Conceptual modeling using domain ontologies: Improving the domain-specific quality of conceptual schemas

Frederik Gailly  
Faculty of Economics, Social and Political Sciences, and the Solvay Business School.  
Vrije Universiteit Brussel  
Pleinlaan 2, 1050 Brussel  
Belgium  
003226291606  
Frederik.Gailly@vub.ac.be

Geert Poels  
Faculty of Economics and Business Administration  
Ghent University  
Tweekerkenstraat 2, 9000 Gent  
Belgium  
00322649831  
Geert.Poels@Ugent.be

ABSTRACT
In this paper an additional step is taken towards ontology-based conceptual modeling by using ontologies that specify conceptualizations of material domains. Using domain ontologies instead of top-level ontologies for ontology-based conceptual modeling, enables the reuse of domain-specific knowledge. The application of domain ontologies to conceptual schema development instead of conceptual modeling language evaluation, ensures the domain-specific quality of the schemas, i.e. the satisfaction of domain-specific axioms in schemas that intend to represent particular situations within the domain of interest. This approach to ontology-based conceptual modeling is implemented by extending existing conceptual modeling languages with domain-specific language constructs that are defined by domain ontologies. The approach is illustrated for enterprise modeling using UML and the Resource Event Agent (REA) enterprise ontology. The usefulness of the approach for the quality assurance of conceptual schemas is demonstrated by evaluating the domain-specific quality of a sample of UML class diagrams intended as enterprise models.

Keywords
ontology-based conceptual modelling; domain ontology; UML profile; OCL; domain-specific quality

1. Introduction
The concept of ontology-driven information system (ODIS) has been attributed to Guarino [1] who describes the role of an explicit formal ontology in such a system as a central one, driving all aspects and components of the system. ODIS may involve the use of the ontology at run-time (i.e. within the information system) or at development-time (i.e. for developing the information system). Guarino envisions two scenarios for ontology-driven information systems development. In the ‘ideal’, but hard to achieve scenario, system developers have a well-equipped ontology library with task and domain ontologies at their disposal. In the ‘realistic’, but less advantageous scenario, they can only consult a generic, top-level ontology consisting of coarse distinctions among the basic entities of the real world, their kinds and relations [1].

Both scenario’s have been investigated more recently in two different contexts. The ontological evaluation of conceptual modeling languages fits Guarino’s ‘realistic’ scenario. The top-level ontologies that are used for this purpose impart real world semantics [2] on conceptual modeling language constructs such that the conceptual schemas articulated with these constructs faithfully convey the meaning of the represented reality. A top-level ontology specifies a conceptualization of the real world in general, without being restricted to any particular material domain (i.e. part of the real world). Studies that have used top-level ontologies for evaluating conceptual modeling languages will be discussed further on in the paper (see Background section).

Guarino’s ‘ideal’ scenario has been further investigated in domain-specific modeling research (DSM) where it is generally accepted that an existing or newly developed ontology can act as a formal specifications of the domain of the language which afterwards can be transformed in a domain-specific metamodel for the language. The ontologies used in this context are domain and task ontologies which specify, respectively, a conceptualization of a material domain (e.g. a software quality ontology) and a generic task (e.g. a measurement ontology) and which have themselves been created by specializing concepts of a top-level ontology.

In this paper Guarino’s ideal scenario is further investigated by demonstrating how domain ontologies can constrain general-purpose modeling languages such that they can be used as domain-specific modeling languages, thereby attaching domain-specific semantics to modeling language constructs. This ontology-driven modeling approach is realized by extending existing general purpose modeling languages with domain-specific profiles that are defined by domain ontologies.

The domain ontologies specify the invariant conditions of the domain of interest which should be respected by any model built for that domain [3]. Therefore, apart from enabling the reuse of domain knowledge, domain ontologies can play an important role in the quality assurance and evaluation of models. Consequently, the paper defines the notion of domain-specific quality as the degree to which a model respects the invariant conditions of a domain, as axiomatized in a domain ontology. Domain-specific quality is assured if schemas instantiate profiles which include the domain-specific constraints.

This approach to ontology-based conceptual modeling is illustrated for enterprise modeling using UML and the Resource Event Agent (REA) enterprise ontology [4-6]. Hence, the paper presents a UML profile for enterprise modeling that is defined by the REA ontology. The paper also demonstrates an application of the approach by using this profile for evaluating the domain-specific quality of a sample of UML class diagrams intended as enterprise models. By verifying the conformance of the diagrams to the profile, domain-specific quality relative to the conceptualization of the enterprise specified by REA is assessed.
Finally our research is also related to conceptual pattern research. A conceptual modeling pattern describes a reproducible solution to a general conceptual design problem [19]. Over the years different patterns have been proposed [20] which must improve the productivity of the modeler. However according to our knowledge pattern researchers have never investigated in detail how the patterns should be incorporated in general purpose modeling languages.

3. Domain ontology-based conceptual modeling approach

A domain axiomatization which is represented by an ontology can be integrated in a modeling language by constraining its syntax. This can be done via the metamodel that specifies the language. An alternative to changing the metamodel of a language is to extend it via a profile (see Figure 1). The UML profiling mechanism is a specific metamodeling technique that specializes a reference metamodel in such a way that the specialized semantics do not contradict the semantics of the reference metamodel. So, contrary to the first approach, the definition of a UML profile does not add new metaclasses and does not modify the standard UML metaclass definitions (e.g. by adding meta-associations). A UML profile is a specific kind of UML package that contains stereotypes. A stereotype is a limited kind of metaclass that cannot be used on its own, but must always be used in conjunction with one of the metaclasses it extends. Extension means that the properties of the metaclass are extended through that of the stereotype. Extensions can be required or non-required; in the latter case the metaclass can be instantiated without instantiating the stereotype that extends the metaclass. Typically, a profile also defines well-formedness rules that are more constraining, but consistent with those already specified by the reference metamodel. In our approach these well-formedness rules are based on the domain ontology axioms and are specified using OCL.

The approach will be further explained in the next sections where we present a concrete implementation of it and apply it to the domain-specific quality evaluation of conceptual schemas.

4. The Resource-Event-Agent ontology

The Resource-Event-Agent ontology (REA) [4, 6] is a domain ontology that originates in McCarthy’s semantic data
model for accounting [5]. The subject domain of REA can be described as ‘the enterprise’. The particular conceptualization of enterprises specified by REA is heavily influenced by REA’s accounting background. Primary attention is paid to what changes the value of the enterprise (i.e. recording these value-affecting events and the value composition of the enterprise is what we call ‘accounting’) and who can be held accountable for this (i.e. accounting enables control of the organization and its members).

So, enterprise reality is described in terms of Resources (having value), Events (affecting this value) and Agents (having custody over the resources and being responsible for the events). The conceptualization includes additional business concepts to predict future value changes (e.g. contracts, terms and commitments) and to specify policies for value creation, transfer and consumption (e.g. business rules).

There are three reasons why we have chosen REA to illustrate a concrete implementation of our approach:

- REA is aligned with a top-level ontology for knowledge representation, i.e. Sowa’s ontology [21], meaning that all REA concepts are specializations of concepts in Sowa’s ontology [6].
- REA is supported by theoretical and empirical evidence. On the one hand it has firm roots in various accounting and micro-economic theories [22]. On the other hand it has also been conceived on the basis of empirical observations [6].
- There is a community of practitioners and researchers for which REA is the specification of a shared conceptualization. The consensual agreement on REA is further evidenced by its acceptance in August 2007 as an international ISO standard [23], i.e. the ISO/IEC 15944-4:2007 accounting and economic ontology.

REA has a limited number of core concepts (Table 1), which are the ontological primitives upon which the other ontological elements (e.g. concept specializations, concept relations) are built. For the purpose of this paper we only need to be concerned with this ontological core. The relations between these concepts (shown in Figure 2 as associations in a UML class diagram model of REA’s core) are the basis for defining an enterprise domain axiomatization, i.e. basic laws of business and economic principles that must hold for any enterprise. Table 2 gives an informal account of these REA axioms. A formal axiomatic system (in OWL) has been presented in [24].

**Table 1. Definition of REA core concepts**

<table>
<thead>
<tr>
<th>Concept</th>
<th>Concept definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>economic resource</td>
<td>A thing that is scarce and has utility for economic agents and is something users of business applications want to plan, monitor and control.</td>
</tr>
<tr>
<td>economic agent</td>
<td>An individual or organization capable of having control over economic resources, and transferring or receiving the control to or from other individuals or organizations.</td>
</tr>
<tr>
<td>economic event</td>
<td>A change in the value of an economic resource that is under the control of the enterprise. Each economic event is either an increment economic event or a decrement economic event.</td>
</tr>
<tr>
<td>commitment</td>
<td>A promise or obligation of economic agents to perform an economic event in the future. Each commitment is either an increment commitment or a decrement commitment.</td>
</tr>
</tbody>
</table>

**Table 2. REA’s enterprise domain axiomatization**

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Axiom description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stockflow axiom</td>
<td>At least one increment economic event and one decrement economic event exist for each economic resource; conversely increment and decrement economic events must affect identifiable economic resources.</td>
</tr>
<tr>
<td>duality axiom</td>
<td>All decrement economic events must be eventually paired in duality relationships with increment economic events and vice-versa.</td>
</tr>
<tr>
<td>participation axiom</td>
<td>Each economic event must be related by a provide relationship to an economic agent and by a receive relationship to an economic agent.</td>
</tr>
<tr>
<td>reciprocal axiom</td>
<td>Each increment commitment must be related by a reciprocity relationship to at least one decrement commitment and vice versa.</td>
</tr>
<tr>
<td>fulfilment axiom</td>
<td>Each commitment must be fulfilled by at least one economic event; increment commitments must be fulfilled by increment economic events and decrement commitments must be fulfilled by decrement economic events.</td>
</tr>
</tbody>
</table>

**Figure 2: UML class diagram model of REA’s core concepts and relations**

5. A UML profile for REA

Figure 3 is a graphical representation of the UML profile that we defined for REA. The notation for an extension is an arrow pointing from a stereotype to the extended metaclass, where the arrowhead is shown as a filled triangle. The REA UML profile was developed using IBM’s Rational Software Modeler, a well known modeling tool based on the open source platform Eclipse. The use of the profile is, however, not limited to Rational Software Modeler as it can be exported XMI which is the Object Management Group (OMG) standard for exchanging models. The REA UML profile can be downloaded from [http://code.google.com/p/reaprofile/](http://code.google.com/p/reaprofile/).

Models developed at the M1 Model Driven Architecture (MDA) level can be reused to define profiles at the M2 level. Consequently we started from a previously developed UML class diagram representation of REA [25] and identified 20 stereotypes to be included in the UML profile for REA. The stereotypes that stand for REA concepts (represented as classes in the UML models of REA) extend the UML metaclass Class (e.g. Commitment, EconomicAgent, EconomicEvent, EconomicResource) or specialize these extensions (e.g. IncrementCommitment, DecrementCommitment, …).
We also introduced stereotypes that stand for the REA concept relations (represented as associations in the UML models of REA). These stereotypes extend the UML metaclass `Association` (e.g. `stockflow`, `duality`, ...) or specialize these extensions (e.g. `inflow`, `outflow`, ...). To realize the effect of these new meta-associations within the profile, OCL constraints are defined that restrict the profile’s metaclasses whose instances can be related by instances of the new meta-associations. For instance, the `outflowConstraint` of the `outflow` stereotype specifies that in M1 models each instance of the `outflow` stereotype is a binary association that relates an instance of the `DecrementEconomicEvent` stereotype with an instance of the `EconomicResource` stereotype.

Other OCL constraints were defined to specify the REA enterprise domain axioms. For instance the stockflow axiom is transformed into two different OCL constraints. The first part requires that in an M1 model every instance of the `EconomicResource` stereotype has at least one association that is an instance of the `outflow` stereotype. For instance, if a UML class `Machine`, representing some type of machine owned by the enterprise, is annotated as being an instance of the `EconomicResource` stereotype, then `Machine` needs at least one association that is an instance of the `outflow` stereotype. Otherwise the enterprise model would not reflect the economic rationale of the enterprise (i.e. machines are used to produce goods of value). The second part of the stockflow axiom requires that every `IncrementEconomicEvent` object (or `DecrementEconomicEvent` object) is related to at least one `EconomicResource` object. For instance, every object of a UML class `AcquireGood` (or `Acquisition`) needs to be linked to an object of some UML class representing a type of goods that can be acquired. So, the multiplicity on the `EconomicResource` side of each instance of the `inflow` (or `outflow`) stereotype in an M1 model should be higher than zero.

6. Evaluating REA domain-specific quality

The concrete implementation of the proposed approach in the previous section allows demonstrating its utility for quality assurance. In this section we present the application of the REA UML profile for the evaluation of the domain-specific quality of a sample of 17 UML class diagrams that were intended as enterprise models. We demonstrate that using our approach, different problems with the domain-specific quality of the models are pinpointed and classified (i.e. identifying which domain constraint is violated) such that they can be evaluated (i.e. deciding whether they reflect real semantic quality problems) and resolved.

The diagrams that constitute the sample were produced by final-year undergraduate business students enrolled in a systems development course as a required output of a development project. Groups of students were assigned to one of two possible development projects. One project involved the development of an enterprise system supporting the planning, operations and administration for a newly created bicycle production company. The other project concerned the development of an inventory management system for a company that sells private flights to enterprises and wealthy customers. In this project, all student groups had to model the complete purchase transaction cycle (i.e. ordering, accepting deliveries and payment) and part of the flight creation transaction cycle (i.e. reservation of articles consumed on board, usage of these articles). The basis for creating the models were narratives detailing the working and listing the data and information requirements of the different transaction cycles.

For these 17 diagrams the domain-specific quality was evaluated by a person different from the course teacher with expertise in enterprise modeling and knowledge of the REA ontology. This expert annotated in each diagram the model elements with the applicable stereotypes of the REA UML profile. In doing so, the expert made sure that REA stereotyped associations connect the REA stereotyped classes they are supposed to connect. Next the expert used the model validation function of the tool to check the OCL constraints that specify REA axioms.

Table 3 gives an overview of the validation results. It lists for each of the 17 diagrams the number of OCL constraints violated (i.e. the numbers before correction in the table). These violations indicate potential problems with the REA-related domain-specific quality of the diagrams. Each case of OCL constraint violation was examined by the course teacher to find out whether it really represents a quality problem (indicating that the diagram can be further improved as a representation of the transaction cycle) (i.e. the numbers after correction in the table), is the result of a deliberate modeling compromise (which might be desirable/acceptable or not), or is a consequence of the intended scope of the diagram. The latter case frequently occurred because REA specifies a conceptualization of an enterprise in its entirety, assuming that all transaction cycles are interconnected to form a value chain, whereas the diagrams evaluated represent isolated transaction cycles. Especially violations of the stockflow axiom because of scoping decisions were frequent in the sample.

In the remainder of this section we comment upon the results of the quality evaluation of two representative diagrams out of the sample evaluated. Model 1 (Figure 4) is a conceptual schema of the bicycle parts purchasing transaction cycle. The validation of Model 1 indicated that two OCL constraints corresponding to REA axioms were violated: `StockflowAxiomTypeIncrement` for the Money «EconomicResource» class and `StockflowAxiomTypeDecrement` for the Parts Stock «EconomicResource» class. According to the course teacher, both violations can be justified. They are the consequence of the scope of the diagram because the purchasing of bicycle parts does not include the consumption of these parts in bicycle assembly or the generation of revenue. So we can say that the domain-specific quality of this diagram is perfect.
result of a modeling compromise because every delivery is always connected to an order, which means that we can also identify the receiving and providing Economic Agents via the order «IncrementCommitment». The course teacher decided, however, that this compromise cannot be justified as the internal agent accountable for ordering articles is not necessarily the same as the internal agent accountable for article delivery (because of the principle of ‘segregation of duties’). Given the project description, also this violation indicates a lack of completeness of the model.

Model 16 (Figure 5) is a student solution for the inventory management system. The description of the project indicated that only one half of the flight creation cycle, namely the reservation and usage of articles consumed on board, had to be modeled. This means that the absence of a «duality» and «receive» association for the ProductUsageReport «DecrementEconomicEvent» class and the absence of «reciprocity» association for the Reservation «DecrementCommitment» is not an incompleteness of the model but the result of a deliberate scoping decision.

Five other OCL constraints are not satisfied. The model does not contain a «reciprocity» association and a «duality» association for respectively the Order «IncrementCommitment» and the Delivery «IncrementEconomicEvent». Both associations were within the scope of the model according to the project description, so these two violations are clear cases of model incompleteness and indicate a lack of semantic quality. Further, the StockflowAxiomTypeDecrement constraint was violated for the ProductUsageReport «DecrementEconomicEvent» because an instance of this class can exist without being linked to an instance of the Stocks Product «EconomicResource» class (i.e. the participation in «outflow» links is optional). This would mean that a system based on the model would support the registration of empty usage product reports, which clearly is wrong, so this is a case of model incorrectness that was identified using the profile. The ReciprocityAxiomDecrement constraint is also violated, pointing at the absence of an «incrementFulfill» association between the Reservation «DecrementCommitment» class and the ProductUsageReport «DecrementEconomicEvent» class. The consequence of this omission is that reservation and usage of products can be registered, but not matched. So if a system is built on the basis of this schema, then it cannot inform its users which product reservations have actually been fulfilled and which have not. This is a clear sign of the schema being incomplete. Finally, the ParticipationAxiomInstance is violated for the Delivery «DecrementEconomicEvent» class because an instance of a Delivery can be created without a connection to a providing and receiving Economic Agent. This violation could have been the

<table>
<thead>
<tr>
<th>Model</th>
<th>classes</th>
<th>relations</th>
<th>Total errors before correction</th>
<th>Total errors after correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>14</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>11</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>14</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>13</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>12</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>7</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>13</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>12</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>9</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>10</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>13</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>9</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>17</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>5</td>
</tr>
</tbody>
</table>

9. Conclusions

In this paper we presented an approach to improve the domain-specific quality of UML class diagrams that are used as conceptual schemas representing real-world situations within some domain of interest. We defined the notion of domain-specific quality as a component of semantic quality that depends on the satisfaction of the invariant conditions that define the domain of interest. These conditions should hold for all conceptual schemas that represent phenomena belonging to that domain. Our approach prescribes the creation of an UML profile defined by a domain ontology that specifies a domain axiomatization in terms of concepts, relations between concepts and the rules that govern these relations (i.e. the invariant conditions that define the domain). Domain concepts and relations are defined as stereotypes extending standard UML metaclasses. Domain axioms are formalized as OCL constraints. A UML profile created this way allows using UML as a domain-specific modeling language. Domain-specific quality can be assured by requiring that conceptual schemas are instantiations of the UML profile. The domain-specific quality of existing UML conceptual schemas can be evaluated by first annotating them with the profile’s stereotypes and next verifying whether the OCL constraints corresponding to the domain axioms are satisfied.
Our contribution is original in the sense that we advocate and
demonstrate conceptual modeling using domain ontologies (e.g.
REA) instead of the top-level ontologies that previous research
has employed to derive ontology-based conceptual modeling
rules. As a consequence we use UML as a domain-specific
conceptual modeling language and not as a general-purpose
conceptual modeling language. Furthermore, given that the
domain ontology that is used to define a domain-specific UML
profile is aligned with a top-level ontology (e.g. REA is aligned
with Sowa’s ontology), the domain semantics imparted on the
UML constructs is at the same time real-world semantics. Our
contribution is also original in the sense that we demonstrated the
use of a domain ontology defined UML profile for quality
evaluation purposes. In particular, we showed that a domain-
specific language profile defined by a domain ontology can be
used to evaluate the domain-specific quality of existing
conceptual schemas that are written in the language that is
extended by the profile.

A limitation to the proposed approach relate to the
availability of formal domain ontologies and software modeling
tools that offer support for the evaluation of OCL expressions.
Whereas Guarino in 1998 was sceptical about the feasibility of
information systems development being driven by domain and
task ontologies (“the availability of off the shelf ontologies to be
used in this way is today extremely limited” [1, p. 12]) we believe
that today the situation is different and tool-supported conceptual
schema construction and evaluation using a standard modeling
language (UML) and based on mid and low-level computational
ontologies that foster reuse and built-in quality can really happen.
In future research we plan to further demonstrate are approach by
using different domain ontologies during the development of a
conceptual schema.

7. REFERENCES
modeling semantics in language syntax. IEEE Transactions
Ontologies and Conceptual Schemas Through Ontology-
Driven Information Systems. Journal of the Association for
Infrastructure for Knowledge Based Enterprise Models.
IEEE Intelligent Systems and Their Applications, 14, 4
(1999), 89-94.
Framework for Accounting Systems in A Shared Data
the Economic Primitives of the Extended-REA Enterprise
Information Architecture. International Journal of
Information System. IEEE Transactions on Software
Engineering, 16, 11 (1990), 1282-1292.
information systems analysis and design grammars.
[10] Guizzardi, G. On Ontology, ontologies, Conceptualizations,
Modeling Languages, and (Meta)Models. IOS press,
Amsterdam, 2007.
[12] Soffer, P. and Hadar, I. Applying ontology-based rules to
conceptual modeling: a reflection on modeling decision
making. European Journal of Information Systems, 16, 5
(2007), 599-611.
Domain Analysis of Domain-Specific Languages. Models in
Software Engineering (2009), LNCS 5421, 332-342.
[14] Assmann, U., Zchaler, S. and Wagner, G. Ontologies, Meta-
role of Domain Ontologies in the Design of Domain-
Specific Visual Languages. Helsinki School of Economics
Printing, 2002.
[16] Silva Parreiras, F. Marrying Ontologies and Model Driven
Engineering Technical Spaces: The TwoUse Approach. In
Proceedings of the PHD Symposium ER 2008 (Barcelona,
Spain, 2008)
[17] Walter, T. and Ebert, J. Combining DSLs and Ontologies
Ontology-Based Framework for Domain-Specific
Languages. Model Driven Engineering Languages and
Systems (2009), LNCS 5795, 408-422.
[19] Batra, D. Conceptual data modeling patterns: Representation
and validation. Journal of Database Management, 16, 2
(2005), 84-106.
[21] Sowa, J. Knowledge Representation: Logical, Philosophical,
and Computational Foundations. Pacific Grove,
[22] Dunn, C. L. and McCarthy, W. E. The REA accounting
model: Intellectual heritage and prospects for progress.
[23] ISO Information technology -- Business Operational View --
Part 4: Business transaction scenarios -- Accounting and
Information Systems: Redesign and Formalization of the
REA Ontology. Business Information Systems (BIS 2007),
Modelling: Improving the Conceptual Representation of the
4801, 407-422.