A Software Measurement Task Ontology

Monalessa Perini Barcellos, Ricardo de Almeida Falbo
Ontology and Conceptual Modeling Research Group (NEMO), Computer Science Department
Federal University of Espírito Santo, Vitória, Brazil
+55-27-4009-2167
{monalessa, falbo}@inf.ufes.br

ABSTRACT
Software measurement is a key process for software project management and software process improvement. There are several process quality models and measurement standards that point out its importance and present good practices for it. Unfortunately, the vocabulary used by these models is diverse. This leads to misunderstanding and problems related to the jointly use of different standards. Aiming at establishing a common conceptualization regarding the software measurement process, we developed a Software Measurement Task Ontology (SMTO), which is grounded on the Unified Foundational Ontology. This task ontology was developed to be used for addressing semantic interoperability problems arising from the jointly use of different measurement-related standards, as well as for supporting semantic integration of software applications supporting measurement.

Categories and Subject Descriptors
D.2.12 [**Interoperability**]

General Terms
Management, Languages.

Keywords
Task Ontology; Software Measurement; Semantic Interoperability

1. INTRODUCTION
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 assessing the quality of software products and the capability of organizational software processes.

There are several standards and maturity models, such as ISO/IEC 12207 [1] and CMMI (Capability Maturity Model Integration) [2], that include measurement as an essential process for organizations to achieve maturity in software development. Besides process quality standards and maturity models, there are also standards and methodologies devoted specifically to assist organizations in defining their software measurement processes, such as ISO/IEC 15939 [3] and PSM (Practical Software Measurement) [4]. In general, these standards define measurement related terms and good practices (activities) that are adopted by the software industry. However, unfortunately, the vocabulary and the activities defined by those standards are diverse. Many times, the same concept is designated by different terms in different proposals. Others, the same term refers to different concepts. The same occurs with the activities. There is not uniformity in naming them and establishing a common set of activities. This causes semantic interoperability problems when an organization decides to jointly use different measurement-related standards.

Semantic interoperability problems also arise when some software applications need to be integrated to support the software measurement process, such as jointly using tools for project management, quality assurance and process management. For properly integrating those applications, someone should take three related aspects into account [5]: data integration, which deals with how the applications share data; functionality integration, which addresses how applications share functionalities/services; and process integration, which is responsible for handling message flows and defining the overall process execution. Thus, semantic integration encompasses the intended meaning of concepts, services and processes [5].

To deal with the interoperability problems aforementioned (between standards and between software applications), we need to establish a common conceptualization regarding the software measurement universe of discourse. Ontologies can be used for that purpose. An ontology is a formal representation of a common conceptualization of a universe of discourse [9]. Ontologies can focus on describing the concepts of a domain (domain ontologies) or describing general tasks (task ontologies) that are independent of domain [9]. A large amount of domain ontologies have been used in various fields [11], including software measurement [6, 7, 8, 13]. These ontologies are useful for dealing with semantic interoperability problems related to the vocabulary used by the standards and to data integration of software applications. However, domain ontologies are not enough to deal with semantic interoperability problems related to the practices/activities advocated by the standards, nor to support functionality and process integration of software applications. In fact, in order to achieve semantic interoperability in a broader sense, it is important to achieve a common understanding regarding both domain and task-related aspects of software measurement.

Despite the increasing use of domain ontologies, the same does not occur with task ontologies [14]. This is a problem, since in functionality and process integration of software applications, task ontologies can be used to assign meaning to services, functionalities, activities, and its related information. In the case of integrating different standards, again task ontologies can be used to assign meaning to activities and best practices.

This paper presents a task ontology for the software measurement process. It is to be used as a reference model for semantically integrating software measurement-related standards, as well as software measurement tools. Since the software measurement process is very complex, the ontology focuses on its main activities, namely measurement planning, execution and results analysis, which are the core of a measurement process.

This paper is organized as follows. Section 2 regards the theoretical background of the paper, discussing briefly software measurement, semantic interoperability problems that arise in this field, and the use of ontologies to deal with these problems. Section 3 presents our software measurement task ontology. Section 4 discusses its use for integrating measurement-related standards. Section 5 discusses some related works. Finally, Section 6 presents the conclusions of this paper.

2. SOFTWARE MEASUREMENT AND SEMANTIC INTEROPERABILITY

Nowadays, software measurement (SMEA) is considered a key process for software project management, software quality and software process improvement. It provides organizations with objective information for decision making that impacts the organization's performance. Successful organizations use measurement as part of their day-to-day activities [4].

The SMEA process involves, among others, activities related to measurement planning, execution and results analysis and reporting [1,2,3,4]. For performing software measurement, initially, it is necessary to plan it. During the measurement planning, information needs are identified and measures able to meet them are identified. For those measures, measurement and analysis procedures must be defined. Once planned, measurement can start. Measurement execution involves collecting and storing data for the defined measures, according to their measurement procedures. Once data are collected, they should be analyzed according to the analysis procedures. Finally, the analysis results should be reported to the interested parties.

SMEA has played an increasingly important role in Software Engineering. As a consequence, most current standards and maturity models, such as ISO/IEC 12207 [1], ISO/IEC 90003 [15] and CMMI [2], includes measurement as an essential process for software organizations. Standardization is very important, since standards provide organizations with agreed and well recognized practices. Standardization is one of the driving forces to achieve interoperability, with the provision of agreed domain conventions, terminologies and practices [13].

However, despite all the efforts in international standardization during the last years, there is no consensus yet on the terminology and practices related to SMEA [13]. Different standards are developed independently, by different groups, in different points in time, without considering the harmonization between them as a priority requirement. As a result, terminological conflicts and inconsistencies appear not only between standards from different bodies, but also within those from the same organization. This problem is amplified as no single standard contains a complete vision of SMEA; each one offers just a partial view of it, e.g. on the measures, on the measurement processes, or on the target entities and measurement goals [13]. For instance, ISO/IEC 90003 recommends its users to see further information regarding SMEA in ISO/IEC 12207 and ISO/IEC 15939, among others standards. In fact, it is a common situation to have an organization jointly using several standards in a software process improvement (SPI) initiative. However, adopting multiple standards and/or models often results in misalignment, creating additional reconciliation efforts [16]. This misalignment is caused largely by the semantic interoperability problems reported above. Thus, we need efforts to harmonize SMEA standards.

Another important issue related to implementing a measurement program in a software organization regards tool support. SMEA is a complex task, and to be properly done, it should be supported by a suite of tools. The integration of these tools is a hard problem. The main difficulty is that generally the systems are not developed thinking in integration. Contrariwise, they generally have their own structural and behavioral models. This heterogeneity is one of the biggest problems in system integration. To solve this problem, it is necessary to resolve syntactic (related to structure) and semantics (related to meaning) conflicts [5]. Semantic integration involves three main aspects: data integration (refers to data exchange), functionality integration (deals with message/service exchange) and process integration (responsible for combining the systems for an adequate support to a process).

An alternative to deal with the semantic interoperability problems related above (including both problems related to software application integration and standard integration) is to establish a common conceptualization about the SMEA universe of discourse. An ontology is a conceptual specification that describes the knowledge of a universe of discourse [10]. Guarino classifies ontologies into [9]: (i) *foundational ontologies*, which describe very general concepts, such as object, event, etc., (ii) *domain ontologies*, which describe the conceptualization related to a generic domain (e.g., medicine, law), (iii) *task ontologies*, which describe the conceptualization related to a generic task (such as diagnosis and sale), and (iv) *application ontologies* that describe concepts dependent on a particular domain and task. Domain ontologies have been widely used in various areas of computer science, but the same does not occur with task ontologies [14].

Task ontologies should capture two major kinds of knowledge [14]: (i) task decomposition, including control flow, and (ii) knowledge roles played by entities from the domain in the fulfillment of the task. These two kinds of knowledge are very inter-related, although they capture different views of a task. In fact, they represent different modeling aspects, i.e. different dimension of modeling that emphasizes particular views of the same portion of the reality. Thus, we need different models for representing them [14]. Martins and Falbo [14] proposed the use of two UML diagrams for representing task ontologies: activity diagrams, capturing task decomposition into sub-tasks and how knowledge roles act in their fulfillment; and class diagrams, modeling the concepts involved and their relations.

In the next section, we present a task ontology that describes aspects of these two perspectives of the SMEA process. It is worthwhile to point out that, although we use the term "task ontology", which is already consecrated in the field of ontologies, in fact we are talking about a process ontology, in the sense that we are interested in describing the SMEA process as a whole, and not tasks with low granularity level. Moreover, we should emphasize that our ontology is a reference ontology, i.e., 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 entities in the universe of discourse, for the purposes of communication, learning and problem-solving. We are not interested in an implementation of this ontology for purposes of reasoning, for instance. On the other hand, as advocated by Guizzardi [10], a reference ontology should be developed taking truly ontological distinctions into account, i.e. a reference ontology should be grounded in a foundational ontology. Thus, our task ontology is developed grounded in the Unified Foundational Ontology (UFO) [11, 12].

As discussed before, a task ontology should capture both behavioral and structural knowledge involved in a task. Since many of the concepts involved in the SMEA process had already been captured by the domain ontologies, we tried, whenever possible, to reuse the Reference Software Measurement Ontology (RSMO) [6, 7, 8], and to maintain our task ontology aligned to the RSMO. A comparison between the structural model of our task ontology and RSMO is presented in Section 5. Finally, due to space limitations, in this paper we present only the core of our ontology. This ontology has several axioms, but they are not discussed in this paper.

3. SMTO: A SOFTWARE MEASUREMENT TASK ONTOLOGY

As a process ontology, SMTO is supposed to answer the following competency questions: (i) Which are the activities of the SMEA process? (ii) Who is responsible for performing them? (iii) How the activities are decomposed into sub-activities? (iv) What is the control flow between them? (v) What are the inputs and outputs of each activity?

Following the guidelines given in [14], for capturing the conceptualization involved in the SMEA process, we developed two conceptual models: a structural conceptual model (Subsection 3.1) and a behavioral model (Subsection 3.2).

3.1 - SMTO Structural Model

Figure 1 presents the structural model of the SMTO. As shown in this figure, a **Measurable Entity** is anything that can be measured, such as processes, artifacts, projects and resources. Given its very general nature, **Measurable Entity** corresponds to *Measurable Universal* in UFO, i.e., a *Universal* that is characterized by at least one *Quality Universal*. The quality universals that characterize a **Measurable Entity** are said **Measurable Elements.** According to UFO [11], a *Quality Universal* is measured by means of *Instruments*. An *Instrument* has a *Quality Structure* that defines qualia (*Quale*) that can be assigned as values of the qualities that are instances of the corresponding *Quality Universal*. *Instrument*, *Quality Structure* and *Quale* are *Abstract Particulars*, whereas *Quality Universal* is a *Universal*. Based on this conceptualization, we consider that **Measure** is an *Instrument* that quantifies **Measurable Elements** (*Quality Universal*). Although not shown in Figure 1, a **Measure** has a **Scale** (*Quality Structure*) associated with it that defines **Scale Values** (*Quale*) [6]. Moreover, a **Measure** is the role played by an *Instrument* when it is used to measure **Measurable Entities** to meet the **Information Needs** of a **Project** or **Organization** (not shown in Fig. 1).

During Measurement Planning, the **Measurement Manager** develops a **Measurement Plan** that registers **Information Needs**, **Measures** and **Operational Definitions of Measures**. Regarding UFO, **Measurement Manager** is a *Role* played by a Person (*Kind*) that is responsible for the **Measurement Plan**. An **Operational Definition of Measure** is a *Relator Universal*. In UFO, relators are mediating individuals with the power of connecting entities. Every instance of a *Relator Universal* is existentially dependent of at least two distinct entities. *Relators* are the foundation for *material relations* [11], such as the relation "defines an operational definition for" between **Organization** and **Measure**. In other words, *material relations* have material structure on their own. The relata of a *material relation* are mediated by *relators* [11]. Thus, the relationships between a *relator* (e.g., **Operational Definition of Measure**) and the entities that it connects (e.g., **Organization** and **Measure**) are *mediation relations*. Mediation is a formal relation that takes place between a relator and the entities it mediates [11]. Thus, **Operational Definition of Measure** is a *Relator Universal* that mediates **Organizations** (not shown in Fig. 1) and **Measures**, defining, among others, the **Measurement Procedure** and the **Measurement Analysis Procedure** to be applied when using this Measure in that Organization. **Measurement Procedure** and **Measurement Analysis Procedure** are a special type of *Kind* in UFO, said *Normative Description*, which defines rules and norms recognized by an organization [12].

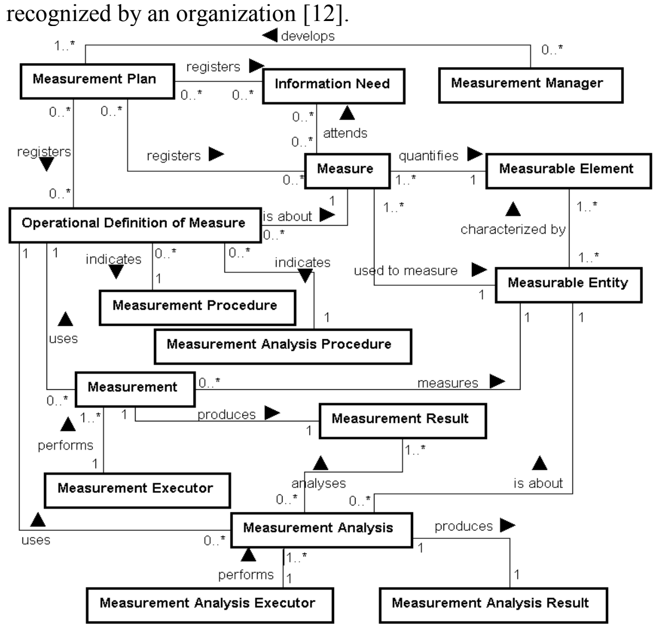

Figure 1. Structural Conceptual Model (Partial) of the SMTO

Some relators have their origin in events. This is the case of **Measurement** and **Measurement Analysis**. Both are *Relator Universals* that are originated by the corresponding events in the behavioral model. The relator universal **Measurement** connects the **Measurable Entity** being measured, the **Measurement Executor**, the **Operational Definition of Measure** used and the **Measurement Result** produced. The relator universal **Measurement Analysis** connects the **Measurable Entity** from which **Measurement Results** are being analyzed, the **Measurement Analysis Executor**, the **Operational Definition of Measure** used and the **Measurement Analysis Result** produced.

Since the **Measurement Executor** can be either Persons or Software Tools, which are *Kinds* that provides different principles of individualization and identity to their instances**, Measurement Executor** is a *Role Mixin* in UFO, i.e., a dispersive (anti-rigid and externally dependent) universal that is common to different roles [11] (in the case, the role played by a Person or a Software Tool that performs a **Measurement**). **Measurement Analysis**, in turn, is performed by a Person (*Kind*), and thus, **Measurement Analysis Executor** is a *Role* (in UFO sense) played by the person that performs **Measurement Analysis**.

3.2 - SMTO Behavioral Model

Figure 2 presents the main activities of the SMEA process and how the concepts from the structural model (Fig. 1) act in this process. Figure 3 shows the detailing of the "Plan Measurement" activity. Although other activities shown in both activity diagrams are also decomposed into sub-activities, in this paper we do not show their detailing in other activity diagrams. In fact, all the activities shown in both diagrams are complex actions in the sense of UFO. According to its ontology of social entities [12], actions are intentional events. Complex actions are actions involving the participation of different objects and agents. In all the activities presented in the SMTO behavioral models we have the participation of one agent and of one or more objects. Thus, all of them are complex actions in the sense of UFO. On the other hand, only some of them are decomposed into other sub-activities, and thus represented as UML's Call Behavior Actions.

Finally, some stereotypes are added to the object flows, in order to capture distinctions made in UFO's ontology of events [12] concerning the types of object participations in actions, namely: creation, indicating that an object is created by the action; change, indicating that some property of the object changed; and usage, when the object is used without changing any of its properties. In Figure 3, there are also some object flows that are not stereotyped. In these cases, we mean that the object was only selected (it was not created nor changed) in the activity.

The SMEA process starts with a **Measurement Manager** performing the "Plan Measurement" activity, aiming at creating a **Measurement Plan**. As a result of performing this activity, a **Measurement Plan** is created, registering the **Information Needs**, **Measures**, and the corresponding **Operational Definition of Measures**. Also as an output of this activity, some new **Information Needs**, **Measures**, and **Operational Definition of Measures** can be created. To perform this activity, existing **Information Needs**, **Measures**, and **Operational Definitions of Measures** are used, as well as information regarding **Measurable Entities**, **Measurable Elements**, **Measurement Procedures** and **Measurement Analysis Procedures**. Figure 3 details this activity.

The first sub-activity of the "Plan Measurement" activity is "Start Measurement Planning". In this sub-activity, the structure of the **Measurement Plan** is defined. This structure can be defined to conform to organizational guidelines, measurement plan templates or organizational policies. Certainly, this activity has inputs. However, since there is not a consensus about what can be used to guide the definition of the measurement plan structure, we chose not to represent any input in the diagram. It is important to mention that, in fact, nothing related to the content of the plan is done in this task. This sub-activity only regards defining the structure of the plan.

The next activity is "Identify Information Needs". This is the first activity that actually is related to the content of the measurement plan, since information needs are the driver for measurement planning. In this activity, Information Needs can be selected among the existing ones or new ones can be defined based on organizational business goals or other information related to the organization that is relevant for measurement. Again, since there is not a consensus about what information can be used to guide the definition of the **Information Needs**, we chose not to represent an input regarding it. In fact, depending on the organization's maturity level, some elements are mandatory. For instance, in high maturity levels (CMMI levels 4 and 5), Information Needs should be defined based on business goals. However, since SMTO aims to address measurement in any maturity level, we decided not to represent Business Goals as an input for this activity.

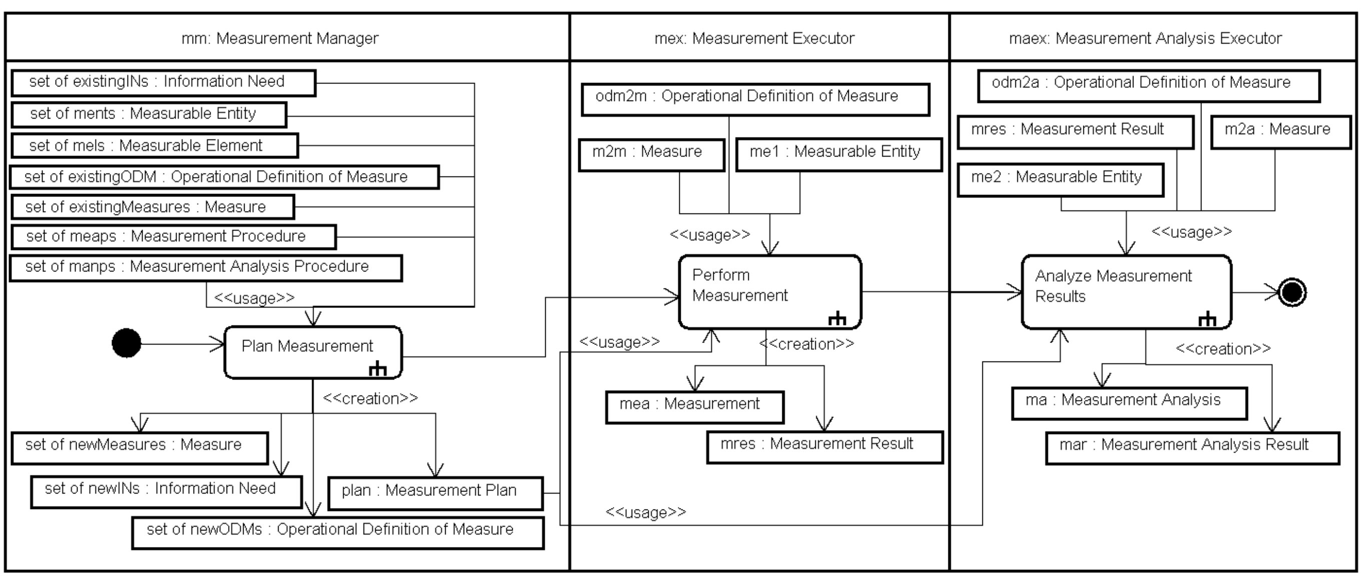

Figure 2. Behavioral Conceptual Model: Overview of the Software Measurement Process.

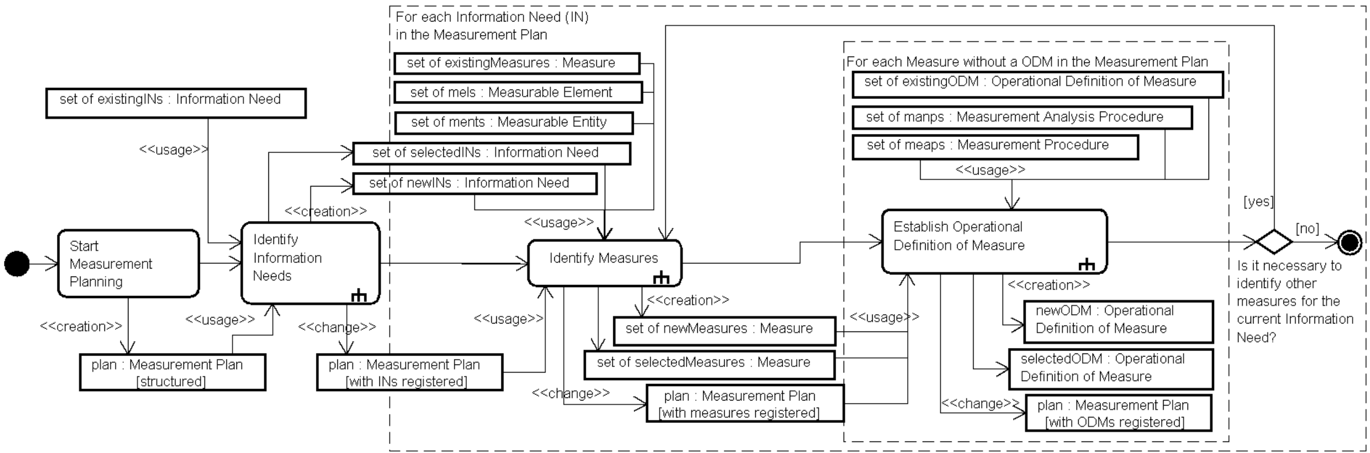

Figure 3. Behavioral Conceptual Model: Sub-activities of the "Plan Measurement" activity.

As outputs of the "Identify Information Needs" activity, we have the **Measurement Plan** with **Information Needs** registered, as well as the new **Information Needs** that may have been created. Existing **Information Needs** that were selected are also registered in the **Measurement Plan**. Thus Selected Information Need is a role (in UFO's sense) played by **Information Needs** that are registered in a **Measurement Plan**.

For each **Information Need** registered in the **Measurement Plan**, the **Measurement Manager** has to "Identify Measures". First, he/she has to select the entity (**Measurable Entity**) and its quality (**Measurable Element**) to be measured. If there are **Measures** already defined for the selected **Measurable Entity** / **Measurable Element** that meet the **Information Need**, these measures can be selected; otherwise, new **Measures** have to be defined. Selected and new measures are, then, registered in the **Measurement Plan**.

For each **Measure** identified in the previous step, the **Measurement Manager** has to "Establish Operational Definition of Measure". Analogously to information needs and measures, if there is an **Operation Definition of Measure** already defined in the organization for the selected **Measure**, this definition can be selected; otherwise, a new **Operation Definition of Measure** has to be defined. When defining a new one, a **Measurement Procedure** and a **Measurement Analysis Procedure** have to be selected or defined. Selected and new operational definitions are registered in the **Measurement Plan**.

Once the **Measurement Plan** is defined, measurement can be performed by a **Measurement Executor** ("Perform Measurement" activity). Although not shown as an activity diagram in this paper, this activity is decomposed into sub-activities. First, a **Measure** should be selected ("Select Measure"). To do so, the **Measurement Executor** should choose the **Measurable Entity** to be measured and the corresponding **Measure** to be used. Once selected the Measure, an **Operational Definition of Measure** should be selected, which indicates the **Measurement Procedure** to be applied (see Fig. 1). This procedure is applied to collect data ("Collect Data" activity), which is then stored ("Store Data"). The *relator* **Measurement** registers the occurrence of the "Perform Measurement" activity, which produces also as an output the **Measurement Result**.

Finally, the "Perform Measurement Analysis" activity can be accomplished. Like the previous activity, this activity is decomposed into sub-activities, starting with selecting the **Measurable Entity** and the corresponding **Measure** to be analyzed. Once selected the Measure, an **Operational Definition of Measure** should be selected, indicating the **Measurement Analysis Procedure** to be applied (see Fig. 1). Taking this procedure into account, the Measurement Results to be analyzed should be selected ("Select Data for Analysis" activity) and analyzed ("Analyze Data" activity), and finally the **Measurement Analysis Results** are stored ("Store Analysis Data" activity). The *relator* **Measurement Analysis** registers the occurrence of the "Perform Measurement Analysis" activity, which produces also as an output the **Measurement Analysis Result**.

4. USING SMTO FOR INTEGRATING MEASUREMENT-RELATED STANDARDS

The proposed ontology was used for integrating some of the main standards and models that address software measurement, namely: ISO/IEC 15939 [3], ISO/IEC 12207 [1], PSM [4], and CMMI [2]. This harmonization of standards[1] can be especially useful for organizations intending to use several them jointly.

Table 1 shows the equivalence between some of the concepts of the SMTO structural model and the terms used in the standards. Some concepts, such as Information Need and Measure, are consensual; others have different terms to designate them; and others are not explicitly addressed by some the standards.

ISO/IEC 12207, ISO/IEC 15539 and PSM define a measurement process as a set of activities that are decomposed into tasks. For each task, ISO/IEC 15939 defines normative clauses. CMMI, in turn, defines the measurement process area in terms of generic and specific goals (GG/SG) that can be achieved by generic and specific practices (GP/SP).

Table 2 shows the equivalence between the activities of the software measurement process as defined in the SMTO and the activities proposed by these standards. Following, we make some comments related to the mapping shown in Table 2.

[1] Although we are talking about standards and models, in this section we use the term standard to designate both, standards and models.

Table 1. Mapping between the main SMTO terms and the corresponding standards' terms.

SMTO	ISO/IEC 15939	ISO/IEC 12207	PSM	CMMI
Measurable Entity	Entity	-	Entity	-
Measurable Element	Attribute	-	Attribute	-
Information Need	Information Need	Information Need	Information	Information Need
Measure	Measure	Measure	Measure	Measure
Operational Definition of Measure	Formal Definition	-	Operational Definition of Measure	Operational Definition of Measure
Measurement Procedure	Measurement Procedure (also Data Collection	Data Collection Procedure	Measurement Method	Data Collection Procedure
Measurement Analysis Procedure	Data Analysis Procedure	Data Analysis Procedure	Analysis Model	Analysis Procedure
Measurement Plan	Measurement Plan	-	Measurement	Measurement Plan
Measurement Result	Data	Data	Data	Measurement Data
Measurement Analysis Result	Information product	Information product	Information	Measurement Result

Table 2. Mapping between SMTO activities and standards' activities.

SMTO	ISO/IEC 15939	ISO/IEC 12207	PSM	CMMI
1- Plan Measurement	Plan the Measurement Process	Measurement Planning	Plan Measurement	GG. Align Measurement and Analysis Activities
1.1- Start Measurement Planning	-	-	-	-
1.2 - Identify Information Needs	Identify information needs	The project shall identify and prioritize the information needs	Identify and prioritize information needs	SP. Establish measurement objectives
1.3- Identify Measures	Select measures	The project shall select and document measures that satisfy the information needs	Select and specify measures	SP. Specify Measures
1.4- Establish Operation Definition of Measure	Define data collection, analysis, and reporting procedures	The project shall define data collection, analysis, and reporting procedures	Select and specify measures	SP. Specify data collection and storage procedures
2- Perform Measurement	Collect Data	The project shall collect, store, and verify data	Collect and process data	SP. Obtain Measurement Data
3- Analyze Measurement Results	Analyze data and develop information products	The project shall analyze data and develop information products	Analyze Data	SP. Analyze Measurement Data

Concerning the "Plan Measurement activity" of SMTO, the "Start Measurement Planning" activity addresses mainly the creation of the Measurement Plan document. As shown in Table 2, the standards do not include a specific activity for this. However, except for ISO/IEC 12207, which does not explicitly mention the Measurement Plan (see Table 1), the proposals address it as an outcome of the measurement planning. Although ISO/IEC 12207 does not cite the document, it defines outcomes that typically are recorded in the Measurement Plan (e.g. information needs and measures). We decided to include this activity for capturing a common practice in real-world organizations, where measurement plans are typically structured a priori.

Regarding the "Identify Information Needs" activity, as discussed in the previous section, there is not a consensus regarding the information used as input to this activity. Considering the standards being analyzed, CMMI says that, among others, strategic, business and project plans, as well as goals, can be source of information needs. PSM lists a set of sources, such as risks and project constraints. ISO/IEC 15939 and ISO/IEC 12207 include tasks to provide information used as input to identify information needs, namely: "*Characterize organizational unit*" (ISO/IEC 15939) and "*The project shall describe the characteristics of the organization that are relevant to measurement*" (ISO/IEC 12207).

Concerning the "Identify Measures" activity, in SMTO, as discussed in the previous section, the identification of measures can be made by selecting existing measures or defining new ones. ISO/IEC 15939 and CMMI make this distinction explicit by means of normative clauses (ISO/IEC 15939) and subpractices (CMMI) (not shown in Table 2).

As for the "Establish Operational Definition of Measure" activity, in PSM there is not a task devoted to establish operational definitions. However, during the task *Select and specify measures*, the measurement construct is defined. It contains detailed information about the measure, including measurement and analysis procedures (called, respectively, *measurement function* and *analysis model* in PSM). The measurement construct joins information regarding measure definition and operational definition of measure.

The "Perform Measurement" and "Analyze Measurement Results" activities are addressed by the standards as a single activity (specific goal in the case of CMMI), which is divided in tasks (specific practices in the case of CMMI) that deal with data collection and analysis aspects.

5. RELATED WORK

There are some works that explore the development of ontologies for the measurement universe of discourse, such as [6, 7, 8, 13]. However, at the best of our knowledge, all of them refer to domain ontologies. Moreover, most works are not committed to a foundational ontology; an exception is the Reference Software Measurement Ontology (RSMO) [6, 7, 8], which is developed grounded in the Unified Foundational Ontology (UFO).

RSMO was used as the basis for the structural conceptual model of our Software Measurement Task Ontology (SMTO). However, it is important to highlight that, although the SMTO structural model is very in line with RSMO, it presents also differences. First, we included new concepts, namely: **Measurement Plan**, **Measurement Manager**, **Measurement Executor** and **Measurement Analysis Executor**. Second, we revised cardinalities of some relations, due mainly to the focus on the task instead of the domain. This was the case, for instance, of the relation "indicates" between **Operational Definition of Measure** and **Measurement Analysis Procedure**. Since in the SMTO there is an activity "Analyze Measurement Results", we consider that an **Operational Definition of Measure** should mandatorily indicate a **Measurement Analysis Procedure**. In RSMO, this relation is optional [8]. Third, we also revised some of the ontological foundations of RSMO. For instance, in RSMO [7], **Measurement** and **Measurement Analysis** are considered *Action Universals* in UFO. We perceive these notions somewhat differently. As discussed in Section 3, *actions* give rise to *relators* that connect the entities involved in the domain. Thus, **Measurement** and **Measurement Analysis** are not *Action Universals*, but *Relators* originated by the corresponding actions, namely "Perform Measurement" and "Perform Measurement Analysis". Another noteworthy difference is related to the ontological foundations for the relations between **Measurable Entity** and **Measurable Element**. In [6], a Measurable Element (a *Quality Universal* in UFO) characterizes a Measurable Entity Type (a *High Order Universal* in UFO); and a Measurable Entity (a *Measurable Universal* in UFO) is an instance of a Measurable Entity Type. However, in UFO, a *Measurable Universal* is a *Universal* that is characterized by at least one *Quality Universal*:

$$\forall x\ Measurable\_Universal(x) \rightarrow Universal(x) \wedge \exists q\ Quality\_Universal(q) \wedge\ characterized\_by(x,q).$$

So, a **Measurable Entity** is characterized by a **Measurable Element**, as shown in Fig. 1. Moreover, we included a relation between **Measure** and **Measurable Entity** (used to measure), in order to capture that a certain **Measure** quantifies **Measurable Elements** of a given **Measurable Entity**, as constrained by the following axiom:

$$\forall mea, mel\ quantifies(mea, mel) \rightarrow\ Measure(mea) \wedge Measurable\ Element(mel) \wedge \exists men\ Measurable\_Entity(men) \wedge characterized\_by(men,mel) \wedge\ used\_to\_measure(mea, mel).$$

Our main intended use of the SMTO is to serve as a reference model for semantically integrating software measurement-related standards, as well as software measurement tools. Concerning these aspects, there are also related works.

Regarding the integration of software measurement-related standards, there are some works that seek to establish mappings between them, such as [17] and [18]. In [17], Oliveira et al. map measurement activities from CMMI-SW (version 1.1), ISO/IEC 15939 (1st edition), IEEE Std 1061, Six Sigma, and PSM. Although this work compares several standards and presents similarities and differences among their activities, the authors do not discuss a consensual view of the software measurement process, nor they use ontologies to support their mappings. Garcia and colleagues [18], on the other hand, use their Software Measurement Ontology (SMO), a domain ontology, to support terminology harmonization between some concepts of ISO/IEC 15939, ISO/IEC 14598-1, IEEE Std 1061, IEEE 610.12 and ISO International Vocabulary of Basic and General Terms in Metrology. Thus, the work presented in [18] is quite similar to the mappings shown in Table 1, although the resulting mapping are different, since the ontologies used are different. Moreover, Garcia et al. only harmonized terms (and not activities/practices) from different standards. At the best of our knowledge, there is not any work that uses a software measurement task ontology for integrating activities described in measurement standards.

Concerning the integration of software measurement tools, we can see it as a special case of Enterprise Application Integration (EAI). As pointed in [5], ontologies are at the heart of the modern approaches for semantic EAI. However, the vast majority of the existing approaches to semantic EAI use only domain ontologies [19]. This is the case, for instance, of ONAR, an ontology-based framework for Enterprise Application Integration [20]. An exception is the work done by Calhau and Falbo [19], who developed a task ontology for the Configuration Management (CM) process, and used it in a semantic integration effort aiming at integrating two CM supporting systems. In this paper, we followed the same development approach as Calhau and Falbo, and as a consequence the structure of our task ontology is quite similar to the one presented in [19]. Moreover, we intend to explore the use of the SMTO for semantically integrating tools supporting the Software Measurement process, following the same approach adopted in [19].

6. CONCLUSIONS

Although the use of task ontologies is still timid when compared to the use of domain ontologies, task ontologies have increasingly received attention [19]. In this paper, we present a task ontology that aims to capture the conceptualization involved in the Software Measurement (SMEA) process. Our purpose in developing this task ontology is to make the best possible description of the SMEA process, seeking for a consensus within the related community. Thereafter, our Software Measurement Task Ontology (SMTO) is a solution-independent specification that seeks to make a clear and precise description of the entities and events in this universe of discourse, for the purposes of communication, learning and problem-solving. In other words, SMTO is a reference ontology. As advocated by Guizzardi [10], a reference ontology should take the ontological distinctions given by a foundational ontology into account. To meet this requirement, we grounded SMTO in the Unified Foundational Ontology (UFO) [11, 12]. For each entity we decided to include in SMTO, an ontological analysis at the light of UFO was done, trying to ground the SMTO entity in a UFO entity. Moreover, we inspected other UFO entities that were related to the one that was perceived as to be the ground for the SMTO entity, and sought for other related entities. This approach aided us to extract tacit knowledge that was implicit in the models.

As knowledge sources for eliciting the knowledge involved in the SMEA process, we used several international standards, such as ISO/IEC 12207 [1] and ISO/IEC 15939 [3], process quality models, such as CMMI [2], books and handbooks devoted to the subject, such as [4], and also our experience in working in this universe of discourse in practice. Moreover, experts in this universe of discourse have also collaborated with us in order to establish a shared conceptualization that is put explicit in the resulting ontology. We also looked for ontologies describing the SMEA universe of discourse, but we only found domain ontologies [6, 7, 8, 13]. The Reference Software Measurement (domain) Ontology (RSMO) developed by Barcellos and colleagues [6, 7, 8] was very useful, especially for defining the SMTO structural model. RSMO is already grounded in UFO, and this made us save some efforts. It is worthwhile to point out, however that, although the SMTO structural model is very in line with RSMO, it presents also some differences, as discussed in the previous section.

The intended use of SMTO is to serve as a reference model for semantically integrating software measurement-related standards, as well as software measurement tools. In this paper, we used it to harmonize the software measurement processes described in ISO/IEC 12207 [1], ISO/IEC 15939 [3], CMMI [2] and PSM [4]. This purpose of this effort is twofold: first, we are exploring the first intended use of SMTO, i.e., to assist in semantically integrating software measurement-related standards; second, this effort served also to evaluate SMTO. Since SMTO is able to map the main standards in the area, we believe that it captures the main conceptualization of the SMEA process. As future work, we intend to use SMTO as a reference model for semantically integrating software measurement tools.

7. ACKNOWLEDGMENTS

This research is funded by the Brazilian Research Funding Agencies FAPES (Process Number 52272362/11) and CNPq (Process Number 483383/2010-4).

8. REFERENCES

[1] ISO/IEC. *ISO/IEC 12207: System and software engineering – Software life-cycle processes*, Second edition, 2008.

[2] Software Engineering Institute. *CMMI for Development, Version 1.3*, Technical Report CMU/SEI-2010-TR-033, 2010.

[3] ISO/IEC. *ISO/IEC 15939: Software Engineering – Software Measurement Process*, Second edition, 2007.

[4] Mcgarry, J., Card, D., Jones, C., Layman, B., Clark, E., Dean, J. and Hall, F. *Practical Software Measurement: Objetive Information for Decision Makers*, Addison Wesley, 2002.

[5] Izza, S. Integration of industrial information systems: from syntactic to semantic integration approaches. *Enterp. Inf. Syst.* 3, 1 (February 2009), 1-57.

[6] Barcellos, M. P., Falbo, R. A., Dal Moro, R. A well-founded software measurement ontology, In *Proceedings of the 6th International Conference on Formal Ontology in Information Systems (FOIS 2010),* 2010, 213-216.

[7] Barcellos, M. P., Falbo, R. A., Rocha, A.R., A well-founded software process behavior ontology to support business goals monitoring in high maturity software organizations, In *Proceedings of the IEEE International EDOC Enterprise Computing Conference Workshops*, 2010, 253-262.

[8] Barcellos, M. P., Falbo, R. A., Rocha, A.R., Establishing a well-founded conceptualization about software measurement in high maturity levels, In *Proceedings of the 7th International Conference on the Quality of Information*, 2010, 467-472.

[9] Guarino, N. Formal Ontology and Information Systems. In: *Formal Ontologies in Information Systems*, IOS Press, 1998, 3 -15.

[10] Guizzardi, G. On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models, In *Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV*, IOS Press, Amsterdam, 2007.

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

[12] 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* (Brazil, 2008), 244-251.

[13] García, F., Bertoa, M.F., Calero, C., Vallecillo, A., Ruiz, F., Piatini, M., Genero, M., Towards a consistent terminology for software measurement. *Information and Software Technology,* 48, 2006, 631–644.

[14] Martins, A.F., Falbo, R.A. Models for representing task ontologies. In *Proceedings of the 3rd Workshop on Ontologies and Their Applications*, Brazil, 2008.

[15] ISO/IEC. *ISO/IEC 90003: Software Engineering – Guidelines for the application of ISO 90001:2000 to computer software*, 2004.

[16] Ferreira, A.L., Machado, R.J., Paulk, M.C. *Size and complexity attributes for multimodel improvement framework taxonomy*, In Proceedings of the *36th EUROMICRO Conference on Software Engineering and Advanced Applications*, 2010, 306-309.

[17] Oliveira, J.M.M., Oliveira, K.B., Belchior, A.D. Measurement process: a mapping among CMMI-SW, ISO/IEC 15939, IEEE Std 1061 and PSM. In *Proceedings of the International Conference on Service Systems and Services Management*, 2006, 810-815.

[18] García, F., Ruiz, F., Calero, C., Bertoa, M.F., Vallecillo, A., Mora, B., Piatini, M., Genero, M., Effective use of ontologies in software measurement. *The knowledge Engineering Review*, 24:1, 2009,23–40.

[19] Calhau, R.F., Falbo, R.A. A configuration management task ontology for semantic integration. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing (ACM-SAC'2012)*, 2012, 348 – 353.

[20] Tektonidis, D., Bokma, A., Oatley, G., Salampasis, M. ONAR - An Ontology-based Service Oriented Application Integration Framework, In *Proceedings of the 1st International Conference on Interoperability of Enterprise Software and Applications*, (Geneva, Switzerland, 2005), 65–74.