Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Information and Software Technology 48 (2006) 631–644
INFORMATION AND SOFTWARE TECHNOLOGY
www.elsevier.com/locate/infsof

Towards a consistent terminology for software measurement

Félix García[a,*], Manuel F. Bertoa[b], Coral Calero[a], Antonio Vallecillo[b], Francisco Ruíz[a], Mario Piattini[a], Marcela Genero[a]

[a]*Alarcos Research Group, Department of Computer Science, University of Castilla-La Mancha, Paseo de la Universidad/4, 13071, Ciudad Real, Spain*
[b]*Department Lenguajes y Ciencias de la Computación, Universidad de Málaga, Spain*

Received 29 March 2005; revised 28 June 2005; accepted 4 July 2005
Available online 22 August 2005

Abstract

Although software measurement plays an increasingly important role in Software Engineering, there is no consensus yet on many of the concepts and terminology used in this field. Even worse, vocabulary conflicts and inconsistencies can be frequently found amongst the many sources and references commonly used by software measurement researchers and practitioners. This article presents an analysis of the current situation, and provides a comparison framework that can be used to identify and address the discrepancies, gaps, and terminology conflicts that current software measurement proposals present. A basic software measurement ontology is introduced, that aims at contributing to the harmonization of the different software measurement proposals and standards, by providing a coherent set of common concepts used in software measurement. The ontology is also aligned with the metrology vocabulary used in other more mature measurement engineering disciplines.

Keywords: Software measurement; Measurement terminology

1. Introduction

Software measurement plays an increasingly important role in Software Engineering. Currently, software metrics are proving to be very effective for building high-quality prediction systems for large database projects [1], understanding and improving software development and maintenance projects [2], assessing and maintaining system quality by highlighting problematic areas [3], determining the best ways to help practitioners and researchers in their work [4], etc. Furthermore, software metrics are important tools to help assess and institutionalize Software Process Improvement in software-intensive organizations. In fact, software measurement is a cornerstone piece of initiatives such as SW-CMM (Capability Maturity Model for Software), ISO/IEC 15504 (SPICE, Software Process Improvement and Capability dEtermination) and CMMI (Capability Maturity Model Integration). The ISO/IEC 90003:2004 standard [5] also emphasizes the importance of measurement in quality assurance and management.

Standardization also plays a key role in Software Engineering, and in particular in the software measurement field. Standards provide organizations with agreed and well-recognized practices and technologies, which assist them to interoperate and to work using engineering methods, reinforcing software engineering as an 'engineering' discipline, instead of a 'craft'. In addition, the Internet is changing the way of doing business today, promoting co-operation and interoperation among individual organizations, which need to compete in a global market and economy, and share information and resources. Standardization is one of the driving forces to achieve that interoperability, with the provision of agreed domain conventions, terminologies and practices.

However, software measurement suffers from the typical symptoms of any relatively young discipline [2]. Despite all the efforts and new developments in research and international standardization during the last decade, software measurement is currently in the phase in which terminology, principles, and methods are still being defined, consolidated, and agreed. In particular, there is no

* Corresponding author.
E-mail address: felix.garcia@uclm.es (F. García).

0950-5849/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2005.07.001

consensus yet on the concepts and terminology used in this field. For instance, software measurement researchers and practitioners have not reached an agreement on the precise meaning of some terms commonly used, such as 'measurement', 'measure', 'metric', 'measurable attribute', etc. Even worse, inconsistencies between the different research measurement proposals often occur.

The situation is not much better if we take a look at the current software engineering international standards produced by the major standardization bodies and organizations, such as IEEE, ISO and IEC. Inconsistencies and terminology conflicts appear not only between standards from different bodies, but also within those from the same organization. Besides, no single standard contains a complete vision of software measurement; all of them offer just partial views of it, e.g. on the metrics, on the measurement processes, or on the target entities and objectives of the measurements. This issue has been recognized by ISO/IEC, which has created a work group for the harmonization of Systems Engineering Standards within its Joint Technical Committee 1 (JTC1: 'Information Technology', www.jtc1.org), and is trying to explicitly include in its directives the procedures that guarantee the consistency and coherency among its standards. ISO's effort to harmonize the measurement terminology started about 5 years ago, with the selection of the terminology of ISO International Vocabulary of Basic and General Terms in Metrology (VIM [6]) for the ISO/IEC TR 14143-3, the ISO/IEC 15939 standard, and for other future SC7 measurement-related documents. Furthermore, there is an agreement since year 2002 between the IEEE Computer Society and ISO-JTC1-SC7 to harmonize their standards, which includes the terminology on measurement. However, from our point of view the situation is still far from being resolved, as we shall later see.

With the goal of contributing to the harmonization of the different software measurement standards and research proposals, this article presents a comparison analysis of the concepts and terms used in them. Commonalities, discrepancies, gaps, and terminology conflicts are identified, and a unifying proposal is presented. In this way, we try to contribute to the provision of a 'consistent' terminology for software measurement.

By consistent we mean both generally agreed (i.e. with consensus) and coherent (i.e. without conflicts and contradictions). ISO defines consensus as a 'general agreement, characterized by the absence of sustained opposition to substantial issues by any important part of the concerned interests' (www.jtc1.org). Consistent, as defined in the Merriam-Webster Online Dictionary (m-w.com), means 'marked by harmony, regularity, or steady continuity: free from variation or contradiction'.

This proposal tries to address the needs of two main audiences. First, software measurement practitioners, who may be confused by the terminology differences and conflicts in the existing standards and proposals. And second, software measurement researchers and standard developers (e.g. international standardization bodies and committees), who do not currently count with a cohesive core set of concepts and terms over which their existing standards could be integrated, or new ones built.

This paper is organized as follows. After this introduction, Section 2 presents an analysis of the current situation, where some of the most representative research proposals and international standards are compared. Then, Section 3 introduces a common ontology for software measurement, which provides a cohesive set of concepts and the relationships between them, and that aims at serving as unifying framework for the rest of the proposals. The comparison analysis between the different proposals and the common ontology is presented in Section 4, using a set of tables that allow the easy identification of the differences and conflicts between them, and the solution proposed by the ontology. Finally, Section 5 draws some conclusions, proposes some suggestions for harmonization, and identifies future research work.

2. Analysis of the current situation

We selected sources from the existing international standards and research proposals that deal with software measurement concepts and terminology.

From IEEE we took IEEE Std. 610.12-1992 (Standard Glossary of Software Engineering Terminology) and IEEE Std. 1061-1998 (IEEE Standard for a Software Quality Metrics Methodology).

From ISO and IEC we selected the ISO/IEC 14598:2001 series (Software engineering-Product evaluation), the ISO VIM (International Vocabulary of Basic and General Terms in Metrology [6]), and the International Standard ISO/IEC 15939 (Software engineering-Software measurement process).

We also included other relevant research proposals related to software measurement, such as the ones by Lionel Briand et al. [2] and by Barbara Kitchenham et al. [7]. The general enterprise ontology proposed by Henry Kim [8] was also considered in the analysis, since it contains a sub-ontology for measurement concepts and terms. Other proposals that make use of measurement terminology (sometimes adapted to their particular domains) were also analyzed, although they were not included in the comparative study because they were either too specific, or clearly influenced by other major proposals already considered.

The first thing we realized is that the different standards and proposals could be basically organized around three main groups, depending on the particular measurement topics they focused on: software measures, measurement processes, and targets-and-goals. The first group of concepts, software measures, deals with the main elements involved in the definition of software measures, including terms such as measure, scale, unit of measurement, etc.

The second group, processes, is related with the actions of measuring software products and processes, including the definition of terms like measurement, measurement result, measurement method, etc. Finally, the third group, target-and-goals, gathers the concepts required to establish the scope and objectives of the software measurement process, e.g. quality model, measurable entity, attribute, information need, etc.

It is worth noting that no single source covers all three groups. Moreover, the set of concepts covered by each source is not homogeneous, even for those sources focusing on the same group. There is a tendency in the sources, however, to converge around these three topic groups as they evolve over time. This is why the different sources we analyzed are presented here in chronological order.

2.1. IEEE Std. 610.12 (1992)

This is a glossary of Software Engineering terminology. It does not include terms specifically related to software measurement, or terms that can be inferred from their common English meaning.

For instance, the terms 'entity', 'attribute' and 'metric' are defined, but without a special focus on software measurement as such. Besides, no other term related to the software measurement was found in this standard.

2.2. VIM (1993)

The International Vocabulary of Basic and General Terms in Metrology covers 120 terms of subjects related to measurement. Although its main target is not software, it has been successfully used by several authors, such as Alain Abran, for defining software measurement concepts [9], and is one of the basis for ISO-JTC1 software measurement harmonization efforts.

The VIM is a very detailed, complete and mature reference. However, its terms remain at a very detailed level for instance, there are no definitions for general terms such as 'metric' or 'measure'. The new version of the VIM, currently in preparation, is expected to deal with the software measurement specific needs.

2.3. IEEE Std. 1061 (1998)

The IEEE Std. 1061–1998 is a standard for a software quality metrics methodology. This standard is a revision of IEEE Std. 1061–1992. It provides a methodology for establishing quality requirements, and for identifying, implementing, analyzing, and validating process and product software quality measures. This methodology applies to all software at all phases of any software life cycle, but without prescribing specific metrics.

This standard covers concepts from all three groups (measures, processes, and target-and-goals), but not in a complete and exhaustive manner-which is not the main goal of the standard, anyway.

Interestingly, we found out discrepancies between this standard and IEEE 610.12, even when they are produced and maintained by the same organization. For example, the IEEE 610.12 definition of 'metric' differs from the definition given in IEEE 1061. This is even explicitly acknowledged in the latter standard.

2.4. Kim (1999)

Henry Kim [8] proposes a formal model of enterprise quality, called 'Ontology of enterprise quality modeling'. This is a global ontology, whose main objective is to help evaluate the conformance of organizations to ISO/IEC 9000 standards. As part of his global ontology, Kim also proposes a measurement ontology.

Although Kim's measurement ontology is not specific to software products and processes, it contains many concepts that can be applied within the context of software measurement. Under this perspective, Kim's proposal mainly focuses on targets-and-goals, including concepts such as 'quality requirement', 'entity', 'enterprise model of quality', and 'measured attribute'. It does not define, however, concepts such as 'measure', 'metric' or 'scale', for instance.

2.5. ISO/IEC 14598 (1999–2001) and 9126 (2001–2004)

ISO/IEC 14598 (Information technology-Software product evaluation) is a series of international standards that provide methods for measurement, assessment and evaluation of software product quality. The different parts of this series set out a generic picture of the process of evaluation, addressing it from the point of view of developers, acquirers and (third party) evaluators.

The standards of ISO/IEC 14598 series are mainly concerned with the set of concepts in the measures group, and partially covering some of the measurement process aspects. ISO/IEC 14598 series makes use of the ISO/IEC 9126 series (Software engineering-Product quality-Parts 1–4), which propose a software product quality model, and metrics for internal quality, external quality, and quality in use.

Both ISO/IEC 9126 and ISO/IEC 14598 series share a common terminology, and are currently under revision. The SQuaRE project [10] has been specifically created to make them converge, trying to eliminate the gaps, conflicts, and ambiguities that they currently present. In fact, ISO/IEC TR 9126-2, 9126-3 and 9126-4 were allowed to be published as Technical Reports between 2002 and 2004 without changing their original terminology, with the agreement that they would be aligned with the new SC7 measurement terms as soon as possible. The ISO/IEC 25000 series will be the end result of this convergence project.

2.6. Kitchenham et al. (2001)

Barbara Kitchenham et al. [7] propose a method for specifying models of software data sets in order to capture the definitions and relationships among software measures.

They propose a conceptual model with three components. First, the generic component defines concepts such as attributes, units, and scale types, independently from other considerations. The development model provides the link between measures and entities of interest. Finally, the project domain represents the data values collected from real projects, linking data values to actual instances of the entities that are defined in the development model domain.

This proposal is mainly concerned with both measures and targets-and-goals, but without considering the measurement process in detail. Besides, their terminology is not completely aligned with the rest of the standards and proposals. For instance, the concept of 'measure' is represented by the term 'DM element measure type', which significantly differs from the terms 'metric' or 'measure' probably the most commonly used terms in the rest of the sources for representing this concept.

2.7. Briand et al. (2002)

Lionel Briand et al. propose the GQM/MEDEA approach for defining measures of product attributes in software engineering. This approach is driven by the experimental goals of measurement, expressed via the GQM [11] paradigm and a set of empirical hypotheses.

This proposal provides a UML class diagram with the concepts involved in the GQM/MEDEA process. Those GQM/MEDEA concepts related to software measurement are mainly concerned with measurement targets-and-goals (e.g. entity, attribute). It does not consider, for instance, the concepts 'measurement' or 'scale', and does not distinguish between base and derived measures either.

One of the specific characteristics of this proposal is that its concepts are not defined, but just presented for their use in the GQM/MEDEA process. This forced us to guess their real meaning when including them in the comparison analysis.

2.8. ISO/IEC 15939 (2002) and PSM (2002)

ISO/IEC 15939 standard identifies the activities and tasks needed to successfully identify, define, select, apply, and improve software measurement within an overall project or organizational measurement structure. It also provides definitions for measurement terms commonly used within the software industry.

The two key components included in this standard are software measurement process and measurement information model. The software measurement process is driven by the information needs of the organization. For each information need, this process produces an information product that tries to satisfy it.

The measurement information model establishes the link between measures and information needs. Measured entities include processes, products, projects, and resources. The model describes how the relevant attributes are quantified, and converted to indicators that provide a basis for decision-making.

ISO/IEC 15939 focuses mainly on the measurement process concepts, although it also covers many of the concepts of the other two groups (measures and targets-and-goals). For measures, it basically rests upon the concepts of ISO/IEC 14598 and ISO/IEC 9126-although changing some of the terms in order to be aligned as much as possible with the ISO VIM. Hence, it does not use the term 'metric', relating directly the terms 'measurement' and 'measure'.

ISO/IEC 15939, together with VIM, has become the standard used by ISO-JTC1 as the basis for its software measurement terminology harmonization efforts [12].

Another key reference, the PSM (Practical Software Measurement) [13], is compatible with ISO/IEC 15939, and therefore uses the same terminology.

2.9. Other models, standards and proposals

Software measurement is also part of other models and standards. Probably the most representative examples include process assessment and maturity models, such as CMMI and ISO/IEC 15504, the Software Lifecycle 12,207 standard, and the ISO/IEC functional size measurement family of standards. They have not been considered in the comparison analysis because either they use the terminology defined by other proposals already included in this study, or they define a very specific vocabulary for particular domains.

This paper does not mention other ISO/IEC measurement related standards and Technical Reports, such as the ISO/IEC 14143 series (Parts 1–6), which are aligned with the VIM, and the four other ISO/IEC standards for Functional Size Measurement methods, including ISO/IEC 19761 (COSMIC-FFP), which is totally aligned with the VIM, too.

- In CMMI, measurement is considered in a key process area, namely 'Measurement and Analysis'. CMMI adopts the ISO/IEC 15939 terminology, in particular 'base measures' and 'derived measures'. It also adapts some terms to process assessment, e.g. 'process measurement' and 'quantitative objective'.
- ISO/IEC 15504 uses its own measurement terminology adapted to the process assessment domain. It includes terms such as 'Software process goal', 'Software process metric', 'Software process target' and 'Software process current measurement'. It also uses 'Measure', 'Measurement' and 'Measurement Techniques', although without providing explicit definitions for these terms.

- The software measurement terminology used in ISO/IEC 12207 is directly adopted from the ISO/IEC 9126 series.
- The ISO/IEC functional size measurement family of standards (including the 14143 series and ISO/IEC 19761, COSMIC-FFP) is totally aligned with the VIM.
- Finally, some authors have tried to adapt the concepts of measurement theory to the realm of software measurement. However, the application of measurement theory has not contributed to alleviate the terminology problems. For instance, measurement theory distinguishes between the concepts 'measure' and 'metric' (a measure is 'an homomorphism which is a structure-preserving mapping from the empirical world to the world of numbers' and a metric is 'a criterion to determine the distance between two entities' [14]). However, this distinction is lost when applied to software measurement; some authors claim that measures can be seen as metrics, and therefore the difference between these two concepts is not essential. The terms measure and metrics are also used in mathematics, but with a very specific mathematical meaning, and not in the general sense used in software measurement (a 'measure' is a function, and a 'metric' is a special kind of measure that serves to calculate distances between points in a metric space).

2.10. Issues found

Summarizing, the following issues were found when the different sources were compared:

- Homonymy (e.g. the term 'metric' means two different things in IEEE 610.12 and IEEE 1061 standards)
- Synonymy (e.g. Briand's term 'measure' and ISO/IEC 14598 term 'metric' seem to represent the same concept)
- Not all terms and concepts appear in all proposals, even if the proposals focus on the same group of concepts (e.g. the term and the concept of 'metric' do not appear in ISO/IEC 15939-they seem to be integrated within other terms and concepts)
- There is no uniform treatment of some of the basic software measurement concepts such as 'base measures', 'derived measures' or 'indicators'. There are no agreements on the approaches to measure them, either (measurement method, measurement function and analysis model)
- The fact that no standard or proposal covers all concepts hinders their integration (same problem you face when you have to integrate different views of a database, but without a common schema).

Since the standards and research proposals were developed by different parties, and on different timelines, it is normal that such differences crop up between them. This is why a harmonization effort that provides a common view and irons out their differences and conflicts is required.

3. A common ontology

With our comparison analysis we pursued the following goals:

- locate and identify synonyms, homonyms, gaps and conflicts;
- generalize the different approaches to measuring attributes;
- provide a smooth integration of the concepts from the three groups, so measurement processes can be built using clearly defined measures, while quality models identify the target and goals of the measurement processes.

A natural approach to achieve such goals is by using a common software measurement ontology, able to identify all concepts, provide precise definitions for all the terms, and clarify the relationships between them. Such a common ontology can serve as the basis for comparing the different standards and proposals, thus helping achieve the required harmonization and convergence process for all of them.

Another important requirement for such a software measurement ontology is that its terms should try to conform to general terminology accepted in other fields-and this includes measurement, which is a quite mature field which has a very rich set of terminology terms. This is also consistent with ISO/IEC and IEEE Computer Society current positions which, in order to ensure both consensus and consistency with the other fields of sciences, made a decision in year 2002 to align their terminologies on measurement with the internationally accepted standards in this field. In particular, ISO-JTC1-SC7 is trying to follow as much as possible the ISO International Vocabulary of basic and general terms on Metrology (VIM).

In this section we present a proposal that we have developed for these purposes, based on an initial proposal by Félix García et al. [15]. Initially, that ontology was created to address the lack of consensus on the Spanish software measurement terms, based on the most representative measurement standards and proposals. Once the Spanish ontology was defined, it was translated into English. Finding the right translation of each Spanish term became a rather difficult task and was done by comparing the different proposals again, and selecting the most appropriate terms in each case. However, once the ontology was created we discovered that it was not fully aligned with the VIM and with the new harmonization efforts happening at ISO. Therefore, the authors of this paper decided to adapt it in order to make it converge with these efforts, and thus the ontology presented here was built.

The resulting software measurement ontology is then based mainly on the ISO VIM and ISO/IEC 15939 standards. It also includes some missing terms in these two documents (e.g. 'quality model') that we think are key in software measurement, and presents some discrepancies with 15939, e.g. the treatment of indicators (this issue will be discussed later in Section 3.2.2).

In particular, the main features and characteristics of the Software Measurement Ontology (SMO) are the following:

- Uses the term 'measure' instead of 'metric'.
- Differentiates between 'measure', 'measurement', and 'measurement result'.
- Distinguishes between base measures, derived measures, and indicators, but considering them all as measures-and generalizing their respective measurement approaches (measurement method, measurement function and analysis model).
- Integrates the software measures with the quality model that defines the information needs that drive the measurement process.

To represent the SMO we have chosen REFSENO (Representation Formalism for Software Engineering Ontologies) [16]. REFSENO provides constructs to describe concepts (each concept represents a class of experience items), their attributes, and relationships. Three tables are used to represent these elements: one table with the glossary of concepts, one table of attributes, and one table with the relationships. REFSENO also allows the description of similarity-based retrievals, and incorporates integrity rules such as cardinalities and value ranges for attributes, and assertions and preconditions on the elements instances.

Several main reasons moved us to use REFSENO for defining our ontology. First, REFSENO was specifically designed for software engineering, and allows several representations for software engineering knowledge-whilst other approaches, e.g. [17–19], only allow representations which are less intuitive for people not familiar with first-order predicate (or similar) logics. In addition, REFSENO has a clear terminology, differentiating between conceptual and context-specific knowledge, and thus enabling the management of knowledge from different contexts. REFSENO also helps building consistent ontologies thanks to the use of consistency criteria. Unlike other approaches, REFSENO uses constructs known from Case-Based Reasoning (CBR). Finally, REFSENO stores experience in the form of documents, and not as codified knowledge.

This results in an important reduction of the learning effort, something typically associated with knowledge-based systems [20].

3.1. The software measurement ontology

The following tables provide a summary of the REFSENO representation of the Software Measurement

Table 1
Definition of the terms in the SMO (I)

SO	Term	Supercon	Definition	Source
1	Information need	Concept	Insight necessary to manage objectives, goals, risks, and problems	15939
1	Measurable concept	Concept	Abstract relationship between attributes of entities and information needs	15939
1	Entity	Concept	Object that is to be characterized by measuring its attributes	15939
1	Entity class	Concept	The collection of all entities that satisfy a given predicate	New
1	Attribute	Concept	A measurable physical or abstract property of an entity, that is shared by all the entities of an entity class	Adapted from 14598
1	Quality model	Concept	The set of measurable concepts and the relationships between them which provide the basis for specifying quality requirements and evaluating the quality of the entities of a given entity class	Adapted from 14598
2	Measure	Concept	The defined measurement approach and the measurement scale. (A measurement approach is either a measurement method, a measurement function or an analysis model)	Adapted from 14598 'metric'
2	Scale	Concept	A set of values with defined properties	14598
2	Type of scale	Concept	The nature of the relationship between values on the scale	15939
2	Unit of measurement	Concept	Particular quantity, defined and adopted by convention, with which other quantities of the same kind are compared in order to express their magnitude relative to that quantity	VIM
2	Base measure	Measure	A measure of an attribute that does not depend upon any other measure, and whose measurement approach is a measurement method	Adapted from 14598 'direct metric'
2	Derived measure	Measure	A measure that is derived from other base or derived measures, using a measurement function as measurement approach	Adapted from 14598 'indirect metric'
2	Indicator	Measure	A measure that is derived from other measures using an analysis model as measurement approach	New

Table 2
Definition of the terms in the SMO (II)

SO	Term	Supercon	Definition	Source
3	Measurement method	Measurement approach	Logical sequence of operations, described generically, used in quantifying an attribute with respect to a specified scale. (A measurement method is the measurement approach that defines a base measure)	Adapted from 15939
3	Measurement function	Measurement approach	An algorithm or calculation performed to combine two or more base or derived measures. (A measurement function is the measurement approach that defines a derived measure)	Adapted from 15939
3	Analysis model	Measurement approach	Algorithm or calculation combining one or more measures with associated decision criteria. (An analysis model is the measurement approach that defines an indicator)	Adapted from 15939
3	Decision criteria	Concept	Thresholds, targets, or patterns used to determine the need for action or further investigation, or to describe the level of confidence in a given result	15939
4	Measurement approach	Concept	Sequence of operations aimed at determining the value of a measurement result. (A measurement approach is either a measurement method, a measurement function or an analysis model)	new
4	Measurement	Concept	A set of operations having the object of determining a value of a measurement result, for a given attribute of an entity, using a measurement approach	Adapted from VIM
4	Measurement result	Concept	The number or category assigned to an attribute of an entity by making a measurement	Adapted from 14598 'Measure'

Ontology by describing their concepts and relationships. For simplicity, we have grouped all the SMO terms in two tables (Tables 1 and 2), their relationships in four tables (Tables 3–6), and have omitted the description of the concepts' attributes. In addition, Fig. 1 shows the graphical representation of the SMO terms and relationships, using the UML (Unified Modeling Language).

The SMO has been organized around four main sub-ontologies:

(1) *Software Measurement Characterization and Objectives*, which includes the concepts required to establish the scope and objectives of the software measurement process.

The main goal of a software measurement process is to satisfy certain information needs by identifying the entities (which belong to entity classes) and the attributes of these entities (which are the focus of the measurement process). Attributes and information needs are related through measurable concepts (which belong to a quality model).

(2) *Software Measures*, which aims at establishing and clarifying the key elements in the definition of a software measure.

A measure relates a defined measurement approach and a measurement scale (which belongs to a type of scale).

Most measures can be expressed in a unit of measurement, and can be defined for more than one attribute. (Nominal measures are examples of measures that cannot be expressed in units of measurement.) Three kinds of measures are distinguished: base measures, derived measures, and indicators.

Table 3
Relationships in the 'software measurement characterization and objectives' sub-ontology

Name	Concepts	Description
Includes	Entity class–entity class	An entity class may include several other entity classes. An entity class may be included in several other entity classes
Defined for	Quality model–entity class	A quality model is defined for a certain entity class. An entity class may have several quality models associated
Evaluates	Quality model–measurable concept	A quality model evaluates one or more measurable concepts. A measurable concept is evaluated by one or more quality models
Belongs to	Entity–entity class	An entity belongs to one or more entity classes. An entity class may characterize several entities
Relates	Measurable concept–attribute	A measurable concept relates one or more attributes
Is associated with	Measurable concept–information need	A measurable concept is associated with one or more information needs. An information need is related to one measurable concept
Includes	Measurable concept–measurable concept	A measurable concept may include several measurable concepts. A measurable concept may be included in several other measurable concepts
Composed of	Entity–entity	An entity maybe composed of several other entities
Has	Entity class–attribute	An entity class has one or more attributes. An attribute can only belong to one entity class

Table 4
Relationships of the 'software measures' sub-ontology

Name	Concepts	Description
Belongs to	Scale–type of scale	Every scale belongs to a type of scale. A type of scale may characterize several scales
Defined for	Measure–attribute	A measure is defined for one or more attributes. An attribute may have several associated measures
Transformation	Measure–measure	Two measures can be related by a transformation function; the kind of function will depend on the scale types of the scales
Expressed in	Measure–unit of measurement	A measure is expressed in one unit of measurement (only for measures whose type is interval or ratio). A unit of measurement is used to express one or more measures of interval or ratio types
Has	Measure-scale	Every measure has a scale. A scale may serve to define more than one measures

Table 5
Relationships in the 'measurement approaches' sub-ontology

Name	Concepts	Description
Calculated with	Derived measure–measurement function	Every derived measure is calculated with one measurement function. Every measurement function may define one or more derived measures
Calculated with	Indicator–analysis model	Every indicator is calculated with one analysis model. Every analysis model may define one or more indicators
Uses	Base measure–measurement method	Every base measure uses one measurement method. Every measurement method defines one or more base measures
Satisfies	Information need–indicator	An indicator may satisfy several information needs. Every information need is satisfied by one or more indicators
Uses	Measurement function–derived measure	A measurement function may use several derived measures. A derived measure may be used in several measurement functions
Uses	Measurement function–base measure	A measurement function may use several base measures. A base measure may be used in several measurement functions
Uses	Analysis model–measure	An analysis model uses one or more measures. A measure may be used in several analysis models
Uses	Analysis model–decision criteria	An analysis model uses one or more decision criteria. Every decision criteria is used in one or more analysis models

(3) *Measurement Approaches*. This sub-ontology introduces the concept of measurement approach to generalize the different 'approaches' used by the three kinds of measures for obtaining their respective measurement results. A base measure applies a measurement method. A derived measure uses a measurement function (which rests upon other base and/or derived measures). Finally, an indicator uses an analysis model (based on a decision criteria) to obtain a measurement result that satisfies an information need.

(4) *Measurement*. It establishes the terminology related with the act of measuring software. A measurement (which is an action) is a set of operations having the object of determining the value of a measurement result, for a given attribute of an entity, using a measurement approach. Measurement results are obtained as the result of performing measurements (actions).

Please notice that these four sub-ontologies are closely related to the three main groups of concepts identified

Table 6
Relationships in the 'measurement' sub-ontology

Name	Concepts	Description
Performs	Measurement–measurement approach	A measurement is the action of performing a measurement approach (the kind of measurement approach will be dictated by the kind of measure used for performing the measurement). A measurement approach may be used for performing several measurements
Produces	Measurement–measurement result	Every measurement produces one measurement result. Every measurement result is the result of one measurement
Is performed on	Measurement–attribute	Every measurement is performed on one attribute of an entity (the attribute should be defined for the entity class of the entity)
Is performed on	Measurement–entity	Every measurement is performed on an entity, through one of its attributes (the attribute should be defined for the entity class of the entity)
Uses	Measurement–measure	Every measurement uses one measure. One measure may be used in several measurements

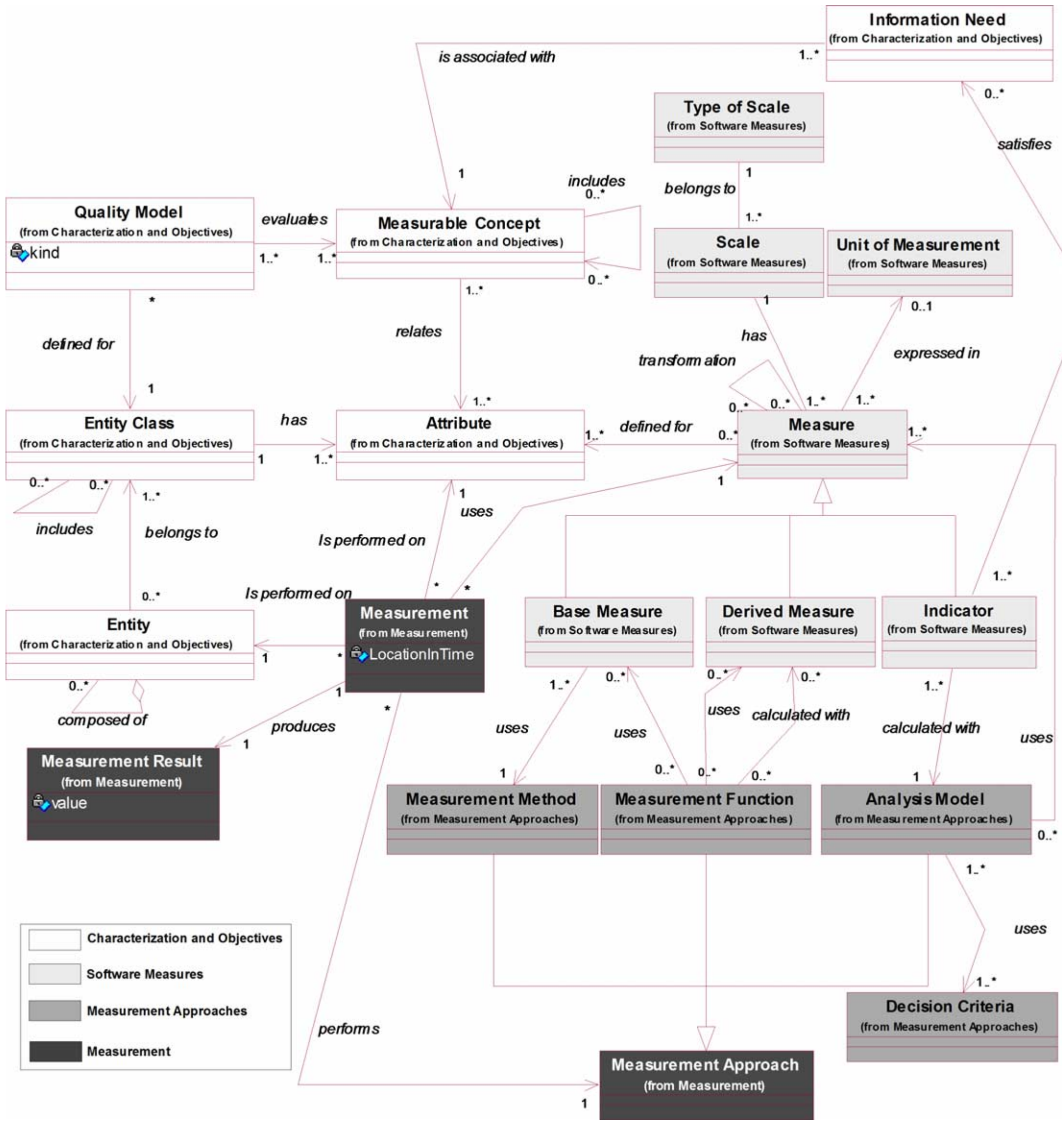

Fig. 1. UML diagram of the software measurement ontology (SMO).

above. Thus, the first sub-ontology corresponds to the target-and-goals group. The software measures sub-ontology corresponds to the measures group. The last two sub-ontologies together cover the measurement process group.

The precise definitions of the concepts included in the Ontology is presented in Tables 1 and 2. These two tables are indexed by their left column, which contains the number of the sub-ontology to which the term belongs. Then, columns two and three show the term being described and its super-concept in the ontology, respectively. Column four contains the definition of the term in the SMO. Finally, column five shows the source (standard or proposal) where the term has been adopted from.

Possible values in the fifth column can be either:

- a reference to a source (e.g. 15939, VIM, 14598), meaning that the term and its definition have been adopted from that source without any changes;
- 'Adapted from (source)', if the term has been borrowed from a source, but its definition has been slightly changed for completeness or consistency reasons;
- 'Adapted from (source) (other term)', if the definition of the term has been borrowed from a source, but that term is known differently in the source;
- new, if the term has been coined for the SMO, or has a new meaning in this proposal.

Section 4 discusses them in more detail.

3.2. Discussion

From all the unresolved issues that we have identified during our analysis, there are two that deserve special discussion: the use of the term metric, currently being disliked by many measurement experts; and the question whether indicators are particular kinds of measures or not.

3.2.1. 'Metric' vs. 'measure'

One of the most controversial issues amongst the software measurement experts nowadays is the use of the term metric. Although widely used and accepted by many practitioners and researchers, this term also counts with many detractors, that argue the following reasons against its use. First, formally speaking a metric is a function that measures the distance between two entities-and therefore it is defined with the precise mathematical properties of a distance. Second, the definition of metric provided by both general and technical dictionaries do not reflect the meaning with which it is informally used in software measurement. Furthermore, metric is a term that is not present in the measurement terminology of any other engineering disciplines, at least with the meaning it is commonly used in software measurement. Therefore, the use of the term 'software metric' seem to be imprecise, while the term 'software measure' seem to be more appropriate to represent this concept.

As a matter of fact, all new harmonization efforts at ISO/IEC and IEEE are trying to avoid the use of metric in order to be aligned with the rest of the measurement disciplines, which normally use the vocabulary defined in Metrology.

In our proposal we finally decided to avoid the use of the term metric.

However, the use of the term measure is not free from controversy in software measurement either-and this is what we personally believe is hindering its wider acceptance by the experts who currently support the use of 'metric'. For instance, ISO/IEC 15939 (so far the standard better aligned with the VIM terminology) defines measure as the result of the measurement. But then, this standard defines the terms base measure and derived measure as particular kinds of measure. As we can see, the concept of measure used in these two latter concepts do not seem to represent the result of the measurement, but the combination of the measurement scale and the measurement method (i.e., what ISO/IEC 14598 defines as metric). The problem with this such a concept is not defined in ISO/IEC 15939!

The situation is not better in ISO/IEC 14598 either. This standard defines metric as the combination of the measurement scale and the measurement method, and defines measure as the result of the measurement. However, it then inconsistently uses direct measure and indirect measure. That is, even when ISO/IEC 14598 defines them as kinds of measures, they are actually used as kinds of metrics.

We have managed to overcome this problem in the SMO by distinguishing between measure and measurement result. We think this is a successful solution since the term measure aggregates the measurement scale and the measurement approach (and therefore base measure and derived measure can be consistently defined and used), while the term measurement result contains the result of the measurement. Thus the term metric needs not be present in our proposal.

3.2.2. Is an indicator a kind of measure?

Both ISO/IEC 15939 and ISO/IEC 14598 introduce the term indicator. Although in both standards this term seems to represent the same concept, the definitions given in them differ significantly (Table 9).

ISO/IEC 15939 does not use any pre defined term (e.g. measure) for defining indicator. It is defined as 'an estimate or evaluation of specified attributes derived from a model with respect to defined information needs'. However, it is closely related with ISO/IEC 15939's base and derived measures.

In ISO/IEC 14598, an indicator is defined as a kind of measure (not a metric, i.e. it is a measurement result in our terms), although it seem to share the same properties of a metric in ISO/IEC 14598 terminology (i.e. a measure in our terms).

What we perceive is that both standards seem to agree that indicators are kinds of measures, but then fail to define this term as such, and use it consistently (in ISO/IEC 14598 it is defined as a measure, but then used as a metric, and in ISO/IEC 15939 it is not even defined as a measure). The reason seem to be again the ill-defined terms metric and measure in both standards (as discussed in the previous section).

In the SMO, our definition of measure allows to define indicators in a consistent way. They share the properties of measures (in SMO terms) because they have a scale and a measurement approach. In case of indicators, the measurement approach is an analysis model. In this sense, indicators are measures (according to the SMO's definition of measure).

4. Comparison analysis

The Software Measurement Ontology was used as the basis for the comparison analysis of the terminology used in the different proposals. The comparison analysis is presented here in four tables, one for each sub-ontology (Table 7–10).

Each table is indexed by its left column, which contains the term of the ontology being described. The second column shows the source (standard or proposal) where the term appears. The proposed definition of the concept represented by the term is shown in column three.

For each term, there may be more than one row in case of conflicts or discrepancies between the different sources. The first row contains (in italics) the definition of the concept according to the SMO. Usually, this term has been adopted from one of the standards, which will appear as the source of the term. The rest of the rows associated to a term describe alternative definitions for that term.

In those cases in which the concept begin represented is named by a different term (synonymy), the term is included in the third column, enclosed in square brackets before the definition of the concept.

It may also be the case of rows with no definition for a term-if the source uses it but does not provide an explicit definition for it.

Table 7
Comparison of the terms in the 'Measurement' sub-ontology

Term	Source	Definition
Measurement approach		Sequence of operations aimed at determining the value of a measurement result. (A measurement approach is either a measurement method, a measurement function or an analysis model.)
Measurement		A set of operations having the object of determining a value of a measurement result, for a given attribute of an entity, using a measurement approach
	15939,VIM	A set of operations having the object of determining a value of a measure
	14598-1	The use of a metric to assign a value (which can be a number or category) from a scale to an attribute of an entity
	Kim	The process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules Measurement can be discussed as the means by which an entity is determined to be of conforming quality
	IEEE 1061	The act or process of assigning a number or category to an entity to describe an attribute of an entity. A figure, extent, or amount obtained by measuring
Measurement result	14598-1	[Measure] the number or category assigned to an attribute of an entity by making a measurement
	15939	[Measure] variable to which a value is assigned as the result of a measurement
	Kitchenham	[Recorded value]
	Kim	[Measurement point] a value for a measured attribute of an entity at a given point in time
	Briand	
	IEEE 1061	(A) A way to ascertain or appraise value by comparing it to a norm. (B) To apply a metric
	IEEE 1061	[Metric value] a metric output or an element that is from the range of a metric

Table 8
Comparison of the terms in the 'software measurement characterization and objectives' sub-ontology

Term	Source	Definition
Information need	15939	Insight necessary to manage objectives, goals, risks, and problems
	Briand	[Corporative objective]
	Kim	[Quality requirement] an organizational constraint that has a bearing on the quality of an entity
Measurable concept	15939	Abstract relationship between attributes of entities and information needs
Entity	15939	Object that is to be characterized by measuring its attributes
	Briand	
	Kitchenham	[Project object occurrence]
	Kim	
	610.12	In computer programming, any item that can be named or denoted in a program. For example, a data item, program statement, or subprogram
Entity class		The collection of all entities that satisfy a given predicate
	Kitchenham	[DM element type]
Attribute		A measurable physical or abstract property of an entity, that is shared by all the entities of an entity class
	14598-1	A measurable physical or abstract property of an entity
	VIM	[Measurable quantity] attribute of a phenomenon, body or substance that may be distinguished qualitatively and determined quantitatively
	15939	Property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means
	610.12	A characteristic of an item; for example, the item's color, size, or type
	Briand	
	Kitchenham	[Generic attribute]
	Kim	[Measured attribute] physical characteristics of entities that are measured. They are attributes that acquire a value through measurement
	IEEE 1061	A measurable physical or abstract property of an entity
Quality model		The set of measurable concepts and the relationships between them which provide the basis for specifying quality requirements and evaluating the quality of the entities of a given entity class
	14598-1	The set of characteristics and the relationships between them which provide the basis for specifying quality requirements and evaluating quality
	Kitchenham	[Development model]
	Kim	[Enterprise quality model] an enterprise model of quality is a model that can be used to answer questions about the quality of the products and processes of an enterprise. It can also be used to identify quality improvement opportunities and suggest means to make improvements
	IEEE 1061	[Metrics framework] a decision aid used for organizing, selecting, communicating, and evaluating the required quality attributes for a software system. A hierarchical breakdown of quality factors, quality sub-factors, and metrics for a software system

Table 9
Comparison of the terms in the 'software measures' sub-ontology

Term	Source	Definition
Measure		The defined measurement approach and the measurement scale. (A measurement approach is either a measurement method, a measurement function or an analysis model)
	14598-1	[Metric] The defined measurement method and the measurement scale
	610.12	[Metric] a quantitative measure of the degree to which a system, component, or process possess a given attribute
	Briand	
	Kitchenham	[DM Element measure type]
	IEEE 1061	[Metric] A function whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which software possesses a given attribute that affects its quality
Information need	15939	Insight necessary to manage objectives, goals, risks, and problems
Scale	14598-1	A set of values with defined properties
	VIM	[Reference-value scale] For particular quantities of a given kind, an ordered set of values, continuous or discrete, defined by convention as a reference for arranging quantities of that kind in order of magnitude
	15939	Ordered set of values, continuous or discrete, or a set of categories to which the attribute is mapped
	Kitchenham	[Generic scale range]
Type of scale	15939	The nature of the relationship between values on the scale
Unit of measurement	15939, VIM	Particular quantity, defined and adopted by convention, with which other quantities of the same kind are compared in order to express their magnitude relative to that quantity
	14598-3	[Unit] A quantity adopted as a standard of measurement
	Kitchenham	[Generic unit]
	Kim	
Base measure		A measure of an attribute that does not depend upon any other measure, and whose measurement approach is a measurement method
	VIM	[Base quantity] one of the quantities that, in a system of quantities, are conventionally accepted as functionally independent of one another
	15939	Measure defined in terms of an attribute and the method for quantifying it. (Note: a base measure is functionally independent of other measures)
	14598-1	[Direct measure] measure of an attribute that does not depend upon a measure of any other attribute
	IEEE 1061	[Direct metric] a metric that does not depend upon a measure of ant other attribute
Derived measure		A measure that is derived from other base or derived measures, using a measurement function as measurement approach
	VIM	[Derived quantity] quantity defined, in a system of quantities, as a function of base quantities of that system
	15939	Measure that is defined as a function of two or more values of base measures
	14598-1	[Indirect measure] a measure of an attribute that is derived from measures of one or more other attributes
Indicator		A measure that is derived from other measures using an analysis model as measurement approach
	15939	An estimate or evaluation of specified attributes derived from a model with respect to defined information needs
	14598-1	A measure that can be used to estimate or predict another measure

Summarizing these results in figures, we can conclude that we analyzed eight sources (five international standards and three academic research proposals), and have presented one common ontology that can be used as starting point to harmonize and join them all.

We identified a total of 20 different software measurement concepts, for which 77 different definitions were found (which gives an average of 3.85 distinct definitions per concept!). The terms with more definitions are attribute (9), measure (7), metric (6), measurement (6), unit (6), entity

Table 10
Comparison of the terms in the 'measurement approaches' sub-ontology

Term	Source	Definition
Measurement method		Logical sequence of operations, described generically, used in quantifying an attribute with respect to a specified scale. (A measurement method is the measurement approach that defines a base measure)
	VIM	Logical sequence of operations, described generically, used in the performance of measurements
	15939	Logical sequence of operations, described generically, used in quantifying an attribute with respect to a specified scale
Measurement function		An algorithm or calculation performed to combine two or more base or derived measures. (A measurement function is the measurement approach that defines a derived measure)
	15939	An algorithm or calculation performed to combine two or more 'base measures'
Analysis model		Algorithm or calculation combining one or more measures with associated decision criteria. (An analysis model is the measurement approach that defines an indicator)
	15939	Algorithm or calculation combining one or more base and/or derived measures with associated decision criteria
Decision criteria	15939	Thresholds, targets, or patterns used to determine the need for action or further investigation, or to describe the level of confidence in a given result
	14598-1	[Rating Level] A scale point on an ordinal scale, which is used to categorise a measurement scale

(5), direct metric (5) and quality model (5). These are of course the most essential concepts, which clearly show the lack of agreement in the original sources for defining the same concepts (even the very basic ones). Besides, we found 28 cases of synonymy, which also confirms the lack of consensus in terminology.

According to the coverage of terms by the different proposals, ISO/IEC 15939 is the one that defines more concepts (16 out of the 20). Then, ISO/IEC 14598 series cover 11 concepts, followed by the proposals by Kitchenham (8), VIM (7), Kim (7), IEEE 1061 (7), Briand (5), and finally IEEE 610.12, which only covers three of the 20 concepts. The high number of terms directly adopted from ISO/IEC 15939 is due to the fact that it is a measurement standard, while other sources and standards define measurement terms only for their particular purposes and application domains.

It is also worth analyzing the origin of the terms proposed in SMO. As previously mentioned in Section 3.1, we can distinguish four different cases:

- *Adopted concepts*, which have been directly taken from a source, without any changes in their definitions or in the terms used to identify them (e.g. entity).
- *Adapted concepts*, which have been borrowed from a proposal but have been changed for consistency or completeness (see the comparison tables to identify the changes). In every adapted entity, either the original term or the original definition could have been changed, and thus we need to differentiate between:
- *Definition-adapted concepts* if the term has been taken from the source, but its original definition has been slightly changed in the SMO (e.g. attribute).
- *Term-adapted concepts* if the definition of the concept has been taken from the source, but the term used in the SMO is different (e.g. measure).
- *New concepts*, which are entities introduced by the SMO, not present in any of the sources (e.g. measurement approach), or whose final meaning does not correspond exactly to any of the sources (e.g. indicator).

Based on this classification, we find that there are three new concepts in the SMO (15%), eight adopted concepts (40%), and nine adapted concepts (45%).

The new concepts are measurement approach, entity class, and indicator. A measurement approach is an abstract concept that represents the generalization of measurement method, measurement function and analysis model. It is abstract in the UML sense, i.e. it is a general concept that has no instances-it just serves for generalizing the common characteristics of a set of other concepts.

The concept entity class represents the class of all entities of the same type. It is required in the ontology because attributes are defined in general for entity classes, and not for particular entities. For example, size is an attribute defined for the class of all C programs.

Finally, although the term indicator is not new strictly speaking, we have consider it here as new: both ISO/IEC 15939 and ISO/IEC 14598 define it, although its meaning in these two standards differs from ours (Section 3.2.2).

We have adopted without changes eight terms (40%), with their corresponding definitions. Six of them are from ISO/IEC 15939 (information need, measurable concept, entity, type of scale, decision criteria, and measurement method), one from the ISO VIM (unit of measurement), and one from ISO/IEC 14598 (scale).

Nine out of the 20 concepts of SMO are adapted (45%). We have six terms adapted from ISO/IEC 14598 (quality model, attribute, measure, base measure, derived measure and measurement result), two from ISO/IEC 15939 (measurement function and analysis model), and one from ISO VIM (measurement).

The terms of the ontology related to software measures have been the most difficult to harmonize. The problem, as mentioned in Section 3.2, is that none of the current proposals provide a successful solution. ISO/IEC 15939 uses the term measure, more in line with VIM and ISO's current preferences, but introduces some confusion with that term, since it is used both as a verb and as a noun. Besides, there is no concept in ISO/IEC 15939 that aggregates the measurement method and the scale used in a measurement action. ISO/IEC 14598 contains such a concept, but the term used, metric, is not aligned with ISO's new requirements. To make things even worse, VIM does not mention neither metric n or measure. With all this, we have tried to harmonize all approaches by avoiding the use of the term metric, the new definition of measure as the aggregation of a measurement approach and a scale (that basically corresponds to ISO/IEC 14598 metric), and the use of measurement as the action that performs a measurement approach to obtain a measurement result. In addition, this definition has the property of generalizing in a natural way the treatment of base measures, derived measures and indicators.

The other adapted concepts (attribute, quality model, measurement function and analysis model) have not been drastically changed. They have just been slightly adapted to be in line and consistent with the rest of the concepts used in our proposal.

5. Conclusions

The increase in international software engineering standardization is a consequence of the continuing growing importance of ICT (Information and Communication Technologies) and ICT-based systems, products and services in the global economy, as well as the growing maturity of the software and system engineering disciplines [21].

However, the lack of a common terminology and inconsistencies between the different standards may seriously jeopardize the usefulness and potential benefits of these standardization efforts. A consistent software measurement terminology may provide an important communication vehicle to companies when interoperating with others.

This article attempts to provide a comparison framework for the problem of consistency of terminology in software measurement. The common vocabulary provided by a common ontology has been used to resolve the problems of completeness and consistency identified in several international standards and research proposals.

Of course, we do not pretend that our proposal resolves all problems and is agreed by all parties, but rather that it serves as a basis for discussion from where the software measurement community can start paving the way to future agreements. What we are completely sure is that without these agreements, all the standardization and research efforts may be wasted, and the potential benefits that they may bring to all users (software developers, ICT suppliers, tools vendors, etc.) may never materialize.

Our future plans include the extension of the ontology to account for most of the concepts in the forthcoming version of the VIM, in order to provide a complete ontology for software measurement, and fully aligned with the VIM beyond the core concepts contemplated in this proposal.

Acknowledgements

The authors would like to acknowledge the clarifications and suggestions made by Prof Azuma and Prof. Abran on earlier versions of this paper. We would also like to thank anonymous reviewers for their comments and suggestions that have helped us to improve this work. This work has been supported by research projects COMPAS (TIC2002-04309-C02-02), MAS (TIC2003-02737-C02-02) and CALIPO (TIC2003-07804-C05-03).

References

[1] S. MacDonell, M. Shepperd, P. Sallis, Metrics for database systems an empirical study, in: Proc. 4th Int'l Software Metrics Symposium (Metrics 97), IEEE Computer Society Press, Albuquerque, May 1997, pages 99–107.

[2] L. Briand, S. Morasca, V. Basili, An operational process for goal-driven definition of measures, IEEE Transactions on Software Engineering 28 (12) (2002) 1106–1125.

[3] D. de Champeaux, Object-Oriented Development Process and Metrics, Prentice-Hall, Upper Saddle River, 1997.

[4] S. Pfleeger, Guest editor's introduction: assessing measurement, IEEE Software 14 (2) (1997) 25–26.

[5] ISO/IEC, International Standard ISO/IEC 90003, Software and Systems Engineering—Guidelines for the Application of ISO/IEC 9001:2000 to Computer Software, International Standards Organization, Geneva, Switzerland, 2004.

[6] ISO, International Standard ISO VIM, International Vocabulary of Basic and General Terms in Metrology, International Standards Organization, Geneva, Switzerland, 2 edition, 1993.

[7] B. Kitchenham, R.T. Hughes, S.G. Linkman, Modeling software measurement data, IEEE Transactions on Software Engineering 27 (9) (2001) 788–804.

[8] H.M. Kim, Representing and Reasoning about Quality using Enterprise Models, PhD thesis, Dept. Mechanical and Industrial Engineering, University of Toronto, Canada, 1999.

[9] A. Abran, A. Sellami, Initial modelling of the measurement concepts in the iso vocabulary of terms in metrology, in: Proc. of IWSM2002, Magdeburg, Germany, 2002.

[10] M. Azuma, SQuaRE: The next generation of the ISO/IEC 9126 and 14598 International Standards series on software product quality, in: Proc. of European Software Control and Metrics (ESCOM), London (England), April, (2001).

[11] V. Basili, D. Rombach, The tame project: towards improvement-oriented software environments, IEEE Transactions on Software Engineering 14 (6) (1998) 758–773.

[12] C. Jones, Making measurement work, CROSSTALK 16 (1) (2001) 15–19.

[13] J. McGarry, D. Card, C. Jones, B. Layman, E. Clark, J. Dean, F. Hall, Objective information for decision makers Practical Software Measurement, Addison-Wesley, MA, USA, 2002.

[14] H. Zuse, A Framework of Software Measurement, Walter de Gruyter, Berlin, 1998.

[15] F. García et al., An ontology for software measurement, Technical Report UCLM DIAB-04-02-2, Computer Science Department, University of Castilla-La Mancha, Spain, 2004.

[16] C. Tautz, C.G. Von Wangenheim, REFSENO: a representation formalism for software engineering ontologies, Technical Report Report No. 15.98/E, version 1.1, Fraunhofer IESE, October, 1998.

[17] A. Gómez-Pérez, Knowledge Sharing and Reuse, CRC Press, FL, USA, 1998.

[18] S. Staab, H.P. Schnurr, Y. Sure, Knowledge processes and ontologies, IEEE Intelligent Systems 16 (1) (2001) 26–34.

[19] M. Uschold, M. Gruninger, Ontologies: principles, methods, and applications, The Knowledge Engineering Review 11 (2) (1996) 93–196.

[20] K.D. Althoff, A. Birk, S. Hartkopf, W. Müller, Managing software engineering experience for comprehensive reuse, in: Proc. of the Eleventh International Conference on Software Engineering, Kaiserslautern, Germany, 1999.

[21] F. Coallier, International standardization in software and systems engineering, CROSSTALK 18 (2003) 15–18.