Chapter 13

# HOW TO MEASURE AND EVALUATE WEB APPLICATIONS IN A CONSISTENT WAY

Luis Olsina, Fernanda Papa, Hernán Molina
*GIDIS_Web, Engineering School, Universidad Nacional de La Pampa, Calle 9 y 110, (6360) General Pico, LP, Argentina,* {olsinal,pmfer,hmolina}@ing.unlpam.edu.ar

## 13.1 INTRODUCTION

A recurrent challenge many software organizations face is to have a clear establishment of a measurement and evaluation of a conceptual framework useful for quality assurance processes and programs. While many useful approaches for and successful practical examples of software measurement programs exist, the inability to clearly and consistently specify measurement and evaluation concepts (i.e., the meta-data) could unfortunately hamper the progress of the software, and Web Engineering as a whole, and could hinder their widespread adoption.

Software and Web organizations introducing a measurement and evaluation program—maybe as part of a measurement and analyses process area and quality assurance strategy (CMMI, 2002)—need to establish a set of activities and procedures to specify, collect, store, and use trustworthy measurement and indicator data sets and meta-data. Moreover, to ensure, for analysis purposes, that measurement and indicator data sets are repeatable and comparable among different measurement and evaluation projects, appropriate meta-data of metrics and indicators should be adapted and recorded.

Therefore, in the present chapter we argue that at least three pillars are necessary to build, i.e., to design and to implement, a robust and sound measurement and evaluation program:

1. a process for measurement and evaluation, i.e., the main managerial and technical activities that might be planned and performed
2. a measurement and evaluation framework that must rely on a sound conceptual (ontological) base
3. specific model-based methods and techniques in order to carry out the specific project's activities

A measurement or evaluation process prescribes or informs a set of main phases, activities, and their input and output that might be considered. Usually, it says what to do but not how to do it; that is, it says nothing about the particular methods and tools in order to perform the specific activities' descriptions. Regarding measurement and evaluation processes for software, the *International Standard Organization* (ISO) published two standards: the ISO 15939 document issued in 2002 (ISO, 2002), which deals with the software measurement process, and the ISO 14598-5 issued in 1998 (ISO, 1998), which deals with the process for evaluators in its part 5. On the other hand, the CMMI (*Capability Maturity Model Integration*) initiative is also worthy of mention as another source of knowledge, in which specific support process areas such as measurement and analyses, decision analyses and resolution, among others, are specified. The primary aim of these documents was to reach a consensus about the issued models, processes, and practices. However, in Olsina and Martin (2004) we observe that very often a lack of consensus exists about the used terminology among the ISO standards.

Considering our second statement, we argue that in order to design and implement a robust measurement and evaluation program, a sound measurement and evaluation conceptual framework is necessary. Very often organizations start measurement programs from scratch more than once because they did not pay too much attention to the way metrics and indicators should be designed, recorded, and analyzed.

A well-established framework has to be built on a sound conceptual base, that is, on an ontological base. In fact, an ontology explicitly and formally specifies the main concepts, properties, relationships, and axioms for a given domain. In this direction, we have built an explicit specification of measurement and indicator meta-data, i.e., an ontology for this domain (Olsina and Martin, 2004). The sources of knowledge for this ontology stemmed from different software-related ISO standards (ISO, 1999, 2001, 2002) and recognized research articles and books (Briand et al., 2002; Kitchenham et al., 2001; Zuse, 1998), in addition to our own experience backed up by previous works on metrics and evaluation processes and methods (Olsina et al., 1999; Olsina and Rossi, 2002).

However, the metrics and indicators ontology itself is not sufficient to model a full-fledged measurement and evaluation framework but rather is

the ground and rationale to building it. In Olsina et al. (2006b), the INCAMI framework (Olsina et al., 2005) is thoroughly analyzed in the light of its ontological roots. INCAMI is an organizational purpose-oriented measurement and evaluation framework that enables consistently saving not only meta-data of metrics and indicators but also values (data sets) for concrete real-world measurement and evaluation projects. It is made up of five main conceptual components, namely: the requirement, measurement, and evaluation of projects definition; the nonfunctional requirements definition and specification; the measurement design and execution; the evaluation design and execution; and the conclusion and recommendation components. We argue that this framework can be useful for different qualitative and quantitative evaluation methods and techniques with regard to the requirements, measurement, and evaluation concepts and definitions (Olsina et al., 2008).

On the other hand, the growing importance the Web currently plays in such diverse application domains as business, education, government, industry, and entertainment have heightened concerns about the quality and quality of delivered Web applications. It is necessary to have not only robust development methods to improve the building process (one of the main aims of this book) but also consistent ways to measure and evaluate intermediate and final products as well. In this sense measurement and evaluation methods and tools that are grounded on the quoted conceptual framework are the third pillar of our proposal.

There are different categories of methods (e.g., inspection, testing, inquiry, simulation, etc.) and specific types of evaluation methods and techniques such as the heuristic evaluation technique (Nielsen et al., 2001), the Web Quality Evaluation Method (WebQEM) (Olsina and Rossi, 2002) as a concept model-centered evaluation methodology for the inspection category, to name just a few. We argue that a method or technique is usually not enough to assess different information needs for diverse evaluation purposes. In other words, it is true that one size does not fit all needs and preferences, but an organization might at least adopt a method or technique in order to know the state of its quality and quality in use for understanding and improving purposes.

In order to illustrate the above three main points, this chapter is organized as follows. In Section 13.2 we present an abridged overview of the state-of-the-art of measurement and evaluation processes as well as a basic process that is akin to our framework. In Section 13.3 we analyze the main components of the INCAMI framework regarding the metrics and indicators ontological base; at the same time, as proof of these concepts, an external quality model to measure and evaluate the shopping cart component of a typical e-commerce site is employed. In Section 13.4, using the specific

models, procedures, and processes, the WebQEM inspection methodology is illustrated with regard to the previous case study. Finally, additional discussions about the flexibility of the framework as well as concluding remarks are drawn in Section 13.5.

## 13.2 OVERVIEW OF MEASUREMENT AND EVALUATION PROCESSES

As previously mentioned, a measurement or evaluation process specifies a set of main phases, activities, their input and output, and sometimes control points that might be considered. Usually, a process says what to do but not how to do it.

For instance, the ISO 14598-5 standard prescribes an evaluation process to assess software quality which is a generic abstract process customizable for different evaluation needs; however, it does not prescribe or inform about specific evaluation methods and tools in order to perform the activities' descriptions.

On the other hand, it is important to remark that no unique ISO standard that integrates in one document the measurement and evaluation process as a whole exists. Instead, there are two separate standards: one for the evaluation process, issued in 1998 (ISO, 1998), and another for the measurement process, issued in 2002 (ISO, 2002). Regarding the former, in an introductory paragraph it says, "The primary purpose of software product evaluation is to provide quantitative results concerning software product quality that are comprehensible, acceptable to and can be dependable on by any interested party"; it continues, "This evaluation process is a generic abstract process that follows the model defined in ISO/IEC 9126."

In the ISO 14598-5 standard, the evaluation process comprises the five activities listed in Figure 13.1 (see ISO, 1998, for a detailed description):

1. *establishment of evaluation requirements*
2. *specification of the evaluation* based on the evaluation requirements and on the product provided by the requester
3. *design of the evaluation,* which produces an evaluation plan on the basis of the evaluation specification
4. *execution of the evaluation plan,* which consists of inspecting, modeling, measuring, and testing the products and/or its components according to the evaluation plan
5. *conclusion of the evaluation,* which consists of the delivery of the evaluation report

*Figure 13.1*. The main activities specified in the ISO 14598-5 evaluation process standard.

The ISO 15939 standard that deals with the measurement process says, "Software measurement is also a key discipline in evaluating the quality of software products and the capability of organizational software processes"; in addition,

> Continual improvement requires change within the organization. Evaluation of change requires measurement. Measurement itself does not initiate change. Measurement should lead to action, and not be employed purely to accumulate data. Measurement should have a clearly defined purpose. . . , This standard defines the activities and tasks necessary to implement a software measurement process ... each activity is comprised of one or more tasks. This International Standard does not specify the details of how to perform the tasks included in the activities.

In this standard two activities (out of four) are considered to be the core measurement process, namely: *plan the measurement process*, and *perform the measurement process.* These two activities are comprised of the following tasks (see Figure 13.2 and also ISO, 2002, for a detailed description):

1. Plan the Measurement Process:
   - *1.1 Characterize organizational unit*
   - *1.2 Identify information needs*
   - *1.3 Select measures*
   - *1.4 Define data collection, analysis, and reporting procedures*
   - *1.5 Define criteria for evaluating the information products and the measurement process*
   - *1.6 Review, approve, and provide resources for measurement tasks*
   - *1.7 Acquire and deploy supporting technologies*
2. Perform the Measurement Process:
   - *2.1 Integrate procedures*
   - *2.2 Collect data*
   - *2.3 Analyze data and develop information products*
   - *2.4 Communicate results*

*Figure 13.2.* The two core measurement processes specified in the ISO 15939 measurement process standard.

Lastly, the CMMI (CMMI, 2002) initiative[1] is also worthy of mention. This initiative specifies support process areas such as *measurement and analyses*, among others. It says, "The purpose of measurement and analysis is to develop and sustain a measurement capability that is used to support management information needs", Figure 13.3 shows the two specific goals

[1] There is a related ISO 15504 initiative named SPICE (*Software Process Improvement and Capability dEtermination*).

for this process area and its specific practices (which can be considered as activities or specific actions).

1 Align Measurement and Analysis Activities
- *1.1 Establish measurement objectives*
- *1.2 Specify measures*
- *1.3 Specify data collection and storage procedures*
- *1.4 Specify analysis procedures*

2 Provide Measurement Results
- *2.1 Collect measurement data*
- *2.2 Analyze measurement data*
- *2.3 Store data and results*
- *2.4 Communicate results*

*Figure 13.3*. The two specific goals and related practices for the CMMI Measurement and Analyses process area.

As the reader could observe in the previous figures, there is in principle no clear integrated proposal about measurement and evaluation activities even though both are closely intertwined, as we discuss in our framework later on. However, a common denominator between activities and tasks outlined in the previous figures can be observed. For instance, there are the definition and specification of requirements, e.g., activities 1 and 2 in Figure 13.1 deal with the establishment and specification of evaluation requirements; tasks 1.1 and 1.2 in Figure 13.2 are about measurement requirements, as is practice 1.1 in Figure 13.3. There are also design activities, i.e., defining, specifying, or ultimately planning activities; then, execution or implementation activities of the designed evaluation or measurement; and lastly, activities about the conclusion and communication of results.

On the other hand, we have been developing the WebQEM methodology since the late 1990s (Olsina et al., 1999; Olsina and Rossi, 2002). The underlying WebQEM process integrates activities for requirements, measurement, evaluation, and recommendations. Figure 13.5 shows the evaluation process, including the phases, main activities, input, and output. This model followed to some extent the ISO's process model for evaluators (ISO, 1998). The main activities are grouped into the following four major technical phases (see Figure 13.4):

1. *Nonfunctional Requirements Definition and Specification*
2. *Measurement and Elementary Evaluation* (both Design and Implementation stages)
3. *Global Evaluation* (both Design and Implementation stages)
4. *Conclusion and Recommendations*

*Figure 13.4.* The four phases underlying the WebQEM methodology and the INCAMI framework. Note that the specific activities are not listed in the figure.

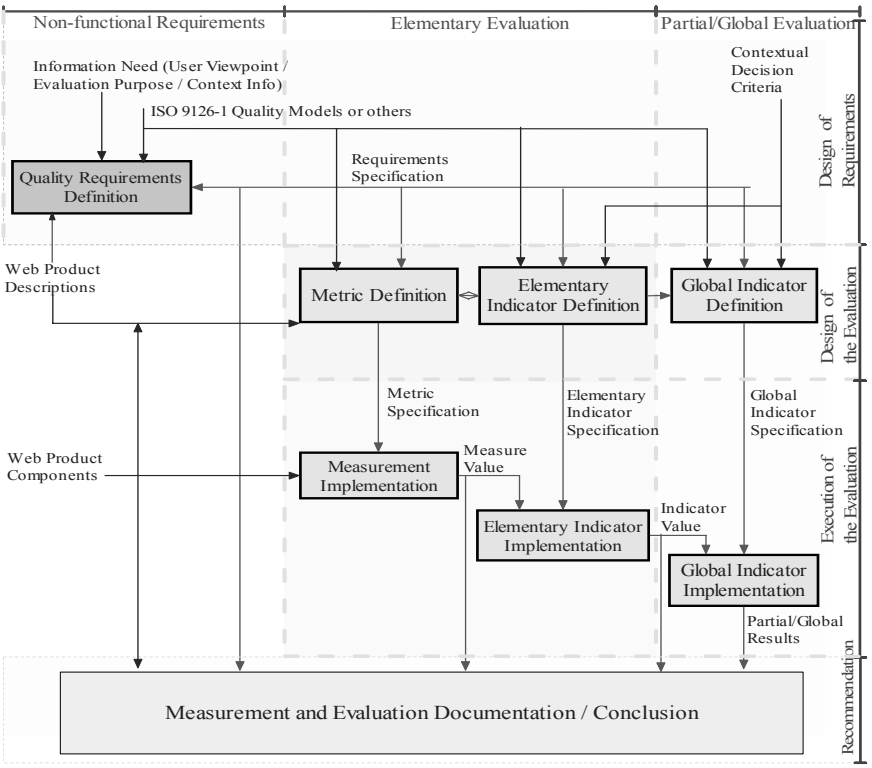



*Figure 13.5*. The basic measurement and evaluation process underlying the WebQEM methodology. The technical phases, main activities, and their input and output are represented (it might be assumed that some activities are iterative).

In the next section we thoroughly discuss the measurement and evaluation framework (the second pillar proposed in Section 13.1) in the light of its conceptual root and the above measurement and evaluation process. As an additional remark, in Olsina and Martin (2004) we observed that very often there is a lack of consensus about the used terminology among the quoted ISO standards, and some terms used mainly for the evaluation domain are missing.

## 13.3 FRAMEWORK FOR MEASURING AND EVALUATING NONFUNCTIONAL REQUIREMENTS

The proposed INCAMI (*Information Need*, *Concept model*, *Attribute*, *Metric,* and *Indicator*) framework (Molina et al., 2004; Olsina et al., 2005) is based upon the assumption that for an organization to measure and evaluate in a purpose-oriented way it must first specify nonfunctional requirements

starting from information needs, then it must design and select the specific set of useful metrics for measurement purpose, and lastly it must interpret the metrics values by means of contextual indicators with the aim of evaluating or estimating the degree to which the stated requirements have been met and, ultimately, to draw conclusions and give recommendations.

As aforementioned, the conceptual framework is made up of five main components: the nonfunctional requirements definition and specification; the measurement design and execution; the evaluation design and execution; the conclusion and recommendation component; and the project definition itself. Currently, most of the components are supported by many of the ontological concepts, properties, and relationships defined in previous works (Olsina and Martin, 2004). For instance, to the nonfunctional requirements definition component, concepts such as *Information Need*, *Calculable Concept*, *Concept Model*, *Entity*, *Entity Category,* and *Attribute* intervene (all these terms are defined and illustrated in Section 13.3.4.1). Some other concepts were added to the framework in order to design and implement it as a Web application (the INCAMI_Tool).

In Sections 13.3.1 to 13.3.3 we give an abridged description of the first three components listed above. In Section 13.3.4 we thoroughly discuss the main terms for these components; in addition, each term is illustrated using as an example the external quality model to assess the shopping cart feature of the www.amazon.com site.

## 13.3.1 Information Need, Concept Model, and Attribute

First, for the nonfunctional requirements definition and specification component, the *Information Need* to a measurement and evaluation *Project* must be agreed upon. Information need is defined as the insight necessary to manage objectives, goals, risks, and problems. Usually, information needs come from two organizational project-level sources: goals that decision makers seek to achieve, or obstacles that hinder reaching the goals; e.g., obstacles involve basically risks and problems. The *InformationNeed* class (see Figure 13.6) has three properties: the *purpose*, the user *viewpoint*, and the *contextDescription*. (Note that from the process standpoint, outlined in the previous section, and particularly for the *Nonfunctional Requirements Definition and Specification* phase, we can represent an activity named *Identify Information Needs* and in turn tasks such as *Establish measurement/evaluation purpose*; *Establish the user viewpoint*; and *Specify the context of the measurement/evaluation.*)

Additionally, the *InformationNeed* class has two main relationships with the *CalculableConcept* and the *EntityCategory* classes, respectively. A calculable concept can be defined as an abstract relationship between attributes of entities' categories and information needs; in fact, internal quality, external quality, cost, etc. are instances of a calculable concept. In turn, a calculable concept can be represented by a *ConceptModel*; for example, ISO 9126-1 specifies quality models for the internal quality, external quality, and quality in use, respectively.

On the other hand, a common practice is to assess quality by means of the quantification of lower abstraction concepts such as *Attributes* of entities' categories. The attribute term can be defined in brief as a measurable property of an *EntityCategory* (e.g., categories of entities of interest to software and Web Engineering are resource, process, product, service, and project as a whole). An entity category may have many attributes, though only some of them may be useful just for a given measurement and evaluation project's information needs.

In summary, this component allows the definition and specification of nonfunctional requirements in a sound and well-established way. It has an underlying organizational strategy that is purpose-oriented by information needs and is concept model-centered and evaluator-driven by domain experts and users.

## 13.3.2 Metrics and Measurement

Regarding the measurement component, purposeful metrics should be selected in the process. In general, each attribute can be quantified by one or more metrics, but in practice just one metric should be selected for each attribute of the requirements tree, given a specific measurement project.

The *Metric* concept contains the definition of the selected *Measurement* or *Calculation Method* and the *Scale* (see Figure 13.8). For instance, the measurement method is defined as the particular logical sequence of operations and possible heuristics specified for allowing the realization of a metric description by a measurement; while the scale is defined as a set of values with defined properties. Thus, the metric *m* represents a mapping *m*: *A->X*, where *A* is an empirical attribute of an entity category (the empirical world), *X* is the variable to which categorical or numerical values can be assigned (the formal world), and the arrow denotes a mapping. In order to perform this mapping, a sound and precise measurement activity definition is needed by explicitly specifying the metric's method and scale. We can apply an *objective* or *subjective* measurement method for *Direct Metrics*; conversely, we can perform a calculation method for *Indirect Metrics*, that is, when a *Formula* intervenes.

Once the metric has been selected, we can perform (execute or implement) the measurement process, i.e., the activity that uses a metric definition in order to produce a measure's value (see Figure 13.5). The *Measurement* class allows the date/time stamp, the information of the owner in charge of the measurement activity, and the actual or estimated yielded value to be recorded.

However, since the value of a particular metric will not represent the elementary requirement's satisfaction level, we need to define a new mapping that will produce an elementary indicator value. One fact worthy of mention is that the selected metrics are useful for a measurement process as long as the selected indicators are useful for an evaluation process in order to interpret the stated information need.

### 13.3.3 Indicators and Evaluation

For the evaluation component, contextual indicators should be selected. Indicators are ultimately the foundation for the interpretation of information needs and decision making. There are two types of indicators: *elementary* and *global indicators* (see Figure 13.9).

In Olsina and Martin (2004) the indicator is described as "the defined calculation method and scale in addition to the model and decision criteria in order to provide an estimate or evaluation of a calculable concept with respect to defined information needs." In particular, we define an elementary indicator as one that does not depend upon other indicators to evaluate or estimate a concept at a lower level of abstraction (i.e., for associated attributes to a concept model). On the other hand, we define a partial or global indicator as one that is derived from other indicators to evaluate or estimate a concept at a higher level of abstraction (i.e., for subconcepts and concepts). Therefore, the elementary indicator represents a new mapping coming from the interpretation of the metric's measured value of an attribute (the formal world) into the new variable to which categorical or numerical values can be assigned (the new formal world). In order to perform this mapping, elementary and global model and decision criteria for a specific user information need should be designed.

Therefore, once we have selected a scoring model, the aggregation process follows the hierarchical structure defined in the concept model, from bottom to top. Applying a stepwise aggregation mechanism, we obtain a global schema; this model lets us compute partial and global indicators in the execution stage. The global indicator's value ultimately represents the global degree of satisfaction in meeting the stated requirements (information need) for a given purpose and user viewpoint.

## 13.3.4 Definition and Exemplification of the INCAMI Terms

In this section (from Sections 13.3.4.1 to 13.3.4.3) we define the main terms that intervene in the above INCAMI framework's components, i.e., the requirement, measurement, and evaluation components. Each one is modeled by a class diagram (Figures 13.6, 13.8, and 13.9), where many (but not all) terms in the diagrams come from the metrics and indicators ontology. Note that for space reasons, we do not define each class attribute and relationships among classes, as is done in Olsina and Martin (2004).

In addition, for illustration purposes, we use an external quality model with associated attributes specified to the shopping cart of Web sites. We have conducted a case study in order to assess the shopping cart feature of the www.amazon.com site (details of this study will be given in Section 13.4).

### 13.3.4.1 Requirements Definition and Specification Model

As shown in Figure 13.6, this model includes all the necessary concepts for the definition and specification of requirements for measurement and evaluation projects. Nonfunctional requirements are the starting point of the measurement and evaluation process, so that a *requirement project* should be defined.

***Definition 13.1.*** *RequirementProject* is a project that allows us to specify nonfunctional requirements for measurement and evaluation activities.

In our example the project *name* is "ExternalQuality_Amazon_05"; the *description* is "requirements for evaluating the external quality for the shopping cart feature of the www.amazon.com site"; with a starting date "2005/12/19" and an ending date "2005/12/30" and in charge of "Fernanda Papa" with the "pmfer@ing.unlpam.edu.ar" *contact* email.

Next, the *information need* should be specified. For this study, a basic information need may be "understand the external quality of the shopping cart component of a typical e-store, for a general visitor viewpoint, in order to incorporate the best features in a new e-bookstore development project."

***Definition 13.2.*** *InformationNeed* is the insight necessary to manage objectives, goals, risks, and problems.

In our example the *information need* is stated by the *purpose* (i.e., to understand), the *user viewpoint* (i.e., a general visitor), in a given *context* of use (e.g., bandwidth constraints, among other contextual descriptions). In addition, an *entity category*, which is the *object* under analysis, and the *calculable concept*, which is the *focus* of *the information need,* must be defined.

***Definition 13.3.*** *Entity Category* is the object category that is to be characterized by measuring its attributes.

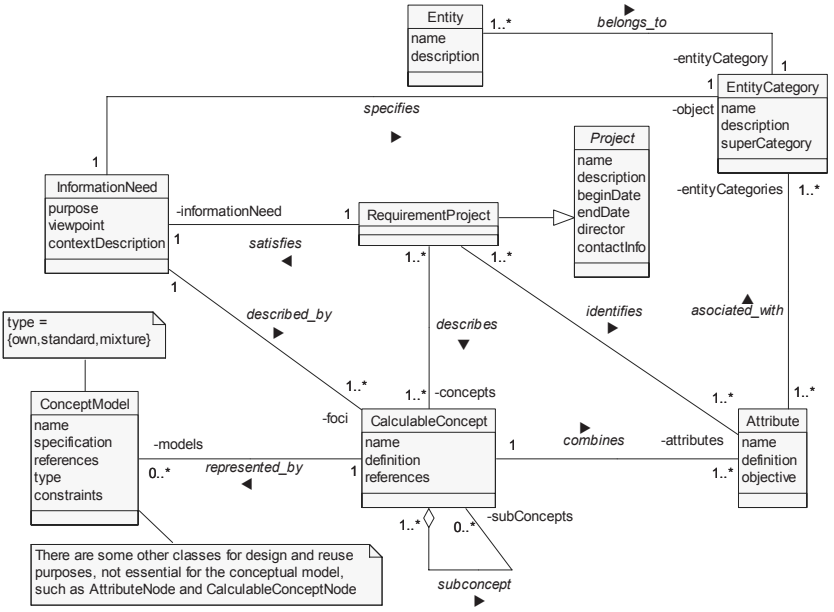



*Figure 13.6.* Key terms and relationships that intervene in the INCAMI requirements component for the definition and specification of nonfunctional requirements.

***Definition 13.4.*** *Entity,* synonym Object, is a concrete object that belongs to an entity category.

Therefore, given the *entity category* (i.e., an e-commerce application, of which *superCategory* is a product), a concrete object *name* that belongs to this category is the "Amazon's shopping cart" Web component.

***Definition 13.5.*** *CalculableConcept,* synonym Measurable Concept in ISO (2002), defines the abstract relationship between attributes of entity categories and information needs.

In the example the *calculable concept name* is "external quality" and its *definition* is "the extent to which a product satisfies stated and implied needs when used under specified conditions" (ISO, 1999). The external quality concept has *subconcepts* such as "usability", "functionality", "reliability", "efficiency", "portability", and "maintainability".

For instance, the "functionality" subconcept is defined in ISO (2001) as "the capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions". In turn, the calculable concept (characteristic) "functionality" is split into five subconcepts (subcharacteristics): "suitability", "accuracy", "interoperability", "security", and "functionality compliance." Suitability is defined as "the capability of the software product to provide an appropriate set of functions for specified tasks and user objectives"; and accuracy as "the capability of the software product to provide the right or agreed results or effects with the

needed degree of precision." See Figure 13.7, where these two subconcepts in the requirements tree are included as "Function Suitability" and "Function Accuracy", respectively (we used the name "function suitability" instead of "suitability" alone, in order to distinguish it from the name "information suitability", which is a subconcept of the Content characteristic).

On the other hand, the calculable concept can be *represented by* a *concept model*.

***Definition 13.6.*** *ConceptModel,* synonym Factor or Feature Model, is the set of subconcepts and the relationships between them, which provide the basis for specifying the concept requirement and its further evaluation or estimation.

As mentioned earlier, INCAMI is a concept model-centered approach; the concept model *type* can be either a standard-based model or an organization own-defined model, or a mixture of both. The concept model used in the example is of the "mixture" *type* that is based mainly on the ISO external quality model (*reference* "(ISO, 1999)"), and the *specification* is shown in Figure 13.11 (note that the model also shows *attributes* combined to the *subconcepts*).

***Definition 13.7.*** *Attribute,* synonym Property, Feature, is a measurable physical or abstract property of an entity category.

Note that the selected attributes are those properties relevant to the agreed-upon information need. The abridged representation in Figure 13.7 shows attribute *names* such as "Capability to delete items" (2.1.2) and "Precision to recalculate after deleting items" (2.2.2), among others.

**2. Functionality**
- 2.1. Function Suitability
  - *2.1.1. Capability to add items from anywhere*
  - *2.1.2. Capability to delete items*
  - *2.1.3. Capability to modify an item quantity*
  - *2.1.4. Capability to show totals by performed changes*
  - *2.1.5. Capability to save items for later/move to cart*
- 2.2. Function Accuracy
  - *2.2.1. Precision to recalculate after adding an item*
  - *2.2.2. Precision to recalculate after deleting items*
  - *2.2.3. Precision to recalculate after modifying an item quantity*

*Figure 13.7.* An excerpt (taken from Figure 13.11) of an instance of the external quality model with associated attributes specified for measurement and evaluation of the shopping cart component; for instance, the 2.1 and 2.2 codes represent specific calculable concepts and subconcepts; and the rest (in italic) are associated attributes to the above subconcepts. The model as a whole is depicted as a requirements tree.

For instance, the "Capability to delete items" attribute is defined (see the field *definition* in the Attribute class in Figure 13.6) as "the capability of the

shopping cart to provide functions in order to delete appropriately items one by one or to the selected group at once."

The INCAMI_Tool, which is a prototype tool that supports this framework, currently implements concept models in the form of requirements trees. It also allows partially or totally previously edited requirements trees to be imported for a new project.

### 13.3.4.2 Measurement Design and Execution Model

The measurement model (see Figure 13.8) includes all the necessary concepts for the design and implementation of the measurement as a part of the *Measurement and Elementary Evaluation* phase shown in Figure 13.4. First, a *measurement project* should be defined.

***Definition 13.8.*** *MeasurementProject* is a project that allows us, starting from a requirement project, to select the metrics and record the values in a measurement process.

Once the measurement project has been created, with similar information as that of a requirement project, the attributes in the requirements tree can be quantified by *direct* or *indirect metrics*.

Consider that for a specific measurement project just one metric should be selected for each attribute of the concept model. In the INCAMI_Tool, each metric is selected from a catalogue (Molina et al., 2004).

On the other hand, note that many measurement projects can rely on the same requirements, for instance, in a longitudinal analysis. In this case the starting and ending dates should change for each project.

***Definition 13.9.*** *Metric*[2] is the defined measurement or calculation method and the measurement scale.

***Definition 13.10.*** *DirectMetric*, synonym Single, Base Metric, is a metric of an attribute that does not depend on a metric of any other attribute.

[2] The "metric" term is used in ISO (1999, 2001) but not in ISO (2002). Furthermore, ISO (1999, 2001) uses the terms "direct measure" and "indirect measure" (instead of "direct" or "indirect metric"), while ISO (2002) uses "base measure" and "derived measure." In some cases we could state that they are synonymous terms, but in others such as "metric", which is defined in ISO (1999) as "the defined measurement method and the measurement scale", there is no term with exact matching meaning in ISO (2002). Furthermore, we argue that the measure term is not synonymous with the metric term. The measure term is defined in ISO (1999) (the meaning we adopted) as "the number or category assigned to an attribute of an entity by making a measurement" or in ISO (2002) as the "variable to which a value is assigned as the result of measurement" reflects the fact of the measure as the resulting value or output for the measurement activity (or process). Thus, we argue that the metric concept represents the specific and explicit definition of the measurement activity.

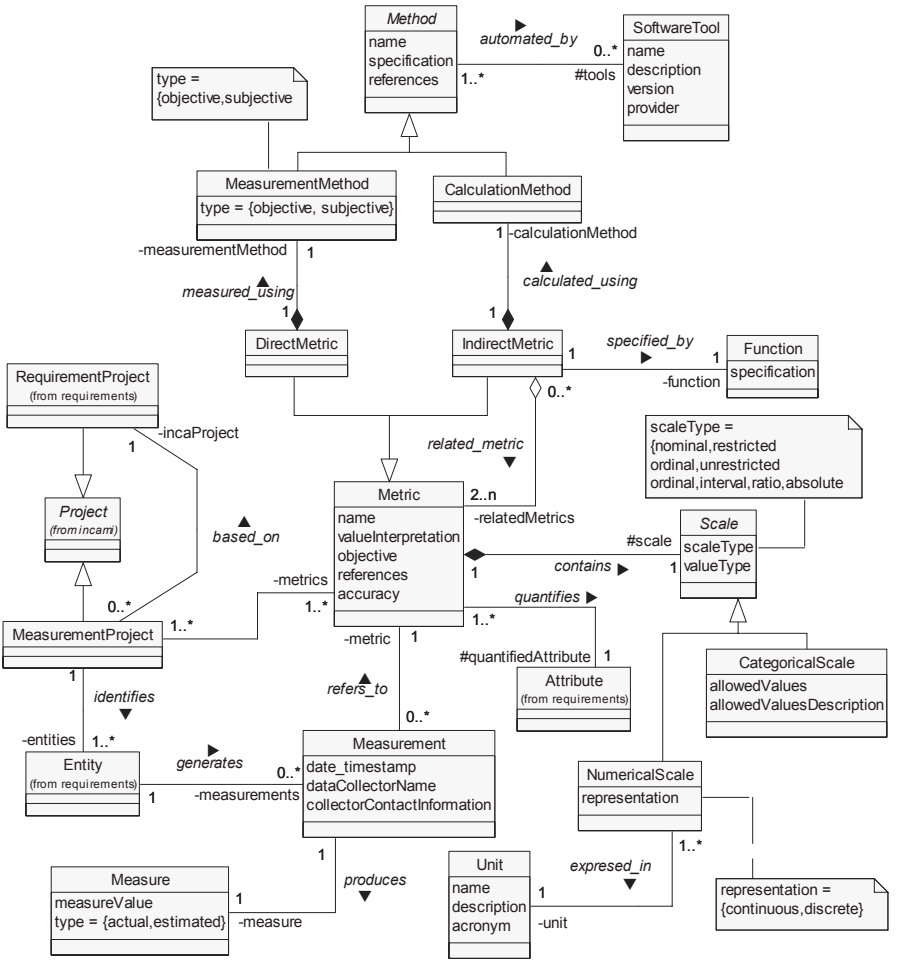



*Figure 13.8.* Key terms and relationships that intervene in the INCAMI measurement component for the definition of metric and measurement concepts.

For example, to the "Capability to delete items" attribute (coded 2.1.2 in Figure 13.7) we designed a direct metric named "Degree of the capability to delete items" that specifies four categories, namely:

0. Does not delete items at all
1. Delete just all at once
2. Delete one by one
3. Delete one by one or delete the selected group at once

***Definition 13.11.*** *IndirectMetric,* synonym Hybrid, Derived Metric, is a metric of an attribute that is derived from the metrics of one or more other attributes.

***Definition 13.12.*** *Function,* synonym Formula, Algorithm, Equation, is an algorithm or formula performed to combine two or more metrics.

There are two key terms in Definition 13.9: *Method* and *Scale*. For the latter, two types of scales have been identified: *Categorical* and *Numerical Scales:*

***Definition 13.13.*** *Scale* is a set of values with defined properties.

The type of scales (*scaletype* attribute in the Scale class in Figure 13.8) depends on the nature of the relationship between values of the scale. The types of scales commonly used in software and Web Engineering are classified into nominal, ordinal (both restricted and unrestricted), interval (and quasi-interval), ratio, and absolute. The scale type[3] of measured and calculated values affects the sort of arithmetical and statistical operations that can be applied to values, as well as the admissible transformations among metrics.

***Definition 13.14.*** *CategoricalScale* is a scale where the measured or calculated values are categories and cannot be expressed in units, in a strict sense.

***Definition 13.15.*** *NumericalScale* is a scale where the measured or calculated values are numbers that can be expressed in units, in a strict sense.

***Definition 13.16.*** *Unit* is a 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.

The *scale type* of the above direct metric (see the example in Definition 13.10) is "ordinal" represented by a *categorical scale* with a "symbol" *value type.* The *allowedValues* for the ordinal categories are from 0 to 3, and the *allowedValuesDescription* are the names of the categories such as "Delete just all at once." Note that because the type of the scale is ordinal, a mapping of categories to numbers can be made, whereas the order is preserved.

As stated earlier, two key terms appear in the metric definition: *method* and *scale*. In the sequel, the method-related terms are defined.

***Definition 13.17.*** *Method,* synonym Procedure, is a logical sequence of operations and possible heuristics, specified generically, for allowing the realization of an activity description.

***Definition 13.18.*** *SoftwareTool,* synonym Software Instrument, is a tool that partially or totally automates a measurement or calculation method.

For example, the INCAMI_Tool, the current prototype tool that supports the WebQEM methodology, allows us to calculate indirect metrics (from direct metrics and parameters) in addition to calculating elementary and global indicators from elementary and global models. A previous tool for WebQEM was the WebQEM_Tool (Olsina et al., 2001). Different

[3] See a deeper discussion about type of scales in Chapter 14, Section 14.2.

commercial tools for data collection of direct metrics are widely well known and available for download.

***Definition 13.19.*** *MeasurementMethod,* synonym Counting Rule, Protocol, is the particular logical sequence of operations and possible heuristics specified for allowing the realization of a metric description by a measurement.

To the exemplified direct metric (see the example in Definition 13.10), the counting rule was clearly specified as well as the measurement method *type*. The type of method can be either "subjective" i.e., where the quantification involves human judgment, or "objective" i.e., where the quantification is based on numerical rules. Generally, an objective measurement method type can be automated or semiautomated by a software tool. Nevertheless, for our example of a direct metric, even though the type is objective, no tool can automate the collection of data, and so a human must perform the task.

***Definition 13.20.*** *CalculationMethod* is the particular logical sequences of operations specified for allowing the realization of a formula or indicator description by a calculation.

***Definition 13.21.*** *Measurement* is an activity that uses a metric definition in order to produce a measure's value.

***Definition 13.22.*** *Measure* is the number or category assigned to an attribute of an entity by making a measurement.

A *measurement* activity must be performed for each metric that intervenes in the project. It allows the *date/time stamp*, the *collector information* in charge of the measurement activity, and the *measure,* the "actual" or "estimated" value *type,* and the yielded *value* itself to be recorded.

Ultimately, for a specific measurement project, at least all the above concepts and definitions of the measurement model are necessary in order to specify, collect, store, and use trustworthy metrics' values and meta-data.

#### 13.3.4.3 Evaluation Design and Execution Model

As introduced in Section 13.3.2, the value of a particular metric will not represent the elementary requirement's satisfaction level. Thus, we need to define a new mapping that will produce an elementary indicator value.

As aforementioned, the selected metrics are useful for designing and performing the measurement process as long as the selected indicators are useful for designing and executing the evaluation process for the stated information need, which is represented specifically in the concept model. The main concepts involved in the elementary and global evaluation are depicted in the model in Figure 13.9.

***Definition 13.23.*** *EvaluationProject* is a project that allows us, starting from a measurement project and a concept model of a requirement project, to select the indicators and perform the calculations in an evaluation process.

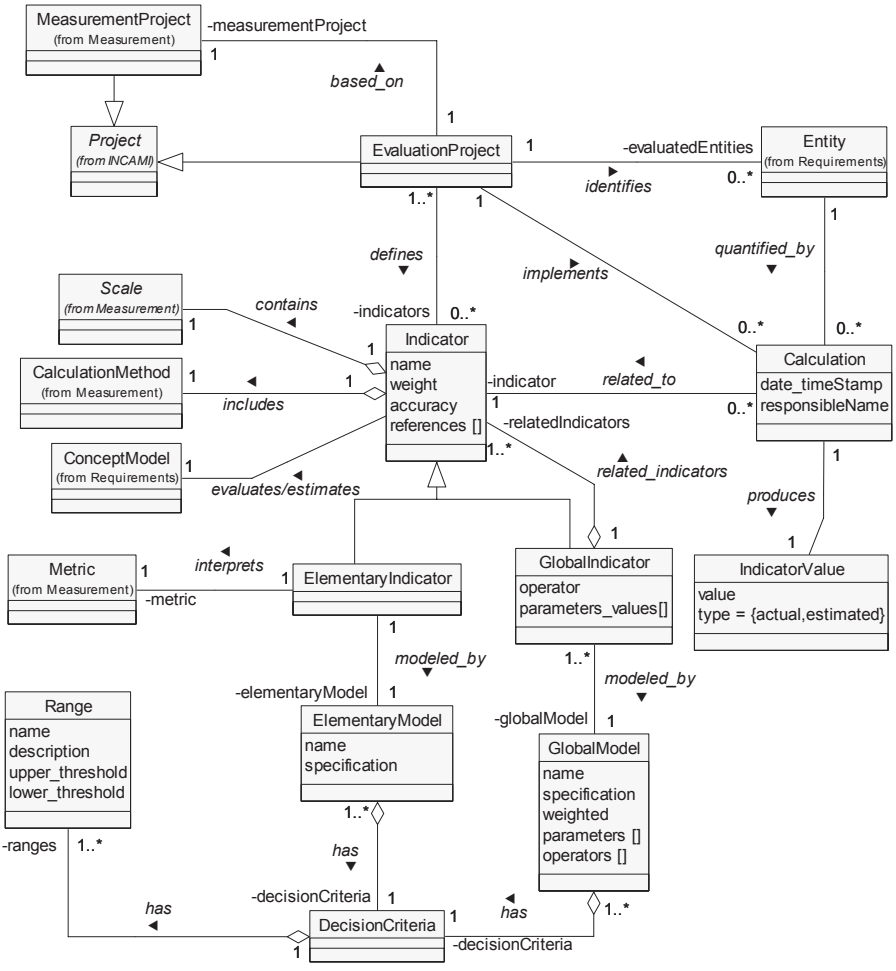



*Figure 13.9.* Key terms and relationships that intervene in the INCAMI.evaluation component for the definition of indicators and related concepts.

Once a measurement project has been created, one or more evaluation projects can in turn be created, relying on the recorded measurement values and meta-data, by adding related information with *indicators*.

***Definition 13.24.*** *Indicator,* synonym Criterion, is the defined calculation method and scale in addition to the model and decision criteria in order to provide an estimate or evaluation of a calculable concept with respect to defined information needs.

***Definition 13.25.*** *ElementaryIndicator,* synonym Elementary Preference, Elementary Criterion, is an indicator that does not depend upon other indicators to evaluate or estimate a calculable concept.

Therefore, an elementary indicator for each attribute of the concept model, i.e., for each leaf of the requirements tree, can be defined. For instance, to the 2.1.2 attribute of Figure 13.7, the *name* of the elementary indicator is "Performance Level of the Capability to Delete Items*"* (CDI_PL).

The elementary indicator interprets the metric's value of the attribute. To this end, an *elementary model* is needed.

***Definition 13.26.*** *ElementaryModel,* synonym Elementary Criterion Function, is an algorithm or function with associated decision criteria that model an elementary indicator.

The *specification* of the elementary model can look like this: CDI _PL = (0.33 * CDI) * 100; where CDI is the direct metric for the *Capability to Delete Items* attribute (see Definition 13.10).

Note that, like a metric, an indicator has a *Scale* (see Definition 13.13). To the above example we considered a *numerical scale* where the *Unit* (see Definition 13.16) can be a normalized "percentage" scale. As mentioned, the elementary indicator interprets the metric's value of an attribute (an attribute as an elementary requirement). Then, the above elementary model interprets the percentage of the satisfied elementary requirement.

***Definition 13.27.*** *DecisionCriteria,* synonym Acceptability Levels, are the 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.

***Definition 13.28.*** *Range* is the threshold or limit values that determine the acceptability levels.

The decision criteria that a model of an indicator may have are the agreed-upon acceptability levels in given ranges of the scale; for instance, it is "unsatisfactory" if the *range* (regarding *lower_threshold* and *upper_threshold*) is "0 to 45", respectively; "marginal" if it is "greater than 45 and less or equal than 70"; otherwise, "satisfactory." A *description* or interpretation for "marginal" is that a score within this range indicates a need for improvement. An "unsatisfactory" rating means change actions must take high priority.

***Definition 13.29.*** *GlobalIndicator,* synonym Global Preference, Global Criterion, is an indicator derived from other indicators to evaluate or estimate a calculable concept.

***Definition 13.30.*** *GlobalModel,* synonym Aggregation Model, Scoring Model, or Function, is an algorithm or function with associated decision criteria that model a global indicator.

In order to enact the concept model (see Definition 13.6) for elementary, partial, and global indicators, an *aggregation model* and *decision criteria* must be selected. The quantitative aggregation and scoring models aim at making the evaluation process well structured, objective, and comprehensible to evaluators. For example, if our procedure is based on a "linear additive scoring model," the aggregation and computing of partial/global indicators (P/GI), considering relatives *weights* (*W*), is based on the following *specification*:

$$\text{P/GI} = (W_1\ \text{EI}_1 + W_2\ \text{EI}_2 + \ldots + W_m\ \text{EI}_m); \tag{13.1}$$

such that if the elementary indicator (EI) is in the percentage scale and unit, the following holds:

$$0 \leq \text{EI}_i \leq 100;$$

and the sum of weights for an aggregation block must fulfill

$$(W_1 + W_2 + \ldots + W_m) = 1$$

if $W_i > 0$; for i = 1, . . ., *m*, where *m* is the number of subconcepts at the same level in the tree's aggregation block (see Figure 13.11).

The basic arithmetic aggregation *operator* for input in Eq. (13.1) is the plus (+) connector. Besides, this model lets us compute partial and global indicators in the execution stage. Other nonlinear aggregation models or functions can be used such as logic scoring of preference (Dujmovic, 1996), fuzzy model, and neural models, among others.

***Definition 13.31.*** *Calculation,* synonym Computation, is an activity that uses an indicator definition in order to produce an indicator's value.

***Definition 13.32.*** *Indicator Value,* synonym Preference Value, is the number or category assigned to a calculable concept by making a calculation.

As a final remark, for a specific evaluation project, all the above concepts and definitions of the evaluation model are necessary in order to specify, calculate, store, and use trustworthy indicator values and meta-data. When the execution of the measurement and evaluation activities for a given project has been performed, decision makers can analyze the results and draw conclusions and recommendations with regard to the established information need. Ultimately, we argue that this framework can be useful for different qualitative and quantitative evaluation methods and techniques with regard to the requirements, measurement, and evaluation concepts and definitions discussed previously.

## 13.4 ASSESSING WEB QUALITY USING WEBQEM: A CASE STUDY

In Section 13.1, we stated that in order to build a robust and clear measurement and evaluation program, at least three pillars are necessary, namely (1) a process for measurement and evaluation, which is outlined in Section 13.2, (2) a measurement and evaluation framework based on an ontological base, which is analyzed in Section 13.3, and (3) specific model-based methods and techniques in order to perform the specific program or project's activities, which are the aim of this section.

While a measurement or evaluation process specifies what to do (i.e., a clear specification of activities' descriptions, input and output, etc.), a method specifies how to do and perform such activities' descriptions relying on specific models and criteria.

As mentioned, there are different categories of methods (for example, categories for inspection, testing, inquiry, simulation, etc.) and specific types of evaluation methods and techniques such as the heuristic evaluation technique, analyses of log files, or concept model-centered evaluation methods, among many others.

In this section we present the Web Quality Evaluation Methodology (WebQEM) (Olsina and Rossi, 2002) as a model-centered evaluation methodology for the inspection category, that is, inspection of concepts, subconcepts, and attributes stemming from a quality or quality-in-use requirement model, among others. In addition, WebQEM relies on the metric and indicator concepts for measurement and evaluation in order to draw conclusions and give recommendations. We have been developing the WebQEM methodology since the late 1990s. It has been used to evaluate Web sites in several domains, as documented elsewhere (Olsina et al., 1999, 2000, 2006a), in addition to evaluating some industrial Web sites.

In order to illustrate WebQEM and its applicability, we conducted an e-business case study by evaluating the external quality of the shopping cart feature of the Amazon Web site, taking into account a general visitor standpoint. Note that users are redirected to the Amazon Web site (www.amazon.com) from the IMDb, the Internet Movie Database Web site (www.imdb.com), when trying to buy a DVD.

### 13.4.1 External Quality Requirements Specification

Many potential attributes, both general and domain-specific, can contribute to the Web's external quality. However, as mentioned earlier, evaluation must be organizational, purpose-oriented for an identified information need. Let us establish that the purpose in the present study is to understand the

external quality of the shopping cart component of a typical e-store, for a general visitor viewpoint, in order to incorporate the best features in a new e-bookstore development project. For this aim, a successful international site such as Amazon was chosen. On the other hand, recall that the ISO 9126-1 standard models the software quality from three related approaches, which can be summarized as follows:

- *Internal quality*, which is specified by a quality model (ISO, 2001; prescribing a set of six characteristics and a set of subcharacteristics for each characteristic) and can be measured and evaluated by static attributes of documents such as specification of requirements, architecture, or design; pieces of source code, and so forth. In the early phases of a software or Web life cycle, we can evaluate and control the internal quality of these early products, but assuring internal quality is not usually sufficient to assure external quality.
- *External quality*, which is specified by a quality model (likewise as in the previous cited model) and can be measured and evaluated by dynamic properties of the running code in a computer system, i.e., when the module or full application is executed in a computer or network simulating the actual environment as closely as possible. In the late phases of a software life cycle (mainly in different kinds of testing, or even in the acceptance testing, or furthermore in the operational state of a software or Web application), we can measure, evaluate, and control the external quality of these late products, but assuring external quality is usually not sufficient to assure quality in use.
- *Quality in use*, which is specified by a quality model (ISO, 2001; prescribing a set of four characteristics) and can be measured and evaluated by the extent to which the software or Web application meets a specific user's needs in an actual, specific context of use.

A point worthy of mention is the important difference between measuring and evaluating external quality and quality in use; see Olsina et al. (2006a) for an in-depth discussion on Web quality and these ISO models. The former generally involves no real users but rather experts, as long as the latter always involves real end users. The advantage of using expert evaluation without extensive user involvement is minimizing costs and time, among other features. Deciding whether or not to involve end users should be carefully planned and justified. On the other hand, without the end user's participation, it is unthinkable to conduct a task testing in a real context of use for quality-in-use evaluation. Nielsen et al. (2001) indicate that it is common for three to five subjects in the testing process for a given audience to produce meaningful results that minimize costs; however, they recommend running as many small tests as possible.

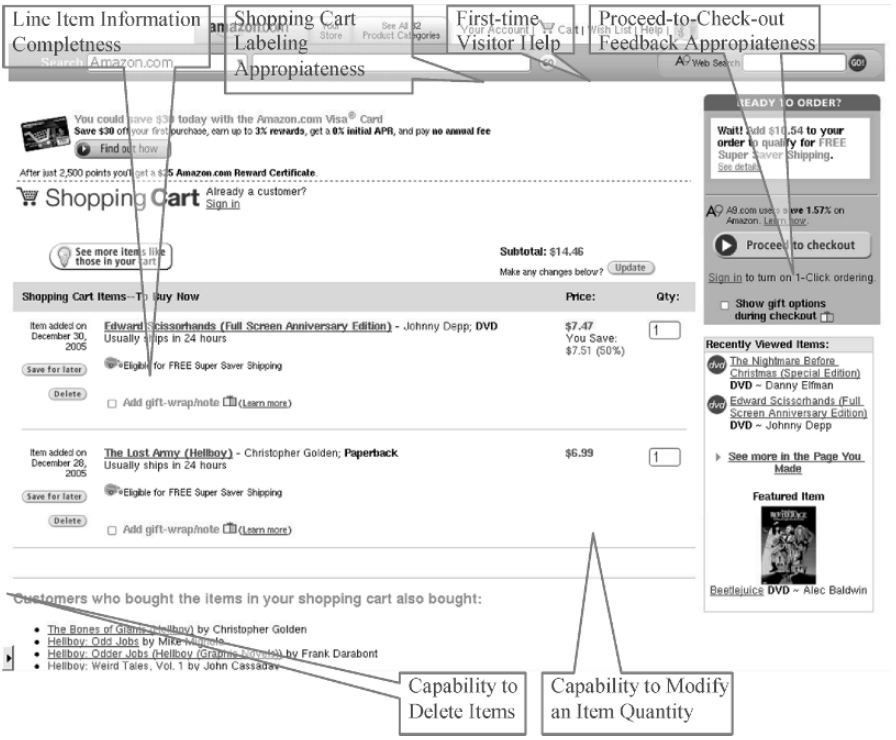



*Figure 13.10.* A screenshot of Amazon's shopping cart page with several attributes highlighted.

Considering the present study, Figure 13.10 shows a screenshot of Amazon's shopping cart page with several highlighted attributes, which intervene in the quality requirements tree of Figure 13.11.

To the external quality requirements definition, we considered 4 main characteristics: *Usability* (1), *Functionality* (2), *Content* (3), and *Reliability* (4), and 32 attributes related to them (see Figure 13.11). For instance, the Usability characteristic splits into subcharacteristics such as *Understandability* (1.1), *Learnability* (1.2), *Operability* (1.3), and *Attractiveness* (1.4).

Instead of previous quoted case studies, we now consider two separate characteristics: *Functionality* and *Content*. The *Functionality* characteristic splits into *Function Suitability* (2.1) and *Accuracy* (2.2), while the *Content* characteristic splits into *Information Suitability* (3.1) and *Content Accessibility* (3.2). As the reader can observe in Figure 13.11, we relate five measurable attributes to the *Function Suitability* subcharacteristic and three to *Function Accuracy*. In the latter subcharacteristic, we mainly consider precision attributes to recalculate values after making supported operations. On the other hand, in Olsina et al. (2006a) we also justified the inclusion of the *Content* characteristic for assessing the Web.

The following categories can help to evaluate information quality requirements of Web sites and applications (see also Lee et al., 2002):

**1. Usability**
- 1.1. Understandability
  - *1.1.1. Shopping cart icon/label ease to be recognized*
  - *1.1.2. Shopping cart labeling appropriateness*
- 1.2. Learnability
  - *1.2.1. Shopping cart help (for first-time visitor)*
- 1.3. Operability
  - *1.3.1. Shopping cart control permanence*
  - *1.3.2. Shopping cart control stability*
  - *1.3.3. Steady behavior of the shopping cart control*
  - *1.3.4. Steady behavior of other related controls*
- 1.4. Attractiveness
  - *1.4.1. Color style uniformity (links, text, etc.)*
  - *1.4.2. Aesthetic perception*

**2. Functionality**
- 2.1. Function Suitability
  - *2.1.1. Capability to add items from anywhere*
  - *2.1.2. Capability to delete items*
  - *2.1.3. Capability to modify an item quantity*
  - *2.1.4. Capability to show totals by performed changes*
  - *2.1.5. Capability to save items for later/move to cart*
- 2.2. Function Accuracy
  - *2.2.1. Precision to recalculate after adding an item*
  - *2.2.2. Precision to recalculate after deleting items*
  - *2.2.3. Precision to recalculate after modifying an item quantity*

**3. Content**
- 3.1. Information Suitability
  - *3.1.1. Shopping Cart Basic Information*
    - *3.1.1.1. Line item information completeness*
    - *3.1.1.2. Product description appropriateness*
  - *3.1.2. Shopping Cart Contextual Information*
    - 3.1.2.1. Purchase Policies Related Information
      - *3.1.2.1.1. Shipping and handling costs information completeness*
      - *3.1.2.1.2. Applicable taxes information completeness*
      - *3.1.2.1.3. Return policy information completeness*
    - *3.1.2.2. Continue-buying feedback appropriateness*
    - *3.1.2.3. Proceed-to-check-out feedback appropriateness*
- 3.2. Content Accessibility
  - *3.2.1. Readability by Deactivating the Browser Image Feature*
    - *3.2.1.1. Image title availability*
    - *3.2.1.2. Image title readability*
  - *3.2.2. Support for text-only version*

**4. Reliability**
- 4.1. Nondeficiency (Maturity)
  - *4.1.1. Link Errors or Drawbacks*
    - *4.1.1.1. Broken links*
    - *4.1.1.2. Invalid links*
    - *4.1.1.3. Reflective links*
  - *4.1.2. Miscellaneous Deficiencies*
    - *4.1.2.1. Deficiencies or unexpected results dependent on browsers*
    - *4.1.2.2. Deficiencies or unexpected results independent of browsers*

*Figure 13.11.* Specifying the external quality requirements tree to the shopping cart component from a general visitor standpoint.

- *Information accuracy.* This subcharacteristic addresses the very intrinsic nature of the information's quality. It assumes that information has its own quality per se. Accuracy is the extent to which information is correct, unambiguous, authoritative (reputable), objective, and verifiable. If a Web site becomes famous for inaccurate information, the Web site will likely be perceived as having little added value and will result in reduced visits.
- *Information suitability.* This subcharacteristic addresses the contextual nature of the information quality. It emphasizes the importance of conveying the appropriate information for user-oriented goals and tasks. In other words, it highlights the quality requirement that contents must be considered within the context of use and the intended audience. Therefore, suitability is the extent to which information is appropriate (appropriate coverage for the target audience), complete (relevant amount), concise (shorter is better), and current (see the specified attributes in Figure 13.11).
- *Accessibility.* It emphasizes the importance of technical aspects of Web sites and applications in order to make Web contents more accessible for users with various disabilities (see the specified attributes in Figure 13.11).
- *Legal compliance.* The capability of the information product to adhere to standards, conventions, and legal norms related to contents and intellectual property rights.

The INCAMI_Tool records all the information for a measurement and evaluation project. Besides the data in the project itself, it also saves to the *InformationNeed* class (see Figure 13.6) the purpose, user viewpoint, and context description meta-data; for the *CalculabeConcept* and *Attribute* classes, it saves all the names and definitions, respectively.

The *ConceptModel* class permits us to instantiate a specific model, that is, the external quality model in our case, allowing evaluators to edit and relate specific concepts, subconcepts, and attributes (the whole instantiated model looks like that in Figure 13.11, and an INCAMI_Tool screenshot of it appears in Figure 13.12).

## 13.4.2 Designing and Executing the Measurement and Elementary Evaluation

As mentioned in Section 13.2, the evaluators should design, for each measurable attribute of the instantiated external quality model, the basis for the measurement and elementary evaluation process by defining each specific metric and elementary indicator for the information needed accordingly.

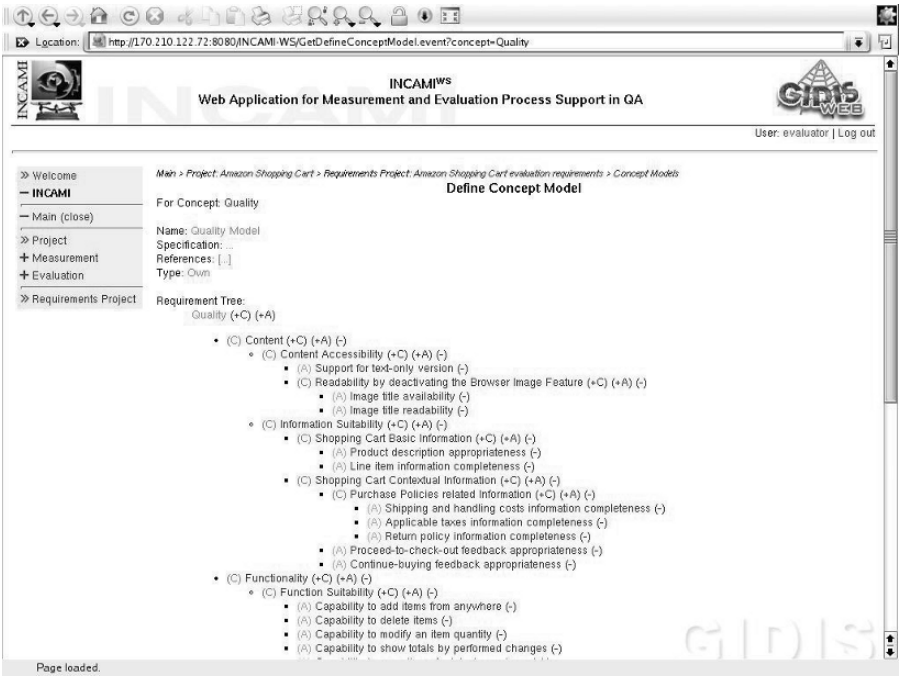



*Figure 13.12.* INCAMI_Tool screenshot to the instantiated concept model. Attributes are labeled with "A" on the left side of the tree; concepts and subconcepts with "C." In addition, "+C" and "+A" mean adding concepts or attributes, respectively, and "-" removing them.

In the design phase we record all the information for the selected metrics and elementary indicators regarding the conceptual schema of the *Metric* and *Elementary Indicator* classes shown in Figures 13.8 and 13.9, respectively. In addition, in Sections 13.4.2 and 13.4.3 the metric and indicator meta-data for the "Capability to delete items" attribute were illustrated. Finally, Figure 13.13 shows the name of the attributes and the name of each metric that quantifies them. Note that we can assign a metric for a given attribute by selecting it from the semantic catalogue (Molina et al., 2004); see the "Assign Metric" link in the figure.

Lastly, in the execution phase, we record for the *Measurement* and *Calculation* classes' instances the yielded final values for each metric and indicator. The data collection for the measurement activity was performed from December 19 to 30, 2005. From the metrics' values, the elementary indicators' values were calculated according to the respective elementary models.

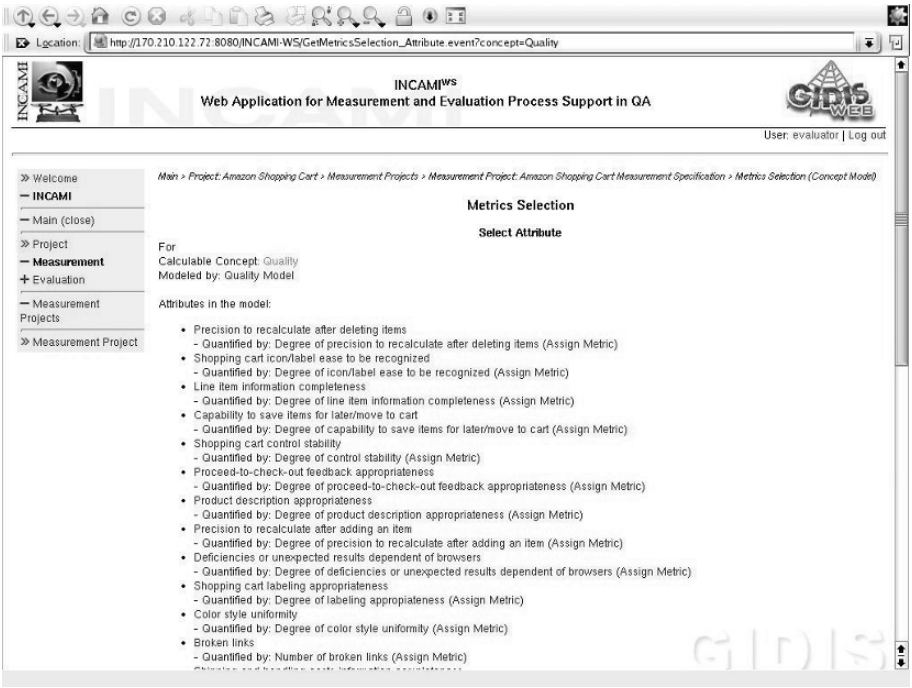



*Figure 13.13.* INCAMI_Tool screenshot of the metric selection process.

Figure 13.14 shows the selection process of a measurement value from a specific measurement project, which will be the input to the respective elementary indicator function in order to produce the indicator value (recall that for the same measurement project we can record measurement values at different times).

Once evaluators have designed and implemented the elementary evaluation, they should consider not only each attribute's relative importance but also whether the attribute (or subcharacteristic) is mandatory, alternative, or neutral. For this global evaluation task, we need a robust aggregation and scoring model, described next.

### 13.4.3 Designing and Executing the Partial/Global Evaluation

In the design of the global evaluation phase we select and apply an aggregation and scoring model (see *GlobalModel* class in Figure 13.9). Arithmetic or logic operators will then relate the hierarchically grouped attributes, subconcepts, and concepts accordingly.

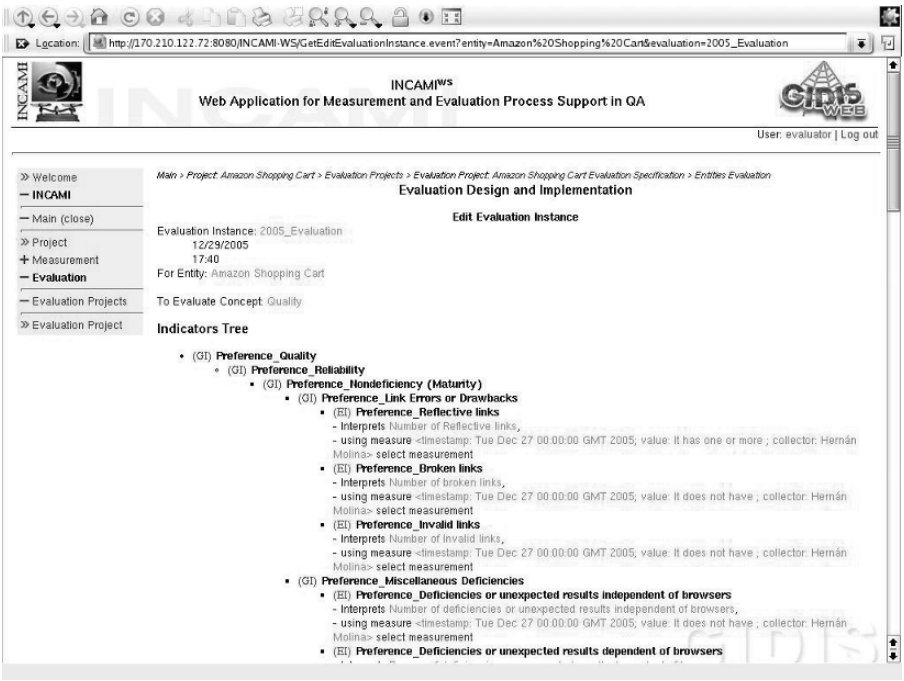



*Figure 13.14.* INCAMI_Tool screenshot of the selection process of a measure value for a given elementary indicator.

As mentioned earlier (see Definition 13.30), the INCAMI_Tool supports a linear additive or a nonlinear multicriteria scoring model (even other models can be used for designing the global evaluation such as fuzzy logic or neural networks not supported currently by the tool). We cannot use the additive scoring model to model input simultaneity (an *and* relationship among inputs) or replaceability (an *or* relationship), however, because it cannot express, for example, simultaneous satisfaction of several requirements as input. Additivity assumes that the insufficient presence of a specific attribute (in an input) can always be compensated for by the sufficient presence of any other attribute. Furthermore, additive models cannot model mandatory requirements; that is, a necessary attribute's or subcharacteristic's total absence cannot be compensated for by another's presence.

A nonlinear multicriteria scoring model lets us deal with simultaneity, neutrality, replaceability, and other input relationships using aggregation operators based on the weighted-power-means mathematical model. This model, called Logic Scoring of Preference (LSP) (Dujmovic, 1996), is a generalization of the additive scoring model and can be expressed as follows:

$$P/GI(r) = (W_1\, EI^r_1 + W_2\, EI^r_2 + \ldots + W_m\, EI^r_m)^{1/r}, \qquad (13.2)$$

where

$$-\infty \le r \le +\infty \; ; \; P/GI\,(-\infty) = \min\,(EI_1\,, EI_2\,, \ldots\,, EI_m),$$
$$P/GI\,(+\infty) = \max\,(EI_1\,, EI_2, \ldots\,, EI_m).$$

The power $r$ is a parameter selected to achieve the desired logical relationship and polarization intensity of the aggregation function. If P/GI($r$) is closer to the minimum, such a criterion specifies the requirement for input simultaneity. If it is closer to the maximum, it specifies the requirement for input replaceability. Equation (13.2) is additive when $r = 1$, which models the neutrality relationship; that is, the formula remains the same as in the first additive model. Equation (13.2) is supra-additive for $r > 1$, which models input disjunction or replaceability, and it's sub-additive for $r < 1$ (with $r! = 0$), which models input conjunction or simultaneity.

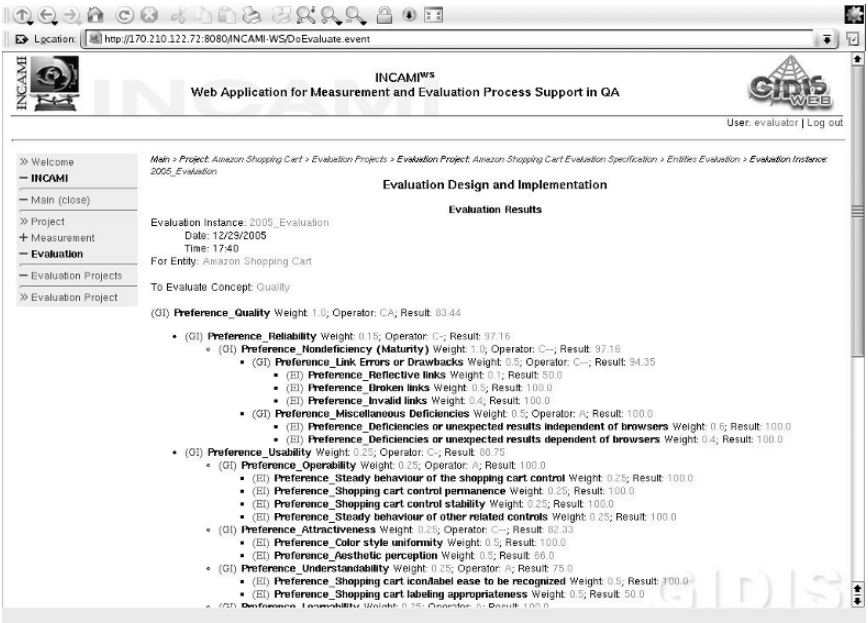
For our case study (as in previous ones), we selected this last model and used a 17-level approach of conjunction–disjunction operators, as defined by Dujmovic. Each operator in the model corresponds to a particular value of the $r$ parameter. When $r = 1$, the operator is tagged with A (or the + sign). The C conjunctive operators range from weak (C–) to strong (C+) quasi-conjunction functions, i.e., from decreasing $r$ values, starting from $r < 1$.

In general, the conjunctive operators imply that low-quality input indicators can never be well compensated for by a high quality of some other input to output a high-quality indicator (in other words, a chain is as strong as its weakest link). Conversely, disjunctive operators (D operators) imply that low-quality input indicators can always be compensated for by the high quality of some other input.

Designing the LSP aggregation schema requires answering the following key basic questions (which are part of the *Global Indicator Definition* task in Figure 13.5):

- What is the relationship among this group of related attributes and subconcepts: conjunctive, disjunctive, or neutral [for instance, when modeling the attributes' relationship for the *Function Suitability* (2.1) subcharacteristic, we can agree they are neutral or independent of each other]?
- What is the level of intensity of the logic operator, from a weak to strong conjunctive or disjunctive polarization?
- What is the relative importance or weight of each element in the aggregation block or group?

Figure 13.15 shows some details of the enacted requirements tree for amazon.com as generated by our tool. Particularly, in the top part of Figure 13.15 we can see LSP operators, weights, and final values for elementary, partial, and global indicators; the bottom part shows only the indicator values and the respective colored bars in a percentage scale.



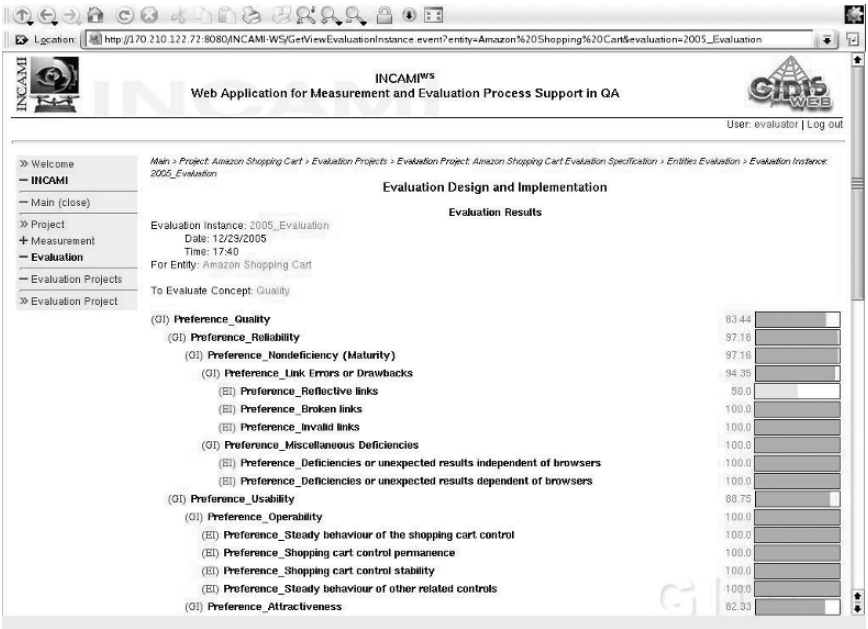




*Figure 13.15.* Once the weights and operators (in this case for the LSP aggregation model) were agreed on, the INCAMI_Tool yields elementary partial and global indicators in the execution phase, as highlighted in the figures. The top figure shows details of weights and operators, while the bottom figure shows just indicator values and the respective colored bars in the percentage scale.

### 13.4.4 Analyzing and Recommending

Once we have performed the final execution of the evaluation, decision makers can analyze the results and draw conclusions and recommendations. As stated in Section 13.4.1, we established (for illustration reasons) that the purpose in this study is to understand the external quality of the shopping cart component of a typical e-store, for a general visitor viewpoint in order to incorporate the best features in a new e-bookstore development project. The underlying hypothesis is that at the level of calculable concepts (characteristics in the ISO 9126 vocabulary) they accomplish at least the satisfactory acceptability range.

Table 13.1 shows the final results for the *Usability*, *Functionality*, *Content*, and *Reliability* characteristics and subcharacteristics, as well as partial and global indicator values for the amazon.com shopping cart.

*Table 13.1.* Summary of Partial and Global Indicators' Values for the Amazon.com Shopping Cart

| Code | Concept/Subconcept Name | Indicator Value |
|---|---|---|
| | *External Quality* | *83.44* |
| 1 | Usability | 88.75 |
| 1.1 | Understandability | 75.00 |
| 1.2 | Learnability | 100.00 |
| 1.3 | Operability | 100.00 |
| 1.4 | Attractiveness | 82.33 |
| 2 | Functionality | 87.61 |
| 2.1 | Function Suitability | 76.40 |
| 2.2 | Function Accuracy | 100.00 |
| 3 | Content | 71.40 |
| 3.1 | Information Suitability | 81.21 |
| 3.1.1 | Shopping Cart Basic Information | 81.70 |
| 3.1.2 | Shopping Cart Contextual Information | 80.47 |
| 3.1.2.1 | Purchase Policies related Information | 77.89 |
| 3.2 | Content Accessibility | 56.79 |
| 3.2.1 | Readability by Deactivating the Browser Image Feature | 67.75 |
| 4 | Reliability | 97.16 |
| 4.1 | Nondeficiency (Maturity) | 97.16 |
| 4.1.1 | Link Errors or Drawbacks | 94.35 |
| 4.1.2 | Miscellaneous Deficiencies | 100 |

The colored quality bars in the bottom part of Figure 13.15 indicate the acceptability ranges and clearly show the quality level each shopping cart feature has reached. In fact, the final indicator value to the external quality of

the Amazon shopping cart was satisfactory getting a rank of 83.44 [that is a similar global indicator value for the study made in late 2004 (Olsina et al., 2006), using the same requirements and criteria, which ranked 84.32%]. Notice that a score within a yellow bar (marginal) indicates a need for improvement actions. An unsatisfactory rating (red bar) means change actions must take high priority. A score within a green bar indicates satisfactory quality of the analyzed feature.

Looking at the *Usability, Functionality, Content,* and *Reliability* characteristics, we can see that the scores fall in the satisfactory level, so that we can emulate these features in a new development project. However, none of them is 100%. For instance, if we look at the *Functionality* characteristic and particularly at the *Function Suitability* subconcept, which ranked 76.40, we can observe that the reason for this score is in part due to the *Capability to Delete Items* (2.1.2) attribute, which is not totally suitable (the indicator value was 66%).

In order to make a thorough causal analysis, we must look at the elementary indicator and metric specification. Regarding the INCAMI_Tool, the following elementary indicator model specification (see Definition 13.26) was edited: CDI _PL = (0.33 * CDI) * 100, where CDI is the direct metric for the *Capability to Delete Items* attribute.

In the example of Definition 13.10, the scale of the direct metric was specified in this way:

1. Does not delete items at all.
2. Delete just all at once.
3. Delete one by one.
4. Delete one by one or delete the selected group at once.

Thus, the resulting indicator value in the execution phase was 66% because the Amazon shopping cart allows users to delete only one item at once, but does not allow the selected group to be deleted at once.

Ultimately, we observe that the state-of-the-art of the shopping cart quality of this typical site is rather high, but the wish list is not empty, because of some weak-designed attributes. Notice that elementary, partial, and global indicators reflect results of these specific requirements for this specific audience and should not be regarded as generalized rankings. Moreover, results themselves from a case study are seldom intended to be interpreted as generalizations (in the sense of external validity).

## 13.5 DISCUSSION AND FINAL REMARKS

Our experience suggests that it is necessary to select metrics for purpose-oriented attributes as well as to identify contextual indicators in order to start and guide a successful measurement and evaluation program. In fact, organizations must have sound specifications of metric and indicator meta-data associated consistently to data sets, as well as a clear establishment of frameworks and programs in order to make measurement and analyses and quality assurance useful support processes to software and Web development and maintenance projects. Ultimately, the underlying hypothesis is that without appropriate recorded meta-data of information needs, attributes, metrics, and indicators, it is difficult to ensure that measure and indicator values are repeatable and comparable among an organization's projects; consequently, analyses and comparisons can be carried out in an inconsistent way as well.

Throughout this chapter we have stated that in order to build a robust and flexible measurement and evaluation program, at least three pillars are necessary: (1) a process for measurement and evaluation (outlined in Section 13.2); (2) a measurement and evaluation framework based on an ontological base (analyzed in Section 13.3); and (3) specific model-based methods and techniques for the realization of measurement and evaluation activities (a particular inspection method was illustrated in Section 13.4).

As a matter of fact, in the present chapter we have emphasized the importance of counting with a measurement and evaluation conceptual framework. The discussed INCAMI framework is based on the assumption that for an organization to measure and evaluate in a purpose-oriented way, it must first specify nonfunctional requirements starting from information needs, then it must design and select the specific set of metrics for measurement purposes, and last it must interpret the metric values by means of contextual indicators with the aim of evaluating or estimating the degree to which the stated information need has been met. Thus, consistent and traceable analyses, conclusions, and recommendations can be drawn.

Regarding other initiatives, the GQM (*Goal-Question-Metrics*) paradigm (Basili and Rombach, 1989) is a useful, simple, purpose-oriented measurement approach that has been used in different measurement projects and organizations. However, as Kitchenham et al. pointed out (2001), GQM is not intended to define metrics at a level of detail suitable to ensure that they are trustworthy, in particular, whether or not they are repeatable. Contrary to our approach, which is based on an ontological conceptualization of metrics and indicators, GQM lacks this conceptual base, and so it cannot assure that measurement values (and the associated meta-data like scale, unit, measurement method, and so forth) are trustworthy and consistent for ulterior analysis among projects.

On the other hand, GQM is a weak framework for evaluation purposes, i.e. GQM lacks specific concepts for evaluation in order to interpret attributes' measures. For instance, elementary and global indicators and related terms are essential for evaluation as shown in the previous sections. Conversely, GQM is more flexible than INCAMI in the sense that it is not always necessary to have a concept model specification in order to perform a measurement project.

In our humble opinion, an interesting improvement to the GQM approach that considers indicators has recently been issued as a technical note (Goethert and Fisher, 2003). This approach uses both the *Balance Scorecard* technique (Kaplan and Norton, 2001) and the *Goal-Question-Indicator-Measurement* method in order to purposely derive the required enterprise goal-oriented indicators and metrics. It is a robust framework for specifying enterprise-wide information needs and deriving goals and subgoals and then operationalizing questions with associated indicators and metrics. It says, "The questions provide concrete examples that can lead to statements that identify the type of information needed. From these questions, displays or indicators are postulated that provide answers and help link the measurement data that will be collected to the measurement goals" (Goethert and Fisher, 2003). However, this approach is not based on a sound ontological conceptualization of metrics and indicators as ours; furthermore, the terms "measure" and "indicator" are sometimes used ambiguously, which can result in data sets and meta-data being recorded inconsistently.

On the other hand, there exist other close initiatives to our research, such as the Kitchenham et al. (2001) conceptual framework as well as the cited ISO standards related to software measurement and evaluation processes. In summary, we tried to strengthen these contributions not only from the conceptual modeling point of view, but also from the ontological point of view, including a broader set of concepts.

Lastly, we argue that the INCAMI framework can be a useful conceptual base and approach for different qualitative and quantitative evaluation methods and techniques with regard to the requirement, measurement, and evaluation concepts and definitions analyzed in Section 13.3. Apart from inspection or *feature analyses* methods (like WebQEM), this framework can be employed for some other methods, such as neural networks and fuzzy logic, when they are intended to measure and evaluate quality, quality in use, and cost, among other calculable concepts.

Finally, due to the importance of managing the acquired enterprise-wide contextual knowledge during measurement and evaluation and during quality assurance projects, a semantic infrastructure that embraces contextual information and organizational memory management is currently being considered in the INCAMI framework. This will be integrated to the

INCAMI_Tool and framework, also making sure that ontologies and the Semantic Web are enabling technologies for our previous (Molina et al., 2004) and current research aims as well.

## ACKNOWLEDGEMENTS

This research is supported by Argentina's UNLPam-09/F037 project, as well as the PICTO 11-30300 and PAV 127-5 research projects.

## REFERENCES

Basili, V., and Rombach, H.D., 1989, The TAME project: Towards improvement-oriented software environments. *IEEE Transactions on Software Engineering*, **14**(6): 758–773.

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

CMMI, 2002, Capability Maturity Model Integration, Version 1.1, CMMI$^{SM}$ for Software Engineering (CMMI-SW, V. 1.1) Staged Representation CMU/SEI-2002-TR-029, CMMI Product Team, SEI Carnegie Mellon University (available online).

Dujmovic, J., 1996, A method for evaluation and selection of complex hardware and software systems. *Proceedings 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise CS* (CMG 96), Vol. 1, pp. 368–378.

Goethert, W., and Fisher, M., 2003, Deriving enterprise-based measures using the balanced scorecard and goal-driven measurement techniques. Software Engineering Measurement and Analysis Initiative, CMU/SEI-2003-TN-024 (available online).

ISO/IEC 14598-5, 1998, Information technology—Software product evaluation—Part 5: Process for evaluators.

ISO/IEC 14598-1, 1999, International standard, information technology—Software product evaluation—Part 1: General overview.

ISO/IEC 9126-1, 2001, International standard, software engineering—Product quality—Part 1: Quality model.

ISO/IEC 15939, 2002, Software engineering—Software measurement process.

Kaplan, R., and Norton, D., 2001, *The Strategy-Focused Organization, How Balanced Scorecard Companies Thrive in the New Business Environment*. Harvard Business School Press, Boston.

Kitchenham, B.A., Hughes, R.T., and Linkman, S.G., 2001. Modeling software measurement data. *IEEE Transactions on Software Engineering*, **27**(9): 788–804.

Lee, Y.W., Strong, D.M., Kahn, B.K., and Wang, R.Y., 2002, AIMQ: A methodology for information quality assessment. *Information & Management,* **40**(2): 133–146.

Molina, H., Papa, F., Martín, M., and Olsina, L., 2004, Semantic capabilities for the metrics and indicators cataloging Web system. In *Engineering Advanced Web Applications*, M. Matera and S. Comai, eds., Rinton Press Inc., Princeton, NJ, pp. 97–109, ISBN 1-58949-046-0.

Nielsen, J., Molich, R., Snyder, C., and Farrell, S., 2001, E-Commerce User Experience, NN Group.

Olsina, L., Godoy, D., Lafuente, G., and Rossi, G., 1999, Assessing the quality of academic Websites: A case study. *New Review of Hypermedia and Multimedia (NRHM) Journal*, **5**: 81–103.

Olsina, L., Lafuente, G., and Rossi, G., 2000, E-commerce site evaluation: A case study. *Proceedings 1st International Conference on Electronic Commerce and Web Technologies* (EC-Web 2000), London, Springer LNCS 1875, pp. 239–252.

Olsina, L., Papa, M.F., Souto, M.E., and Rossi, G., 2001, Providing automated support for the Web quality evaluation methodology. *Proceedings 4th Workshop on Web Engineering, at the 10th International WWW Conference*, Hong Kong, pp. 1–11.

Olsina, L., and Rossi, G., 2002, Measuring Web application quality with WebQEM. *IEEE Multimedia*, **9**(4): 20–29.

Olsina, L., and Martin, M., 2004, Ontology for software metrics and indicators. *Journal of Web Engineering,* **2**(4): 262–281, ISSN 1540-9589.

Olsina, L., Papa, F., and Molina, H., 2005, Organization-oriented measurement and evaluation framework for software and Web Engineering projects. *Proceedings International Congress on Web Engineering* (ICWE05), Sydney, Australia, Springer, LNCS 3579, pp. 42–52.

Olsina, L., Covella, G., and Rossi, G., 2006, Web quality. Chapter 4 in *Web Engineering,* E. Mendes and N. Mosley, eds., Springer, New York, ISBN 3-540-28196-7.

Olsina, L., Papa, F., and Molina, H., 2008, Ontological support for a measurement and evaluation framework. To appear in the *Journal of Intelligent Systems*.

Zuse, H., 1998, *A Framework of Software Measurement*, Walter de Gruyter, Berlín.