

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/221216075>

Establishing a Well-Founded Conceptualization about Software Measurement in High Maturity Levels

Conference Paper · September 2010

DOI: 10.1109/QUATIC.2010.84 · Source: DBLP

CITATIONS

13

READS

43

3 authors, including:



[Monalessa Perini Barcellos](#)

Universidade Federal do Espírito Santo

68 PUBLICATIONS 250 CITATIONS

[SEE PROFILE](#)



[Ricardo de Almeida Falbo](#)

Universidade Federal do Espírito Santo

172 PUBLICATIONS 1,661 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



INTEROPERABILIDADE SEMÂNTICA DE INFORMAÇÕES EM SEGURANÇA PÚBLICA [View project](#)



Standards Harmonization [View project](#)

Establishing a Well-founded Conceptualization about Software Measurement in High Maturity Levels

Monalessa Perini Barcellos^{1,2}, Ricardo de Almeida Falbo¹, Ana Regina Rocha²

¹Department of Computer Science,
Federal University of Espírito Santo
Vitória – Brazil

²COPPE
Federal University of Rio de Janeiro
Rio de Janeiro – Brazil

monalessa@inf.ufes.br, falbo@inf.ufes.br, darocha@cos.ufrj.br

Abstract - Software measurement is a key process for software process improvement. Measurement provides organizations with the objective information they need to make informed decisions that impact their business performance. Nowadays, there are several process quality models and standards that point out the importance of software measurement, such as CMMI. Unfortunately, the vocabulary used by those models concerning software measurement is diverse. This leads to misunderstanding and problems related to the jointly use of different standards. In this paper, we present a fragment of a Software Measurement Ontology (SMO) with focus on measurement at high maturity levels. In order to establish a basic conceptualization regarding this domain, the Unified Foundational Ontology was used to ground SMO.

Keywords - Software Measurement, Software Measurement Ontology, Domain Ontologies, Foundational Ontology.

I. INTRODUCTION

Nowadays, software measurement is recognized as a key process for software project management and software process improvement. Several process quality standards and maturity models, such as ISO/IEC 12207 [1] and CMMI [2], includes software measurement as an essential process for organizations achieve maturity in software development. Moreover, depending on the organization's maturity level, software measurement is performed in different ways. In the initial maturity levels, such as the levels 2 and 3 of CMMI, the focus is on developing and sustaining a measurement capability that is used to support project management information needs [2]. In high maturity levels, such as CMMI levels 4 and 5, measurement is performed aiming the statistical process control, in order to understand the process behavior and to support process improvement efforts [2, 3].

There are also standards and methodologies devoted specifically to assist organizations in defining their software measurement process, such as ISO/IEC 15939 [4] and PSM [5]. These standards provide some definitions for measurement related terms that are commonly used by the software industry. Unfortunately, the vocabulary used by those standards, and as a consequence by the software organizations, is diverse. Many times, the same concept is designated by

different terms in different proposals. Others, the same term refers to different concepts. To deal with these problems, it is important to establish a common conceptualization regarding the software measurement domain. In this context, a domain ontology can be used for human communication, promoting common understanding among knowledge workers [6]. More specifically, in this context, we need a domain reference ontology, i.e., a domain ontology that is constructed 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 [7]. A domain reference ontology is a special kind of conceptual model representing a model of consensus within a community. It is a solution-independent specification with the aim of making a clear and precise description of domain entities for the purposes of communication, learning and problem-solving. Aiming fidelity to reality and conceptual clarity, ideally domain ontologies should be built based on foundational ontologies [8].

In order to deal with the problem of vocabulary diversity in related standards, we developed a Software Measurement Ontology (SMO), which is partially presented in this paper. The focus here is on measurement at high maturity levels, more specifically on operational definitions for the measures. Operational definitions are crucial for measurement in high maturity levels because they address two important aspects: communication, concerning what has been measured, how it was measured, among others; and repeatability, which regards if the measurement can be repeated, given the same definition, to get the same results [2]. This domain reference ontology was built based on the Unified Foundational Ontology (UFO) [9, 10]. Besides, the SMO was developed based on the vocabulary used in several standards and specific requirements of software measurement at high maturity levels. These requirements were identified in a study based on systematic review of the literature.

This paper is organized as follows. Section II discusses software measurement and operational definition of measures. Section III presents the concepts of the Unified Foundational Ontology considered relevant for this paper. Section IV presents an overview of the Software Measurement Ontology and details its fragment that treats operational definitions of measures. Section V discusses related works, and Section VI presents our conclusions and future work.

II. SOFTWARE MEASUREMENT AND OPERATIONAL DEFINITION OF MEASURES

Software Measurement is a primary support process for managing projects. It is also a key discipline in evaluating the quality of software products and the performance and capability of organizational software processes [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) to consider 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.

Operational definitions of measures must be established according to their intended use. E.g., measures used in high maturity levels for analyzing process performance must apply statistical process control techniques. Thus, their operational definitions should include such techniques as procedures for analyzing collected data. This is not the case for measures used in initial maturity levels, where their intended use is to support traditional project monitoring and control.

Measurement repeatability is related to the accuracy and completeness of the operational definitions applied. If an operational definition of measure is imprecise, ambiguous or poorly documented, probably different people will understand the measure in different ways. As a consequence, it is likely that they will collect invalid data, perform incomparable measurements or incorrect analysis, making the measurement inconsistent and inefficient [11].

In high maturity levels, as said before, data are used for statistical process control. In this context, the quality of the operational definitions is even more important [12]. In order to analyze the behavior of its processes, an organization has to get a certain volume of data (greater than the volume required in the initial levels). Moreover, it is necessary to form homogeneous data groups. This requires data to be collected in a consistent way, and measurement consistency is directly related to the quality of the operational definitions.

Software measurement is a relatively young discipline. The terminologies used by different measurement approaches and standards are diverse, and the problem of terminology harmonization still needs to be solved in this domain [13]. Thus, we need to establish a common conceptualization of the software measurement domain, including aspects related to measurement in high maturity levels. In this context, special attention should be given to operational definitions of measures and their main elements.

A domain ontology can be used to establish a common conceptualization about certain domain. Thus, a software measurement ontology, providing a coherent set of concepts, relations and axioms constraining their interpretation, is of great value [14]. Furthermore, as discussed in the introduction of this paper, we are interested in a domain reference ontology, grounded in a foundational ontology. Thus for developing our Software Measurement Ontology, we decided to use the Unified Foundational Ontology (UFO) [9, 10], briefly presented in the next section.

III. THE UNIFIED FOUNDATIONAL ONTOLOGY

UFO is a foundational ontology that has been developed based on a number of theories from Formal Ontology, Philosophical Logics, Philosophy of Language, Linguistics and Cognitive Psychology. It is composed by three main parts. UFO-A is an ontology of *endurants*. A fundamental distinction in UFO-A is between *Particulars (Individuals)* and *Universals (Types)*. Particulars are entities that exist in reality possessing a unique identity, while Universals are patterns of features, which can be realized in a number of different particulars [9]. UFO-B is an ontology of *perdurants* (events). UFO-C is an ontology of social entities (both endurants and perdurants) built on the top of UFO-A and UFO-B. One of its main distinctions is between *agents* and *objects*. Agents are capable of performing actions with some intention, while objects only participate in events [10].

A complete description of UFO falls outside the scope of this paper. However, in the sequel we give a brief explanation of its concepts that are important for this paper. This description is based on [9, 10, 14]. These concepts belong to UFO-A and UFO-C parts. Figure 1 shows a fragment of UFO-A. The concepts that are directly used here are shown detached in grey.

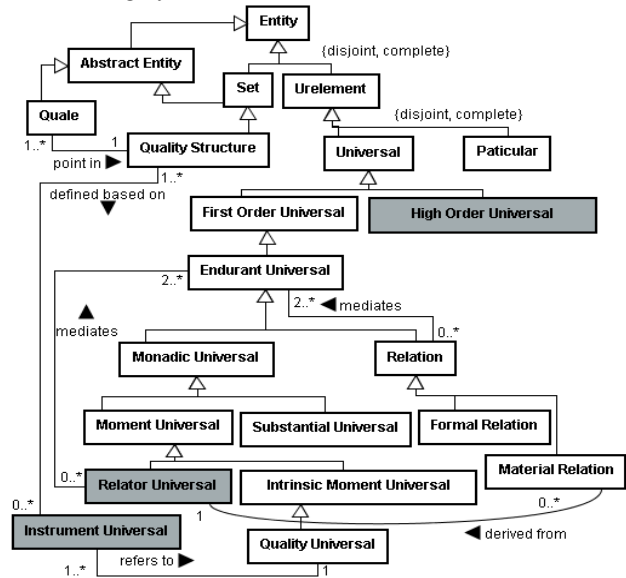


Figure 1. An UFO-A fragment focusing on universals.

An *entity* is something perceivable or conceivable. It is the most general concept in UFO. *Universals* are patterns of features that can be realized in a number of different entities (e.g., Person). *Particulars* are entities that exist in reality, possessing a unique identity (e.g., the person Mary). The model depicted in Figure 1 focus on universals. Universals can be *first order universals*, i.e., universals whose instances are particulars, or *high order universals*, which are universals whose instances are also universals. *Endurant universals* are universals that persist in time maintaining their identity. Endurant universals can be monadic universals or relations. *Monadic universals*, in turn, can be further categorized into substantial universals and moment universals (properties). A *moment*¹ is an endurant that is existentially dependent of another endurant, in the way, for example, that the color of an apple depends on the apple in order to exist. Existential dependence can also be used to differentiate intrinsic and relational moments. *Intrinsic moments* are dependent of one single endurant (e.g., color). *Relators* depend on a plurality of endurants (e.g., an employment) and, for this reason, provide the material connection between these endurants. In other words, we can say that they are the foundation for material relations such as “working at”. Thus, material relations require relators in order to be established. *Formal relations*, in contrast, hold directly between individuals.

A *quality universal* is an intrinsic moment universal that is associated with a *quality structure*. A quality structure can be understood as a measurement structure (or a space of values) in which individual qualities can take their values. A *quale* is a point (a value) in a quality structure. For instance, the quality universal Weight is associated to a space of values that is a liner structure isomorphic to the positive half-line of the real numbers. For the same quality universal, there can be potentially many quality structures associated with it, but a quality structure is always associated with a unique quality universal. An *instrument* is used to associate a quality universal to values (qualia) in a quality structure. For a given quality universal, there can be different quality structures associated with different instruments.

Figure 2 shows a fragment of UFO regarding *substantial universals*. While persisting in time, substantials can instantiate several substantial universals. Some of these types a substantial instantiates necessarily (i.e., in every possible situation) and define what the substantial is. These are the types named *kind* (for general substantials) and *subkind*. Taking into account kinds, an important distinction in UFO is between agents and objects. According to UFO-C, an *object kind* is a non-agentive substantial universal. Its instances (objects) do not act. They can only participate in actions. Object kinds can be categorized into *physical object* (e.g., Book) and *social object* (e.g., Language). A *normative description kind* is a social object kind whose instances define one or more rules/norms recognized by at least one social agent (e.g., a method describing a set of directives on how to

perform some activity within an organization). An *agent kind* is a substantial universal that is capable to perform actions with some intention. Agent kinds can also be further categorized into *physical agent* (e.g., Person) and *social agent* (e.g., Team). *Organization kind* is a specialization of social agent kind

Intentional Moment Universal is a special kind of intrinsic moment universal that are inherent to agents and have a propositional content called *Proposition*. Intentional moments in which the intentionality is “intending something” are called *Intention*. An intention characterizes a situation desired by the agent (e.g., an organization *O* can have the intention “to be successful”). The propositional content of an intention is a *Goal* (e.g., the propositional content of the intention “to be successful” could be “to be among the ten best software organizations of its country”).

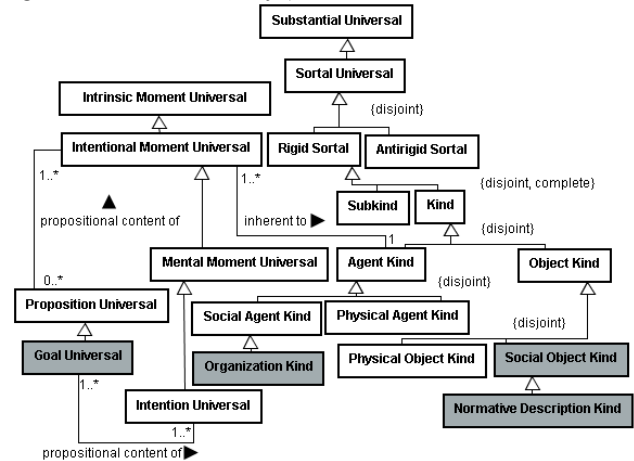


Figure 2. An UFO fragment including concepts from UFO-A and UFO-C.

IV. A SOFTWARE MEASUREMENT ONTOLOGY

For developing the Software Measurement Ontology (SMO), we used the SABiO (Systematic Approach for Building Ontologies) method [15]. SABiO prescribes an iterative process comprising the following activities: (i) *requirement specification* that concerns to identify the competency questions that the ontology should be able to answer; (ii) *ontology capture* that regards capturing relevant concepts, relations and properties, and building conceptual models modeling them; (iii) *ontology formalization*, which comprises identifying constraints and defining axioms in First Order Logic; (iv) *integration of existing ontologies*, which involves searching for existing ontologies to be reused and integrated to the ontology being developed; (v) *ontology evaluation*, for evaluating the ontology to check whether it satisfies its requirements; (vi) *ontology documentation*, which deals with documenting the ontology development process.

Since the software measurement domain is strongly related to the domains of software processes and organizations, we looked up to ontologies in these domains. We decided to use the software process ontology described in [10] that is already grounded in UFO. Concerning the domain of software organizations, we decided to reuse the software organization ontology proposed by Villela et al. [16]. This ontology,

¹ The word moment in UFO-A is derived from the german term *Momente* and it bears no relation to the notion of time instant. It is related to the ways things are.

however, was not developed grounded in a foundational ontology, and thus we had to, first, reengineer it [17].

Since the scope of the SMO is very complex, we applied a decomposition mechanism allowing building the ontology in parts. Thus, SMO was divided into six sub-ontologies. The *Measurable Entities & Measures* sub-ontology is the core of the SMO and it is presented in [14]. 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 handles 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 [14].

Due to space limitation, in this paper we discuss part of the Operational Definition of Measure sub-ontology, presenting some of its competency questions, conceptual models and axioms. Also, its evaluation is briefly discussed.

The Operational Definition of Measures Sub-ontology

An Operational Definition of Measure (ODM) regards defining in details how a measure must be collected and analyzed, taking into account measurement goals. Thus, this sub-ontology should be able to answer the following competency questions:

- QC1. Which are the ODMs of a given measure in an organization?
- QC2. Which are the goals taken into account by an ODM?
- QC3. According to an ODM, when a measure should be measured?
- QC4. According to an ODM, which is the organizational role that is responsible for measuring a measure?
- QC5. According to an ODM, how often a measure should be measured?
- QC6. According to an ODM, when collected data of a measure should be analyzed?
- QC7. According to an ODM, which is the organizational role that is responsible for analyzing the collected data of a measure?
- QC8. According to an ODM, how often the collected data of a measure should be analyzed?
- QC9. According to an ODM, which is the measurement procedure indicated to measure a measure?
- QC10. According to an ODM, which is the measurement analysis procedure indicated to analyze the collected data of a measure?

Figure 4 shows the conceptual model that addresses the competency questions CQ1 to CQ8. The concepts from other ontologies or sub-ontologies are identified as follows: SOO – Software Organization Ontology [16]; SPO – Software

Process Ontology [10]; MEM – Measurable Entities & Measures sub-ontology [14]; and MG – Measurement Goals Sub-ontology. The distinctions made in UFO are shown as stereotypes in the concepts of the SMO, indicating that they are subtypes of concepts of UFO, as defined in [9]. When a concept does not have a stereotype, it means that the concept is of the same type of its super-type.

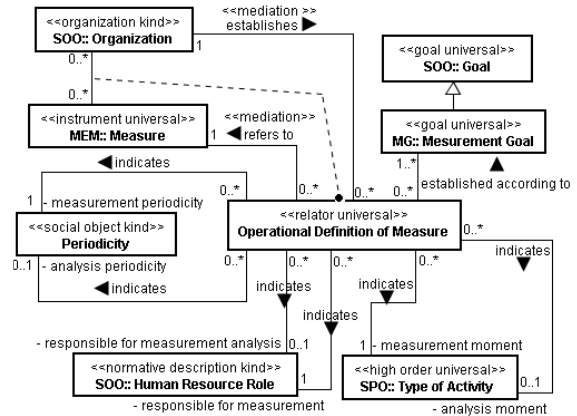


Figure 3. Fragment I of the Operational Definition of Measure sub-ontology.

An *Operational Definition of Measure* (ODM) details some aspects related to the collection and analysis of a *Measure* in an *Organization*. An *Operational Definition of Measure* is existentially dependent of both *Organization* and *Measure*, and it corresponds to a relator in UFO. The relations that take place between a relator and the endurants it mediates are called *mediations* in UFO. Therefore, the relations *refers to* and *establishes* are mediation relationships.

An organization establishes ODMs taking into account *Measurement Goals*. A *Measurement Goal* is related to the intention for which software measurement actions are planned and performed (e. g., monitoring the critical process performance). It is a concept from the Measurement Goals sub-ontology and it is a specialization of *Goal*, a concept from the Software Organization Ontology [16]. The relations between *Measure*, *Organization* and *Measurement Goal* are not presented in Figure 3 because they involve concepts and relations of these other ontologies, which are not addressed in this paper. However, it is worthwhile to point out that ODMs are constrained by the measurement goals of the organization.

An ODM should indicate [11, 12, 18]: (i) the moment when measurement should occur (*measurement moment*). This is established in terms of the activity (*Type of Activity*) of the software process during which measurement should occur (e.g., Requirements Specification Approval); (ii) the *measurement periodicity*, that is, the frequency with which measurement should be performed (e.g. monthly, weekly, in each occurrence of the activity designated as measurement moment); (iii) the organizational role (*Human Resource Role*) responsible for performing the measurement (*responsible for measurement*) (e.g., requirement engineer); (iv) the moment when collected data for the measure should be analyzed (*analysis moment*). Analogously to the measurement moment, analysis moment is established in terms of the activity (*Type*

of Activity) of the software process during which the analysis should occur; (v) the *analysis periodicity*, that is, the frequency with which the measurement analysis should be performed; and (vi) the organizational role responsible for analyzing the collected data for the measure (*responsible for measurement analysis*).

Figure 4 shows the conceptual model that addresses the competency questions CQ9 and CQ10. As shown in this figure, in addition to the elements discussed previously, an ODM should also indicate the *Measurement Procedure* and the *Measurement Analysis Procedure* to be followed in order to guide data collection and analysis, respectively. Both are *Procedures* defining rules and norms recognized by the organization, and thus, they represent a normative description in UFO.

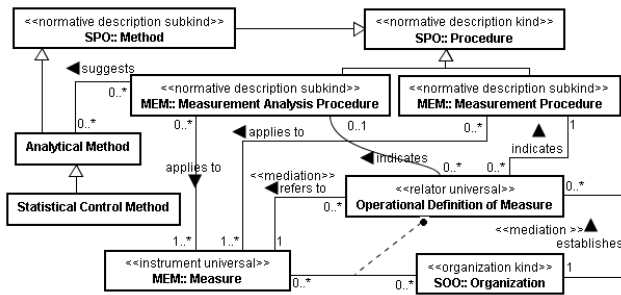


Figure 4. Fragment II of the Operational Definition of Measure sub-ontology.

Measurement and measurement analysis procedures apply to certain measures (*applies to* relationships). Therefore, an ODM can only indicate procedures that apply to the measure it refers to. Several constraints such this one were identified during the development of the SMO. However, they are not captured by the conceptual models. Thus, we defined axioms to make them explicit. The following axiom (A1) captures the constraint described above: if an operational definition of measure *odm* refers to a measure *m* and indicates a measurement procedure *mp*, then *mp* must apply to *m*.

$$(\forall odm \in \text{Operational Definition of Measure}, m \in \text{Measure}, mp \in \text{Measurement Procedure}) (\text{referTo}(odm, m) \wedge \text{indicates}(odm, mp) \rightarrow \text{appliesTo}(mp, m)).$$

Measurement analysis procedures can suggest the use of analytical methods for representing and analyzing the measured values. *Analytical Method* is sub-kind of *Method*, which describes systematic procedures for performing an activity [10]. Histograms and bar charts are examples of analytical methods. Analytical methods that use principles of statistical control to represent and analyze values are said *Statistical Control Methods*. The XmR and mXmR charts [4] are examples of statistical control methods. At high maturity levels, measurement analysis procedures should indicate the use of statistical control methods.

Evaluating the ODM Sub-ontology

For evaluating the SMO ontology as a whole, we adopted two strategies. First, we checked if the ontology was able to answer the competency questions posed to it (verification).

Second, we validated it with domain experts by using them as basis for defining a strategy to support organizations to obtain and maintain measurement repositories suitable for statistical process control (SPC), as well as to perform measurements appropriately in this context. This strategy is composed of three components [19]: the SMO itself, an Instrument for Evaluating the Suitability of a Measurement Repository to SPC, and a Body of Recommendations for Software Measurement. The instrument and the body of recommendations have already been evaluated by experts and used in real cases. The preliminary results point out to its usefulness and also to an agreement of the vocabulary used.

Regarding the ontology verification, we checked it manually, since SMO is a reference ontology and it is not implemented in any computational language. Thus, during ontology verification, we related the concepts, relations and axioms of the SMO to the competency questions answered by them, as well as we used individuals (extracted from measure repositories of organizations) to evaluate if the ontology was actually able to represent concrete situations of the real world. Table 1 shows an example of the evaluation of the ODM sub-ontology, considering the competency question CQ9. Table 2 shows one of the instantiations we performed.

TABLE 1. ONTOLOGY EVALUATION.

QC	Concept A	Relation	Concept B	Axioms
CQ9	Operational Definition of Measure	indicates	Measurement Procedure	A1
	Operational Definition of Measure	refers to	Measure	
	Measurement Procedure	applies to	Measure	

The relation *indicates* is the main responsible for answering QC9. However, the axiom A1 must also hold. A1 establishes a relationship between the measurement procedure indicated in an ODM and the measurement procedures that apply to this measure.

TABLE 2. ONTOLOGY INSTANTIATION.

Concept	Instance
ODM	ODM-01
Measure	Change Requirements Rate
Organization	Org X
Measurement Goal	Monitoring the critical processes performance.
Measurement Procedure	Calculating the change requirements rate in the period. It is the ratio between the number of approved requirements which were changed in the period and the number of approved requirements of the project.
Measurement Periodicity	In each occurrence of the activity designated as measurement moment.
Measurement Moment	Preparing data to monitoring the project.
Responsible for Measurement	Requirement Engineer.
Measurement Analysis Procedure	(i) Plotting the measured values. (ii) Comparing the process performance in the project and the organizational performance process. For this, the measured values must be represented in a control chart whose limits are provided by the process performance baseline. If the measured values are within the control limits (baseline limits), then the process performance is in accordance with expected. Otherwise, corrective actions are needed.

TABLE 2. ONTOLOGY INSTANTIATION (CONT).

Concept	Instance
Statistical Control Method	XmR chart.
Measurement Analysis Periodicity	In each occurrence of the activity designated as measurement analysis moment.
Measurement Analysis Moment	Monitoring the project.
Responsible for Measurement Analysis	Requirement Engineer.

V. RELATED WORKS

Concerning the domain of software measurement, there are some initiatives committed with ontology-based modeling and formalization of this domain. Two of them are the ones described in [13] and [20]. These works are focused on the basic aspects of measurement and are very in line with our Measurable Entities & Measures sub-ontology. A comparison between these proposals and this sub-ontology can be found in [14]. However, these works did not focus on measurement aspects related to high maturity levels and they did not properly address the operational definition of measures. Furthermore, as a rule, such initiatives are not committed to the use of a foundational ontology as their basis, and, consequently, they rely on models of low expressivity.

VI. CONCLUSIONS

Nowadays, there are several process quality standards and maturity models that address software measurement. However, the vocabulary used is diverse and some software measurement aspects are not treated, especially aspects related to software measurement in high maturity levels. Aiming to provide a common vocabulary to the software measurement domain, in several maturity levels, we developed a Software Measurement Ontology (SMO). Since we were interested in a reference domain ontology [7], we developed SMO grounded in the Unified Foundational Ontology [9, 10]. This paper presented a SMO sub-ontology: the Operational Definition of Measure (ODM) sub-ontology.

Although several researchers argue in favor of using a foundational ontology as basis for developing domain ontologies [8, 10], few works have explored this use. This is the case of the software measurement domain, in which the proposed ontologies are, in general, lightweight ontologies. We chose UFO because it has been used to evaluate, re-design and integrate (meta) models of conceptual modeling languages, as well as to evaluate, re-design and give real-world semantics to domain ontologies [10].

Currently, the SMO is been 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 carry out process improvement practices, especially in high maturity levels.

REFERENCES

- [1] ISO/IEC 12207 - Systems and Software Engineering - Software Life Cycle Process, 2008.
- [2] CMMI Product Team, CMMI for Development Version 1.2, Software Engineering Institute, Pittsburgh USA, 2006.
- [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] 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.
- [6] M. Uschold and R. Jasper, "A Framework for understanding and classifying ontology applications", In *Proceedings of IJCAI Workshop on Ontologies and Problem-Solving Methods*, Stockholm, Sweden, p. 1-11, 1999.
- [7] 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.
- [8] N. Guarino, "Formal ontology and information systems", In *Proceedings of International Conference in Formal Ontology and Information Systems*, pp 3-15, 1998.
- [9] G. Guizzardi, *Ontological Foundations for Structural Conceptual Models* Universal Press, The Netherlands, 2005.
- [10] G. Guizzardi, R.A Falbo and R.S.S. Guizzardi, "Grounding software domain ontologies in the Unified Foundational Ontology (UFO): the case of the ODE software process ontology", In *Proceedings of the XI Iberoamerican Workshop on Requirements Engineering and Software Environments*, 244-251, 2008.
- [11] B. Kitchenham, R. T. Hugues and S. G. Linkan, "Modeling software measurement data", *IEEE Transactions on Software Engineering*, v. 27, n. 9, p. 788-804, 2001.
- [12] B. Kitchenham, C. Kutay, R. Jeffery and C. Connaughton, "Lessons learned from the analysis of large-scale corporate databases", In *Proceedings of the 28th International Conference on Software Engineering (ICSE'06)*, Shanghai, China, p. 439-444, 2006.
- [13] 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.
- [14] M. P. Barcellos, R. A. Falbo, R. Dalmoro, "A well-founded software measurement ontology", In *Proceedings of the 6th International Conference on Formal Ontology in Information Systems (FOIS 2010)*, Toronto - Canadá, 2010.
- [15] R.A. Falbo, C.S. Menezes and A.R.C. Rocha, "A systematic approach for building ontologies", In *Proceedings of the 6th Ibero-American Conference on Artificial Intelligence, LNCS*, vol. 1484, 1998.
- [16] K. Villela, A.R. Rocha, G.H. Travassos et al., "The use of an enterprise ontology to support knowledge management in software development environments", *Journal of the Brazilian Computer Society*, Brazil, 11(2): 45-59, 2005.
- [17] M.P. Barcellos, R.A. Falbo, "Using a foundational ontology for reengineering a software enterprise ontology", In *Proceedings of the Joint International Workshop on Metamodels, Ontologies, Semantic Technologies, and Information Systems for the Semantic Web*, 2009.
- [18] B. Rackzinski and B. Curtis, "Software data violate SPC's underlying assumptions", *IEEE Software*, v. 25, n. 3, pp. 49 - 50, 2008.
- [19] M.P. Barcellos, A.R.C. Rocha, R.A. Falbo, "An ontology-based approach for software measurement and suitability measures basis evaluation to apply statistical software process control in high maturity organizations", In *Proceedings of the ER2009 PhD Colloquium*, 2009.
- [20] M.A.Martin, L. Olsina, "Towards an ontology for software metrics and indicators as the foundation for a cataloging web system", *First Latin American Web Congress*, 2003.