FACULTEIT ECONOMIE EN BEDRIJFSKUNDE

TWEEKERKENSTRAAT 2
B-9000 GENT
Tel. : 32 - (0)9 – 264.34.61
Fax. : 32 - (0)9 – 264.35.92

WORKING PAPER

Positioning REA as a Business Domain Ontology

Frederik Gailly [1]

Wim Laurier [2]

Geert Poels [3]

May 2007

2007/460

[1] Frederik.Gailly@ugent.be, Faculty of Economics and Business Administration, Ghent University {correspondent author}

[2] Wim.Laurier@ugent.be, Faculty of Economics and Business Administration, Ghent University

[3] Geert.Poels@ugent.be, Faculty of Economics and Business Administration, Ghent University

D/2007/7012/31

Abstract Ontologies are a popular research direction in different domains as is also the case in information systems research. In the beginning of this millennium Geerts and McCarthy (2002) proposed an ontology for the business context which was an extension of the basic REA-model which had already proven its use as a semantic accounting information model. In the recent papers of Geerts and McCarthy the focus is primarily on defining and theoretically justifying the content of this newly proposed enterprise ontology. In this paper we elaborate on more practical issues related to the REA-ontology. A lot of confusion exists about what ontologies are and for what purpose they can be used. In this paper we investigate how we could classify the REA-ontology and the REA-ontology applications. This analysis clarifies the application potential of the REA-ontology but also emphasizes that a generally accepted, explicit and formal specification is needed in order to improve the usability. In the case of the REA-ontology this means that the ontology should be more unambiguously to interpret by business experts and the applicability for ontology-driven system development and ontology driven systems should be improved.

This paper proposes a new REA-ontology specification that uses an UML profile for graphically representing ontologies (OMG 2006). This specification of the ontology in a single graphical representation formalism is more complete than previously available representations, without compromising its ability to be understood by business professionals. At the same time it can easily be transformed into a more formal representation which can be understood by machines. Having a machine readable representation of the REA-ontology is a necessary prerequisite for the successful application of the REA-ontology in business modeling, software engineering, knowledge representation and interoperability creation.

Keywords: REA-ontology, REA-ontology applications, OWL, UML, business modeling

I Introduction

Around the end of the nineties and the beginning of the millennium, Geerts and McCarthy (1999; 2002) extended the Resource Event Agent semantic accounting model into a comprehensive enterprise information architecture which they propose as an ontology for enterprises. Since then a lot of research has been conducted in different sub-domains of Computer Science which further explore the development, use, evaluation, etc. of ontologies. These research findings may indeed support the lifting of REA onto a higher ontology level. By this we mean, amongst others, formalizing REA in order to make it machine readable and determining the correct position of REA as a business domain ontology.

Different business domain ontologies have been proposed but it is not always clear what the intended use of these ontologies is. If we look at existing ontology engineering methodologies (for an overview see (Gómez-Pérez et al. 2004)) one of the first steps in the ontology development process is specifying the intended use of the ontology. This was not explicitly done for the REA-ontology. The basic REA-model was gradually extended because the developers believed that the REA-ontology should have some natural implementation advantages over more traditional kinds of accounting conceptualizations. These advantages relate to the increasing need in enterprise information systems for shared communication and increased ontological commitment. However, Geerts and McCarthy (1999; 2002) only partly state how these advantages can be realized in possible ontology applications.

This paper intends to position the REA-ontology as a business domain ontology by taking into account generally accepted ontology research, related business domain ontology research and recent developments in the REA-ontology research (or so called REA design science research). In this paper we first classify the REA-ontology and REA-ontology applications according to well known ontology classification schemes in order to develop a clear understanding of the intended use of the REA-ontology. This analysis results in the identification of the possible application domains for the REA-ontology. Furthermore the importance of having a generally accepted formal specification of the REA-ontology is also emphasized. In the second part of the paper, a further step is taken to enhance the applicability of the REA-ontology. We present the development of a representation of the REA-ontology that uses an UML profile for graphically representing ontologies (OMG 2006). This specification of the ontology in a single representation formalism is more complete than previous representations, should still be easy to understand by business professionals and can also be easily transformed into a machine-readable representation.

The paper is structured as follows: Section II classifies the REA-ontology based on the richness of its internal structure and the subject of its conceptualization. In order to further clarify the intended use of the REA-ontology, section III analyses some recent REA-ontology applications. Based on this analysis some key issues are identified that when solved would facilitate the realization of further REA-ontology applications. As a first step towards solving some of these issues, in section IV the REA-ontology is graphically represented using the UML OWL profile. Section V illustrates how this new representation can be used for business modeling which is one of the possible applications of the REA-ontology. Section VI ends with conclusions and future work.

II Classification of the REA-ontology

A first step in positioning the REA-ontology is classifying the ontology. Different classification schemes for ontologies have been proposed (a good overview can be found in (Gómez-Pérez et al. 2004)). The overall focus of our research project is primarily on improving the applicability of the REA-ontology and therefore we will classify the REA-ontology based on the richness of its internal structure and on the subject of its conceptualization. This classification must also make it easier to define the intended use of the REA-ontology and compare it with the intended use of the other business domain ontologies.

Subject of the Conceptualization

Using an ontology classification scheme which is based on the subject of the conceptualization is very helpful for providing a clear definition of the proposed ontology. The most cited definition of an ontology is the definition by Gruber: “an ontology is an explicit specification of a conceptualization” (Gruber 1993, p. 199). A conceptualization is an intensional semantic structure which encodes the implicit rules constraining the structure of a piece of reality (Guarino and Giaretta 1995). Ontologies can be used to represent explicitly the semantics of structured and semi-structured information enabling automatic support for maintaining and accessing information (Fensel 2001). The Gruber definition was modified slightly by Borst (1997) who added that the specification must be formal and the conceptualization should be shared. Formal means that a machine must be able to process the specification and shared indicates that the knowledge captured by the ontology is the consensus of a community of experts (Gómez-Pérez et al. 2004).

Based on the subject of the conceptualization different types of ontologies can be distinguished (van Heijst et al. 1997): representation ontologies, top-level ontologies, domain ontologies and application ontologies. The REA-ontology is not intended as a *knowledge representation ontology* (or ontology language) like the Knowledge Interchange Format (KIF) or the web ontology language (OWL), although such representation ontologies can be used to formalize the content of the REA-domain ontology. Furthermore the REA-ontology is not a *top-level ontology* as its Universe of Discourse is not the real world, but only a part of it: one or more enterprises in a business context. REA should therefore be a specialization of a top-level ontology. Top-level ontologies describe very general concepts that are reusable across domains. The REA-developers use the SOWA classification of ontological categories in one of their papers (Geerts and McCarthy 2002) to provide definitions for the different REA-concepts and relations.

Domain ontologies specify a conceptualization of a selected part of reality (i.e. a domain) (Guarino 1998). They describe the concepts that exist in a domain, the classification of the concepts, the relations between the concepts and their axioms (i.e. basic propositions assumed to be true). Based on the subject of the conceptualization the REA-ontology can be classified as a business domain ontology. Business domain ontologies have as Universe of Discourse business, which is "the activity of providing goods and services involving financial, commercial and industrials aspects" (Cognitive Science Laboratory 2006). It should be noted that the REA-ontology does not support all these business related aspects. The REA-ontology, as an event-ontology (Allen and March 2006), focuses on the creation and transfer of economic value and does not include all concepts for capturing the organizational structure and management of the business. Commercial aspects include for instance marketing strategies. Industrial aspects may also include geographical details. Neither of these concepts are included in the REA-ontology. This does not mean that the REA-ontology needs to be further extended, however this proves that the integration of REA with other appropriate ontologies is an important issue and should also be addressed when investigating the usability of the REA-ontology.

In order to use business domain ontologies in actual implementations, *application ontologies* that fine-tune the business domain ontology to a specific application are needed. REA is not an application ontology because its intended use is not limited to one particular application (not even a broadly defined application such as accounting). In ontology research the contradiction between generality (or ontology reusability) and specificity (or ontology usability) has been recognized and some ontology engineering techniques have been proposed that take this contradiction into account during the development of an ontology (Guarino 1998).

There are many potential applications of business domain ontologies. It is generally accepted that ontologies are used to improve communication between humans or computers. Uschold and Jasper (1999) specify this use further into the following three areas: to assist in communication between human agents, to achieve interoperability among computer systems or to improve the process and/or quality of software engineering. Different business domain ontologies have been proposed with accompanying specializations in business application ontologies which have been implemented in one or more of these areas. Broadly, the intended use of these business domain ontologies may be grouped into two popular application areas: *business modeling* and *e-collaboration*. The Toronto Virtual Enterprise ontology (TOVE) (Fox 1992), the Enterprise Ontology (EO) (Ushold et al. 1998), the E^3-value ontology (Gordijn 2002) and the Business Model Ontology (Osterwalder 2004) can all be considered as *business modeling ontologies*. The main purpose of this type of business domain ontology is supporting communication between people. However business modeling ontologies can also support interoperability by providing a translation between different modeling methods (e.g. EO). Additionally, business modeling ontologies can also improve the software engineering process by assisting the process of identification of requirements (e.g. E^3-value).

E-collaboration ontologies focus on supporting the collaboration between and within enterprises. A widely used group are the Products and Services Categorization Standards (PSCS) (Hepp et al. 2005). They provide frameworks to identify products and services in global markets and are used to support the information exchange between customers and suppliers, and among different suppliers, which is required in a B2B context where an effective communication between machines is necessary. Well-known examples are UNSPSC (United Nations 2007), NAICS (U.S. Census Bureau), e-cl@ss classification and product description (eCl@ss) and the RosettaNet Technical Dictionary (RosettaNet). Another well-known example of e-collaboration ontologies are the XBRL taxonomies which focus the semantic integration of financial information from heterogeneous sources.

Important to notice is that the distinction between the different kinds of business domain ontologies is not strict. Following the lack of a clear specification of the intended use of the REA-ontology, it is impossible to assign the REA-ontology to one specific group of business domain ontologies. There have been REA-ontology applications that use a specialization of the REA business domain ontology to support business modeling. But the REA-ontology has also been specialized such that it facilitates e-collaboration. In section III some recent REA-ontology applications related to both uses will be further analyzed.

Richness of the internal structure

The second classification dimension classifies different types of ontologies according to the richness of their internal structure (Lassila and McGuiness 2001). Many forms of specifications of conceptualizations exist which are all referred to as ontologies. Figure 1 visualizes different types of lightweight and heavyweight ontologies in a continuous line.

The richness of the internal structure of the REA-ontology has been evolving a lot since the publication of the original REA-model. Geerts and McCarthy still use in their papers a combination of textual descriptions of the concepts, the relations of the concepts and the axioms, and graphical representations of partial views on the ontology in a conceptual modelling language. The appearance and subsequent popularity of the Unified Modelling Language (UML) has driven the move from Entity Relationship (ER) representations of REA (as in McCarthy (1982)) to UML representations. Conceptual modeling languages like UML and ER have rich representation systems, are widely used in practice and are well supported by tools. In systems development practice, models developed using such languages are widely used to communicate the user/expert view of the domain and information (system) requirements between project stakeholders (Davies et al. 2006). As such they are designed to be easily understood and they do not require users to have a profound know-how of knowledge representation languages.

Comparing to Figure 1, the REA-ontology contains a semantically rich internal structure that is not limited to the IS-A relations as found in a typical thesaurus. However, many details of the REA-ontology internal structure are not explicitly specified. Like Geerts and McCarty (1999) we recognize the importance of better specifying the REA-ontology by further developing its ontological engineering aspects. An important step to take here is formalizing the REA-ontology in an ontology representation language and transforming its mix of textual and graphical representations into a more coherent, formal representation. There have been some efforts by researchers (Bialecki 2001; Borch et al. 2003; Geerts 2004) to represent the REA-ontology in machine readable form but none of these formalizations is widely known or generally accepted.

<< insert figure 1 >>

In previous work (Gailly and Poels 2007b) we proposed a formalization of the REA-ontology in OWL. This formalization was the result of a reengineering methodology which uses well known ontology engineering principles. In our future research this reengineering process will be repeated in order to further improve the resulting specification of the ontology and make the specification more generally accepted. The first iteration described in Gailly and Poels (2007a; 2007b) focused on developing a single UML representation of the REA-ontology that makes the specification more explicit and improves the existing partial representations employing various informal and semi-formal representation formats. An additional advantage of the new conceptual representation of the REA-ontology is that it could be used to generate a formal representation in OWL using UML-to-OWL mapping rules. The next iterations of our reengineering process will focus more on the content of the

REA-ontology and will require feedback from the REA-ontology community (to result in a 'shared' conceptualization).

Important to notice is that Chou (2006) has also recognized the importance of using an ontology engineering method for the development of an accounting ontology. Chou used the REA model as a starting point for the development of a general accounting knowledge model and also started with the ontological formalization of the REA model using OWL. However, his work has a different focus than ours because it aims at developing an accounting application ontology that can be used to give semantic meaning to core accounting data. Our research follows more the view of the REA-ontology developers, which see the REA-ontology as a business domain ontology leading to a wide variety of applications (of which accounting is of course one of the more important).

It is our belief that existing business domain ontologies must be further improved in order to fully exploit their application potential. The general approach taken in our research is to incorporate general ontology research into existing business domain ontology research which in most cases is not executed by ontology researchers but by business experts. Similar approaches have been followed for other business domain ontologies. For instance, the eClassOWL project has as goal the ontologizing of the eCl@ss e-collaboration ontology which must result in an OWL lite representation of eCl@ss (Hepp 2006). Additionally, parts of the XBRL taxonomies have been transformed into OWL in order use the reasoning mechanisms of OWL which are not supported by XML and XML-schema (Lara et al. 2007).

III Classification of REA-ontology applications

The first research goal of this paper is positioning the REA-ontology as a business domain ontology. This can be done conceptually (see previous section), but also by looking at existent REA-ontology applications and investigating how the REA-ontology is used in practice.

REA-ontology application is defined here in a very broad sense and refers to applications that make use of or benefit somehow from the ontology. Some of these applications are only theorized and not actually put into practice but exploring these applications can help us in identifying the intended use of the REA-ontology.

The classification of the REA-ontology applications uses the framework for understanding and classifying ontology applications of Uschold and Jasper (1999). The framework identifies different application scenarios which are characterized by five key dimensions: the intended purpose or benefits of the application, the role of the ontology, the actors required to implement the scenario, the supporting technologies and the maturity level. In the rest of this section we will use this framework to describe some published REA-ontology application experiences. An overview of the different applications can also be found in table 1. We limit ourselves largely to applications that have been proposed or developed after the publication of the REA ontology extension (i.e. Geerts and McCarthy (1999; 2002). Applications that are mainly based on the original REA model as in McCarthy (1982) are not discussed.

REA accounting model in education:

References: (Dunn et al. 2005; McCarthy 2003)

Intended purpose or benefits: The REA-ontology is used as a conceptual framework for teaching accounting information systems. It integrates the teaching of accounting transactions structures, commitments and business policy specification, business process engineering and enterprise value chain construction.

Role of the ontology: The ontology acts as a reference and provides reliable and objective information to those who want to learn more about the underlying structure of the accounting information systems domain. The REA-ontology is also used as a meta-model to generate reference models for different types of business (or transaction) cycles, and thus provides an instrument to teach the main elements and structure of these cycles.

Actors: Business/accounting information system professors and students

Supporting technologies: During the business/accounting information systems courses conceptual modeling tools are used for the development of REA-models which can be transformed into relational database models which can be implemented in simple relational database systems. There has also been developed an online education tool that can be used by AIS professors for teaching the REA concept relations and what the consequences are of the cardinality restrictions on the different REA concept relations (Geerts et al. 2002).

The maturity level: This is definitely the most mature REA-ontology application. A considerable group of accounting information systems courses use the REA-ontology and different textbooks haven been published which teach the accounting information systems domain by means of the REA-ontology.

<u>Model-driven design using REA business domain ontology:</u>

References: Model-driven design using business patterns (Hruby 2006, 2005), A Model Driven Architecture for REA based systems (Borch et al. 2003)

Intended purpose or benefits: Improve the design of applications by using the REA-ontology. Using the REA-ontology makes the development of the application more straightforward and software applications based on REA contain more and more correct business knowledge.

Role of the ontology:

- REA is used as a language ensuring unambiguous communication and understanding among all participants of the software development process

- Borch et al. (2003) illustrate how the REA-ontology can be used in the model-driven development of systems. The REA-ontology is used to develop a platform independent XML-model which can be automatically transformed into a java enterprise application.

Actors: Business Analysts, Business Modelers and all other participants of the development process (users, consultants and application developers)

Supporting technologies: MDA-technologies (UML, QVT, …), and XML technologies can support this application. For example a REA UML profile can be used to support the REA-ontology-driven business application development. According to our knowledge this UML profile is currently not available.

Maturity level: Using patterns for software development is a well-known and frequently used technique (e.g. design patterns). Using patterns for modeling an application (i.e. analysis patterns) has never been that popular, but is recently getting more attention. The application of Borch et al (2003) provided first insights of using the REA-ontology for ontology-driven software engineering but is only theorized in the paper and it should be further investigated how domain ontologies such as REA can be integrated into model-driven engineering approaches to develop a real proof of concept. Recently the use of ontologies in an MDA context is getting more and more attention (Assmann et al. 2006; Gasevic 2006; OMG 2006).

<u>Supply Chain Collaboration:</u>

References: Open-EDI business transaction ontology (ISO 2006), UN/CEFACT Modeling Methodology (Hofreiter et al. 2006) and Internet Supply Chain Collaboration (Haugen and McCarthy 2000)

Intended purpose or benefits: Using the REA-ontology should make it easier to establish supply chain collaboration via internet technologies. E-collaboration can be realized by the

REA-ontology by providing standard business scenarios and the necessary services to support them in order to establish quickly and cost effectively short term relationships between businesses. The UN/CEFACT Modeling Methodology focuses more on the global choreography of the collaboration between business partners. It enables to capture business knowledge independent of the underlying implementation technology, like Web Services or ebXML.

Role of the ontology: The REA-ontology is used for supporting interoperability within and between enterprises. It provides an ontological framework for specifying the concepts and relations involved in business transactions and scenarios.

- The Open-EDI business transaction ontology is based on the REA-ontology and can be used for creating interoperability between different enterprise applications. The ontology is used as an interchange format.
- In the application of Haugen and McCarthy (2000) the REA-ontology is used by the different trading partners and provides a computer readable model of the classes, relationships and functions that are involved in supply chain collaboration. The role of the ontology is creating interoperability between the REA-based systems of a supply chain.
- The UN/CEFACT Modeling Methodology uses parts of the REA ontology to specify a global choreography of a business collaboration serving as an ”agreement” between the participating business partners in the respective collaboration.

Actors: the trading partners, the communication provider, system developers … **.**

Supporting technologies: Conceptual modeling languages like UML, XML for representing the semantic model, formal ontology languages

Maturity level: There exist different standards instead of just one and it is generally believed that standardization efforts with a specific focus on a industry sector may be more successful.

A further degree of harmonization is certainly needed. Open-EDI also require in many cases long and costly negotiations and as a result successful EDI implementations have been realized in what could be called 'closed trading relationships', i.e. long-lasting trading relationships, involving a high number of transactions, between parties that have a high level of trust and possibly a close coordination of the parties' business processes.

The REA-ontology for knowledge representation

References: Augmented Intensional Reasoning in Knowledge-Based Accounting Systems (Geerts and McCarthy 2000), An XML architecture for Operational Enterprise Ontologies (Geerts 2004)

Intended purpose or benefits: By adding semantics into the business applications additional information can be more easily extracted from the business.

Role of the ontology: The different systems use the operational REA-ontology at run-time for adding semantics to the enterprise schema and data. Additionally the REA-ontology is also used as a language for the representation of ontological scripts which can be used for searching different enterprise systems. As such Geerts positions the REA-ontology also as a type of representation ontology.

Actors: Systems developers, auditors

Supporting technologies: Different XML-technologies can be used for the description of the REA-ontology (XML-schema and XSLT), the ontological scripts (XSLT) and the enterprise data (XML-documents).

Maturity level: Ontologies can be used for improving the performance of information retrieval. Geerts uses well-known XML-technologies for illustrating this. A lot of the techniques that are described by Geerts are now supported by newly developed ontology representation languages. Using OWL for representing the ontology makes it possible to represent the REA-ontology and the enterprise data in one knowledge base which can be very easily queried by using the OWL Query language (OWL-QL) or the Semantic Web Rule Language (SWRL) for which Geerts defines his own representation language in XSLT.

The classification of the REA-ontology applications provides some proof that REA can be used for a wide series of applications: education, business modeling, software engineering, knowledge representation, information retrieval and various e-collaboration applications. Apart from education and (to some extent) inter/intra-enterprise modeling, many of the proposed applications are only theorized or implemented with an illustrative (toy) example rather than providing a convincing proof of concept. It is clear that in order to fully exploit the potential of REA as a business domain ontology, a generally accepted, explicit and formal specification of the REA ontology is needed which is reusable across different types of business applications.

The degree of formalization required depends of course on the type of application. In an educational context a formal representation of the REA-ontology is less desirable than in ontology driven information system (engineering) contexts. In order to make a formal and explicit specification of the REA-ontology also usable for application contexts which do not require a high degree of formality, the formal representation should be easily transformable into a graphical, semi-formal and easy-to-understand representation.

A general agreement about a formal and explicit specification of the REA-ontology will also make the realization of the currently theorized applications more straightforward and can make the REA-ontology more useful for application areas which are currently not explored by the REA-ontology community.

IV Specification of the REA-ontology using UML OWL profile

In this section we will present a further step towards a more formal and explicit specification of the REA-ontology, which is the second research goal of this paper. This work builds further upon Gailly and Geerts (2007b) where a formal specification of the REA-ontology in OWL was presented. In this paper we develop a graphical representation of this OWL representation. This representation is more formal and explicit than the currently available REA-ontology representations, without losing the advantages that graphical modeling languages offer for non-experts in knowledge representation.

In Gailly and Poels (2007a; 2007b), the REA-ontology was reengineered following some well known ontology engineering principles. The 'version' of the REA-ontology considered corresponds closely to the REA-ontology as presented in the most recent papers of Geerts and McCarthy (2005; 2006), but there are some points of difference (e.g. provide/receive participation relationships as in Hruby (2006) instead of the more common inside/outside participation relationships). Important to notice is that the reengineering of REA focused on a formal specification of the REA ontology. However, the content and theoretical background of REA was not questioned, nor changed. The OWL code for the REA-ontology is available upon request.

In this paper a next step is taken in the REA reengineering process by graphically representing the formally specified REA-ontology using an UML-derived graphical language for representing ontologies. The four-layered modeling metapyramid developed by the Object Management Group (OMG) and the UML profiling mechanism make it possible to extend UML for specific application domains. The Ontology Definition Metamodel (ODM) specification (OMG 2006) recently adopted by OMG uses this extension mechanism in order to support both graphical conceptual modeling and ontology development in several knowledge representation and ontology languages. The ODM specification thus provides a coherent framework for visual ontology creation based on the Meta Object Facility (MOF) (highest level or M3 level in the metapyramid) and UML (M2 level in the metapyramid). The ODM framework is especially useful for formalizing the REA ontology because the resulting representation should be easy to understand by the REA-ontology community whose members are in most cases not familiar with formal representation languages like OWL. Additionally, following this specification also makes sure that some of the known differences between conceptual modeling languages and ontology representation languages are taken into account (de Bruyn et al. 2005).

Accordingly, in this paper we extend our previous work (Gailly and Poels 2007a, 2007b) by using the UML profile for OWL proposed in the ODM specification for the graphical representation of the REA-ontology. This profile respects the structure of the OWL metamodel and reuses standard UML2 notation when the constructs have the same intuitive semantics as OWL (in some cases stereotyped UML constructs are used). For a complete overview of the OWL UML profile we refer to chapter 14 of the ODM specification (OMG 2006). The end result of our OWL-to-UML translation can be found in the Appendix. In what follows we present partial views over the developed UML class diagram in order to facilitate presentation and discussion.

Figure 2 shows a UML class diagram representation of the basic Resource-Event-Agent constellation at the business process level. Economic reciprocity, which would normally be captured by axiomatized duality/reciprocal relations between events, is not included in this view as it would require the inclusion of a mirror-image R-E-A constellation. In the diagram the basic REA concepts `Economic Agent`, `Economic Event` and `Economic Resource`, their type images, and `Commitment` are represented by <<owl:Class>> stereotyped UML classes. A constraint has been added which explicitly shows that these seven classes are disjoint. Something cannot be an instance of more than one class at the same time.

The representation of the relations between the concepts is more complex and different representations have been proposed which all serve a different focus. In figure 2 the relations between the classes are represented by bidirectional associations with a role name at both sides. In fact this means that two inverse OWL objectproperties are defined for the two connected OWL classes. For example, for the binary association between `Economic Resource` and `Economic Event` a `stockflow` OWL objectproperty is defined with as domain an `Economic Resource` and as range an `Economic Event` [1]. The `inverseOfStockflow` OWL objectproperty is the OWL inverseOf objectproperty of the `stockflow` objectProperty.

[insert figure 2]

Notice that Figure 2 looks very similar to the usual UML class diagrams showing the basic REA pattern (as in the papers of Geerts and McCarthy) but REA-experts will certainly remark the absence of the duality and reciprocity relations. These relations are still present in the OWL specification but are specified between specializations of `Economic Event` (see figure 3) and `Economic Commitment` (see Appendix). By specializing the basic REA concepts we add semantics to the REA-ontology representation which is not present in the UML class diagrams that we found in the REA-ontology literature sources, but which is usually described in text. For example figure 3 adds specializations to the `Economic Event` concept and to the `stockflow` association. For OWL classes the OWL subclass construct is used which is represented by a UML specialization. Additionally the OWL UML profile also uses constraints for depicting that a specialization is complete and disjoint.

The specialization of the relations between the classes is less straightforward. In order to represent this more clearly a different approach than in figure 2 is followed to represent objectProperties. In figure 3 the OWL objectProperties are represented by two inverse <<owl:objectProperty>> stereotyped association classes which can be specialized by the UML specialization construct which in this case denotes an OWL subProperty. Important to notice is that figure 3 does not add the restriction that the specialization of the stockflow relation and its inverse are total and complete because this is not supported by OWL 1.0. However future versions of OWL will probably support this feature.

Adding specialization structures to the graphical REA-ontology representation makes it possible to add constraints which were in the original REA sources captured by informally described axioms. For example in figure 3 the multiplicities on the `(inverseOf)inflow` and `(inverseOf)outflow` clearly state that every `Increment Economic Event` and `Decrement Economic Event` must affect one identifiable `Economic Resource`. The other REA axioms are also captured in this new representation by adding multiplicity constraints to the objectProperties. A thorough discussion of the choices that were made can be found in previous work (Gailly and Poels 2007a). The main difference with other REA representations is that we clearly distinguish between axioms that can be defined at the operational level and the knowledge level. For example the duality axiom that defines that all events effecting an outflow must be eventually paired in duality relationship with events effecting an inlow and vice-versa, can only be modeled at a knowledge level because it could for example be possible that in reality the purchase of a product has not yet been paid by the company.

[insert figure 3]

The same approach can be followed for the `participate` relation between an `Economic Event` and an `Economic Agent`. We did not decide to specialize an `Economic Agent` into an `Economic Agent Provider` and an `Economic Agent Receiver` because this it just a temporal distinction. In the REA-ontology an economic agent can be a provider and a receiver for receiver for different events. The `participate` relation can be specialized into a `provide` and `receive` relation. Figure 4 represents these specializations using the OWL UML profile. The used constructs are similar to figure 3 and also make it possible to visually represent the REA axiom that stipulates that every economic event must have always one receiver and one provider. This axiom does not fully comply with the participate axiom in the Geerts and McCarthy (2005; 2006) paper but represents the interpretation of Gailly and Poels (2007a) who changed the axiom because they also followed Hruby (2006) who uses provide and receive relationships instead of inside and outside participate relationships.

[insert figure 4]

V REA as a business modeling ontology

Like already mentioned in the beginning of this paper, ontology-driven business modeling is one of the possible applications of the REA-ontology. Apart from the aforementioned standardization efforts (open-EDI, UN/CEFACT) and the use of the REA-ontology in educations there are few documented accounts of REA ontology-driven business modeling in practice. In this section we investigate how a formal representation of the REA-ontology can be used for the modeling of a simple business process, what the benefits are of using this more formal approach and what problems still will arise when using a formal language like OWL for business modeling. To illustrate our ideas we use the well known knowledge representation framework Protégé, OWL and a simple REA-structured Enterprise schema which was taken from Geerts (2004) and is shown in figure 5.

[insert figure 5]

The first step was introducing in Protégé the OWL specification of the REA-ontology developed in previous work (Gailly and Poels 2007b) and which also corresponds to the graphical representation described in the previous section. An overview of the resulting owl classes and owl objectProperties in the Protégé environment is shown in figure 6 by means of an owlViz and ontoViz visualization which are both popular ontology visualization plugins for Protégé. The owlViz visualization gives an overview of the classification of the REA-concepts in OWL and also indicates the disjointness constraints (by means of a ¬). The ontoViz gives a graphical overview of the OWL objectProperties. In order to keep the visual representation conveniently arranged the graphical representations offer only a partial view and do not include the inverseOf OWL objectproperties. At this stage a Protégé plug-in that uses the UML OWL profile for graphically representing an ontology is not yet available but this seems to be a very useful and easy to realize addition to the ODM project in the future.

[insert figure 6]

The next step consists in using the REA OWL specification to model an enterprise (we take the example of figure 5). The availability in Protégé of the semantic description of the REA-ontology concepts, relations and axioms (as a result of the first step) guides the development of the enterprise schema in Protégé. The Protégé OWL individual editor is used for this purpose and provides for every REA-concept a specific form which makes it very straightforward to instantiate this concept and make sure that the mandatory and optional relations with other concept instantiations are included. For instance, figure 7 shows the Protégé form for an increment economic event. For the `Increment Economic Event` *purchase* the `Provider` (*vendor*), the `Receiver` (*receiving clerk*), the `dual Decrement Economic Event` (*cash disbursement*) and the `Economic Resource` which is increased (*inventory*) can be specified. Important to notice is that this form also takes into account the axioms by stipulating that it is necessary (red box) to provide a `Receiver`, a `Provider` and an `Economic Resource`.

The main advantage of using Protégé for business modeling is that a formal specification of the enterprise schema is now available that can be used in different application contexts. For instance in the REA-application described by Geerts (2004) the REA-ontology, the enterprise schema and the enterprise data can be easily implemented in OWL which is a much richer representation language than the combination XML-schema, XSLT and XML. Also in the MDA approach to software engineering the formal specification in OWL of the enterprise schema could act as Computation Independent Model (CIM) which is a representation of the problem domain and which can be developed by instantiating a CIM meta-model which corresponds to the REA business domain ontology (Assmann et al. 2006). In our future research we will further explore these opportunities.

[insert figure 7]

A major issue when using ontologies for modeling is the difference in interpretation of some basic language characteristics between ontology languages and information modeling languages (de Bruyn et al. 2005). In future research we plan to further investigate these differences and more critically evaluate the appropriateness of using formal languages like OWL business modeling purposes.

VI Conclusions

After 25 years of REA research one the most challenging research directions will be improving the usability of the REA-ontology. This paper groups three related research efforts which all use existing ontology research outcomes in order make the REA-ontology more applicable. Together they all contribute to position the REA-ontology as a business domain ontology.

First the current state of the REA-ontology was analyzed by using three well-known ontology classification frameworks. The classification based on the subject of the conceptualization is very useful for providing a definition for business domain ontologies and helped us in comparing the REA-ontology to other existing business domain ontologies. Next, the richness of the internal structure of the REA-ontology was examined. At this stage we can conclude that the REA-ontology can be considered as a business domain ontology which can be, and has been specialized in different applications ontologies. The current richness of the internal structure has been improved a great deal by the REA-developers but in order to make this internal structure more generally accepted and less confusing a formal specification is needed. The last classification schema takes a slightly different approach by focusing on the REA-ontology applications and as such gives further insight into the intended use of the REA-ontology. Different applications of the REA-ontology have been proposed and it this stage the educational use of REA is without any doubt the most successful application. However the REA ontology offers additional opportunities for business modeling, knowledge representation and retrieval, and ontology-driven information systems (engineering), all of which require additional research to put REA into practice. The review of the currently known REA applications also stresses the importance of having a shared machine-readable specification of the REA ontology ready and available.

A second research effort was the development of a REA-ontology specification in a graphical representation format which is more expressive than the currently available REA specifications in UML or ER and that can easily be formalized without adding considerable complexity for business experts. Like concluded in the previous part, one of the critical success factors of the REA-ontology is the availability of a formal specification which can be easily used in applications and which is generally accepted by the REA-community. A formal specification offers a lot of benefits compared to a specification in text or a less formal graphical conceptual modeling language but has as major drawback that the formal specification is hard to understand and interpret. This makes it hard to come to a general accepted formalization of the REA-ontology. This paper uses a compromise between the formal ontology language OWL and the graphical conceptual modeling language UML by using the OWL UML profile which is part of the ODM specification recently adopted by OMG.

Finally this paper also contains a preliminary proof of concept to persuade the REA research community of the usefulness of having a formal specification. A formal specification in OWL of the REA-ontology was imported in the Protégé knowledge representation environment and afterwards used for representing a simple REA-structured enterprise model. This example clearly shows that with ontology-driven business modeling it is very easy to specify a correct business model in a formal language that is machine-readable and that can subsequently be used for knowledge representation, ontology-driven information systems engineering.

In future research we plan to use the formal representation of the REA-ontology for the development of specific proofs of concept for the different REA-ontology application domains. The success of these applications depends in al large extent on the existing formal specification of the REA-ontology. As a result we also plan to further improve the REA-ontology by repeating the ontology reengineering process proposed in Gailly and Poels (2007a; 2007b) . The new graphical representation of REA proposed in this paper will certainly facilitate this process and we hope that this unified and uniform REA representation will enable the REA research community to contribute to the shared understanding, dissemination and further improvement of this Silver Anniversary but evergreen enterprise ontology.

---

[1] We could also have chosen to do it differently and use `Economic Event` as domain and `Economic Resource` as range. The choice we made was rather arbitrary, but so would be the other choice as the term 'stockflow' does not indicate the intended direction.

References

Allen, G. N., and S. T. March. 2006. The effects of state-based and event-based data representation on user performance in query formulation tasks. *Mis Quarterly* 30 (2):269-290.

Assmann, U., S. Zchaler, and G. Wagner. 2006. Ontologies, Meta-Models, and the Model-Driven Paradigm. In *Ontologies for Software Engineering and Software Technology*, edited by C. Calero, F. Ruiz and M. Piattini.

Bialecki, A. 2001. REA ontology. http://www.getopt.org/ecimf/contrib/onto/REA/,

Borch, S. E., J. W. Jespersen, J. Linvald, and K. Osterbye. 2003. A Model Driven Architecture for REA based systemsPresented at Proceedings of the Workshop on Model Driven Architecture: Foundations and Applications, Enshede, The Netherlands.

Borst, W. N. 1997. Construction of Engineering Ontologies. Centre for Telematica and Information Technology,

Chou, C.-C. 2006. Using ontological methodology in building the accounting knowledge model – REAPPresented at 2006 AAA mid-year meeting - 2006 AI/ET Workshop

Cognitive Science Laboratory. 2006. WordNet: Princeton University.

de Bruyn, j., R. Lara, A. Polleres, and D. Fensel. 2005. OWL DL vs. OWL Flight: Conceptual Modeling and Reasoning for the Semantic Web. In *World Wide Web Conference (WWW 2005)*. Chiba, Japan: ACM.

Dunn, C. L., J. O. Cherrington, and A. S. Hollander. 2005. *Enterprise Information Systems: A Pattern Based Approach*: McGraw-Hill.

eCl@ss. 2007. *eCl@ss classification and product description* [cited 2007]. Available from http://www.eclass.de/.

Fensel, D. 2001. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*: Springer-Verslag.

Fox, M. S. 1992. The TOVE Project: A Common-sense Model of The Enterprise. In *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, edited by F. Belli and F. Radermacher. Berlin, Germany: Springer-Verslag, 25-24.

Gailly, F., and G. Poels. 2007a. Ontology-driven Business Modelling: Improving the Conceptual Representation of the REA-ontology. FEB Working paper series, FEB WP 07/444, Faculty of Economics and Business Administration, Ghent University, http://www.feb.ugent.be/fac/research/WP/Papers/wp_07_444.pdf.

———. 2007b. Towards Ontology-driven Information Systems: Redesign and Formalization of the REA Ontology. Lecture Notes in Computer Science 4439, Presented at 10th International Conference on Business Information Systems (BIS 2007), Poznan, Poland.

Gasevic, D. 2006. *Model driven architecture and ontology development*. 1st ed. New York: Springer.

Geerts, G. 2004. An XML Architecture for Operational Enterprise Ontologies. *Journal of Emerging Technologies in Accounting* 1:73-90.

Geerts, G., and W. E. McCarthy. 1999. An Accounting Object Infrastructure for Knowledge Based Enterprise Models. *IEEE Intelligent Systems and Their Applications* 14 (4):89-94.

———. 2005. The Ontological Foundation of REA Enterprise Information Systems.

———. 2006. Policy-Level Specification in REA Enterprise Information Systems. *Journal of Information Systems* Fall.

Geerts, G. L., and W. E. McCarthy. 2000. Augmented Intensional Reasoning in Knowledge-Based Accounting Systems. *Journal of Information Systems* 14 (2):127.

———. 2002. An Ontological Analysis of the Economic Primitives of the Extended-REA Enterprise Information Architecture. *International Joural of Accounting Information Systems* 3 (1):1-16.

Geerts, G. L., B. A. Waddington, and C. E. White. 2002. Stevie: A Dynamic, Between-Instructor Collaborative Internet Tool for Learning Cardinalities. *Journal of Information Systems* 16 (1):75-89.

Gómez-Pérez, A., M. Fernández-López, and O. Corcho. 2004. *Ontological Engineering*: Springer-Verslag.

Gordijn, J. 2002. Value based requirements engineering: Exploring innovative e-commerce ideas, Vrije Universiteit Amsterdam.

Gruber, T. R. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition* 5:199-220.

Guarino, N. 1998. Formal Ontology and Information SystemsPresented at Proceedings of FOIS'98, Trento, Italy.

Guarino, N., and P. Giaretta. 1995. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing.*, edited by N. Mars. Amsterdam: IOS press, 25-32.

Haugen, R., and W. E. McCarthy. 2000. REA: A Semantic Model for Internet Supply Chain CollaborationPresented at Proceedings of the Business Objects and Component Design and Implementation Workshop VI: Enterprise Application Integration.

Hepp, M. 2007. *eClassOWL* 2006 [cited 2007]. Available from http://www.heppnetz.de/eclassowl/.

Hepp, M., J. Leukel, and V. Schmidt. 2005. A Quantitative Analysis of eCl@ss, UNSPSC, eOTD, and RNTD: Content, Coverage, and Maintenance. In *IEEE International Conference on e-Business Engineering (ICEBE 2005)*. Beijing, China, 572-581.

Hofreiter, B., C. Huemer, P. Liegl, R. Schuster, and M. Zapletal. 2006. UN/CEFACT'S Modeling Methodology (UMM): A UML Profile for B2B e-Commerce. In *Advances in Conceptual Modeling - Theory and Practice (ER 2006)*. Tucson, Arizona, VS: Springer, 19-31.

Hruby, P. 2005. Role of Domain Ontologies in Software FactoriesPresented at OOPSLA 2005 Workshop on Software Factories, San Diego, California.

———. 2006. *Model-driven design using business patterns*. New York: Springer.

ISO. 2006. Information technology -- Business Operational View -- Part 4: Business transaction scenarios -- Accounting and economic ontology (ISO/IEC 15944-4): ISO.

Lara, R., I. Cantador, and P. Castells. 2007. XBRL taxonomies and OWL ontologies for investment funds. In *Advances in Conceptual Modeling - Theory and Practice*, edited by J. F. Roddick, R. Benjamins, S. S.-S. Cherfi, R. Chiang, C. Claramunt, R. Elmasri, F. Grandi, H. Han, M. Hepp, M. Lytras, V. B. Misic, G. Poels, I.-Y. Song, J. Trujillo and C. Vangenot: Springer.

Lassila, O., and D. L. McGuiness. 2001. The Role of Frame-Based Representations on the Semantic Web. In *Technical Report*. Stanford, California: Knowledge System Laboratory, Stanford University.

McCarthy, W. E. 1982. The REA Accounting Model: A Generalized Framework for Accounting Systems in A Shared Data Environment. *The Accounting Review* july:554-578.

———. 2003. The REA Modelling Approach to Teaching Accounting Information Systems. *Issues in Accounting Education* 18 (4):427-441.

OMG. 2006. Ontology Definition Metamodel: OMG Adopted Specification (ptc/06-10-11): Object Management Group.

Osterwalder, A. 2004. The Business Model Ontology - a proposition in a design science approach, Ecole des Hautes Etudes Commerciales, University of Lausanne, Lausanne.

RosettaNet. 2007. *RosettaNet Technical Dictionary (RTND)* [cited 2007]. Available from http://www.rosettanet.org/.

U.S. Census Bureau. 2007. *North American Industry Classification System (NAICS)* [cited 2007]. Available from http://www.census.gov/epcd/www/naics.html.

United Nations. 2007. *The United Nations Standard Products and Services Code* 2007 [cited 2007]. Available from http://www.unspsc.org/.

Ushold, M., and R. Jasper. 1999. A Framework for Understanding and Classifying Ontology ApplicationsPresented at IJCAI Workshop on Ontologies and Problem-Solving Methods.

Ushold, M., M. King, S. Moralee, and Y. Zorgios. 1998. The Enterprise Ontology. *The Knowledge Engineering Review: Special Issue on Putting Ontologies to Use* 13 (1):31-89.

van Heijst, G., A. T. Schreiber, and B. J. Wielinga. 1997. Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies* 46 (2-3):183-292.

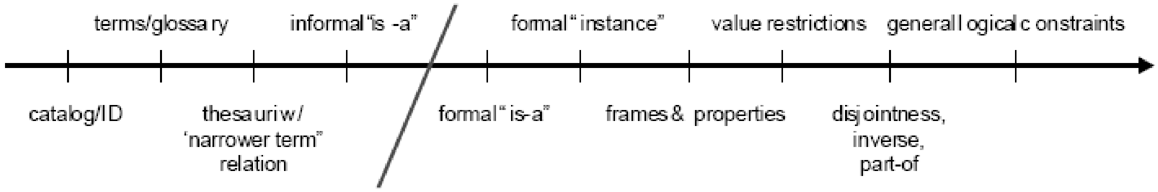


Figure 1: Classification of ontologies based on the richness of the internal structure (Lassila and McGuiness 2001)

Ontology application	Intended Purpose or benefits	Role of the ontology	Actors	References
Education	The ontology acts as a reference ontology for understanding business domain which improves communication between teacher and student	• conceptual framework for teaching accounting information systems • meta-model to generate reference models for different types of business (or transaction) cycles	• Business/accounting information system professors • Students	(Dunn et al. 2005; McCarthy 2003)
Model-driven design	Using the REA-ontology makes the development of the application more straightforward and software applications based on REA contain more and more correct business knowledge.	• modeling language • meta-model • reference ontology	Participants in the system development process	(Borch et al. 2003; Hruby 2005, 2006)
Supply chain collaboration	Using the REA-ontology makes it easier to establish supply chain collaboration via internet technologies (e-collaboration)	The REA-ontology is used as an ontological framework for specifying the concepts and relations involved in business transactions and scenarios.	Trading partners	(Haugen and McCarthy 2000; Hofreiter et al. 2006; ISO 2006)
Knowledge representation	By adding semantics to the business applications additional information can be more easily extracted from the business.	The different systems use the operational REA-ontology at run-time for adding semantics to the enterprise schema and data	Systems developers, auditors	(Geerts 2004; Geerts and McCarthy 2000)

Table 1:REA-ontology applications

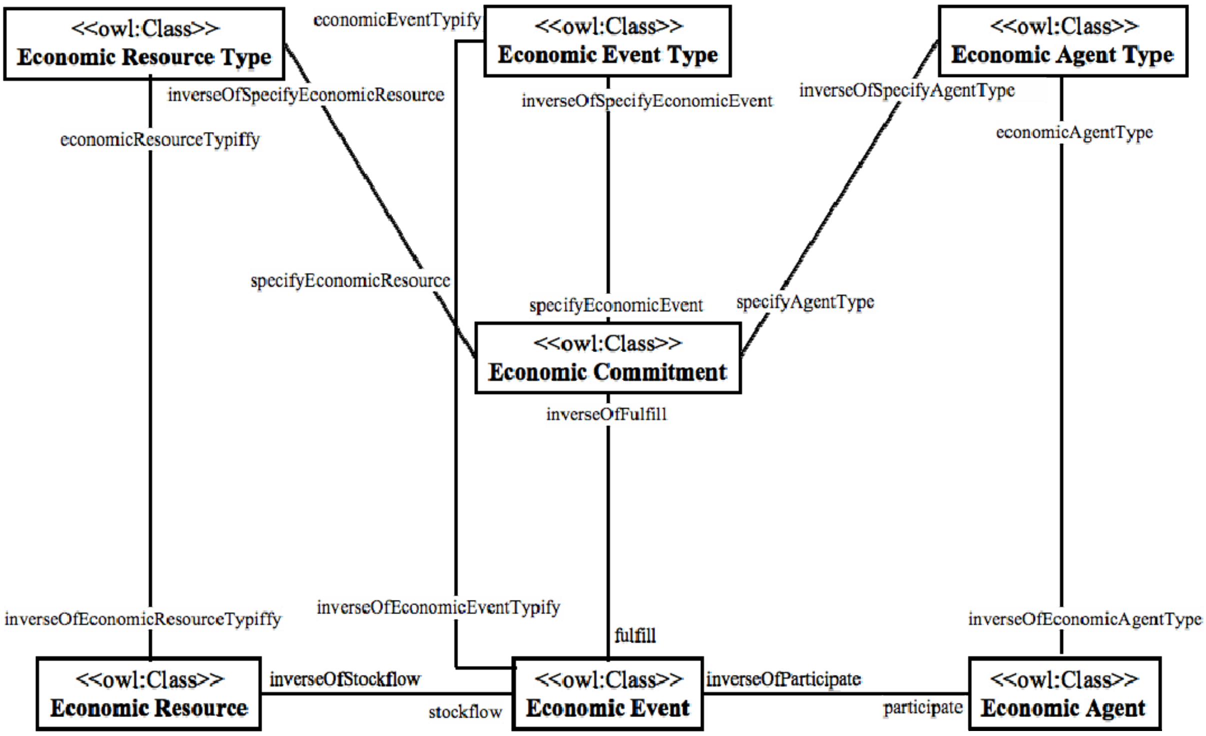


Figure 2: Basic R-E-A constellation representation with OWL UML profile

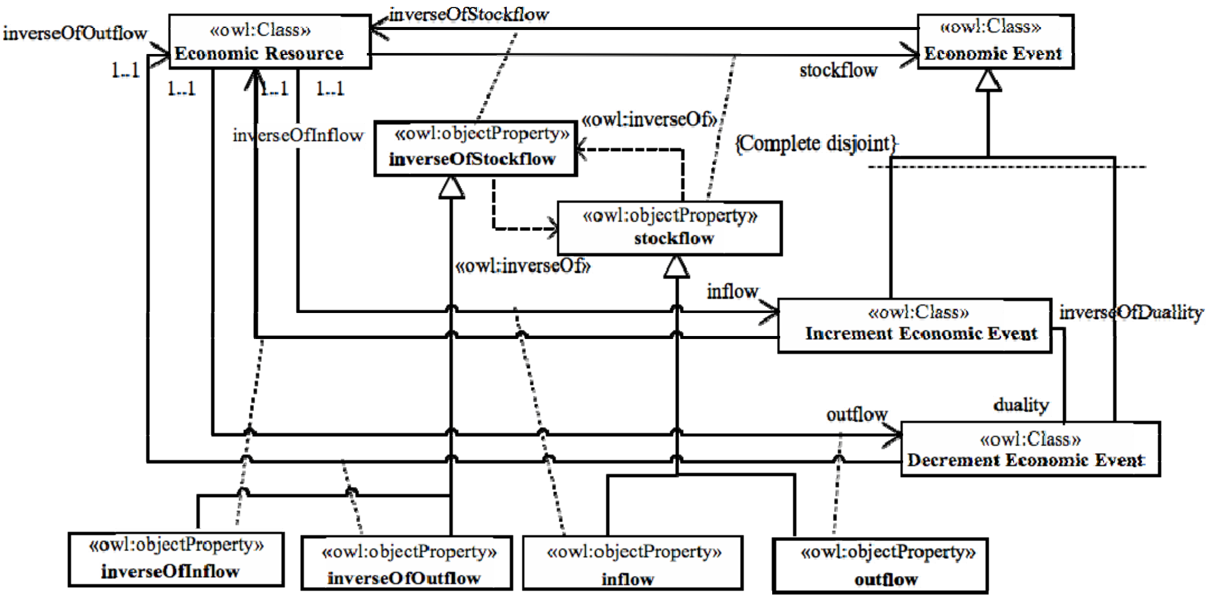


Figure 3: `Economic Event` and `stockflow` specialization representation with OWL UML profile

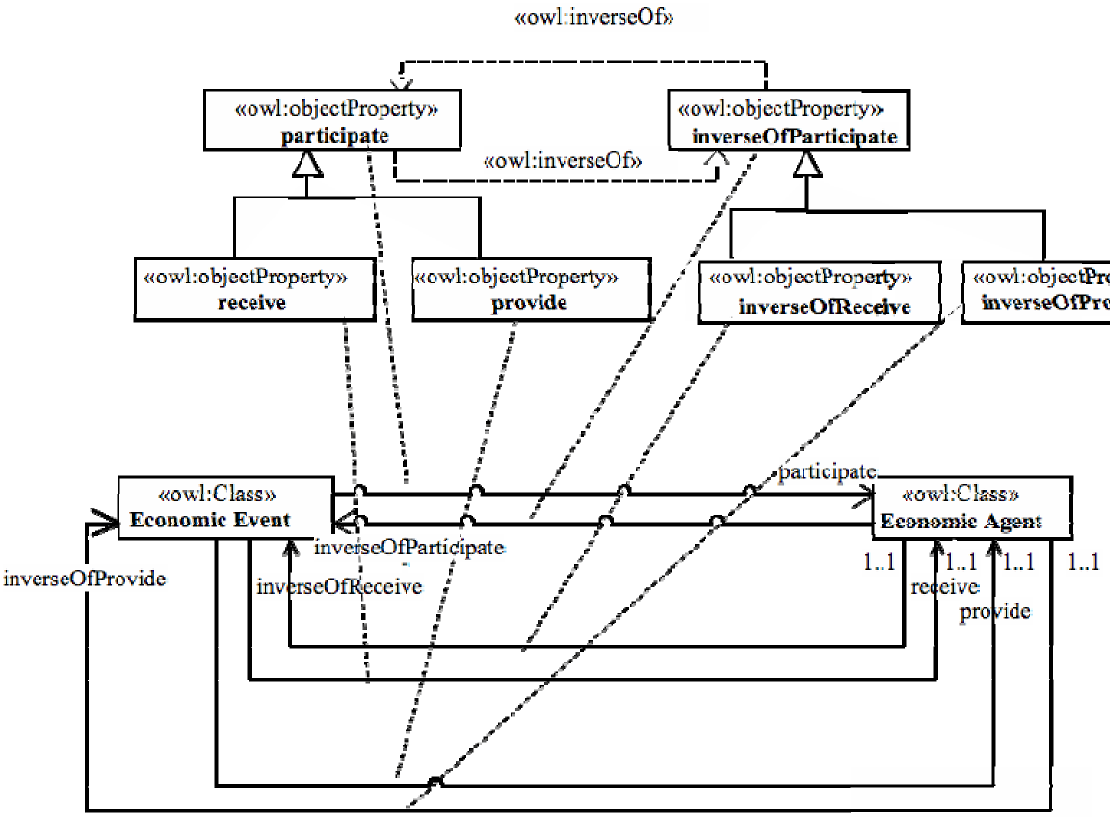


Figure 4: `participate` specialization representation with OWL UML profile

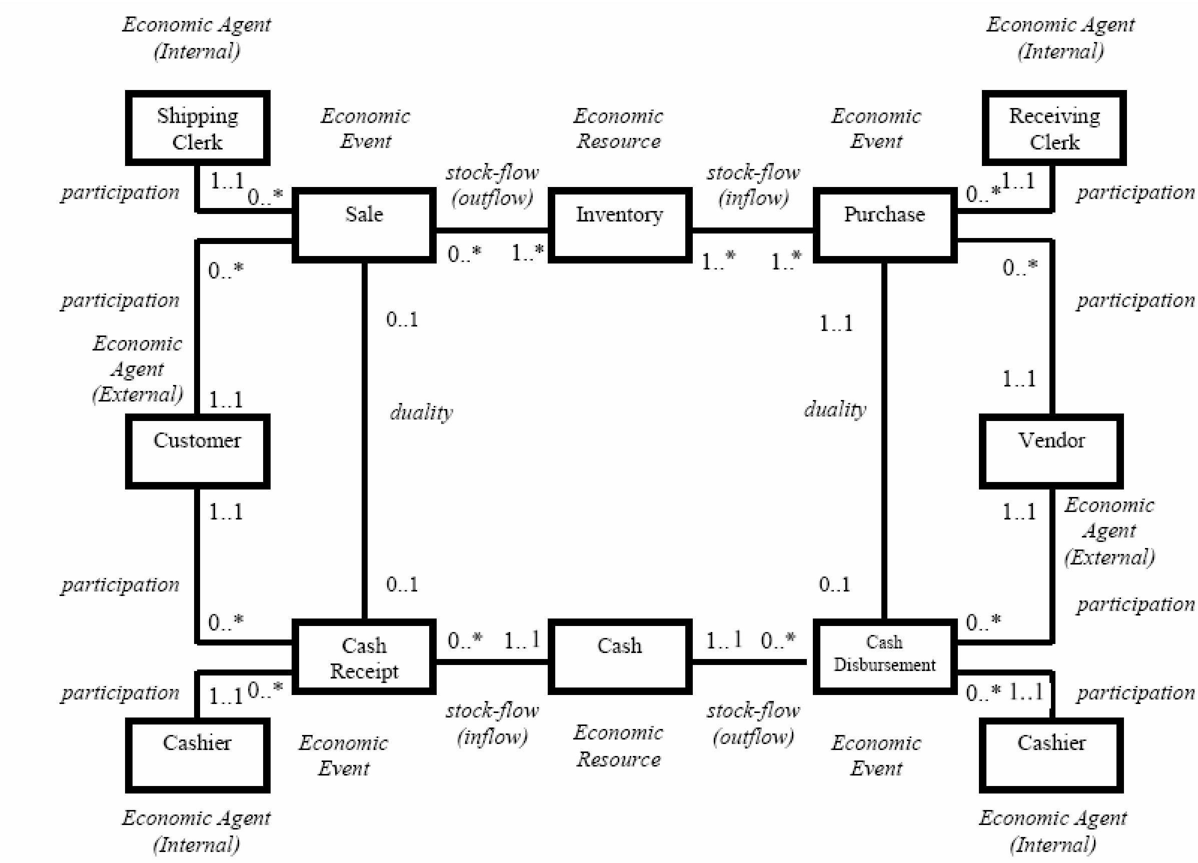
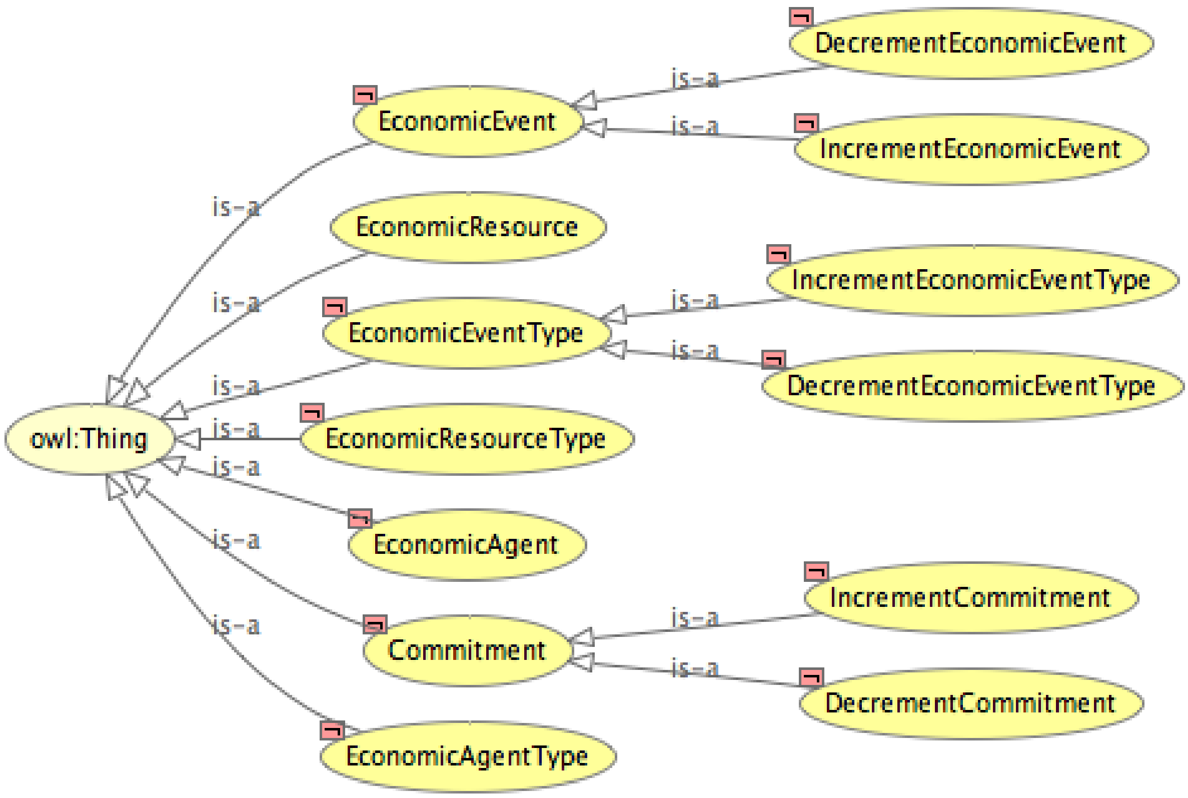
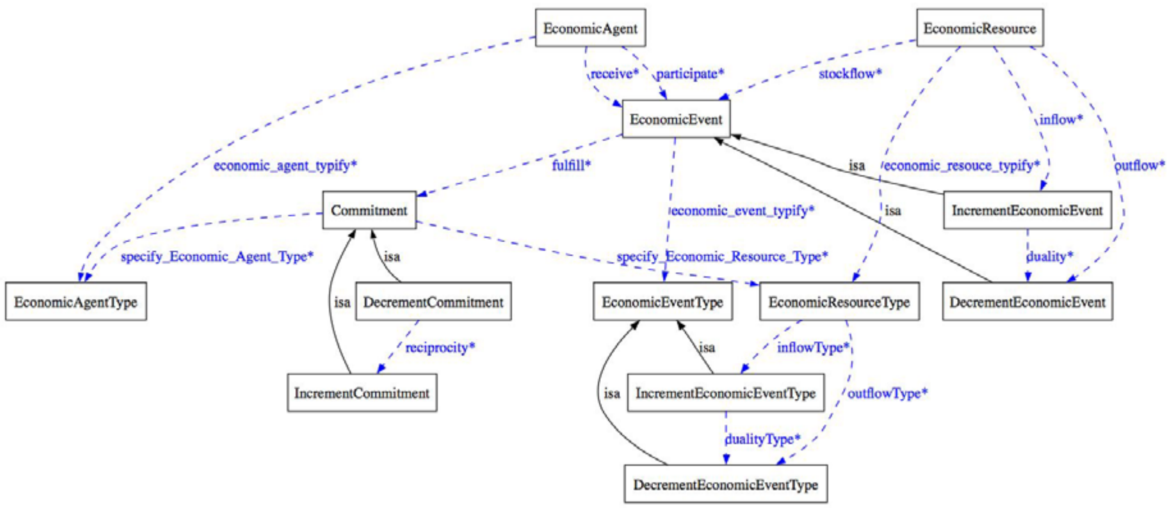


Figure 5: REA-structured Enterprise Schema (Geerts 2004)



(a)



(b)

Figure 6: OwlViz (a) and OntoViz (b) visualization REA-ontology (partial view)

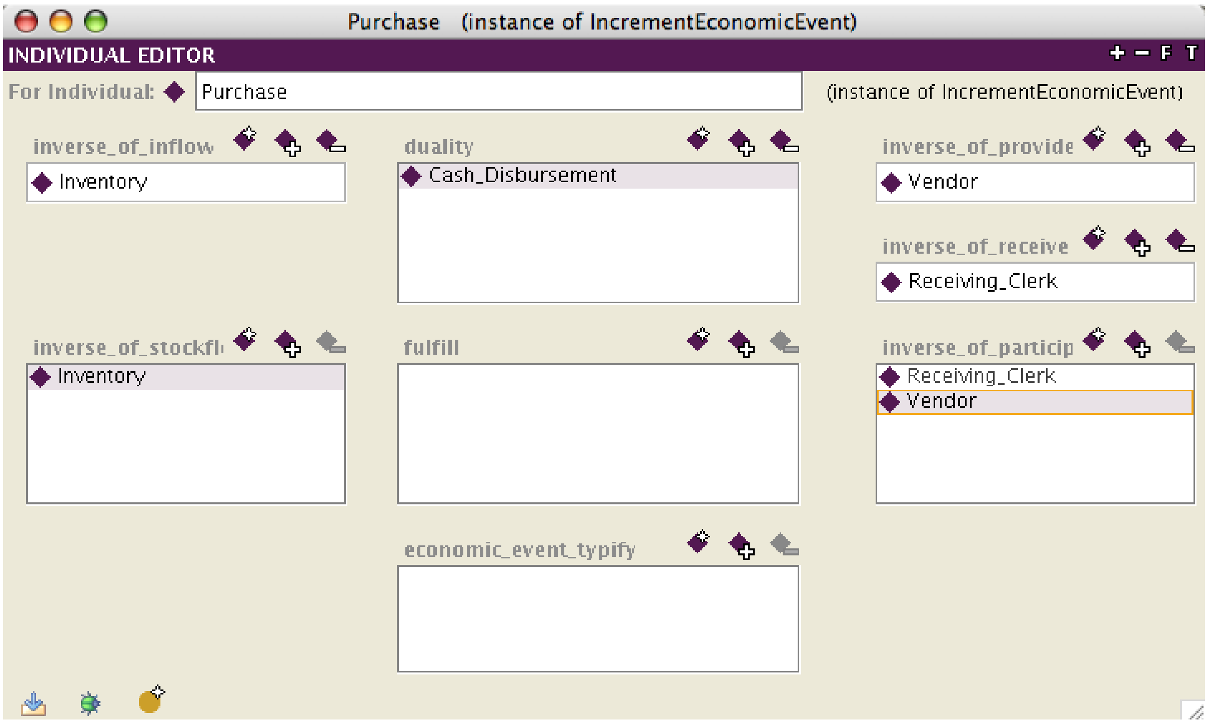
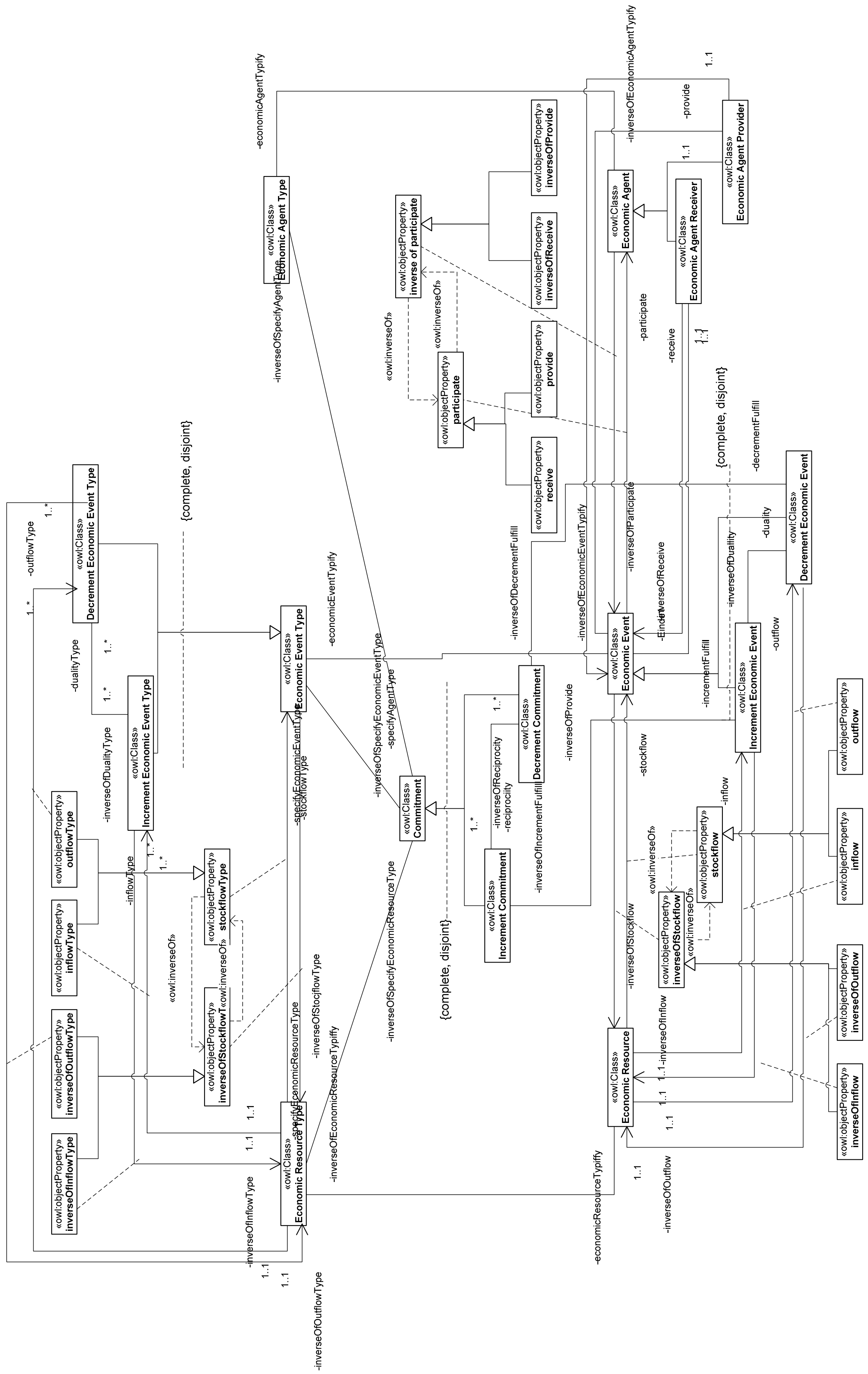


Figure 7: Protégé form for an increment economic event

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
   <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
   <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
   <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
   <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
   <!ENTITY REA "http://XXX/REAontology/R-REA-ontology.owl#" >
]>
<rdf:RDF xmlns="http://users.ugent.be/~fgailly/REA-ontology/Example#"
    xml:base="http://users.ugent.be/~fgailly/REA-ontology/Example"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:REA="http://users.ugent.be/~fgailly/REAontology/R-REA-ontology.owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#">
<owl:Ontology rdf:about="">
<owl:imports  rdf:resource="http://users.ugent.be/~fgailly/REAontology/R-REA-ontology.owl"/>
</owl:Ontology>
<REA:Economic_Resource rdf:ID="Cash">
   <REA:inflow rdf:resource="#Cash_Receipt"/>
</REA:Economic_Resource>
<REA:Decrement_Economic_Event rdf:ID="Cash_Disbursement">
   <REA:inverse_of_provide rdf:resource="#Cashier"/>
   <REA:inverse_of_receive rdf:resource="#Vendor"/>
</REA:Decrement_Economic_Event>
<REA:Increment_Economic_Event rdf:ID="Cash_Receipt">
   <REA:inverse_of_provide rdf:resource="#Customer"/>
   <REA:inverse_of_receive rdf:resource="#Cashier"/>
   <REA:inverse_of_inflow rdf:resource="#Cash"/>
</REA:Increment_Economic_Event>
<REA:Economic_Agent rdf:ID="Cashier">
   <REA:receive rdf:resource="#Cash_Receipt"/>
   <REA:provide rdf:resource="#Cash_Disbursement"/>
</REA:Economic_Agent>
   <REA:Economic_Agent rdf:ID="Customer">
   <REA:receive rdf:resource="#Sale"/>
   <REA:provide rdf:resource="#Cash_Receipt"/>
</REA:Economic_Agent>
<REA:Economic_Resource rdf:ID="Inventory">
   <REA:outflow rdf:resource="#Sale"/>
   <REA:inflow rdf:resource="#Purchase"/>
</REA:Economic_Resource>
<REA:Increment_Economic_Event rdf:ID="Purchase">
   <REA:inverse_of_provide rdf:resource="#Vendor"/>
   <REA:inverse_of_receive rdf:resource="#Receiving_Clerk"/>
   <REA:inverse_of_inflow rdf:resource="#Inventory"/>
</REA:Increment_Economic_Event>
<REA:Economic_Agent rdf:ID="Receiving_Clerk">
   <REA:receive rdf:resource="#Purchase"/>
</REA:Economic_Agent>
<REA:Decrement_Economic_Event rdf:ID="Sale">
   <REA:inverse_of_provide rdf:resource="#Shipping_Clerk"/>
   <REA:inverse_of_receive rdf:resource="#Customer"/>
   <REA:inverse_of_outflow rdf:resource="#Inventory"/>
</REA:Decrement_Economic_Event>
<REA:Economic_Agent rdf:ID="Shipping_Clerk">
   <REA:provide rdf:resource="#Sale"/>
</REA:Economic_Agent>
<REA:Economic_Agent rdf:ID="Vendor">
   <REA:receive rdf:resource="#Cash_Disbursement"/>
   <REA:provide rdf:resource="#Purchase"/>
</REA:Economic_Agent>
</rdf:RDF>
```

Figure 8: OWL specification of an example REA-structured Enterprise Schema

Appendix


«owl:Class» Economic Agent Type
«owl:Class» Economic Agent
«owl:Class» Economic Agent Receiver
«owl:Class» Economic Agent Provider
«owl:objectProperty» inverse of participate
«owl:objectProperty» inverseOfProvide
«owl:objectProperty» inverseOfReceive
«owl:objectProperty» participate
«owl:objectProperty» provide
«owl:objectProperty» receive
«owl:Class» Decrement Economic Event Type
«owl:Class» Increment Economic Event Type
«owl:Class» Economic Event Type
«owl:Class» Commitment
«owl:Class» Decrement Commitment
«owl:Class» Increment Commitment
«owl:Class» Economic Event
«owl:Class» Decrement Economic Event
«owl:Class» Increment Economic Event
«owl:Class» Economic Resource Type
«owl:Class» Economic Resource
«owl:objectProperty» outflowType
«owl:objectProperty» inflowType
«owl:objectProperty» inverseOfOutflowType
«owl:objectProperty» inverseOfInflowType
«owl:objectProperty» stockflowType
«owl:objectProperty» stockflow
«owl:objectProperty» inverseOfStockflow
«owl:objectProperty» outflow
«owl:objectProperty» inflow
«owl:objectProperty» inverseOfOutflow
«owl:objectProperty» inverseOfInflow
{complete, disjoint}