context, the quality of the operational definitions is even more important, because it is necessary to get a certain volume of data (greater than the volume required at initial levels) and to form homogeneous data groups [16]. This requires data to be collected in a consistent way, and measurement consistency is directly related to the quality of the operational definitions.

B2, RM1 e RM2 are based on the same elements of the RSMO (Measurable Entity Type, Measurable Entity, Measurable Element, Measure, Measurement Unit, Scale, Operational Definition of Measure and other concepts shown in Figure 2 that are related to Operational Definition of Measure by *indicates* relations), and they can be used in a complementary way for dealing with the same issue. RM1 and RM2 can be used to evaluate and correct existing measures and B2 can be used to help organizations to define new measures.

D. Using the RSMO to identify recommendations

The RSMO conceptualization captures a consensual view and includes relations and constraints that organizations interested in software measurement should respect. During the development of the BRSM, for each aspect, we verified if the relations and constraints presented in RSMO regarding that aspect were addressed by the identified recommendations. Relations and constraints not addressed were incorporated to BRSM as new recommendations. An example is the definition of the *measure moment* as an activity of the process defined for the project or an activity of an organizational process, in B2. Other example is *(B3): Decompose measurement goals when they are large. For measurement goals related to process performance analysis, it is recommended to establish measurement goals for each critical process that will be submitted to SPC*. This recommendation is based on the whole-part relation between goals.

V. EVALUATING THE COMPONENTS OF THE STRATEGY

Each component of the strategy was evaluated individually. The RSMO evaluation started with a verification activity, where we checked if the concepts, relations and axioms are able to satisfy the ontology requirements (described by a set of competency questions). Table 5 shows a small fragment of the table used for verification purposes, considering a competency question established for the Measurement Goals sub-ontology.

TABLE 5. EXAMPLE OF ONTOLOGY VERIFICATION

COMPETENCE QUESTION	CONCEPT A	RELATION	CONCEPT B	AXIOMS
Which measurable elements characterize a measurable entity?	Measurable Entity	is instance of	Measurable Entity Type	A1
	Measurable Element	characterizes	Measurable Entity Type	

After that, as a form of validation, we instantiated the RSMO concepts with real elements extracted from measurement repositories of organizations, in order to verify if the ontology was able to represent concrete situations of the real world. Table 6 shows part of an instantiation made to the Measurable Entity & Measure sub-ontology.

TABLE 6. EXAMPLE OF ONTOLOGY VALIDATION

CONCEPT	INSTANCE
Measurable Entity Type	Standard Software Process
Measurable Entity	Requirements Management Process of the organization Org
Measurable Element	Requirements stability

Although the RSMO had been conceptually evaluated, it would be important to evaluate its use in real situations. Since the IESMR and the BRSM were built based on the RSMO, the evaluation of these components also served to evaluate the RSMO.

Considering the IESMR, it was used for evaluating the measurement repositories of three organizations [14]. First, we checked if the identified requirements were correct. We checked if a measure that fulfilled the IESMR requirements could be effectively used in SPC and if a measure that did not fulfill the requirement was really unsuitable for SPC. For this, we plotted the data collected for the measures to be evaluated in control charts. As result, we observed that the measures considered suitable for SPC according to IESMR could be correctly plotted in control charts and they provided useful information about the process performance. By contrast, the measures considered unsuitable by IESMR could not be plotted in control charts or they did not describe the process performance. We also verified if the IESMR content was correct. One of the organizations whose repository was evaluated by IESMR carried out the actions suggested and, after that, the behavior of the critical processes was successfully analyzed using SPC techniques.

Finally, the BRSM was evaluated by experts using peer reviews. The reviewers were appraisers that are able to perform assessments in high maturity levels, since they have knowledge about SPC and practical experience with high maturity organizations. Next, recommendations were used by some organizations in practical situations.

The IESMR and BRSM evaluations served as an indirect evaluation of the RSMO. The common vocabulary proposed by RSMO was recognized and accepted by people involved in the evaluations. Despite of different people profiles (some were experts in SPC, others beginners), they agreed about the vocabulary and the conceptualization adopted.

VI. Related works

Regarding the software measurement domain, there are some initiatives committed with ontology-based modeling and formalization of this domain, such as the one proposed by Martín and Olsina [17] (called here MO ontology) and the one proposed by Bertoa, Vallecillo and García [18] (called by them SMO). These works are focused on the basic aspects of measurement and are very in line with our Measurable Entities & Measures sub-ontology, since they are inspired in some common sources, such as ISO/IEC 15939 [18].

Concerning the similarities, there are many common concepts, such as concepts related to measurable entities (entities in the MO-ontology and SMO), measures (metrics in the MO-ontology, and measures in SMO), measurable elements (attributes in MO-ontology and SMO). However, there are also many differences. For instance, several concepts present in RSMO are not explicit in these ontologies, such as Goal specializations. Although not shown in the models presented in this paper, RSMO distinguishes between types of measurement goals, in order to make explicit the application of a measure according the goal to which it is associated. This information is relevant for defining a measure appropriately. Besides, since those works did not focus on measurement aspects related to high maturity levels, they also did not address many concepts addressed by RSMO, such as process performance baseline, process capability and specified process performance. RSMO addresses high maturity, especially in the Software Process Behavior sub-ontology and part of the Operational Definition of Measures sub-ontology. A detailed discussion about the differences between RSMO and those proposals can be found in [10].

Although the use of UFO as a basis to our ontology is not discussed in this paper, we can state that it was of great value and a distinction point of our work when compared with these other two. It allowed identifying several problems and drove the ontology engineering process, making explicit ontological commitments that were implicit and elucidating conceptual mistakes. Some discussions in this line can be found in [10, 11, 12, 19].

Regarding the use of measurement ontologies as a basis for developing tools and techniques to support software measurement, García and colleagues [25] developed some applications of their software measurement ontology, including its use for defining a data quality model for Web portals. They also discuss the use of their ontology for analyzing differences and similarities between measurement standards. Although these authors make a broad discussion about the use of their ontology, discussions do not include details with respect to the synergy between elements of proposals.

With respect to the evaluation of measures for SPC, Tarhan and Demirors [20, 21] defined measure usefulness to SPC as a requirement for choosing a process to be put under SPC. Nevertheless, since the focus of their approach is the selection of processes to SPC, the approach is limited to evaluating measures. These authors themselves state that the study of the usefulness of measures considering only the attributes defined by their proposal is not enough to select the most appropriate measures to SPC [20].

Concerning the recommendations for software measurement, there are several works addressing software measurement process, such as [1, 4, 22]. However, there is still no consolidated set of guidelines on how to perform measurement in order to reach high maturity levels. As mentioned before, models like CMMI [2] do not provide sufficient guidance in this context. With respect to high maturity, we found few studies that deal with measurement. The initiatives of Dumke and colleagues [23, 24] have particular relevance. However, these works deal more specifically with the application of data in SPC than with the measures definition and data collection aiming at SPC.

Although some related works have been presented in this section, we do not found one that discusses the use of a software measurement ontology as basis to establish and maintain integration between elements of a strategy. Besides, the development of RSMO based on a foundational ontology (UFO) is also a differential of our proposal.

VII. Conclusions And Future Work

In this paper we discussed the use of a reference domain ontology as a basis for defining a strategy for software measurement aiming at SPC. The strategy is composed by three components: RSMO, IESMR, and BRSM. RSMO provided the common vocabulary and the knowledge used for developing the other two components. The use of RSMO contributed for harmoniously integrating the other two components and allowed refining their content. On the other hand, the evaluations of the IESMR and the BRSM served as an additional evaluation of the RSMO.

As pointed by several authors, such as Guizzardi [6], domain reference ontologies are useful to assist humans in tasks such as meaning negotiation and consensus establishment. This is the case of the use of RSMO for developing the strategy presented in this paper. There are other initiatives that also focus on developing ontologies for the Software Measurement domain, such as [18, 25]. However, at the best of our knowledge, none of them used the proposed ontology to establish a strategy for measurement in both initial and high maturity levels.

Regarding the IESMR and the BRSM, we are aware that we need to use them in a larger number of actual situations, in order to make a better evaluation. We gave tutorials regarding the recommendations present in the BRSM, and currently it is being used by three organizations that are starting SPI programs. The initial results of this practical evaluation are being obtained. IESMR is being used for evaluating the measurement repository of an organization that is starting to apply high maturity practices.

Finally, RSMO is being used as a conceptual specification for developing and integrating tools and measurement repositories of the High Maturity Environment at LENS (Software Engineering Laboratory) in COPPE/UFRJ. This environment aims to support software organizations to accomplish process improvement practices, especially at high maturity levels. RSMO is also being used as a reference conceptual model for developing tools to support software measurement in ODE (Ontology-based Development Environment) [26] at NEMO (Ontology & Conceptual Modeling Research Group) in UFES.

ACKNOWLEDGMENTS

This research is funded by the Brazilian Research Funding Agencies FAPES (Process PRONEX Number 52272362/2011) and CNPq (Process Number 483383/2010-4).

REFERENCES

[1] J. McGarry, D. Card, C. Jones, B. Layman, E. Clark, J. Dean, F. Hall, Practical Software Measurement: Objective Information for Decision Makers, Addison-Wesley, Boston USA, 2002.

[2] CMMI Product Team, CMMI for Development Version 1.3, Software Engineering Institute, Pittsburgh USA, 2010.

[3] W.A. Florac, A.D. Carleton, Measuring the Software Process: Statistical Process Control for Software Process Improvement, Addison-Wesley, Boston USA, 1999.

[4] ISO/IEC 15939, Systems and Software Engineering – Measurement Process, 2007.

[5] F. García, M., Bertoa, C. Calero, A, Vallecillo, Ruíz, F., M. Piattini, M. Genero, “Towards a consistent terminology for software measurement”, Information and Software Technology, 48 (8), 631-644, 2006.

[6] G. Guizzardi, On Ontology, ontologies, Conceptualizations, Modeling Languages and (Meta)Models, In O. Vasilecas, J. Edler, A. Caplinskas (Org.). Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV. IOS Press, Amsterdam, 2007.

[7] IEEE (1998). Std 1061 – IEEE Standard for a Software Quality Metrics Methodology.

[8] G. Guizzardi, Ontological Foundations for Structural Conceptual Models. Universal Press, The Netherlands, 2005.

[9] A. C. Bringuente, R. A. Falbo, G. Guizzardi, “Using a foundational ontology for reengineering a software process ontology”, Proceedings of the XXVI Brazilian Symposium on Data Base, 2011.

[10] M. P. Barcellos, R. A. Falbo, R. Dal Moro, “A Well-Founded Software Measurement Ontology”, Proceedings of the 6th International Conference on Formal Ontology in Information Systems, pp. 213-226. IOS Press, Toronto, Canada , 2010.

[11] M. P. Barcellos, R. A. Falbo, A. R., Rocha, “Establishing a Well-founded Conceptualization about Software Measurement in High Maturity Levels”, Proceedings of the 7th International Conference on the Quality of Information and Communications Technology, Oporto, Portugal, 467-472, 2010.

[12] M. P. Barcellos, R. A. Falbo, A. R., Rocha, “A Well-founded Software Process Behavior Ontology to Support Business Goals Monitoring in High Maturity Software Organizations”, Proceedings of the IEEE 5th Joint VORTE-MOST Workshop, Vitória-ES, Brasil, 253-262, 2010.

[13] R. A. Falbo, C. S. Menezes, A. R. Rocha, “A systematic approach for building ontologies”, Proceedings of the 6th Ibero-American Conference on Artificial Intelligence, LNCS, vol. 1484, 1998.

[14] M. P. Barcellos, A. R., Rocha, R. A. Falbo, “Evaluating the Suitability of a Measurement Repository for Statistical Process Control”, Proceedings of the 4th International Symposium on Empirical Software Engineering and Measurement, Bolzano-Bozen, Italy, 1-10, 2010.

[15] ISO/IEC 15504, Information Technology– Process Assessment, 2003.

[16] B. Kitchenhan, C. Kutay, R. Jeffery and C. Connaughton, "Lessons learned from the analysis of large-scale corporate databases", Proceedings of the 28th International Conference on Software Engineering (ICSE’06), Shanghai, China, 439-444, 2006.

[17] M. A. Martin, L. Olsina, “Towards an ontology for software metrics and indicators as the foundation for a cataloging Web system”, Proceedings of the First Latin American Web Congress (LA-WEB'03) - Santiago – Chile, 2003.

[18] M.F. Bertoa, A. Vallecillo, F. García, An Ontology for Software Measurement, In: C. Calero, F. Ruiz, M. Piatini (Eds.), *Ontologies for Software Engineering and Software Technology*, Springer-Verlag Berlin Heidelberg, 2006.

[19] M. P., Barcellos, R. A., Falbo, “Using a Foundational Ontology for Reengineering a Software Enterprise Ontology”, Proceedings of the Joint International Workshop on Metamodels, Ontologies, Semantic Technologies, and Information Systems for the Semantic Web (MOST-ONISW 2009). Lecture Notes in Computer Science, 5833, 179-188, 2009.

[20] A. Tarhan, O. Demirors, “Investigating Suitability of Software Process and Metrics for Statistical Process Control”, Lecture Notes in Computer Science, 4257, 88-99, 2006.

[21] A. Tarhan, O. Demirors, “Assessment of software process and metrics to support quantitative understanding”, Lecture Notes in Computer Science, 4895, 102-113, 2008.

[22] V. R. Basili, H. D. Rombach, G. Caldiera, “Goal Question Metric Paradigm”, Encyclopedia of Software Engineering, 2 Volume Set, John Wiley & Sons, Inc, 1994.

[23] R. Dumke, I. Côté, O. T. Andruschak, Statistical Process Control (SPC) - A Metric-based Point of View of Software Processes Achieving the CMMI Level Four. Technical Report, Dept. of Computer Science, University of Magdeburg, Germany, 2004.

[24] R. Dumke, R. Braugarten, M. Blazey, H.Hegewald, D. Reitz, K. Drichter, Software Process Measurement and Control - A Measurement-based Point of View of Software Processes. Technical Report, Dept. of Computer Science, University of Magdeburg, Germany, 2006.

[25] F. García, F. Ruiz, C. Calero, M. F. Bertoa, A. Vallecillo, B. Mora, M. Piattini, “Effective use of ontologies in software measurement”, The Knowledge Engineering Review, Vol. 24:1, 23–40, 2009.

[26] R. A. Falbo, F. B. Ruy, R. Dalmoro, “Using ontologies to add semantics to a software engineering environment”, Proceedings of the 17th International Conference on Software Engineeringi and Knowledge Engineering, Tapei, China, 151-156, 2005.