An Ontology-based Approach for Software Measurement and Suitability Measurement Repository Evaluation to Apply Statistical Software Process Control in High Maturity Organizations

Monalessa Perini Barcellos[1,2],
Ana Regina C. da Rocha (advisor)[1], Ricardo de A. Falbo (advisor)[2]

[1] COPPE, Federal University of Rio de Janeiro – Brazil
[2] Ontology and Conceptual Modeling Research Group (NEMO), Federal University of Espírito Santo – Brazil
{monalessa, falbo}@inf.ufes.br; darocha@cos.ufrj.br

Abstract. The escalating demands on the development of software products require software organizations to produce mature software processes that are capable of providing the required levels of quality and productivity. The implementation of statistical process control (SPC) in performance process analysis uses data collected during the course of the project to analyze the behavior of organization processes, identifying actions that are needed for the stabilization and improvement of those processes. An essential element for the SPC application is the suitability of the measures being used. This paper presents the approach proposed in a doctorate thesis to support organizations obtain and maintain measurement repository suitable for SPC, as well as to perform measurements appropriate in this context. The approach is composed by an Instrument for Evaluating the Suitability of a Measurement Repository to SPC, a Software Process Measurement Ontology and a Body of Recommendations for Software Measurement.

Keywords: Software Measurement, Software Measurement Ontology, Foundational Ontology, Statistical Process Control, High Maturity Levels.

1 Introduction

The requirements of the industry of software products and services have increased the organizations' interest in process improvement. There are several frameworks that support definition and institutionalization of programs of this nature, in which measurement plays an important role, such as CMMI and ISO/IEC 15504. While the process improvement evolves in an organization, the organizational maturity level has a tendency to increase too. In high maturity[1], process improvement based on traditional measurement is not enough. It is necessary to carry out the statistical process control (SPC) to get to know the behavior of the process, to determine its

[1] High maturity is characterized by the highest levels of maturity models such as CMMI levels 4 and 5.

performance in previous executions and so predicting its performance in current and future projects, checking if they will be able to achieve the established objectives. Using SPC in this context is only possible for organizations that carry out measurement appropriately. However, this is one of the biggest difficulties for organizations that desire to achieve high maturity [1]. In spite of models and standards saying what it is necessary in the highest maturity levels, they do not guide the organizations about how carrying out the actions (among their measurement) that lead the high maturity [2]. Considering this situation, the doctorate thesis described here aims to define a strategy to support organizations that desire to achieve high maturity in their software process in order to obtain and maintain measurement repository suitable for SPC, as well as to perform measurements appropriate in this context. Considering that the thesis's approach is related strongly to software measurement, it was necessary to establish the vocabulary to be used in it, in order to allow the common understanding of the approach. Besides, this vocabulary should consider basic and high maturity measurement aspects. We do not found a vocabulary suitable for identified needs, so we decided to define a Software Process Measurement Ontology (SPMO) able to attend them.

In this paper we present briefly the strategy proposed in the thesis and, as soon as the focus of discussion here is the SPMO, this is discussed in more details (Section 2). In section 3 we present the expected contributions and the proposal for evaluation of the results, specifically in the context of the SPMO.

2 The Approach

Software organizations that want to perform Statistical Process Control (SPC) usually are in one of the following situations: *(i)* it achieved the initial maturity levels and it has measurement repository with data collected along fulfilled projects; or *(ii)* it is starting a process improvement program and wants to initially define a measurement repository and to perform measurement in a way that is appropriate to SPC. The strategy proposed in this thesis must contemplate both situations and, for that, it has three main components:

(a) *Instrument for Evaluating the Suitability of a Measurement Repository to SPC (IESMB):* the goal of this component is to evaluate existing measurement repository and to determine its suitability to SPC, identifying corrective actions that can be done to make the measurement repository suitable to SPC, when necessary.

(b) *Software Process Measurement Ontology (SPMO):* the goal of this component is to capture the conceptualization involved in the software measurement domain, including traditional and high maturity aspects.

(c) *Body of Recommendations for Software Measurement (BRSM):* its goal is to supply guidelines on how to perform measurement suitable for SPC.

According to the proposed strategy, organizations that achieved initial maturity levels can use the IESMB in order to evaluate and adapt, when possible, their measurement repository to SPC. On the other hand, organizations that are starting a measurement program and look for high maturity can use BRSM and SPMO to define their measurement repository and measurement program in a suitable way for SPC. Besides, it is important to note that performing SPC requires frequent data collection for measures and sometimes the definition of new measures. Then, when new data are collected for measures and/or new measures are defined, the measurement repository suitability must be reevaluated (using IESMB) and new measures must be defined and collected appropriately (using BRSM and SPMO).

Considering related works, in context of measures evaluation for SPC, Tarhan and Demirors [3] defined measure usefulness to SPC as requirement to choose a process to SPC. Nevertheless, as far as we noticed, the work about measure evaluation is superficial and it does not offer guidelines to correct a measure or to avoid new useless measures to SPC. Regarding software measurement ontology, Ferreira et al. [4] proposed a software measurement ontology and metamodel. However, they did not include high maturity aspects and they used a foundational ontology as basis, likewise in the thesis described here.

2.1 The Software Process Measurement Ontology

Software Measurement is considered a recent discipline and still did not establish agreed standards. Terminologies, concepts, principles and methods were defined in the last decade, but there is no consensus mainly about measurement concepts and terminologies. There are duplications and inconsistencies on the most common terms like measure, metric and measurement [4]. Besides, the proposals found in the literature do not offer a complete vision of software measurement, since they do not include specific aspects of the measurement in the context of high maturity organizations.

Domain ontologies can be used to define a common vocabulary for sharing and reusing of knowledge about some domain. As defended by Guarino [5], ideally domain ontologies should be built based on Foundational Ontologies. Besides, fidelity to reality and conceptual clarity are fundamental quality attributes to conceptual models in general and, in particular, to domain ontologies. The use of a foundational ontology as a basis for building domain ontologies contributes to achieve these attributes [6]. Thus, we decided to use the Unified Foundational Ontology (UFO) [6] as basis for building the SPMO. UFO [6] 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. UFO makes distinctions and provides guidelines for building and evaluating conceptual models, giving them real-world semantics. It is composed by three main parts: UFO-A, an ontology of endurants, that is the core of UFO; UFO-B, an ontology of perdurants (events); and UFO-C, an ontology of social entities (both endurants and perdurants) built on the top of UFO-A and UFO-B. A fundamental distinction in UFO-A is between *Individuals* (entities that exist in reality, possessing a unique identity, e.g. the person Mary) and *Universals* or *Types* (patterns of features which can be realized in a number of different individuals, e. g. Person). Due to space limitations, we do not discuss UFO here.

The SPMO is based on terminologies used in the literature and on specific requirements of software process measurement in high maturity organizations that were identified during studies based on systematic reviews and in some practical experiences. The building of the SPMO follows the process defined by SABiO (*Systematic Approach for Building Ontologies*) [7], whose activities are: Requirement Specification, Ontology Capture, Ontology Formalization, Integration of Existing Ontologies, Ontology Evaluation and Documentation.

In the development of the SPMO, the requirement specification involved the definition of competency questions (CQs), such as the following ones: Which indicators can be used to analyze an objective? What are the information needs considered by an indicator? Based on the CQs, concepts were captured and grounded on UFO, being represented in models UML, textual descriptions and first-order logic axioms. The SPMO was integrated with a Software Process Ontology [8] and a Software Enterprise Ontology which, before the integration, were reengineered at light of UFO [9]. Figure 1 presents a small fragment of the SPMO. In this paper, the distinctions made in UFO are shown in the concepts of the SPMO as stereotypes, indicating that they are subtypes of concepts of UFO, in an approach analogous to the one defined in [5].

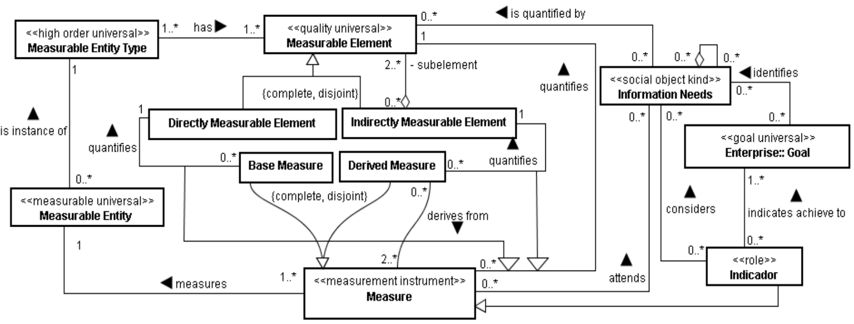

Fig. 1. A SPMO Fragment.

A *Measurable Entity Type* is a universal high order in UFO, meaning that its instances are universals (e.g., Organization, Process and Project). A measurable entity type indicates which measurable elements can be used to measure entities of this type. A *Measurable Element* is a property (quality universal in UFO) of a measurable entity type through which *Measurable Entities* of this type can be described. It can be *directly* (e.g., cost) or *indirectly measurable* (e.g., productivity). Measurable elements are quantified by measures. A *Measure* is a measurement instrument that is used to associate a value to a measurable element. When a measure quantifies a directly measurable element, it is said a *Base Measure* (e.g., number of defects). When a measure quantifies an indirectly measurable element it is said a *Derived Measure* (e. g., defects density). Organizations carried out measurement to attend their *Objectives* and/or *Information Needs*. An organization with the goal of "reducing operational costs in 10%" can, for example, take as information need "knowing costs with rework in projects". Measures are used to attend information needs and when a measure is used to indicate the achievement of an objective, it plays the role[2] of *Indicator*.

During the development of the SPMO, several restrictions were identified and, since the models did not capture several of them, we defined axioms to make them explicit. For instance, the following axiom holds: if an indicator *ind* considers an information need *inf* and *ind* quantifies the measurable element *mel* then the information need *inf* is quantified by the measurable element *mel*.

$$(\forall ind \in Indicator, inf \in Information\ Needs, mel \in Measurement\ Element)$$
$$(considers(ind,inf) \wedge quantifies(ind, mel) \rightarrow isQuantifiedBy(inf, mel))$$

[2] According to UFO, it is a possible role that a substance sortal can play along its history. An entity plays a role in a certain context, demarcated by its relations with other entities. Sortal universals are entities that describe a standard of characteristics that can be carried out in a number different from individuals and, besides, provide a principle of individualization, persistence and identity.

3 Expected Contributions and Results Evaluation

Once that SPMO treats high maturity software measurement aspects and it is grounded on a foundational ontology (characteristics that we did not found in others proposals) we believe that the SPMO is a contribution itself. Besides, the works carried out until now produced other contributions, like the reengineering of a Software Enterprise Ontology at the light of UFO, for integration to the SPMO [9] and the evolution of some aspects of UFO concerning generic concepts related to the measurement context.

The main expected result related to the SPMO is that its conceptualization supports understanding and performing software process measurement in traditional and high maturity contexts. To evaluate this, as part of this thesis, SPMO is being used as basis for building BRSM (described in Section 2). Besides, the SPMO is being used together with the BRSM for building a measurement repository suitable for high maturity, which will be used at LENS (Software Engineering Laboratory) in COPPE/UFRJ. Both experiences will supply information about the usefulness of the SPMO. In the near future, it will also be possible to evaluate how SPMO supports interoperability between tools that will be developed in High Maturity Environment at LENS, context where this thesis is being developed.

References

[1] Wang, Q., Li, M.: Measurement and Improving Software Process in China. In Proceedings of International Symposium on Empirical Software Engineering, pp.183-192 (2005).

[2] Card, D. N.: Statistical Techniques for Software Engineering Practice. In Proceedings of the 26th International Conference on Software Engineering, pp. 722-723 (2004).

[3] Tarhan, A., Demirors, O.: Investigating Suitability of Software Process and Metrics for Statistical Process Control. Lectures Notes in Computer Science, Vol. 4257, pp. 88-99 (2006).

[4] Ferreira, M., Garcia F., Ruiz, F., et al. : Medición del Software: Ontología y Metamodelo. Informe Técnico UCLM -TSI-001, Universidad de Castilla-la Mancha, Ciudad Real, Spain (2006).

[5] Guarino, N.: Formal Ontology and Information Systems. In Proceedings of International Conference in Formal Ontology and Information Systems, pp 3-15 (1998).

[6] Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Universal Press, The Netherlands, ISBN 90-75176-81-3 (2005).

[7] Falbo, R. A.: Experiences in Using a Method for Building Domain Ontologies. In Proceedings of the 4th International Conference on Quality Software, pp. 162-169 (2004).

[8] Guizzardi, G., Falbo, R. A., Guizzardi, R. S. S.: 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, pp 244-251 (2008).

[9] Barcellos, M. P., Falbo, R. A.: Using a Foundational Ontology for Reengineering a Software Enterprise Ontology. The Joint International Workshop on Metamodels, Ontologies, Semantic Technologies, and Information Systems for the Semantic Web (2009).