Software Qual J (2012) 20:201–228
DOI 10.1007/s11219-011-9136-9

A conceptual modeling quality framework

H. James Nelson · Geert Poels · Marcela Genero · Mario Piattini

Published online: 19 April 2011

Abstract The goal of any modeling activity is a complete and accurate understanding of the real-world domain, within the bounds of the problem at hand and keeping in mind the goals of the stakeholders involved. High-quality representations are critical to that understanding. This paper proposes a comprehensive Conceptual Modeling Quality Framework, bringing together two well-known quality frameworks: the framework of Lindland, Sindre, and Sølvberg (LSS) and that of Wand and Weber based on Bunge's ontology (BWW). This framework builds upon the strengths of the LSS and BWW frameworks, bringing together and organizing the various quality cornerstones and then defining the many quality dimensions that connect one to another. It presents a unified view of conceptual modeling quality that can benefit both researchers and practitioners.

Keywords Conceptual modeling · Quality · System development process

1 Introduction

Accurate representations are critical for understanding the highly complex problem domains that exist in today's organizations. Conceptual modeling creates representations,

H. J. Nelson (✉)
Department of Management, Southern Illinois University, Carbondale, IL 62901-4627, USA
e-mail: nelson.j@cba.siu.edu

G. Poels
Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, 9000 Ghent, Belgium
e-mail: geert.poels@UGent.be

M. Genero · M. Piattini
Department of Technologies and Information Systems, University of Castilla-La Mancha, Paseo de la Universidad, 4, 13071 Ciudad Real, Spain
e-mail: marcela.genero@uclm.es

M. Piattini
e-mail: mario.piattini@uclm.es

abstractions that have much of the complexity found in real-world problem domains removed. The result is a representation that contains the critical information that is needed for designing and applying effective organizational strategies, process and knowledge management, and a necessary foundation for constructing an organization's databases and information systems. The discipline of conceptual modeling is gaining more relevance with the emergence of Model Driven Architecture (MDA) that promotes the use of models at all steps of a software development project from initial planning through the delivery of the software on a given platform (Wagelaar and Van Der Straeten 2007).

Despite the importance of having accurate representations, the notion of quality in conceptual modeling is still immature and is a rapidly evolving concept. Early research simply lists desirable properties that can be found in "good" conceptual representations (Batini et al. 1991). Definitions, when given, are vague and complicated, and there is no underlying structure that helps to understand how the properties relate to one another. The result is a set of *ad hoc* evaluations of representations, and little agreement about what makes a representation "good" (Moody et al. 1998).

There are many frameworks that attempt to bring order and enlightenment to the quality of conceptual modeling representations and to the quality of the conceptual modeling process. These range from rigorous theory-based frameworks to speculative frameworks. Two frameworks stand out from the mass: the first is a framework introduced by Lindland, Sindre, and Sølvberg (LSS) (Lindland et al. 1994) based on Morris' semiotics theory, and the second is a framework developed by Wand and Weber based on Bunge's ontological theory (the Bunge–Wand–Weber representational model: BWW) (Wand and Weber 1990a). Although both frameworks have solid theoretical foundations, they look at quality in conceptual modeling from two different perspectives. The LSS framework focuses on the product of conceptual modeling whereas the BWW framework focuses on the process of conceptual modeling. Significant empirical research has been conducted using both of these frameworks, but the results obtained are fragmentary and hard to integrate. Proposals made to improve the quality of conceptual representations involve (somehow) changes made to the conceptual modeling process. Vice versa, proposals made to improve the conceptual modeling process should be ultimately evaluated by their effect on the end result, i.e., the conceptual representation. To further guide research on conceptual modeling quality, a framework is needed to better understand how the modeled domain, conceptual model, and conceptual representation interact with the modelers involved in conceptual modeling and the process they follow, and how this all relates ultimately to representation quality.

In this paper, we propose a Conceptual Modeling Quality Framework (CMQF). This framework is useful for evaluating not only the end result of the conceptual modeling process, the conceptual representation, but also the quality of the modeling process itself. This comprehensive and unified framework integrates the LSS and the BWW frameworks and can serve as a guide for both researchers and practitioners. For example, many studies focus on the quality of the conceptual model (such as the Entity-Relationship Model) or the quality of the conceptual representation (such as an Entity-Relationship diagram that represents a specific situation in a problem domain). Relatively few studies examine the quality of the modelers' *knowledge* of the conceptual model or representation. Fewer still examine the quality of the process of conceptual modeling and focus exclusively on product quality.

The CMQF is, at this point, a theoretical framework. However, it can serve as a theoretical basis and an organizing framework for to ground future research on quality in conceptual modeling. In addition, it can serve as a base for developing quality assessment tools and techniques. This foundation may assist practitioners in developing quality improvement and assessment programs.

The remainder of the paper is arranged as follows. The next section lays the foundation for the CMQF with an examination of the LSS and the BWW quality frameworks. This is followed by a discussion of the combined and enhanced framework cornerstones and their interrelationships. We then describe the layers and quality dimensions of the CMQF, followed by an example of the use of the CMQF. We finish with our conclusions and suggestions for further research.

2 Foundations

One of the earliest complete frameworks for exploring conceptual representation quality was developed by Lindland et al. (1994) and subsequently underwent numerous extensions and refinements (Krogstie and Sølvberg 2003). For instance, in Krogstie et al. 2006, it was extended to include evaluations of the appropriateness of modeling languages. The framework, shown in Fig. 1, is based on the assumption that the modeling domain, the model externalization, the modeler's knowledge, and so on can be considered as sets of statements (or facts) in a language. Under that assumption, quality is defined as a comparison between these sets. For example, semantic quality is determined by comparing the set of statements of the modeling domain to the set of statements in the model externalization.

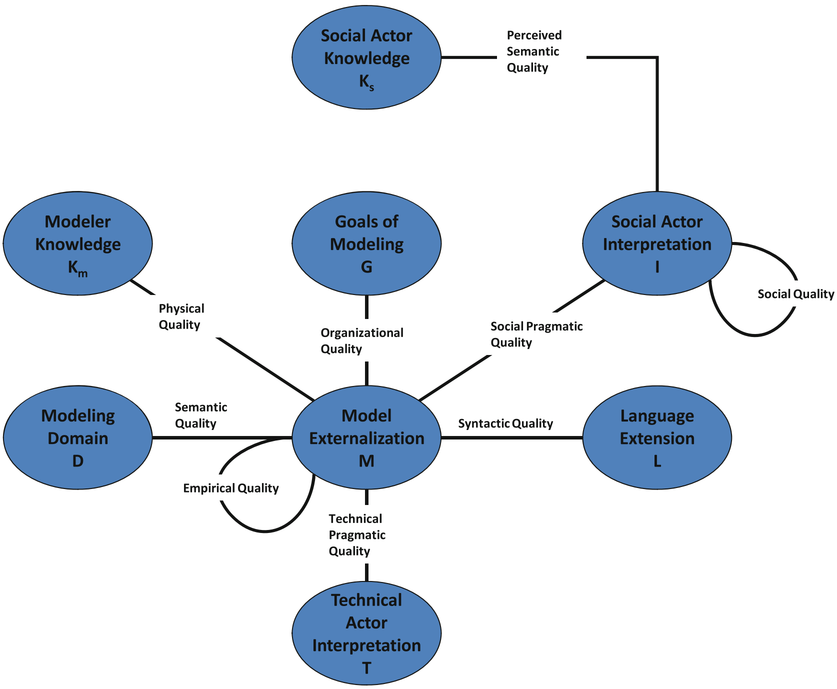

Fig. 1 The LSS conceptual representation quality framework (Krogstie 2003)

The sets of statements that form the cornerstones of the LSS quality framework are defined as follows (Krogstie et al. 2006): The *Modeling Domain* is the set of statements that are correct and relevant about the situation at hand. It can be considered as the content that must be captured in a conceptual representation. The *Goals of Modeling* are the normally organizationally motivated goals of the modeling task. They answer the question why conceptual modeling is undertaken. The *Language Extension* is the set of all statements that are possible to make given the modeling language's syntax. The *Model Externalization* is the conceptual representation defined as the set of all statements that are made given someone's view (perceived reality, shaped by the organizational goals of the modeling process) of the modeling domain and made using statements in the language extension. The *Social Actor Knowledge* is the relevant knowledge of the modeling domain of all the stakeholders who are involved in some way in the modeling process. The *Modeler Knowledge* is the relevant knowledge of the modeling domain of those modelers who are actively involved in the modeling process, so modeler knowledge is a subset of social actor knowledge. The *Social Actor Interpretation* contains the statements in the conceptual representation as understood by all the stakeholders involved in modeling. The *Technical Actor Interpretation* contains the statements in the conceptual representation that can be interpreted by the modeling tools used in modeling.

The various quality dimensions of the LSS framework have their foundations in the theory of semiotics. Semiotics itself consists of a theory of codes and a theory of sign production that is used to convey meaning about things or states of things in the world. Stamper (Stamper 1992; Stamper et al. 2000) used semiotic theory to develop a six-layer framework to help classify and understand the impact of information technology: the social, pragmatic, semantic, syntactic, empiric, and physical layers. The LSS quality framework is based on these six layers.

In the LSS framework, conceptual modeling quality is defined through equations on pairs of sets of statements, defined above (Moody et al. 2003). Given two cornerstone sets *A* and *B*, the quality of *B* can be defined in relation to *A*. If there are statements in *A* that are correct and relevant for B, but not contained in *B*, then *B* is incomplete with respect to those statements. The statements in *B* that are not contained in *A* are invalid (i.e., incorrect or irrelevant statements). If *B* is complete and valid with respect to *A*, then *B* can be said to be of high quality. Alternatively, if *B* is complete and valid with respect to *A*, then the quality of *B* is assured.

For example, the syntactic quality of the conceptual representation (Model Externalization M) is assessed by comparing the set of statements in the representation to the set of statements in the Language Extension L which is the set of all statements that are allowable given a modeling language's vocabulary and grammar. The representation is syntactically valid if all statements in the representation are in the language extension. Those statements in the representation that are not in the language extension are syntactically invalid. However, not all statements in the language extension should be contained in the model externalization; only those that are relevant and correct for the model externalization. Which language extension statements are relevant and correct for the model externalization cannot be determined without referring to other framework cornerstones. Other conceptual modeling quality dimensions explicitly defined in the LSS framework are organizational quality, social quality, (social and technical) pragmatic quality, semantic and perceived semantic quality, empirical quality and physical quality, as described below.

- *Syntactic quality*: How well the representation corresponds to the language extension.
- *Semantic quality*: How well the representation corresponds to the domain. There are two semantic goals; *validity* which means that all statements made in the model externalization are correct and relevant to the domain, and *completeness* which means that the model externalization contains all the statements that would be correct and relevant about the domain.
- *Perceived semantic quality* is the similar correspondence between the audience interpretation of a model externalization and their knowledge of the domain.
- *Pragmatic quality* is the correspondence between the audience interpretation and the model externalization itself. Social pragmatic quality represents how well the stakeholders understand the conceptual representation. Technical pragmatic quality captures the degree to which modeling tools can interpret the representation. Feasibility is also defined for pragmatic quality.
- *Social quality*: The agreement among participant interpretations.
- *Physical quality*: The degree to which modelers were able to externalize their domain knowledge. How well the conceptual representation expresses the tacit domain knowledge of the modelers.
- *Empirical quality*: The "readability" of a representation defined as the range of errors that occur across many readings of the same representation or errors across many different ways of representing the same model externalization.

To further understand and analyze the LSS framework, its cornerstones can be rearranged based on Searle's (1995) distinction between physical and social reality. This is embodied in two propositions: there is a physical "real world" that is composed of things and those things have properties by which they are known (Russell 1903), and there is a separate cognitive representation of the real world constructed by our perceptions (Johnson-Laird 1983). This splits the LSS framework into those cornerstones that exist in the real world and those that exist in the mind, as shown below in Fig. 2. The Modeling Domain, Language Extension, and the Model Externalization are all artifacts that exist in the real world. Social Actor Knowledge, Modeler Knowledge, Technical Actor Interpretation, and Social Actor Interpretation all exist in the mind as social or institutional knowledge. While the goals of modeling also exist in the mind, they actually form the context that all other cornerstones exist within (and therefore this cornerstone is not explicitly shown in Fig. 2). The distinction between "real world" and "in the mind" cornerstones indicates that there are different types of quality dimensions. Some dimensions like syntactic quality are a physical reality, others like social quality are a social reality, whereas still others like pragmatic quality are defined in relation to concepts in different worlds. We will discuss the implication of these distinctions in a subsequent section, where our CMQF framework is presented.

Although the LSS framework has been updated and extended several times, it remains a relatively product-focused view, focusing on the quality of the conceptual representation rather than the process of conceptual modeling (Krogstie et al. 2006). Wand and Weber (1988, 1990a, 1995), Wand and Wang (1996) took a process-oriented approach and created an alternative quality framework based on Bunge's (1977) ontology. Known as the Bunge–Wand–Weber (BWW) framework, it considers the process of conceptual modeling as it moves from the real world through a number of transformations, eventually arriving at a conceptual modeling script: the information system. An information system, in the BWW framework, is "an artifactual representation of a real-world system, as perceived by someone" (Wand and Weber 1990b, p. 62). The ontological framework is shown below in Fig. 3.

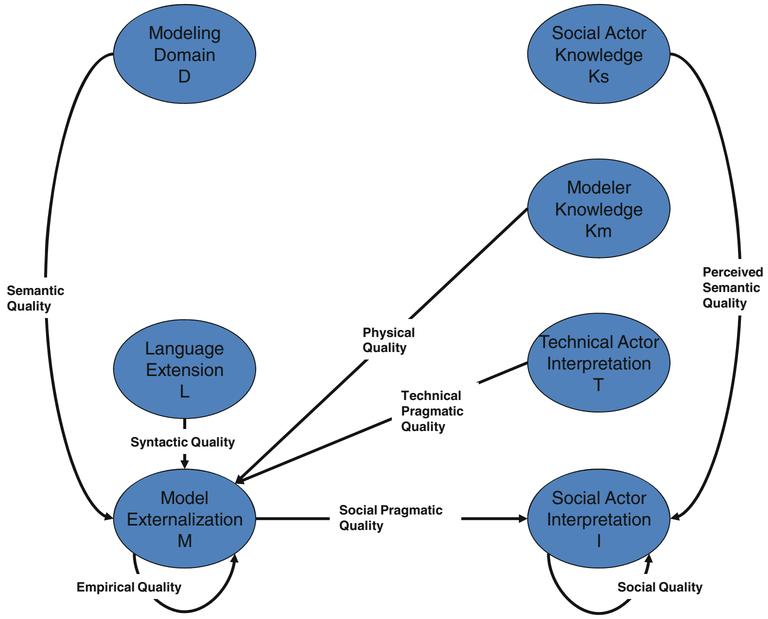

Fig. 2 The LSS framework arranged by physical and social reality

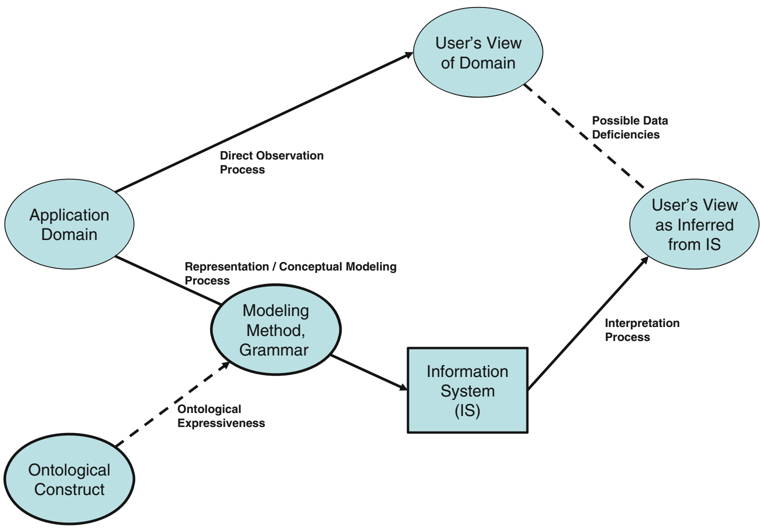

Fig. 3 The BWW ontological representation framework

The BWW framework begins with the *Application Domain*, also known as the real-world system (Wand and Wang 1996). Analysts perceive the application domain and from that perception use the *representation/conceptual modeling process* to create the *Information System*. The *Information System* is a conceptual modeling script, the end product of the modeling process (Wand and Weber 2002). The modeling process uses the *Modeling*

Grammar, a set of modeling language constructs derived from or mapped onto ontological constructs, and an associated set of rules that show how to combine these constructs into representations of the real world (Wand and Weber 2002). The *Ontological Construct* is the philosophical basis for modeling things in the real world (Wand and Weber 1990b). A system of ontological constructs (e.g., events, actions, states, and so on) can be used to categorize the basic things in the real world that could be in the representation.

The intended users of the information system use the *direct observation process* on the application domain to produce the *User's View of Domain*. This view is filtered by the user's particular domain needs, requirements, and so on and is a representation of known aspects of real-world system, as perceived by users (Wand and Wang 1996). Although not explicitly shown in Fig. 3, analysts may perceive an application domain via the users' view of the domain, e.g., through interviewing users. Once the information system is made accessible, users also use the *interpretation process* to create the *User's View as Inferred from the Information System* (Wand and Wang 1996). The BWW framework has two conceptual modeling quality checks: the ontological expressiveness of the modeling method and grammar, and the possible data deficiencies found by comparing the user's view of the application domain to their view of the domain as represented by the information system.

Similar to the LSS framework, the BWW framework has elements that exist in the real world and elements that exist in the mind. Rearranging the elements according to Searle (1995) produces Fig. 4, below.

The BWW framework has many of the same cornerstones as the LSS framework with some very important differences. The LSS Language Extension is the set of all statements that can be made using the language syntax. For most languages, this is an infinite number of statements (Lindland et al. 1994). The BWW framework has this as the modeling grammar: the set of rules from which the set of modeling statements can be made. However, it also contains the modeling method and the ontological construct. Ontological constructs provide a theoretical basis for modeling languages. Separating Ontological

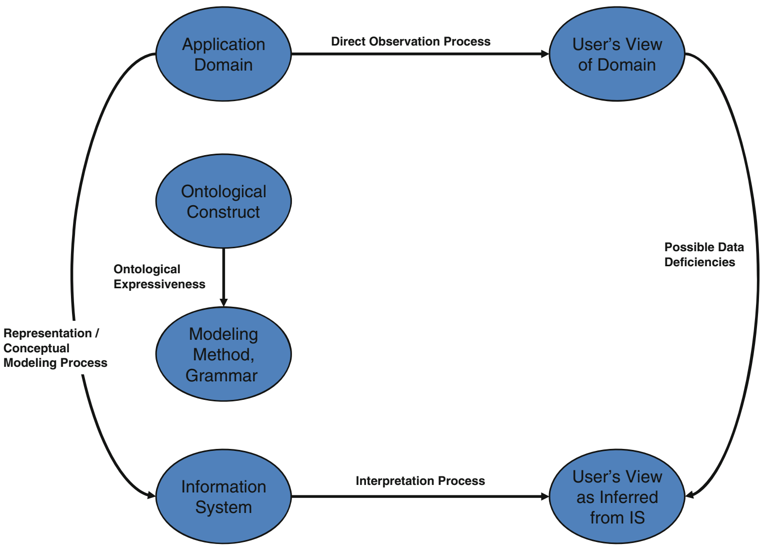

Fig. 4 The BWW framework rearranged

Construct and Modeling Grammar allows a single ontology to serve as the basis for a number of different modeling languages. Similarly, there can be multiple methods that guide the application of a modeling grammar.

Searle's distinction between physical and social reality provided the first "vertical" organizing method for the quality cornerstones. A second "horizontal" organizing method can be seen in the BWW framework and is described by Bunge (Bunge 1977). Physical reality, and by extension cognitive/social reality, can be divided into the domain, reference framework (or ontology), modeling language, and resulting model (i.e., the conceptual representation of the domain) (Recker et al. 2007). By rearranging both the LSS and the BWW frameworks vertically and horizontally, we can see that they have similar cornerstones. BWW seems to be more complete than LSS as it has a cornerstone for the reference framework in the physical reality (i.e., Ontological Construct). Both frameworks miss cornerstones for the reference framework and modeling language in the social reality, as shown in Fig. 5.

The LSS framework, focusing on the product of conceptual modeling, and the BWW framework, focusing on the process of conceptual modeling, each have elements that are important to include in any attempt to describe conceptual modeling quality. A combined framework with the extensive relationships and semiotic foundation of the LSS framework and the additional Ontological Construct cornerstone and ontological foundation of the BWW framework should have considerable utility for informing research about conceptual modeling quality: where research has been already conducted and where there are gaps in important quality dimensions that have yet to be explored. The next section presents a combined and extended set of quality cornerstones, combining cornerstones that are similar and filling the empty cells of Fig. 5 (i.e., where cornerstones were missing in both LSS and BWW).

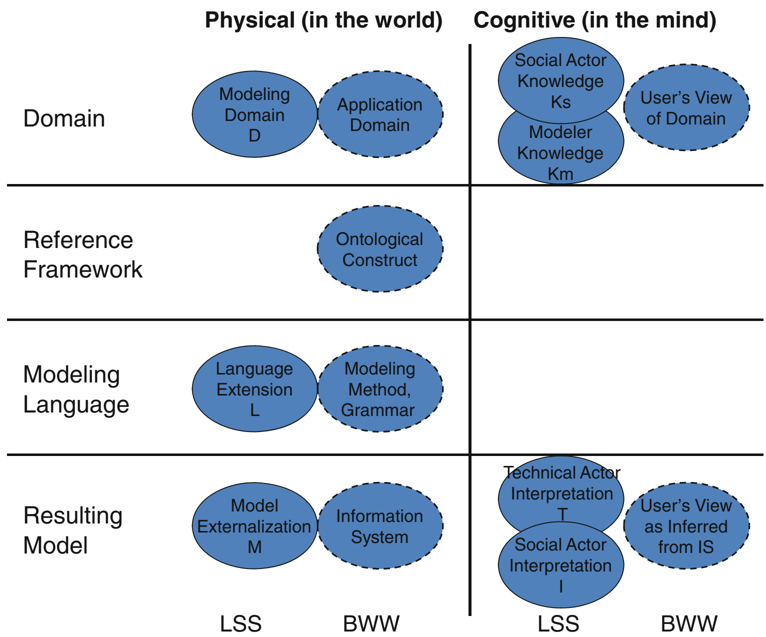

Fig. 5 Combined LSS and BWW frameworks

3 Quality cornerstones

Rearranging the combined LSS and the BWW frameworks "vertically" (i.e., distinguishing physical and social reality) and "horizontally" (i.e., distinguishing domain, reference framework, modeling language and conceptual representation) gives eight possible cornerstones. These cornerstones can be thought of as either sets of statements that constitute physical artifacts or represent cognitive artifacts used in or resulting from conceptual modeling (as in the LSS framework) or as sets of states of the conceptual modeling process (as in the BWW framework). The complete framework and all relationships are shown in Fig. 6 and explained below.

Physical Domain. The physical domain is the real-world universe of discourse, meaning all of the things and/or phenomena that are of interest to the users and the conceptual modelers. It corresponds to the LSS modeling domain. It is the real world in the BWW framework (Wand and Weber 1990b).

Domain Knowledge. The domain knowledge is the understanding of the real-world universe of discourse by both users and modelers involved in the process. It corresponds to the social actor's knowledge and its subset, and the modeler's knowledge (LSS) created by their perception of the application domain (BWW). This view consists of those elements shaped by the user's and/or modeler's context that they consider meaningful to the situation at hand.

Physical Model. A model can be thought of as a "lens for the mind." It filters and shapes the way the world is seen. A model focuses on the things that are important to the problem at hand and to discard those things that have nothing to do with the problem. The model is the Ontological Construct in the BWW framework but is not present in the

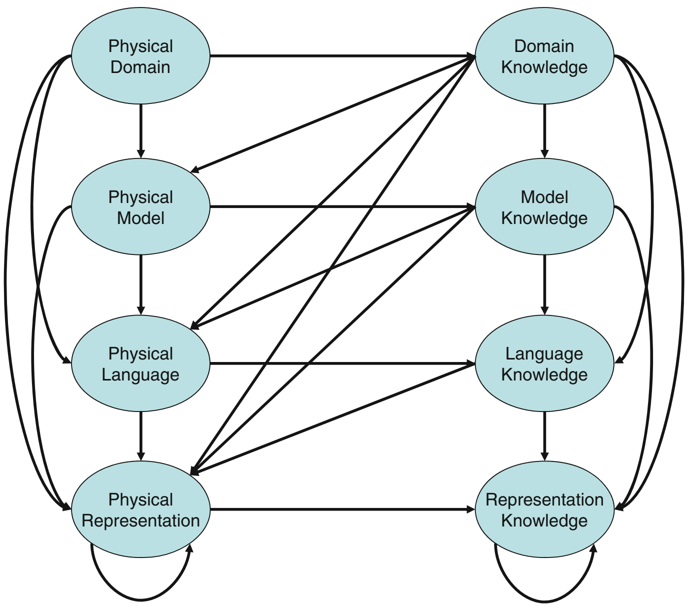

Fig. 6 Complete framework of conceptual modeling quality cornerstones

LSS framework. This model is preferably explicated and possibly formalized using first-order logic axioms that define the domain ontology.

Model Knowledge. This is the model as understood by the users and modelers who are involved in the modeling process. It is their mental model as shaped by the physical model. It is the "lens of the mind." There is no corresponding element in the BWW framework nor does it exist in the LSS framework.

Physical Language. The language consists of the grammar and the vocabulary that are used in combination with the model. This cornerstone corresponds to the Modeling Method and Grammar construct of the BWW framework and to the Language Extension element of the LSS framework.

Language Knowledge. This is the language as understood by those modelers who are actively involved in the modeling process. It is not included in the LSS and BWW frameworks.

Physical Representation. The physical representation corresponds to the LSS Model Externalization or the BWW Information System. It is the user's description rendered into a formalized (data flow diagram, ER diagram, UML diagram, or coded information system) model-based representation. A representation is built for use by the user whose view of the application domain is captured in the design (Wand and Wang 1996).

Representation Knowledge. This is the users' cognitive interpretation of the physical representation. It corresponds to the LSS Social Actors' Interpretation or the BWW Users' View of the Information System.

The cornerstones identified in Fig. 6 contain (in one form or another) all of the elements of the LSS and the BWW frameworks. Three LSS elements that are explicit in their framework are found implicitly in the CMQF. The LSS Goal of Modeling element is part of the overall context of the modeling exercise and helps to determine what is interesting, useful, economical, and so on. It is related to the Domain, to the Model, and to the Language framework cornerstones. The LSS framework also has a Technical Actor Interpretation element, which is included, if needed, in the Representation Knowledge element. The LSS Modeler Knowledge is a subset of the Social Actor Knowledge, hence included in the Domain Knowledge cornerstone.

4 The conceptual modeling quality framework

The framework cornerstones discussed above form the foundation of the Conceptual Modeling Quality Framework (CMQF) that we propose here and provide the grounding for the discussion of the conceptual modeling quality layers and quality dimensions that follows. The CMQF is a combination and an extension of both the LSS and the BWW quality frameworks. The similarities and differences are noted and help clarify the quality dimensions.

Conceptual modeling quality in the LSS framework is achieved through a correspondence between two framework elements with the goal being completeness and validity. With the elements being defined as sets of statements, completeness is achieved if (for example) the physical representation contains all the statements about the physical domain that are correct and relevant (Lindland et al. 1994). Relevance is critical, since it is clear that the physical domain will contain far more statements than the physical representation. It is the context of the modeling exercise (or the Goal of Modeling in LSS) that determines what is relevant or not. The representation is an abstraction of the relevant part of the domain with its purpose being clarity and understanding.

Conceptual modeling quality in the BWW framework is also achieved through a correspondence between two framework elements. The elements are considered to be state spaces, where the state of an element is the set of all values of all of its attributes as allowed by the set of laws defined in the ontological construct (Wand and Wang 1996). Quality is achieved by mapping between these state spaces. For example, for the physical domain to be properly represented by the physical representation, every lawful state (a state allowed by the ontology) in the physical domain should be mapped to at least one lawful state in the physical representation. Every lawful state in the physical representation should, in principle, be mapped back to a lawful state in the physical domain. In this way, a representation will be complete, unambiguous, meaningful, and correct.

The eight elements and their associated quality types are split into four "layers" that roughly follow the conceptual modeling process. These layers are the physical layer, knowledge layer, learning layer, and development layer. These layers are shown in Fig. 7, outlined in Table 1, and described in detail below. In Table 1, Quality Type (for example, *Semantic Quality*) is the name given to the Conceptual Modeling Quality Framework quality dimensions. Quality Type is defined as a relationship between the Quality Reference and the Object of Interest. The Object of Interest is the quality framework cornerstone that is being examined across the quality dimensions. The Quality Reference is the quality framework cornerstone to which the Object of Interest is being compared for completeness and validity. For example, the *Physical Model* (the object of interest) can be said to have *Semantic Quality* if it is complete and valid relative to the *Physical Domain* (the quality reference) within the defined conceptual modeling context, following the goals of the organization (or individual or group of individuals) performing the modeling process.

4.1 The physical layer

The first of the five layers is the physical layer. This layer contains the physical, observable elements of the quality framework, distinct from their cognitive counterparts. It is the explicated and shared layer elements of the framework rather than the tacit and individual

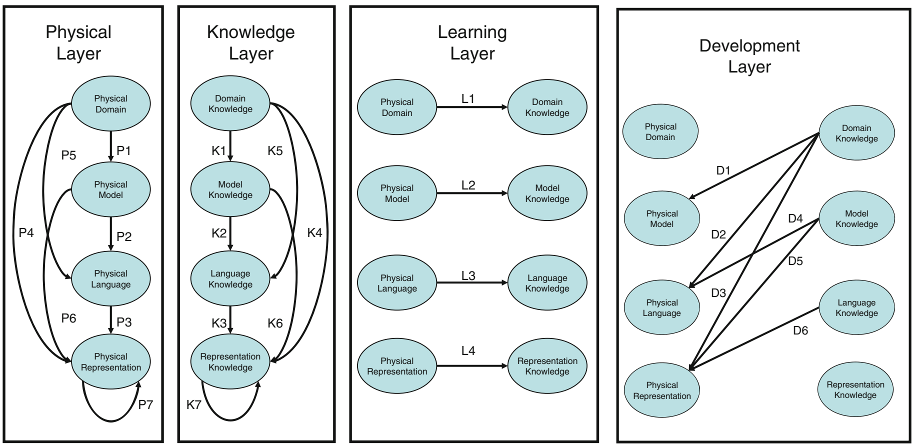

Fig. 7 The CMQF quality layers

Table 1 Quality types and their associated quality cornerstones

Label	Quality type	Quality reference	Object of interest
P1	Model-domain appropriateness	Physical domain	Physical model
P2	Ontological quality	Physical model	Physical language
P3	Syntactic quality	Physical language	Physical representation
P4	Semantic quality	Physical domain	Physical representation
P5	Language-domain appropriateness	Physical domain	Physical language
P6	Intensional quality	Physical model	Physical representation
P7	Empirical quality	Physical representation	Physical representation
K1	Perceived model-domain appropriateness	Domain knowledge	Model knowledge
K2	Perceived ontological quality	Model knowledge	Language knowledge
K3	Perceived syntactic quality	Language knowledge	Representation knowledge
K4	Perceived semantic quality	Domain knowledge	Representation knowledge
K5	Perceived language-domain appropriateness	Domain knowledge	Language knowledge
K6	Perceived intensional quality	Model knowledge	Representation knowledge
K7	Perceived empirical quality	Representation knowledge	Representation knowledge
L1	View quality	Physical domain	Domain knowledge
L2	Pedagogical quality	Physical model	Model knowledge
L3	Linguistic quality	Physical language	Language knowledge
L4	Pragmatic quality	Physical representation	Representation knowledge
D1	Applied domain—model appropriateness	Domain knowledge	Physical model
D2	Applied domain—language appropriateness	Domain knowledge	Physical language
D3	Applied domain knowledge quality	Domain knowledge	Physical representation
D4	Applied model—language appropriateness	Model knowledge	Physical language
D5	Applied model knowledge quality	Model knowledge	Physical representation
D6	Applied language knowledge quality	Language knowledge	Physical representation

knowledge layer elements. The physical layer plays a central role in the conceptual modeling process as the process transforms concepts from the physical domain to the final physical representation. The physical elements, the artifacts, are learned, analyzed, and manipulated by the modelers as they try to understand the domain. The Physical Layer has seven quality types, described below.

P1: *Model-Domain Appropriateness*. The physical model must be appropriate to the domain being modeled and for the ultimate use of the physical representation. For example, the Entity-Relationship Model is appropriate for conceptualizing a data-oriented domain but is not appropriate for a more process-oriented domain where a workflow model (Georgakopoulos et al. 1995) would be used

P2: *Ontological Quality*. The physical, external language (the grammar and the vocabulary of the language) must be appropriate for expressing the concepts of the physical model and for ultimately encoding the concepts in the physical representation (St. Amant et al. 2006)

P3: *Syntactic Quality*. All of the elements in the final physical representation must be able to be derived from the vocabulary and the grammar of the physical language (Lindland et al. 1994)

P4: *Semantic Quality*. The final representation must accurately and completely capture the meaning of the physical domain, within the constraints of the modeling task at hand (Moody and Shanks 2003)

P5: *Language-Domain Appropriateness*. The language must be powerful enough to parsimoniously express anything in the physical domain (Krogstie 2003; Wand and Weber 1993). The language must allow users to create a faithful representation of the physical domain with a minimum of resources (Wand and Weber 2002)

P6: *Intensional Quality*. The physical representation should remain true to the mindset and the meanings defined by the physical model. For example, the ontological foundations for the use of relationships, and what a relationship really *is*, are found in the physical model (Wand et al. 1999)

P7: *Empirical Quality*. This is a measure of the readability of a conceptual representation (Krogstie et al. 2006). The physical representation is both object of interest and quality reference. The empirical quality of physical representation (e.g., a UML class diagram) X is determined by comparing the representation to some "ideal" physical representation Y. In this case, "ideal" means that the two representations are informationally equivalent, but the reference representation has optimal readability. For example, X and Y are representations of the same UML class diagram, without line crossings (Batini et al. 1991) and with symbols of the same size)

4.2 The knowledge layer

The knowledge layer of quality types parallels the physical layer of quality types. However, where the physical layer exists in the "real world," the knowledge layer exists only cognitively, in the minds of the stakeholders involved in the conceptual modeling process and in the process of using the final representation. Knowledge layer quality types refer to the quality of the model, language, and representation, but not to the quality of the knowledge itself (for which there are the learning layer quality types). Whereas the physical layer quality types of model, language, and representation quality are defined objectively, i.e., independent from the stakeholder involved in evaluating quality, the corresponding knowledge layer quality types recognize a "subjective" notion of quality, i.e., quality as perceived by the user or modeler. When an objective measurement of some physical layer quality type is not possible, the corresponding knowledge layer quality type may be assessed as an approximation.

A difficulty in exploring the various quality types in the knowledge layer lies in the many possible sets of knowledge that may exist based on the perception of the stakeholders. The learning layer below shows how (for example) the real-world physical domain is transformed into cognitive domain knowledge through an stakeholder's perceived view of the domain. Each stakeholder may have a different view based on variations in context, experience, view, and so on. Each knowledge component may represent a single stakeholder's knowledge or the collective knowledge of a set of stakeholders. Any examination of the qualities of the knowledge layer must explicitly note the source of the knowledge.

The knowledge layer parallels the physical layer described above, so the individual quality types will not be described in detail. However, one example can serve to illustrate the layer:

The Perceived Semantic Quality (*K4*) of a physical representation is the correspondence between the stakeholder's knowledge about the domain as derived from the real-world domain itself and knowledge about the domain as derived from the real-world representation. For instance, a user may evaluate the validity and completeness of a conceptual representation (e.g., an Entity-Relationship diagram that acts as a conceptual data model for some domain within the enterprise, such as customer relationship management) by comparing the subjective understanding of the representation (i.e., what is believed to be represented) to the knowledge of the domain (i.e., what is believed to be in the domain, so should be represented). This type of quality has been identified as a surrogate for semantic quality (Krogstie et al. 2006).

4.3 The learning layer

Each element of the physical layer cornerstones has a corresponding element in the knowledge layer cornerstones. All stakeholders using the knowledge cornerstones need to learn them from the physical cornerstones. The learning quality layer measures how well that learning, interpretation, and/or understanding takes place.

A common element to all of the learning quality types is perception. The stakeholder's first exposure to anything in the physical world begins with his/her perception of that world's artifacts. Context drives the stakeholder's perceptions. It determines which things are meaningful and which things will fade into the environment. Our only contact with the real world is through our perceptions that filter and shape the way we see the things of the world (Borgida et al. 1985; Wand and Weber 1988). Ideally, our perception of the world is perfect. In practice, this is rarely the case.

Each of the quality types in the learning layer has, to a greater or lesser extent, a grounding in perception. What is perceived, and how it is learned, depends on the quality cornerstone. The quality types of the learning layer are described more fully below.

- *L1:* *View Quality*. The stakeholder must have a complete and valid understanding of the real-world domain as it relates to the problem at hand. Whether the real-world physical domain exists or not, or the domain knowledge is a product of perception or is a social construction (Wand and Weber 2002), the quality of an stakeholder's domain knowledge is based on the quality of his/her learning
- *L2:* *Pedagogical Quality*. The stakeholder must have the proper mindset as defined by the physical model's paradigm (Kuhn 1970)
- *L3:* *Linguistic Quality*. The stakeholder must have a mastery of the modeling language (the vocabulary and the grammar). Linguistic quality is of primary concern to modelers, but also users of conceptual representation need to master the basics of the employed modeling language in order to understand the conceptual representation
- *L4:* *Pragmatic Quality*. This addresses the comprehension of the final physical representation by the stakeholders: analysts who must use the representation to create the information system and other users who must understand what the representation is modeling. Pragmatic quality captures the extent to which the stakeholder completely and accurately understand the statements in the representation that are relevant to them (Bolloju and Leung 2006)

4.4 The development layer

The elements of the physical layer have their developmental roots in the knowledge layer. A developer's knowledge is used to create the subsequent physical artifacts. For example, a domain expert's knowledge is used to create the physical model, the physical language, and the physical representation. A modeler's knowledge can be used to develop the language and the representation. The development layer's quality types measure how well this knowledge has been used to create the physical elements.

For example, the Entity-Relationship Model did not spring into being from nothing, but rather from a deep understanding of the domain, the semantics of the domain to be modeled, and the needs of the analysts for understanding the real world (Chen 1976). In information systems development, whether the end result is a conceptual representation such as a data flow diagram or ER diagram or a complete information system, a thorough understanding of the domain, the model, and the language is critical. The accurate and complete transfer of that knowledge (from Domain Knowledge to the creation of the Physical Model, from Model Knowledge to the creation of the Physical language, and so on) directly affects the quality of the final Physical Representation.

D1: *Applied Domain—Model Appropriateness*. "The meaning of modeling constructs should be found in models of reality" (Wand et al. 1999, p. 494). The physical model (ontology, mindset) being developed must be appropriate to the modeler's conceptual understanding of the domain. This quality type is especially relevant to, for example, the development of domain ontologies

D2: *Applied Domain—Language Appropriateness*. Any modeling (or even programming) language being developed must be appropriate to the modeler's knowledge of the real-world domain

D3: *Applied Domain Knowledge Quality*. Knowledge of the domain is fundamental to all disciplines. The lack of domain knowledge and the lack of its correct application results in "inelegant problem-solving strategies" (Khatri et al. 2006; Shaft and Vessey 1995). The quality of the final representation is directly dependent on the accurate communication and application of that knowledge (Evermann 2005)

D4: *Applied Model—Language Appropriateness*. The modeling language being developed must be appropriate to the developer's knowledge of the particular mindset or ontology it will be based upon

D5: *Applied Model Knowledge Quality*. Knowledge of the model that underlies the language and the domain is important to the quality of the final representation. For example, an incomplete or incorrect understanding of the Object-Oriented Model (misinterpreting object-oriented concepts as procedural programming concepts) can lead to a hybrid of object and procedural code in the final representation (Nelson et al. 2001)

D6: *Applied Language Knowledge Quality*. The modeler uses the modeling language, the vocabulary and the grammar, to create the physical representation. While the modeler's knowledge of the language may be incomplete (he or she may not be familiar with some of the language's constructs), accurate knowledge of the language and its application are critical to the quality of the physical representation (Gemino and Wand 2003)

5 Example: quality of a conceptual representation

In this section, we present an example of the use of the Conceptual Modeling Quality Framework. The example illustrates how the CMQF may be used by practitioners to examine the quality of a physical representation. A physical representation is the last of a series of transformations from the physical domain through the physical model and physical language, using domain knowledge, model knowledge, and language knowledge. The representation is a primary tool for documenting and understanding the areas of interest in the domain, and its quality will impact the effectiveness and efficiency of the use that is made of the representation. For instance, a conceptual data model with low empirical quality (i.e., low pragmatic quality) can easily be misunderstood by database designers and thus lead to poor database designs. Here, we illustrate the utility of the CMQF quality dimensions by highlighting the quality dimensions that directly affect the conceptual representation and different techniques available for ensuring and testing conceptual modeling quality, concentrating on the quality of conceptual models developed using the Unified Modeling Language (UML) introduced by Jacobson et al. (1999). Unfortunately, it is not currently possible to directly measure the quality of any of the framework cornerstones. “While the set algebra definitions of various quality goals provide some clarity as to what the different quality levels mean, it is really only syntactic quality (of the levels in the original framework), which has any hope of being objectively measured, as both the problem domain and the minds of the stakeholders are unavailable for formal inspection. This would, however, be a general challenge for any framework trying to address model evaluation beyond the syntactic level” (Krogstie et al. 2006, p. 94).

UML has gone through considerable development and many modifications since its introduction in 1995, leading to the present version: UML V2.1.2 (OMG 2007a, b). UML is used for the illustration as it is frequently used as a modeling language for conceptual representations and has become the focus of a great deal of conceptual modeling quality research. The Physical Representation quality dimensions of interest are shown below in Fig. 8. Other quality dimensions from the knowledge layer, in particular K3, K4, K6, and K7, are also relevant to the physical representation, but entail an indirect assessment of the physical qualities, via their knowledge counterparts. They are not used in the illustration below.

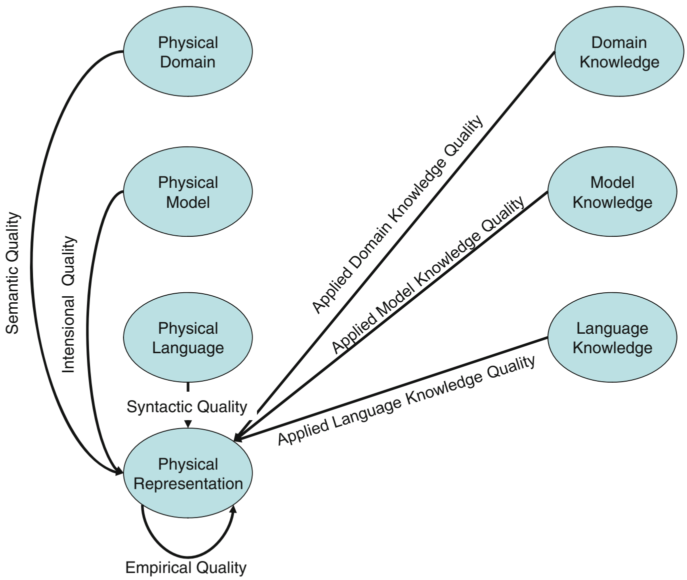

Fig. 8 The quality dimensions leading to the physical representation

5.1 Syntactic quality (P3)

All of the elements in the final physical representation must be able to be derived from the vocabulary and the grammar of the physical language. The syntactic quality (P3) of UML diagrams can be improved through the use of controls (Unkelkar 2005). For example, UML class diagram controls include verifying the elements allowed for a class diagram, the annotations for each class, the annotations on all of the relations, the association of all of the relations, that the line of each of the association relations has one (open) arrow, and that the cardinality constraints are indicated. The application of Object-Oriented Reading Techniques (Conradi et al. 2003) can help inspectors detect defects in UML diagrams, improving defect detection rates to between 68% and 98% in UML diagrams. Also automated techniques for identifying the syntactic correctness of UML diagrams may be used (van Amstel et al. 2007).

Many UML modeling tools will flag syntactic errors; sometimes the diagram can simply not be saved when it is not syntactically correct. Figure 9 shows an example of a UML class diagram built using the Objecteering 6.1 tool (http://www.objecteering.com). The diagram is intended as a conceptual representation of a real-world situation in which the factories that a company owns each employ one or more laborers, who are employees of the company. The Objecteering tool flags two syntactic errors: (1) the modeler has apparently switched the minimum and maximum multiplicity constraints on the factory worker role (i.e., minimum many, maximum one); (2) there is a cycle in the inheritance graph (i.e., a factory laborer is a back-office worker, who is an employee, who is again a factory laborer).

Figure 10 shows another example, this time a UML activity diagram that is a conceptual representation of a simple order-to-cash business process. The diagram does not satisfy the grammatical rules of UML as the decision node after the receive order action has only one outgoing control flow arrow, so the representation does not specify what action to perform if the guard condition [order accepted] is not fulfilled.

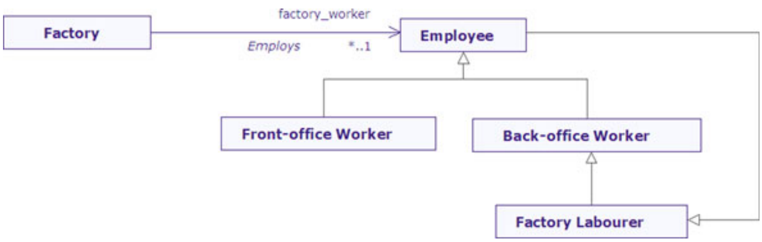

Fig. 9 Conceptual representation (as a UML class diagram) of factories employing laborers that are company employees, with syntactic errors

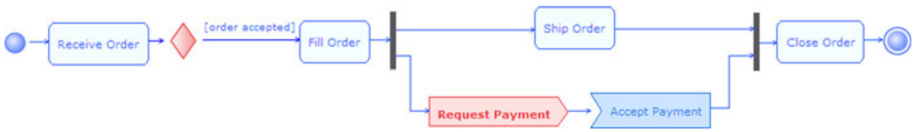

Fig. 10 Conceptual representation (as a UML activity diagram) of an order-to-cash business process, with a syntactic error

5.2 Semantic quality (P4)

The final representation must accurately and completely capture the meaning of the physical domain, within the constraints of the modeling task at hand. There has been considerable research that focuses on validating the semantic quality (P4) of UML diagrams versus the application domain, particularly against requirements documents. One method is to apply formal semantics to the many UML diagrams (Deng et al. 2005) to arrive at a "common denominator" so that consistency between the diagrams and the requirements documents may be achieved. Formal semantics may also be employed to verify the behavioral semantic quality of UML statechart diagrams (Kong et al. 2009).

In general, knowledge of the physical domain is required to evaluate semantic quality. Some physical domains are generally well enough understood to identify incorrectness or incompleteness in conceptual representations. For instance, Fig. 11 shows a UML class diagram intended as a conceptual representation of part of a library domain. There are two clear cases of incorrectness in the diagram: (1) a library's books must be on loan all the time (i.e., minimum multiplicity of one of the borrower role); and (2) a book can be borrowed by more than one library member at the same time (i.e., maximum multiplicity of many of the borrower role). If it is the intent to further elaborate the diagram as a conceptual data model for developing a computerized library system, then the alternative diagram shown in Fig. 12 provides a better starting point. The diagram shown in Fig. 12 has higher semantic quality as it represents the library domain constraint that books can be borrowed by at most one library member at a time, while showing that over time they can be borrowed many times by the same or different library members.

5.3 Intensional quality (P6)

The physical representation should remain true to the mindset and the meanings defined by the physical model. The physical model sets out the rules, the paradigm or "mindset," that defines how the world is to be viewed. For example, the Object-Oriented Model views the world as composed of objects, their attributes, and methods. It is possible to use a modeling

Fig. 11 Conceptual representation (as a UML class diagram) of books on loan with library members that is semantically incorrect

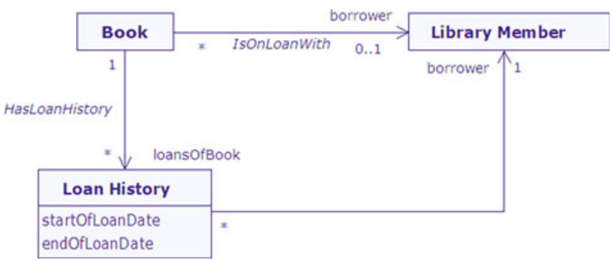

Fig. 12 Alternative conceptual representation of books currently on loan with library members and loan histories

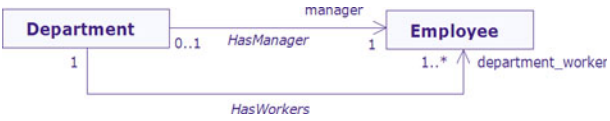

Fig. 13 Conceptual representation (as a UML class diagram) in which being the manager of a department is an optional property of employees

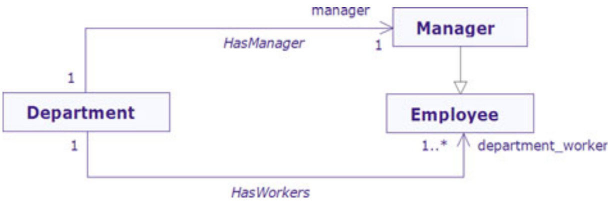

Fig. 14 Informationally equivalent representation without optional properties

language that was developed to support a particular mindset (such as UML supporting OO) in a manner inconsistent with that mindset. For example, modeling guidelines based on mappings between UML and Bunge's (1977) ontology may be used (Evermann and Wand 2006) to keep the representation within Bunge's ontology. One of these modeling guidelines prohibits the use of optional properties (i.e., relationships or attributes) in conceptual representations, the reason being that if some property is optional for a particular instance, then that instance represents a thing belonging to a different functional schema than the things that are represented by the instances for which the property is mandatory (Gemino and Wand 2005). A UML representation that remains true to the mindset defined by the Bunge ontology would distinguish these functional schemas by using one class for the instances that posses the property and another class for the instances that do not, where the former class is a subclass of the latter. For instance, Fig. 13 shows that, although every employee works in a department, not every employee needs to manage a department. This representation is not consistent with the physical model provided by Bunge's ontology as the property of being the manager of a department is optional for employees. A physical representation with higher intensional quality (assuming an ontological commitment to Bunge's ontology) is given in Fig. 14, which shows that managers are the subset of employees that manage departments. To manage a department is only a property of managers, not of employees in general.

5.4 Empirical quality (P7)

This is a measure of the readability of a conceptual representation and is a function of the representation itself. Generally speaking, a physical representation that exhibits high empirical quality (P7) will be more maintainable and more understandable than a similar representation that has low empirical quality. Esthetics metrics (Kiewkanya and Muenchaisri 2005) have been developed for use on UML class diagrams and sequence diagrams. These metrics are a good indicator of the maintainability of such diagrams. Readability guidelines (Ambler 2005) and layout recommendations (Purchase et al. 2002) improve the overall quality of the UML diagrams themselves.

While Fig. 15 is a syntactically correct UML activity diagram representing the same order-to-cash process as Fig. 10 (and improves the syntactic quality (P3) of Fig. 10 as now

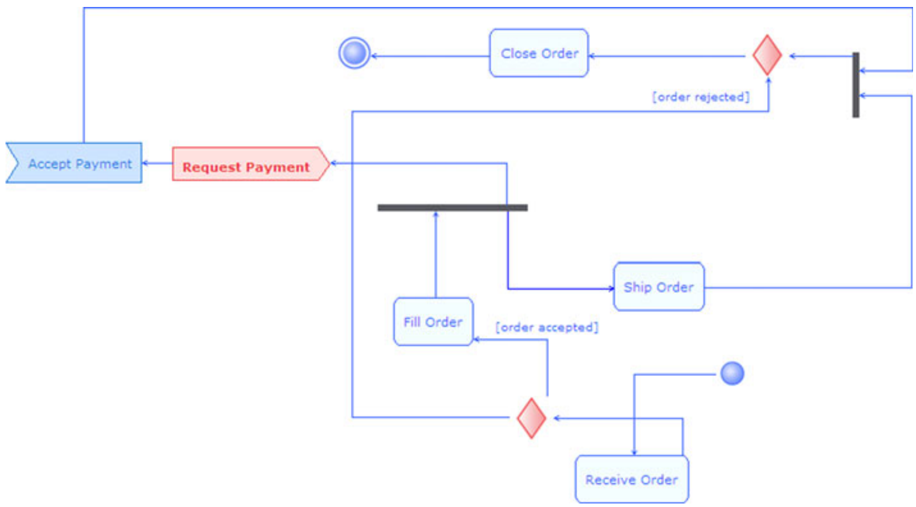

Fig. 15 Conceptual representation (as a UML activity diagram) of an order-to-cash process with low empirical quality

also the complement of the [order accepted] guard is included), it is of lower empirical quality (P7) because the consistent left-to-right control flow ordering of Fig. 10 is abandoned. The readability of the diagram is lower because the reader has to move in different directions when tracing the control flow of the represented business process. Furthermore, the shapes of the send signal action (i.e., request payment) and accept change event action (i.e., accept payment) symbols suggest a natural left-to-right ordering (as both shapes are based on arrows pointing to the right and the send signal action arrow fits into the accept change event action arrow), whereas here an opposite and unnatural right-to-left ordering was used. Empirical quality would be further reduced by replacing the verb labels of the actions with noun labels (e.g., order received instead of receive order).

5.5 Applied domain knowledge quality (D3)

The quality of the final representation is directly dependent on the accurate communication and application of domain knowledge. Khatri et al. (2006) demonstrate the effect of application domain knowledge and conceptual modeling knowledge (IS domain knowledge) on different types of schema understanding tasks: syntactic and semantic comprehension tasks and schema-based problem-solving tasks. Their research provides guidelines for organizations that have relatively scarce resources in a particular application domain. Gerard (2005) showed that domain-specific knowledge structures support the conceptual modeling task. Other researchers have shown that different application domains may share common knowledge structures, and that the reuse of models (capturing domain knowledge) from one domain to another is feasible and beneficial (Maiden and Sutcliffe 1992). For instance, Snoeck and Poels (2000) demonstrate this principle of analogical reuse by presenting a conceptual model of product usage (also called object allocation (Lung and Urban 1995)) that was derived from generalizing domain-specific models for libraries and hotels, and that can be applied to domains like the car rental business or flight reservations.

Through analogical reuse, modelers apply domain knowledge contributed by many domain experts that is "stored" in generic domain models or patterns.

Despite these research efforts, most conceptual modeling texts do not emphasize domain-specific patterns nor analogical reuse, and conceptual modeling notations are considered as general tools that must be mastered independently of the application domain. Poels et al. (2007) and Poels (2009) showed that UML class diagrams that are used as conceptual representations of enterprise transaction cycles (e.g., order-to-cash, procure-to-pay, produce-to-stock, etc.) are better understood by business users, both objectively (in terms of performance in model comprehension tasks) and subjectively (in terms of perceived ease of understanding), if these diagrams are consistently constructed using resource-event-agent patterns.

Figures 16 and 17 illustrate these effects of domain pattern use on pragmatic quality (L4), empirical quality (P7), and perceived empirical quality (K7). Figure 16 is a conceptual representation of an instance of the acquisition/payment transaction cycle that involves the ordering, obtaining, and paying for external consulting services. The analyst that constructed the diagram applied his knowledge of enterprise transaction cycle structures by employing three left-to-right ordered resource-event-agent constellations of classes and associations (i.e., Consulting Service Type—Order Consulting Services—Supervisor/Consulting Firm, Consulting Services—Get Consulting Services—Clerk/Consulting Firm, and Account—Pay Consulting Services—Cashier/Consulting Firm). Figure 17 is a syntactically correct UML class diagram that is informationally equivalent to Fig. 16 (so does not differ from Fig. 16 with respect to semantic quality (P4)), but lacks clear signs of applied domain knowledge in the sense that the common resource-event-agent structures that build up enterprise transaction cycles are not recognizable in the diagram. Comparing both diagrams, one can visually appreciate the difference in readability between both diagrams (i.e., higher empirical quality (K7) of Fig. 16 due to higher

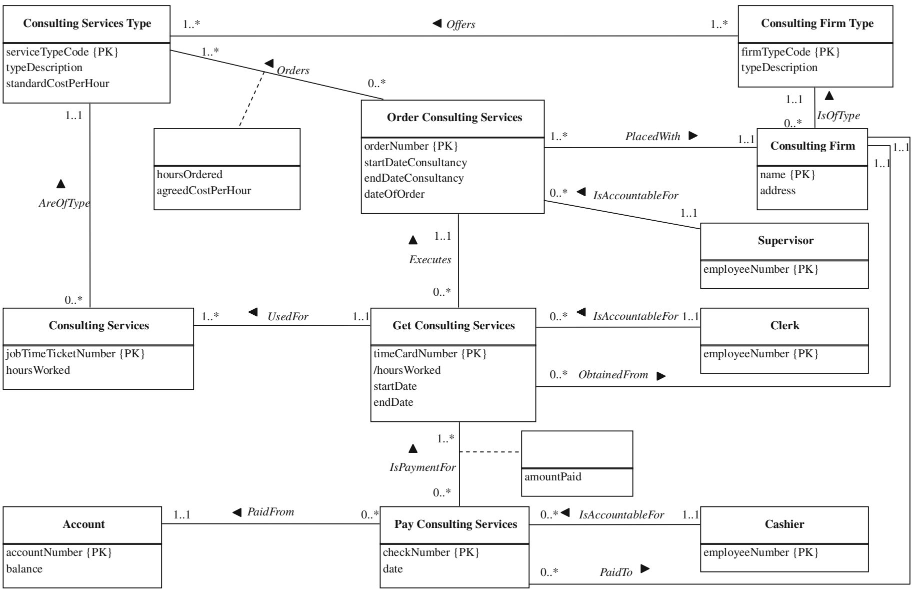

Fig. 16 Transaction cycle representation (as a UML class diagram) constructed using a domain-specific pattern (resource-event-agent conceptual modeling structures) (based on (Poels 2009))

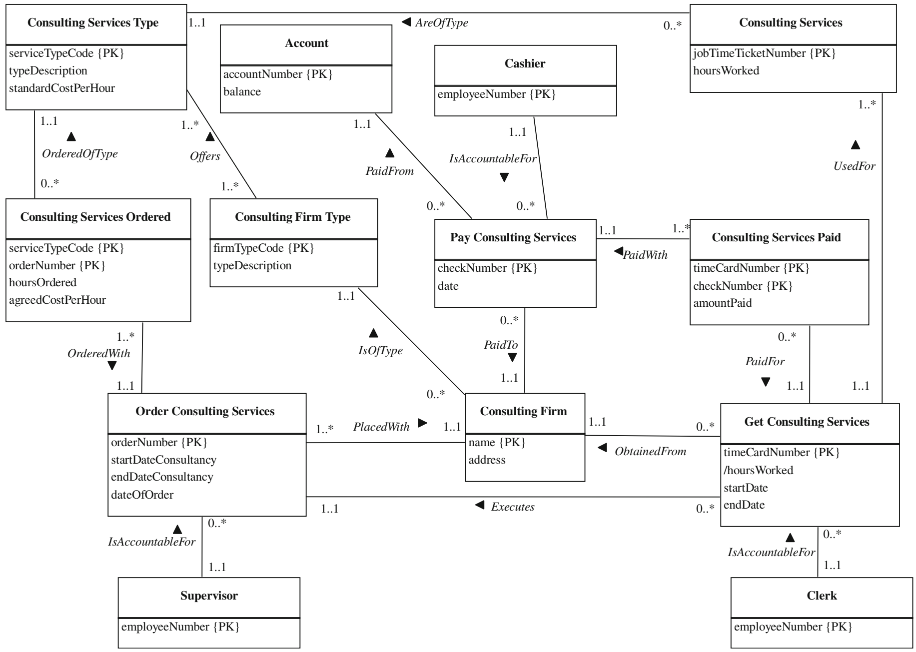

Fig. 17 Informationally equivalent transaction cycle representation (as a UML class diagram) without resource-event-agent structures (based on (Poels 2009))

applied domain knowledge quality (D3)). Furthermore, as demonstrated in (Poels, et al., 2007), business users like accountants and auditors get a far better understanding of the transaction cycle modeled if a diagram like Fig. 16 is used (i.e., higher pragmatic quality (P7)). Moreover, they perceive diagrams with clear traces of applied domain knowledge as easier to interpret (i.e., higher perceived empirical quality (K7)) (Poels 2009).

5.6 Applied model knowledge quality (D5)

Knowledge of the model, the paradigm or "mindset," that underlies the language and the domain is important to the quality of the final representation. Physical representations are created with analysts looking through the "lens" of the model. That mindset or paradigm guides the analyst in focusing on the various important parts of the domain: for example, the objects in the Object-Oriented Model or the activities in the Procedural Model. Training programs must be adapted to place differential emphasis on various ontological concepts (Armstrong and Hardgrave 2007) to assist software developers who have difficulty applying the appropriate ontology in their software development efforts. Where confusion arises, the trainer can explicitly describe the differences in the application of similar ontological concepts.

Suppose that there is a need for a conceptual representation that focuses on the control flow aspects (e.g., sequencing of actions, alternative paths of execution, parallel paths of execution, synchronization points, etc.) of the order-to-cash process that was also the subject of Figs. 10 and 15. This conceptual representation could for instance serve as the basis for designing an executable workflow model that is used by a workflow engine to

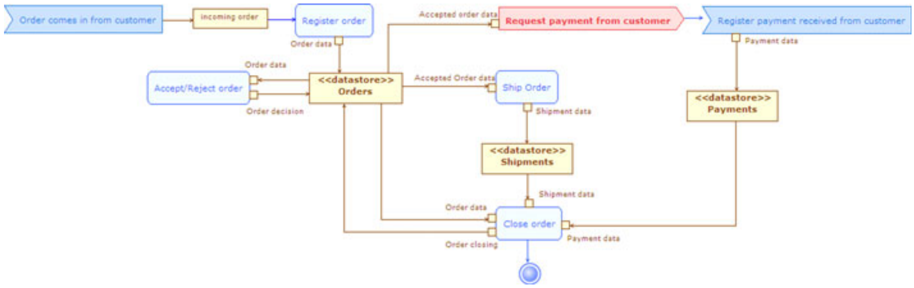

Fig. 18 Conceptual representation (as a UML activity diagram) of an order-to-cash process intended as a workflow model, low applied model knowledge quality

enact the order-to-cash process. Assume further that there is a versatile modeler trained in various aspects of process modeling and the use of UML activity diagrams for business process modeling. Assume also that this modeler has thoroughly analyzed the order-to-cash process. Figure 18 is the modeling result obtained by this modeler. The diagram is a syntactically correct (P3) and semantically valid (P4) representation of the order-to-cash process. The empirical quality (P7) is reasonable given the left-to-right, top-down ordering of the flows (it is surely higher than the empirical quality of Fig. 15). Both the applied domain knowledge quality (D4) and applied language knowledge quality (D6) seem to be fine. The problem is, however, with applied model knowledge quality (D5) as the diagram clearly evidences a data flow/processing mindset instead of a workflow/control flow mindset. So, the modeler has applied his data flow modeling knowledge to solve the problem rather than his control flow modeling knowledge. As a result, a conceptual representation is obtained that may provide a good basis for developing a database system supporting the order-to-cash process, but is less suitable for developing a workflow system because the control flow aspects of the order-to-cash process are not emphasized and only implicitly present in the diagram.

5.7 Applied language knowledge quality (D6)

Accurate knowledge of the language and the ability to apply it correctly are critical to the quality of the physical representation. Even though the analyst may have expert knowledge of the modeling language and the physical representation may be syntactically correct, typographical errors, inattention to detail or other errors in its application can affect the quality of the final representation (Bolloju and Leung 2006). Knowledge of the modeling language is critical for creating high-quality conceptual representations (Lange et al. 2006). Figures 19 and 20 are UML class diagrams intended as conceptual representations of marriages between men and women. Let us assume that the reality represented pertains to a society, civilization, or legal system in which a marriage between one man and one woman is the only possible or allowable configuration. Let us further assume that if the representation is going to be used as a conceptual data model (e.g., for automating the population register of a city), then the date of the wedding is a relevant property of marriage to be included in the representation. Let us also simplify the data requirements such that there is no need for keeping historical information on marriages (i.e., who was married to whom in the past). The diagram in Fig. 19 is produced by a modeler that correctly understood marriage as a mutual property of man and woman and therefore chose to represent the date

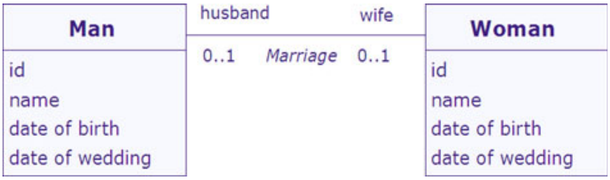

Fig. 19 Conceptual representation of marriage with date of wedding as an attribute of the man and woman classes

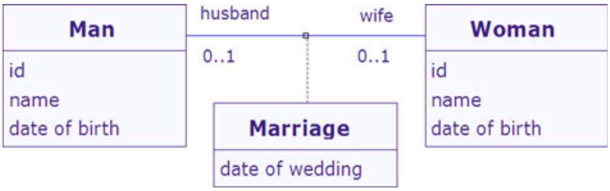

Fig. 20 Conceptual representation of marriage with date of wedding as an attribute of the marriage association class

of wedding as a property of both man and woman. The diagram in Fig. 20 is produced by a modeler that applied his knowledge of UML to represent the date of wedding as a property of the marriage by means of an association class. Both diagrams are syntactically and semantically correct (P3 and P4) and probably have the same empirical quality (P7) and, from an object orientation point of view, less than optimal applied model knowledge quality (D5) (as man and woman may be generalized to person such that person properties can be inherited). Their intensional quality (P6) may differ depending on the chosen mindset, e.g., in the Bunge ontology, properties (e.g., marriage) cannot have properties (e.g., date of wedding). The solution in Fig. 20 has, however, higher applied language knowledge quality (D6) as the modeler has chosen a specific and more appropriate language construct to represent the date of the wedding. The representation in Fig. 20 suffers from UML language construct overload because attributes are used to represent both intrinsic properties (e.g., id, name, date of birth) and mutual properties (i.e., date of wedding) (Wand and Weber 1993, 1995), which results in lower ontological clarity, and possibly a negative impact on pragmatic quality (L4).

6 Summary and conclusions

High-quality representations are critical for producing high-quality information systems. They are vital for understanding the user's domain and system requirements when developing new information systems and are a knowledge repository for future analysts when the information system must be modified, extended, or repaired. High-quality representations make it easier for analysts to discover and correct errors earlier in the development process, and errors are far less expensive to correct early rather than later when the information system development process has moved into the coding, implementation, and maintenance stages.

However, the use of a conceptual representation comes at the end of a long chain of development, from the domain and its understanding, to the development of an ontological model that codifies a view of the domain, to the language containing the grammar and vocabulary for creating the representation, and finally to the representation itself. This chain must also be high quality, for any errors or ambiguities in the model or language will eventually appear in the representation.

Two frameworks, Lindland, Sindre, and Sølvberg (LSS) and Bunge–Wand–Weber (BWW), have been developed that attempt to clarify and focus attention on various parts of the conceptual representation development chain. Each has been fairly successful in describing the various interrelationships between conceptual modeling elements. However, each has missing elements and relationships that make a complete understanding difficult. This paper describes a framework that combines the LSS and the BWW frameworks into a single comprehensive Conceptual Modeling Quality Framework (CMQF). Our framework contains eight quality cornerstones and four “layers” containing twenty-four quality types defined as relationships between pairs of cornerstones.

Future research will examine each of the quality dimensions and quality cornerstones individually and in combination to help improve the state of the art of conceptual modeling. A significant area of future research includes the development of specific metrics on each of the quality dimensions. These metrics will be useful to both researchers and to practitioners to assure the quality of conceptual representations and the development of sound models and languages.

Acknowledgments This research has been funded by the following projects: MEDUSAS (CDTI-MICINN and FEDER IDI-20090557), ORIGIN (CDTI-MICINN and FEDER IDI-2010043(1-5)), PEGASO/MAGO (MICINN and FEDER, TIN2009-13718-C02-01), EECCOO (MICINN TRA2009_0074) and MECCA (JCMM PII2I09-0075-8394).

References

Ambler, S. (2005). *Elements of UML 2.0 style*. Cambridge: Cambridge University Press.

Armstrong, D. J., & Hardgrave, B. C. (2007). Understanding mindshift learning: The transition to object-oriented development. *MIS Quarterly, 31*(3), 453–474.

Batini, C., Ceri, S., & Navathe, S. B. (1991). *Conceptual database design: An entity-relationship approach*. Redwood City: Benjamin/Cummings.

Bolloju, N., & Leung, F. (2006). Assisting novice analysts in developing quality conceptual models with UML. *Communications of the ACM, 49*(7), 108–112.

Borgida, A., Greenspan, S., & Mylopoulos, J. (1985). Knowledge representation as the basis for requirements specifications. *IEEE Computer, 15*(4), 82–90.

Bunge, M. A. (1977). *Ontology I: The furniture of the world* (Vol. 3). Dordrecht: Reidel.

Chen, P. P. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems, 1*(1), 9–36.

Conradi, R., Mohagheghi, P., Arif, T., Hedge, L. C., Bundle, G. A., & Pedersen, A. (2003). *Object-oriented reading techniques for inspection of UML models—an industrial experiment.* Paper presented at the European conference on object-oriented programming (ECOOP'03), Darmstadt, Germany.

Deng, M., Stirewalt, R. E. K., & Cheng, B. H. C. (2005). Retrieval by construction: A traceability technique to support verification and validation of UML formalizations. *International Journal of Software Engineering, 15*(5), 837–872.

Evermann, J. (2005). Towards a cognitive foundation for knowledge representation. *Information Systems Journal, 15*, 147–178.

Evermann, J., & Wand, Y. (2006). Ontological modeling rules for UML: An empirical assessment. *Journal of Computer Information Systems, 47*, 14–29.

Gemino, A., & Wand, Y. (2003). Evaluating modeling techniques based on models of learning. *Communications of the ACM, 46*(10), 79–84.

Gemino, A., & Wand, Y. (2005). Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties. *Data & Knowledge Engineering, 55*(3), 301–326.
Georgakopoulos, D., Hornick, M., & Sheth, A. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases, 3*, 119–153.
Gerard, G. J. (2005). The REA pattern, knowledge structures, and conceptual modeling performance. *Journal of Information Systems, 19*(2), 57–77.
Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The unified software development process*. Reading: Addison-Wesley.
Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness*. Cambridge, MA: Harvard University Press.
Khatri, V., Vessey, I., Ramesh, V., Clay, P., & Park, S.-J. (2006). Understanding conceptual schemas: Exploring the role of application and is domain knowledge. *Information Systems Research, 17*(1), 81–99.
Kiewkanya, M., & Muenchaisri, P. (2005). *Measuring maintainability in early phase using aesthetic metrics.* Paper presented at the 4th international conference on software engineering, parallel and distributed systems (SEPADS 2005), Salzburg, Austria.
Kong, J., Zhang, K., Dong, J., & Xu, D. (2009). Specifying behavioral semantics of UML diagrams through graph transformations. *Journal of Systems and Software, 82*(2), 292–306.
Krogstie, J. (2003). Evaluating UML using a generic quality framework. In L. Favre (Ed.), *UML and the unified process*. Hershey, PA: Idea Group Publishing.
Krogstie, J., Sindre, G., & Jørgensen, H. (2006). Process models representing knowledge for action: A revised quality framework. *European Journal of Information Systems, 15*, 91–102.
Krogstie, J., & Sølvberg, A. (2003). *Information systems engineering: Conceptual modeling in a quality perspective*. Trondheim, Norway: Kompendiumforlaget.
Kuhn, T. S. (1970). *The structure of scientific revolutions*. Chicago: The University of Chicago Press.
Lange, C. F. J., Chaudron, M. R. V., & Muskens, J. (2006). UML software architecture and design description. *IEEE Software, 23*(2), 40–46.
Lindland, O. I., Sindre, G., & Sølvberg, A. (1994). Understanding quality in conceptual modeling. *IEEE Software, 11*(2), 42–49.
Lung, C. H., & Urban, J. E. (1995). *An approach to the classification of domain models in support of analogical reuse.* Paper presented at the SIGSOFT symposium on software reusability, Seattle.
Maiden, N. A. M., & Sutcliffe, A. G. (1992). Exploiting reusable specifications through analogy. *Communications of the ACM, 35*(55–64).
Moody, D., & Shanks, G. (2003). Improving the quality of data models: Empirical validation of a quality management framework. *Information Systems, 28*, 619–650.
Moody, D., Shanks, G. G., & Darke, P. (1998). *Improving the quality of entity relationship models—experience in research and practice.* Paper presented at the 17th international conference on conceptual modeling, Singapore.
Moody, D., Sindre, G., Brasethvik, T., & Sølvberg, A. (2003). *Evaluating the quality of information models: Empirical testing of a conceptual model quality framework.* Paper presented at the International Conference on Software Engineering, Portland, OR.
Nelson, H. J., Monarchi, D. E., & Nelson, K. M. (2001). *Ensuring the "goodness" of a conceptual representation.* Paper presented at the European Conference on Software Measurement and Control (FESMA-DASMA), Heidelberg, Germany.
OMG. (2007a). *OMG unified modeling language (OMG UML) infrastructure V2.1.2*. Object Management Group.
OMG. (2007b). *OMG unified modeling language (OMG UML) superstructure V2.1.2*. Object Management Group.
Poels, G. (2009). *Understanding business domain models: The effect of reorganizing resource-event-agent conceptual modeling structures*. Unpublished FEB Working Paper. Ghent University.
Poels, G., Maes, A., Gailly, F., & Paemeleire, R. (2007). The pragmatic quality of resources-events-agents diagrams: an experimental evaluation. *Information Systems Journal*. doi:10.1111/j.1365-2575.2007.00253.x.
Purchase, H., Allder, J., & Carrington, D. (2002). Graph layout aesthetics in UML diagrams. *Journal of Graph Algorithms and Applications, 6*(3), 255–279.
Recker, J., Rosemann, M., & Krogstie, J. (2007). Ontology- versus pattern-based evaluation of process modeling languages: A comparison. *Communications of the AIS, 20*, 774–799.
Russell, B. (1903). *The principles of mathematics*. New York: W. W. Norton and Company.
Searle, J. R. (1995). *The construction of social reality*. New York: The Free Press.

Shaft, T. M., & Vessey, I. (1995). The relevance of application domain knowledge - the case of computer-program comprehension. *Information Systems Research, 6*(3), 286–299.
Snoeck, M., & Poels, G. (2000). Improving the reuse possibilities of the behavioral aspects of object-oriented domain models. *Lecture Notes in Computer Science (LNCS), 1920*, 423–439.
St. Amant, R., McBride, S., & Ritter, F. (2006). *AI support for building cognitive models.* Paper presented at the twenty first national conference on artificial intelligence (AAAI-06), Menlo Park, CA.
Stamper, R. (1992). *Signs, organisations, norms and information systems.* Paper presented at the 3rd Australian conference on information systems, Wollongong.
Stamper, R., Liu, K., Hafkamp, M., & Ades, Y. (2000). Understanding the roles of signs and norms in organizations—a semiotic approach to information systems design. *Behavior & Information Technology, 19*(1), 15–27.
Unkelkar, B. (2005). *Verification and validation for quality of UML 2.0 models*. Hoboken, NJ: Wiley.
van Amstel, M. F., Lange, C. F. J., & Chaudron, M. R. V. (2007). *Four automated approaches to analyze the quality of UML sequence diagrams.* Paper presented at the 1st IEEE international workshop quality-oriented reuse of software (QUORS'07), Beijing.
Wagelaar, D., & Van Der Straeten, R. (2007). Platform ontology for the model driven architecture. *European Journal of Information Systems, 16*(4), 362–373.
Wand, Y., Storey, V. C., & Weber, R. (1999). An ontological analysis of the relationship construct in conceptual modeling. *ACM Transactions on Database Systems, 24*(4), 494–528.
Wand, Y., & Wang, R. (1996). Anchoring data quality dimensions in ontological foundations. *Communications of the ACM, 39*(11), 86–95.
Wand, Y., & Weber, R. (1988). *An ontological analysis of some fundamental information system concepts.* Paper presented at the international conference on information systems (ICIS), Minneapolis, MN.
Wand, Y., & Weber, R. (1990a). An ontological model of an information system. *IEEE Transactions on Software Engineering, 16*(11), 1282–1292.
Wand, Y., & Weber, R. (1990b). *Towards a theory of the deep structure of information systems.* Paper presented at the international conference on information systems, Copenhagen, Denmark.
Wand, Y., & Weber, R. (1993). On the ontological expressiveness of information systems. *IEEE Transactions on Software Engineering, 16*(11), 217–237.
Wand, Y., & Weber, R. (1995). On the deep structure of information-systems. *Information Systems Journal, 5*(3), 203–223.
Wand, Y., & Weber, R. (2002). Research commentary: Information systems and conceptual modeling—a research agenda. *Information Systems Research, 13*(4), 363–376.

Author Biographies

H. James Nelson is an assistant professor of Management Information Systems at Southern Illinois University, Carbondale. He received his BS in Computer Science from California Polytechnic State University, San Luis Obispo, and his MS and PhD in Information Systems from the University of Colorado, Boulder. His research interests include developing theoretically grounded models of information systems quality, investigating how people make IT paradigm shifts, and determining the business value of information technology.

Geert Poels is head of Ghent University's Information and Knowledge Management research group. He holds Master degrees in business engineering and computer science and a PhD in business administration. His research interests are software metrics, conceptual modeling, ontology, business process management, and service science. His papers were published in IEEE Transactions on Software Engineering, Journal of Systems and Software, Information and Software Technology, Software and Systems Modeling, Data & Knowledge Engineering, Journal of Information Systems, European Journal of Information Systems, and Information Systems Journal.

Marcela Genero is an Associate Professor at the Department of Information Systems and Technologies at the University of Castilla-La Mancha, Ciudad Real, Spain. She received her MSc degree in Computer Science in the Department of Computer Science of the University of South, Argentine in 1989, and her PhD at the University of Castilla-La Mancha, Ciudad Real, Spain in 2002. Her research interests are: empirical software engineering, software metrics, conceptual data models quality, database quality, quality in product lines, quality in MDD, etc. Marcela Genero has published in prestigious journals (Journal of Software Maintenance and Evolution: Research and Practice, L'Objet, Data and Knowledge Engineering, Journal of Object Technology, Journal of Research and Practice in Information Technology, Information and Software technology, Empirical Software Engineering) and conferences (CAISE, E/R, MODELS/UML, ISESE, METRICS, OOIS, SEKE, etc.). She edited with Mario Piattini and Coral Calero the books titled "Data and Information Quality" (Kluwer, 2001) and "Metrics for Software Conceptual Models" (Imperial College, 2005). She participated in several program committees (CAISE, METRICS, ISESE, ESEM, ICSM, etc.). She is member of the International Software Engineering Research Network (ISERN).

Mario Piattini CISA, CISM, CGEIT, and CSQE is a professor in the Department of Computer Science at the University of Castilla-La Mancha, in Ciudad Real, Spain. He has a MSc and PhD in Computer Science from the Politechnical University of Madrid and is a Certified Information System Auditor Manager by ISACA (Information System Audit and Control Association). Author of several books and papers on databases, software engineering, and information systems, he leads the ALARCOS research group of the Department of Computer Science at the University of Castilla-La Mancha, in Ciudad Real, Spain. His research interests are: advanced database design, database quality, software metrics, software maintenance, and security in information systems.