JOURNAL OF INFORMATION SYSTEMS
Vol. 22, No. 2
Fall 2008
pp. 219–248

Positioning and Formalizing the REA Enterprise Ontology

Frederik Gailly
Ghent University

Wim Laurier
Ghent University

Geert Poels
Ghent University

ABSTRACT: Recent Resource, Event, Agent (REA) research has focused on defining and theoretically justifying the ontology's contents. Here, we elaborate on more practical issues related to REA. First, we classify REA and its applications using ontology classification schemes and application frameworks. This analysis clarifies REA's application potential but also reveals weaknesses that may impede its operationalization. Next, we propose a new REA ontology specification that uses a Unified Modeling Language (UML) profile for graphically representing ontologies. This new specification is more complete and precise than previously available specifications, without compromising understandability. It can easily be transformed into a machine-readable representation for automatic processing, which is a prerequisite for the successful application of REA in business modeling, software engineering, knowledge representation, and interoperability creation. The paper ends with a proof of concept application in which a formal Ontology Web Language (OWL) specification of REA is fed into the Protégé knowledge representation tool and subsequently used for the development of an enterprise schema.

Keywords: REA; ontology; ontology applications; OWL; UML; business modeling; knowledge representation.

Data Availability: REA Enterprise ontology OWL formalization URL: http://www.managementinformation.ugent.be/REAontology/REAontology.owl.

I. INTRODUCTION

In this paper, we respond to David et al.'s (2002) call for more design science research in accounting information systems by further developing and evaluating the Resource, Event, Agent (REA) enterprise ontology. The origin of the REA ontology lies in the semantic data model for accounting that was proposed in the early eighties by McCarthy (1982). This accounting information model was extended in the late nineties by Geerts and McCarthy (1999; 2002) into a comprehensive enterprise information architecture.

We acknowledge the helpful comments of Mette Jaquet, the participants of the REA-25 conference, the three anonymous reviewers, and Guido Geerts, Guest Editor of the JIS Special Issue on REA.

Geerts and McCarthy (1999, 2002) used notions from domain-specific ontology research to present REA as an "ontology" for enterprises. The most cited definition of an ontology is the definition by Gruber: "an ontology is an explicit specification of a conceptualization" (Gruber 1993, 199). A conceptualization is an intentional 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, thereby 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).

Since the beginning of the millennium, a considerable amount of research has been conducted in different sub-domains of computer science which further explores the development, use, evaluation, etc. of domain-specific ontologies. Our research intends to evaluate and further develop the REA ontology as an enterprise ontology by taking into account generally accepted ontology research. The ultimate goal of our research is a formal REA ontology specification that can readily be operationalized, using the appropriate representation formats and transformations, for use in a variety of enterprise ontology applications like business modeling, model-driven system development, system interoperability creation, and business knowledge storage and retrieval.

David et al. (2002) provide an overview of REA design science research and classify the contributions made by this research using the Information Technology (IT) research framework of March and Smith (1995). The March and Smith IT research framework distinguishes two dimensions: research activities and research outputs. The research activities include both design science research activities (build and evaluate) and natural science research activities (theorize and justify). The different possible research outputs are constructs, models, methods, and instantiations.

The two REA design science contributions that we intend with this paper are indicated in Table 1. As we wish to position the REA ontology as an enterprise ontology, we first evaluated the currently available REA ontology specifications (constructs and models) and REA ontology applications (instantiations) using ontology classification schemes and frameworks that have been proposed in ontology research. This analysis enables us to determine where the REA ontology currently stands and what its deficiencies are as a business domain ontology. It also helps identifying the future REA design science activities that are required to address these deficiencies. As a result of our evaluation, we will stress the importance of further developing a generally accepted formal specification of the REA ontology in order to improve its applicability in various enterprise ontology application scenarios.

The second contribution of the paper is already a step toward a more formally specified REA ontology. We used existing ontology engineering principles and technologies to restructure and formalize the current REA ontology specification in order to enhance its applicability. Our newly proposed 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 easily be transformed into a machine-readable representation.

The paper is structured as follows: the next section classifies the REA ontology based on two ontology quality/classification dimensions: the richness of its internal structure and the subject of its conceptualization. In order to further clarify the intended position of the REA ontology as a business domain ontology, Section II also analyses some recent REA

TABLE 1
Research Contributions

	Evaluate	**Build**
Constructs Models Methods	Research Contribution 1a: REA ontology classification based on: • richness of internal structure • subject of conceptualization	Research Contribution 2a: Developing a formal REA ontology specification
Instantiations	Research Contribution 1b: REA application classification based on: • Guarino IS ontology framework • ontology application framework (Ushold and Jasper)	Research Contribution 2b: Business modeling proof of concept application of the formally specified REA ontology

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, Section III presents a new formal specification of the REA ontology, represented using the UML profile for the Web Ontology Language (OWL). Section IV illustrates how this formal specification of the REA ontology in OWL can be used for business modeling which is one of the possible applications of the REA ontology. Section V ends with conclusions and future work.

II. CLASSIFYING THE REA ENTERPRISE ONTOLOGY AND ITS APPLICATIONS

The first research goal of this paper is positioning the REA ontology as an enterprise ontology. We start this section describing the ontology classification schemes and application frameworks that will be applied to the REA ontology and REA ontology applications.

Ontology Classification Schemes and Frameworks for Ontology Applications

Determining the position of a proposed ontology in the "ontology landscape" can be done conceptually or by studying existing ontology applications and investigating how the ontology is actually used in practice. For the conceptual point of view, different classification or quality evaluation schemes for ontologies have been proposed (a good overview can be found in Gómez-Pérez et al. [2004]). As the overall focus of our research project is on improving the applicability of the REA ontology, the most relevant classification or evaluation dimensions are those that relate to an ontology's ability to be applied in practice.

The ontology evaluation framework that we use is the framework proposed by Burton-Jones et al. (2005) who adopted the semiotic framework of Stamper et al. (2000), which is a theoretical model of quality derived from linguistics and organizational semiotics principles. Guided by the Burton-Jones et al. (2005) framework, we selected two classification/evaluation dimensions: the subject of the REA ontology's conceptualization and the richness of its internal structure.

The subject of conceptualization dimension determines to what extent the REA ontology can be used as a domain ontology for enterprises. It can be considered as an attribute of the pragmatic quality dimension of the Burton-Jones et al. (2005) ontology quality framework. Pragmatic quality refers to an ontology's usefulness for users and their agents, irrespective of syntax and semantics. The richness of the internal structure determines what the expressiveness of the REA ontology specification is, which is an indicator of the syntactic quality of the ontology. The syntactic quality dimension measures the quality of the ontology according to the way it is written.

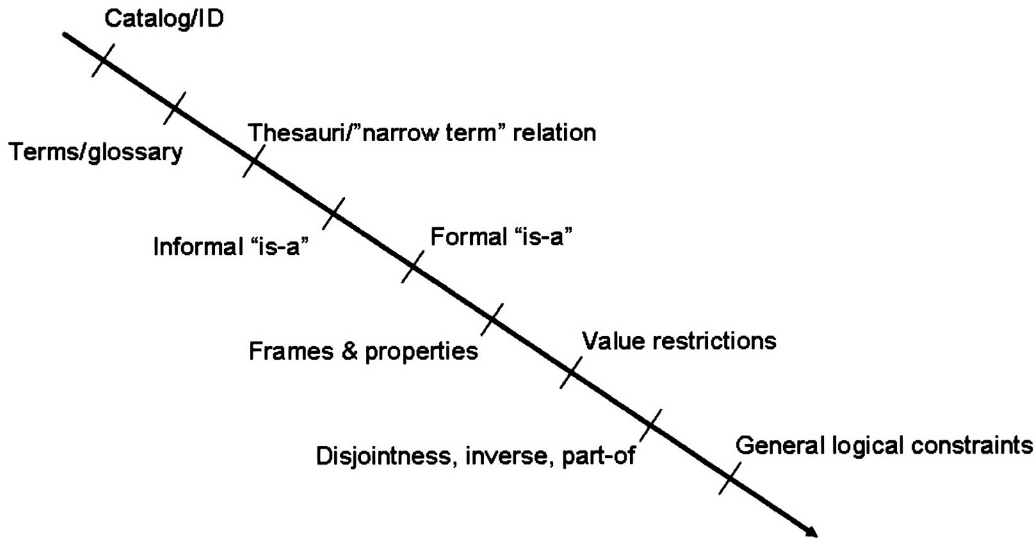
The Burton-Jones et al. (2005) framework contains two additional quality dimensions: the semantic and social quality dimensions. The semantic quality dimension evaluates the meaning of the different concepts of the ontology by referring to a more general top-level ontology as the evaluation benchmark. The social quality dimensions reflect the extent to which the ontology is used in practice. The REA ontology's social quality will be evaluated by means of the ontology application frameworks described further in this sub-section. At this stage, we did not evaluate the semantic quality of the REA ontology because of our focus on applicability. We acknowledge that semantic quality may affect applicability, but the primary aim of our current research is to make the REA ontology "as it stands" more applicable, without making fundamental changes to the ontology's content.

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. *Representation ontologies* are languages that have been specifically proposed for specifying ontologies. *Top-level ontologies* (also called *upper-level* or *core ontologies*) describe very general concepts that are reusable across domains, i.e., top-level ontologies are not domain-specific. *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). Given the existence of well-established top-level ontologies, specifying domain ontologies as specializations of these top-level ontologies helps assuring the (semantic) quality of the domain ontologies. Finally, in order to use domain ontologies in actual implementations, *application ontologies* that fine-tune the domain ontology to a specific application are needed. Application ontologies add specificities to a domain ontology which are only relevant for the application considered, but are not shared across the entire domain.

The second ontology classification dimension that will be applied to the REA ontology 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 ontologies in a continuous line. This spectrum was developed by a panel of top-level ontology researchers (Welty et al. 1999) and provides a semantic richness or expressiveness path for ontologies. Ontologies that incorporate more of the listed constructs possess a richer internal structure. Depending on the type of the intended application, the ontology needs to be more or less expressive and an ontology language that is sufficiently formal to express the required level of expressiveness is needed to specify the ontology.

For the evaluation of the REA ontology applications (social quality), two frameworks will be used. The Uschold and Jasper (1999) framework identifies different application scenarios which are characterized by five key dimensions: intended purpose or benefits of the application, role of the ontology, actors required to implement the scenario, supporting technologies, and maturity level. The Uschold and Jasper framework is very general and can be used to describe ontology applications in different types of domains.

FIGURE 1
Classification of Ontologies Based on the Richness of the Internal Structure (Lassila and McGuiness 2001)



Additionally we will use the framework of Guarino (1998) which classifies ontology applications specifically for information systems. Guarino distinguishes two dimensions that determine the impact ontologies can have on information systems:

- A temporal dimension which makes the distinction whether an ontology is used at development time or at run-time.
- A structural dimension which determines which of three main Information System (IS) components—being application programs, information resources like database and/or knowledge bases, and user interfaces—is affected by the use of the ontology.

Classification of the REA Ontology

The REA ontology is not intended as a *knowledge representation ontology* (or ontology language) for specifying other ontologies (although its use as a domain-specific modeling language for specifying application models has been explored by Hruby [2006]). 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. Moreover, Geerts and McCarthy (2002) have used the SOWA classification of ontological categories as a top-level ontology for providing definitions for the different REA-concepts and relations. This means that the REA ontology can be considered as a specialization of the SOWA ontology.

Based on the subject of the conceptualization, the REA ontology can thus be classified as a *domain ontology* because it only describes a part of the real world, more specifically the business domain. Business domain ontologies have business, which is "the activity of providing goods and services involving financial, commercial and industrials aspects" (Cognitive Science Laboratory 2006), as Universe of Discourse. In its description of an enterprise doing business, the REA ontology focuses on the events that occur within business. These events relate to the creation and transfer of economic value in the conversion and exchange processes of an enterprise. Because of its accounting background, the REA ontology does not emphasize business concepts that are only indirectly related to value creation, transfer, and consumption, like, for instance, marketing strategies. The incompleteness of the REA ontology as a domain ontology for business "as a whole" does not necessarily mean that the REA ontology needs to be extended. It does show that the integration of REA with other business domain ontologies is an important issue and should also be addressed when investigating the usability of the REA ontology.

Finally, 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). REA is intended to support a wide range of business applications as we will show in the next sub-section. REA-based application ontologies used for different applications should be based on the generally accepted REA domain ontology.

The second classification dimension classifies different types of ontologies according to the richness of their internal structure (Lassila and McGuiness 2001). 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. The REA ontology defines for instance axioms that further constrain the intended meaning of the ontology and that allow for semantic reasoning or inference (e.g., the existence of a concept of some type implies the existence of concepts of other types). However, many details of the REA ontology internal structure and underlying logic are not explicitly specified (e.g., disjointness of concepts).

Like Geerts and McCarthy (1999), we recognize the importance of better specifying the REA ontology by further developing its ontological engineering aspects. Formalizing the REA ontology in an ontology representation language and transforming its mix of textual and graphical representations into a more coherent, formal representation will move REA toward a higher maturity level called "formal ontology" (Guarino and Poli 1995). There have been some efforts by researchers (Bialecki 2001; Chou 2006; Geerts 2004) to represent the REA ontology in a machine-readable format, but none of these formalizations is widely known or generally accepted.

Classification of REA Ontology Applications

REA ontology application is defined here in a very broad sense and refers to applications that make use of or benefit from the ontology. An overview of the different applications can be found in Table 2. We limit ourselves largely to applications that have been proposed or developed after the publication of the REA ontology extensions (i.e., Geerts and McCarthy 1999; 2002). Applications that are mainly based on the original REA model as in McCarthy (1982) are not discussed.

Four different application areas were identified: education, systems design, e-collaboration, and knowledge representation and retrieval.

First, The REA ontology is used as a *conceptual framework for teaching accounting information systems*. It integrates the teaching of accounting transaction structures, commitments and business policy specification, business process engineering, and enterprise value chain construction (McCarthy 2003). 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 system. The REA ontology is also used by business/accounting information system professors as a meta-model to generate reference models for different types of business (or transaction) cycles, and thus provides an instrument for teaching the main elements and structures of these cycles.

TABLE 2
REA Ontology Applications

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	• McCarthy (2003) • Dunn et al. (2005)
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	• Geerts et al (1996) • Nakamura and Johnson (1998) • 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 application 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 and McCarthy (2000) • Geerts (2006) • Sedbrook et al. (2007)

In the education context, the REA ontology is primarily used at development time (temporal dimension) for the development of REA models which can be transformed into relational database models (structural dimension). This is definitely the most mature REA ontology application. A considerable group of accounting information systems courses use the REA ontology, and different textbooks have been published which teach accounting information systems by means of the REA ontology.

A second application is *model-driven design using the REA enterprise ontology*. In this type of application, the REA ontology is used by business analysts and business modelers during the development process (temporal dimension) of business software applications (structural dimension). The ontology can be used as a language ensuring unambiguous communication and understanding among all participants of the software development process (Hruby et al. 2006; Hruby 2005; Nakamura and Johnson 1998). More ambitiously, the REA ontology can be used for the model-driven development of systems. In Geerts et al. (1996), an overview is given of some domain-specific CASE tools that use the REA ontology at different stages of the development of a system or database. Nowadays, model-driven development of systems must fit within Object Management Group's (OMG) Model-driven Architecture (OMG 2003), which describes the use of a Computation Independent Model (CIM) as an abstraction of the system from the end-user's viewpoint. The CIM is a representation of the problem domain focusing on systems requirements and can be developed by using the REA ontology as a reference (Borch et al. 2003).

Third, 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. Supply chain collaboration applications use ontologies for the integration of different information sources (structural dimension) at run-time (temporal dimension). Following Guarino, two degrees of use can be distinguished: ontology-aware and ontology-driven. Ontology-aware e-collaboration can be realized using the REA ontology, by providing standard business scenarios and the necessary services to support them in order to establish short-term relationships between businesses quickly and cost effectively. In this context, REA developers have been very active in different international standardization efforts for e-collaboration systems (e.g., ISO Open-EDI initiative, UN/CEFACT, OAG, eBTWG). An example of an REA ontology-driven e-collaboration system was developed by Haugen and McCarthy (2000). In this application, 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.

Finally, ontologies can also be used for *knowledge representation and retrieval.* By adding semantics to the business applications, additional information can be extracted from the business more easily. The systems proposed by Geerts and McCarthy (2000) and Geerts (2004) use an operational REA ontology at run-time (temporal dimension) for adding semantics to the enterprise schema and data (structural dimension). Sedbrook et al. (2007) explore the use of the REA knowledge level extensions for the management of business policies. Geerts (2004) uses well-known XML technologies to illustrate how ontologies can be used for information retrieval. Many techniques that are described by Geerts (2004) 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 SWRL, for which Geerts (2004) defines his own representation language in XSLT. The exploration of the Semantic Web technologies OWL and SWRL for specifying business policies (as in Sedbrook et al. 2007) is a recent development.

REA as a Business Domain Ontology

The classification of the REA ontology applications is in line with the results of the conceptual analysis that positioned REA as a semantically rich business domain ontology that can be used for many enterprise applications. However, 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 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, because the educational context requires formats that are suited for recall by the human mind, while the information systems context requires machine-readable formats. Machines require a format that is based on formal logic. Human minds require succinct representations that can be extended by context-specific reasoning or association. 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 that currently are not explored by the REA ontology community.

III. FORMALLY SPECIFYING THE REA ENTERPRISE ONTOLOGY

The previous sections showed that the REA ontology can serve different purposes and that the basic REA domain ontology can be used to specialize different application ontologies. The language used (graphical or formal) for best representing the ontology depends on the intended application. To ensure the usability of an ontology, a mechanism should be present that allows smooth transformations of the ontology specification from one representation formalism into another. This section proposes a formal specification of the REA ontology using a graphical representation format that can easily be understood by business domain experts but that can also be seamlessly transformed into a machine-readable specification.

First, we present the ontology engineering methodology and associated principles and techniques that we use to restructure and formalize the current REA ontology specification. Next, we discuss related work that fits our research goal. This part also includes an overview of our previous REA formalization and representation activities. Finally, we present our new formal specification of the REA ontology.

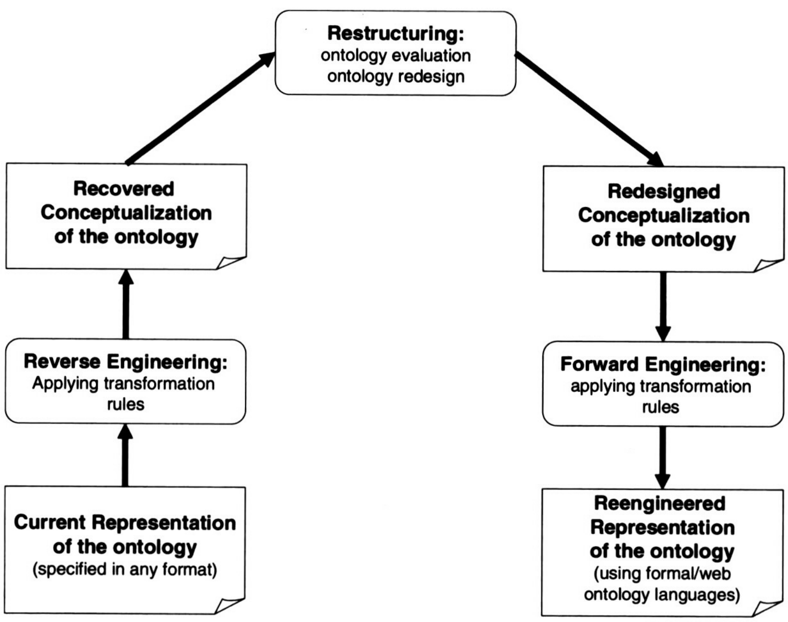
Ontology Engineering and Conceptual Modeling

The success of ontology applications depends in large extent on the quality of their conceptual backbone: the domain ontology. Similar to the relation between product and process quality in software engineering, the quality of a proposed domain ontology depends on the methods used for developing that ontology. Ontology research has proposed methodological guidelines for developing ontologies. Domain ontologies are often developed *ad* *hoc* since their developers focus on the content of the ontology rather than on the representation, formalization, and computational correctness of the ontology. It is our position that ontology content and format are equally important. Therefore, we propose to apply sound ontology engineering principles to existing domain ontologies in order to restructure and formalize them, which should eventually result in increased applicability.

An ontology engineering methodology can provide guidelines for different purposes. It can prescribe a stepwise ontology development process, provide decision rules to follow during the modeling of the ontology, and address various design, representation, and management aspects of the ontology (Jarrar 2005). Figure 2 gives an overview of the methodology, previously proposed in Gailly and Poels (2007a, 2007b), that we used to restructure and formalize the REA ontology specification.

The proposed methodology follows the development steps identified in the METHONTOLOGY framework (Fernández-López et al. 1997) which has its roots in software engineering and knowledge engineering methodologies. METHONTOLOGY is especially useful for our purposes because it has an ontology reengineering extension (Gómez-Pérez and Rojas 1999). The METHONTOLOGY framework defines ontology reengineering as "the process of retrieving and transforming a conceptual model of an existing and implemented ontology into a new, more correct and complete conceptual model, which is reimplemented" (Gómez-Pérez and Rojas 1999, 142). Three main activities are identified in this reengineering process: reverse engineering, restructuring, and forward engineering. During reverse engineering, the conceptualization of the domain ontology is recovered starting from the currently available specification in whatever representation format available. In the following restructuring phase, the recovered conceptualization is redesigned by

FIGURE 2
A Business Domain Ontology Reengineering Methodology—Process, Activities and Artifacts



applying ontology evaluation and redesign techniques. The resulting redesigned conceptualization is subsequently transformed into a reengineered, formal ontology representation during the forward engineering phase.

The importance of using conceptual modeling languages and tools for ontology modeling has been recognized recently. Different authors have proposed UML class diagrams for representing ontologies (Gasevic 2006; Guizzardi and Wagner 2004; Spaccapietra et al. 2004; Kogut et al. 2002; Cranefield and Purvis 1999), but also other less known modeling languages like Object Role Modeling (ORM) (Spyns 2005) have been used for ontology engineering. We agree with Spaccapietra et al. (2004) that conceptual modeling languages (e.g., ER, UML, ORM, etc.) and knowledge representation or ontology languages (e.g., KIF, RDF(S), OWL, etc.) should cooperate during the ontology development process and that they should be used for the tasks for which they are best suited. Conceptual modeling languages can be very useful for the conceptualization of the ontology because they offer representations that closely fit the human perception of the world (Mylopoulos 1998), especially in a business context where these techniques and tools are known (or can easily be learned) by the business domain experts (Davies et al. 2006). The use of conceptual modeling languages may therefore result in an improved readability and understandability of the ontology content. The main purpose of conceptual models is to facilitate or improve the communication concerning domain-specific information between analysts and users (Topi and Ramesh 2002; Wand and Weber 2002). Therefore, most conceptual modeling languages offer graphical representations with well-defined syntax and semantics that are specifically aimed at improving the conceptual model's pragmatic quality (i.e., the degree to which it is understood by its intended audience [Lindland et al. 1994]). When working with the conceptualizations of business domain ontologies (both original/recovered and redesigned), they seem to be more useful than the more formal ontology languages.

Having representations in multiple languages (conceptual modeling and ontology) increases the usability of the ontology (Jarrar 2005). However, transformations between conceptual modeling representations and formal representations in an ontology language are not always straightforward. Ontology developers that use conceptual modeling languages must be aware of the fact that ontologies are a special kind of model (Assmann et al. 2006). Conceptual models are application-specific, while domain ontologies should be as generic and task-independent as possible (Spyns et al. 2002). Furthermore, ontologies describe the reality while conceptual models prescribe the structure or behavior of reality and reality is constructed according to the model (Assmann et al. 2006). This issue is related to the difference in the conceptual modeler and ontology modeler view of the world. In conceptual modeling, something does not exist (i.e., is not part of the Universe of Discourse) unless it is represented in the model ("Closed World Assumption"). In ontology engineering, something is false only if it can be proven to contradict other information in the ontology. This "Open World Assumption" is a very important characteristic of ontologies and must be taken into account when a representation of the conceptualization of an ontology is developed using conceptual modeling languages.

The Ontology Definition Metamodel (ODM) specification of the Object Management Group (OMG) partly solves this problem by providing semantically correct mapping rules in both directions between the UML and Entity Relationship (ER) conceptual data modeling languages (as well as topic maps) and the knowledge representation (or web ontology) languages RDF and OWL (OMG 2006). The four-layered modeling metapyramid developed by the OMG and the UML profiling mechanism is used by Operational Data Model (ODM) 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 (Gasevic 2006). It may happen that a construct that is available in an ontology representation language is not available in a conceptual modeling language. It is, for instance, impossible to represent complex classes that are developed using basic set operations like union, intersection, and complement in UML, whereas these operations are available in OWL. Additionally OWL classes can be enumerated (i.e., defined through exhaustive enumeration of their elements) which is a feature not supported by UML. The ODM specification has solved these shortcomings by including graphical icons for ontology-specific primitives.

Further caution is needed when interpreting ontology models because a construct of a conceptual modeling language may represent different semantics in a knowledge representation language. For example, OWL considers cardinalities as restrictions and not as constraints as in UML. The interpretation of a cardinality as a restriction allows for reasoning applications such as inference (e.g., inferring the existence of a yet unknown thing based on the currently available knowledge or inferring that two known things are actually the same thing). However, for applications in the field of conceptual modeling, software engineering, and database design, cardinalities should be interpreted as constraints that must be satisfied by models. Fortunately, solutions exist (e.g., the "Unique Naming Assumption"; see de Bruijn et al. [2005]) that constrain the usual knowledge representation semantics of ontology language constructs such that they correspond to conventional conceptual modeling semantics, which is particularly useful when developing application ontologies for the aforementioned fields.

Related Research

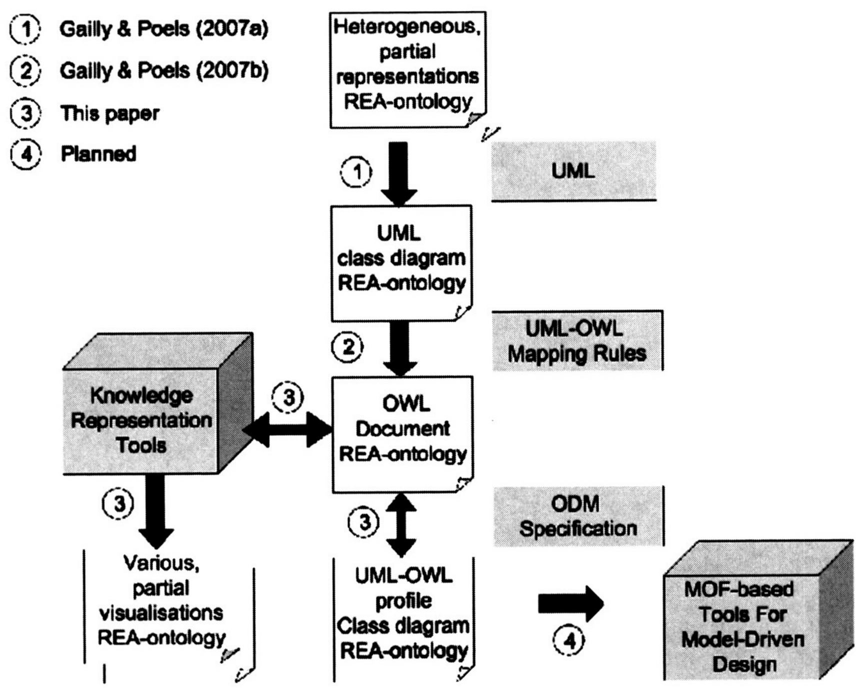
Other authors have recognized the importance of formalizing business domain ontologies which in most cases were not developed by ontology researchers but by business experts. For instance, the eClassOWL project has as main goal the formalization in OWL-lite of the eCl@ss e-collaboration ontology (Hepp and de Bruijn 2007). The ontology formalization approach used in this project was specifically developed for deriving OWL and RDF-S ontologies from hierarchical categorization schemas like eCl@ss and UNSPSC. The main advantage of the approach is that the formalization can be executed automatically which is extremely useful for categorization schemes that contain thousands of concepts. As already stated in Section II, the REA ontology is semantically richer than a simple categorization schema, and as a result, this ontology reengineering methodology cannot be used for REA. Parts of the XBRL taxonomies have also been transformed into OWL in order to use the reasoning mechanisms of OWL which are not supported by XML and XML-schema (Lara et al. 2007).

There have been previous attempts at formalizing the REA ontology. Geerts (2004) explored how XML technologies can be used for the operationalization of the REA enterprise ontology and therefore formalized the ontology with XML Schema. This language was chosen because of its wide acceptance and the availability of different XML tools. Bialecki (2001) also explored how ontology languages and tools could be used for the formalization of the REA ontology. As part of the E-Commerce Integration Meta-Framework project, he formalized the REA ontology with Resource Description Framework (Schema) (RDF/S) using the Protégé knowledge representation tool. It should be noted that Bialecki (2001) based his formalization primarily on textual descriptions of the REA ontology provided by Geerts and McCarthy (1999, 2002). He did not start by developing a conceptualization of the REA ontology in some kind of graphical language.

Chou (2006) has also recognized the importance of using an ontology engineering method for the development of an accounting ontology. Chou (2006) 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 more applications than just accounting.

The main difference with these previous attempts to formalize the REA ontology is that we recognize the importance of an explicitly and completely specified conceptualization that is represented in a uniform, graphical, and easy-to-understand format, as a starting point for the elaboration of a formal representation for an ontology. Following the approach shown in Figure 3, the partial REA ontology specifications that exist in various kinds of representation formats (e.g., text, table entries, conceptual modeling diagrams) must be

FIGURE 3
REA Ontology Transformations and Tools



unified (i.e., ontology reverse engineering or recovering the conceptualization) and restructured in a new conceptual representation (i.e., ontology restructuring or modeling the conceptualization) before developing a formal specification in a knowledge representation or ontology language (i.e., ontology forward engineering or ontology formalization).

Figure 3 gives an overview of the different steps in our research and indicates how the different ontology reengineering phases proposed by the METHONTOLOGY framework have been applied. In a first step, a new conceptual representation of the REA ontology was designed as a single, comprehensive UML class diagram (reverse engineering and ontology redesign) that included specifications of the REA ontology concepts, concept relations and axioms (Gailly and Poels 2007a, 2007b). The "version" of the REA ontology considered corresponds closely to the REA ontology as presented in the most recent papers of Geerts and McCarthy (2002; 2006), but there is a notable point of difference. Instead of referring to economic agents as *inside* or *outside* agents, we prefer the terminology used in Hruby et al. (2006) where economic agents can have *provide* or *receive* relationships with economic events. We prefer the provide/receive dichotomy because it can be applied to characterize the participation of economic agents in economic events in both conversion and exchange processes, whereas the inside/outside distinction is natural for exchange processes (where some outside party like a customer or supplier is involved) but is artificial for conversion processes. In a next step, the UML class diagram used as a conceptual representation of the REA ontology was translated in OWL to create a formal ontology representation (forward engineering) that can be processed by machines (Gailly and Poels 2007b).

Important to notice is that these first two steps focused on obtaining a formal specification of the REA ontology. However, the content and theoretical background of the REA ontology was not questioned, and with the exception of introducing provide and receive relationships, barely changed. The OWL code for the REA ontology is publicly available at http://www.managementinformation.ugent.be/REAontology/REAontology.owl. The OWL code also includes textual definitions of the REA ontology concepts and can be considered as a complete specification of the ontology using a single representation formalism. The third step in the research is new and is presented in the next sub-section. The fourth step is planned and is discussed as part of future work in Section V.

A Graphical Representation of the Formally Specified REA Ontology

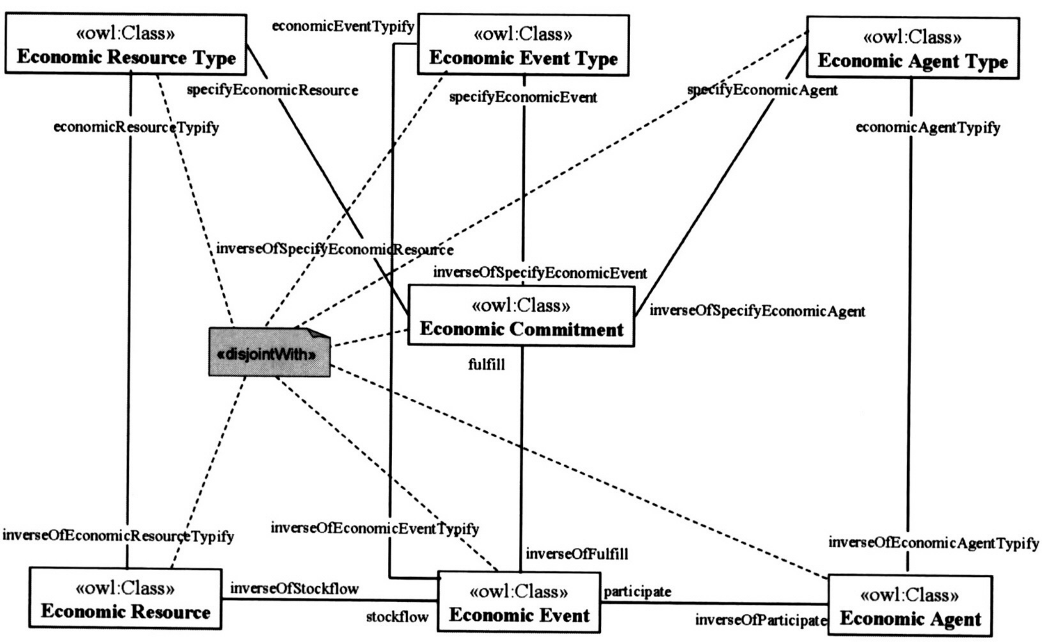
This section shows how the formal REA ontology specification in OWL that was introduced by Gailly and Poels (2007b) can be transformed easily back into a graphical form using an UML-derived language for representing ontologies. The obtained graphical representation is more complete and precise than the currently available REA ontology representations, without losing the advantages that graphical modeling languages offer for non-experts in knowledge representation. This makes the newly proposed graphical representation useful for subsequent iterations of ontology reengineering where the content of REA is evaluated (e.g., in order to assure the "sharedness" of the underlying business conceptualization or to integrate REA with other business domain ontologies). Additionally, this new representation is advantageous for using the REA ontology in a model-driven engineering context (e.g., as input for the design of a meta-model for a REA-based domain-specific modeling language).

A graphical representation of the formally specified REA ontology was obtained by using the UML profile for OWL proposed in the ODM specification (OMG 2006). 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 starting point for applying the OWL-to-UML mapping of the profile is our previously developing OWL code for the REA ontology. Important to notice is that the REA diagrams in this paper use the stereotypes provided by the UML profile for OWL as much as possible for graphically representing the REA ontology. Using the UML OWL profile stereotypes will in some cases make the diagrams more complex and less readable but allows an easy transformation from the graphical conceptualization to the formal representation. The end result of the OWL-to-UML translation can be found in the Appendix B. In what follows, we present partial views over the developed UML class diagram in order to facilitate presentation and discussion. Also, major modeling decisions taken in Gailly and Poels (2007a; 2007b) will be explained.

Figure 4 shows an UML (OWL profile) class diagram that represents 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 of Economic Agent, Economic Event and Economic Resource, their type images, and Commitment are represented by «owl:Class» stereotyped UML classes. In the OWL code of the REA ontology, each of these seven concepts was specified by an OWL class. As these OWL classes are pairwise disjoint, a disjointness constraint has been added to the UML diagram (meaning something cannot be an instance of more than one of the classes at the same time). The UML OWL profile disjointness constraint is a short notation to indicate that all REA concepts are disjoint subclasses of an OWL thing.

FIGURE 4
Basic REA Constellation Representation with OWL UML Profile



The representation of the relations between the concepts is more complex and alternative solutions have been proposed. In Figure 4, the relations between the classes are represented by bidirectional associations with a role name at both ends. In fact, this choice stems from the OWL specification of the REA ontology where relations between concepts are specified by means of the OWL objectProperty construct. As an objectProperty is unidirectional, each relation was specified as a pair of inverse OWL objectProperties (meaning that the domain of one objectProperty is the range of the other objectProperty, and vice versa). Consequently, a pair of inverse OWL objectProperties can be represented as a bidirectional association in UML. For example, for formalizing the stockflow relation between economic resources and economic events, the OWL code of the REA ontology includes the stockflow OWL objectProperty with the OWL class Economic Resource as domain and as range, the OWL class Economic Event and the inverseOfStockflow objectProperty with domain Economic Event and range Economic Resource. Important to notice is that in OWL, setting a domain and range means defining an axiom that is used in reasoning. In our example, this means every object that has a stockflow objectproperty can be considered as an economic resource by the reasoner. The inverseOfStockflow objectProperty is formally defined as the inverse of the stockflow objectProperty via the OWL inverseOf property. Note that the naming of these two inverse objectProperties was arbitrarily chosen as the term "stockflow" and does not indicate a natural reading direction from resource to event or vice versa.

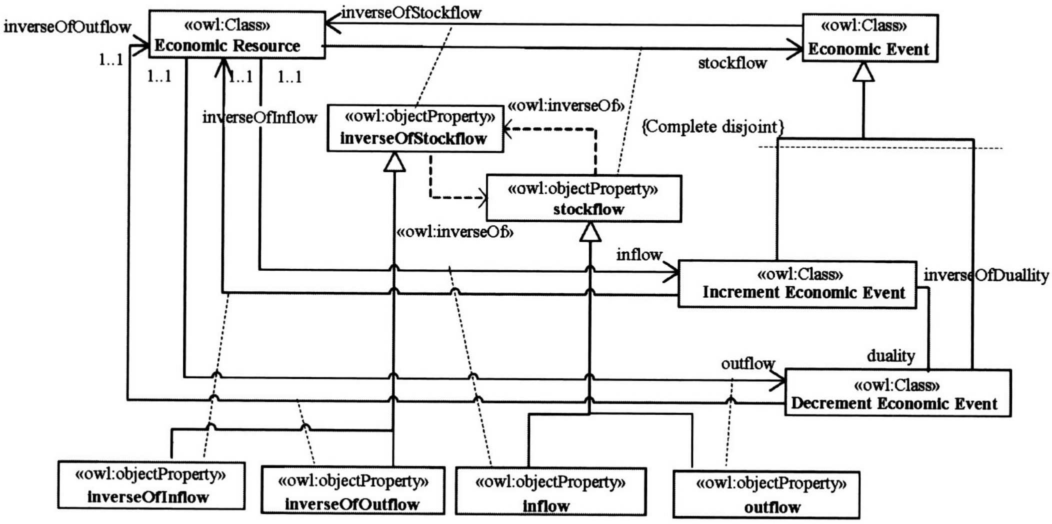
Figure 4[1] looks very similar to the usual UML class diagrams showing the basic REA pattern (as in the papers of Geerts and McCarthy [2002, 2006]). Notice that Figure 4 does not show duality and reciprocity relations. These relations are still present in the OWL version of the REA ontology but are specified between specializations of Economic Event (see Figure 5) and Economic Commitment (see Appendix B). By specializing some of the basic REA concepts and relations, we make implicit semantics of the REA ontology explicit. For example, Figure 5 adds specializations to the Economic Event concept and the stockflow relation. Current REA ontology specifications make a distinction between inflow and outflow stockflow relations and increment and decrement economic events.[2] Our REA ontology model unambiguously specifies that when the stockflow results in an inflow of the economic resource, the economic event is an increment event and when the stockflow results in an outflow, the event is a decrement.

The specialization of economic events into increment and decrement events is complete and disjoint (turning economic event into an abstract class), meaning that every event instance is classified as either an increment event or a decrement event. This mutually exclusive choice reflects the *trading partner view* (as opposed to the *independent view*, see Hruby et al. [2006]) on the REA ontology that is taken by Geerts and McCarthy (2002, 2006). From the perspective of the enterprise (in a business context), an economic event results either an inflow or in an outflow, but not in both simultaneously. From an independent point of view the same event (e.g., a cash receipt) can be an increment event for one business partner (e.g., the seller) and a decrement event for the other business partner (e.g.,

[1] In figure 4 the specify relationships connect commitment and economic resource type, economic event type and economic agent type. However it should be noted that in the REA ontology the specify relationship represents a relationship between a commitment and a basic REA concept or its type. For instance, it could be that an order already specifies a specific product that will be sold and not a type of product.

[2] In this paper we use the terms inflow and outflow and not, respectively, stock and flow which was used by McCarthy in the original REA paper for the role names of the stockflow relationship. We consider inflow and outflow as more generic and also want to make sure that no confusion arises with the role names of the stockflow relationship.

FIGURE 5
Economic Event and Stock Flow Specialization Representation with OWL UML Profile



the buyer). Our choice of trading partner view is an important design decision underlying our formalization effort.

The specialization of the commitment and economic event type concept (see Appendix B) is similar to that of economic event and makes it possible to explicitly specify that increment events fulfill increment commitments, decrement events fulfill decrement commitments, increment commitments specify increment economic event types, and decrement commitments specify decrement economic event types.

In the OWL formalization of the REA ontology, the OWL subclass construct is used to represent a specialization of an OWL class. The OWL code also specifies that all the OWL class specializations are complete and disjoint. In addition, the UML profile representation of the OWL REA ontology specification shows these completeness and disjointness constraints. The specialization of the relations between the classes (like stockflow) is less straightforward in OWL. In order to explain this more clearly, a different approach than the one followed in Figure 4 is used in Figure 5 to represent objectProperties. OWL objectProperties are represented in Figure 5 by two inverse «owl:objectProperty» stereotyped association classes that are linked to unidirectional UML relationships (where the direction is indicated by arrowheads). UML association classes are classes and thus can be specialized. This representational choice stems from the OWL code where an objectProperty can be declared as a specialization of another objectProperty via the OWL subProperty property. Important to notice is that Figure 5 does not specify 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, so that this constraint can be added to our OWL formalization of the REA ontology.

Using specialization structures makes it possible to add constraints which were formerly captured by informally described axioms in the original REA specifications. For example, in Figure 5 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. This constraint is an interpretation and further formalization of part of the stockflow axiom (see REA Ontology Axioms in Appendix A) which states "inflow and outflow events must affect identifiable resources." From this informal axiom statement, using plural forms for concepts, it cannot be derived whether each event must affect at least one resource or exactly one resource. By choosing the second option, we constrain the granularity of an economic event (or, to put it differently, events that affect more than one resource are not atomic and can be further decomposed).

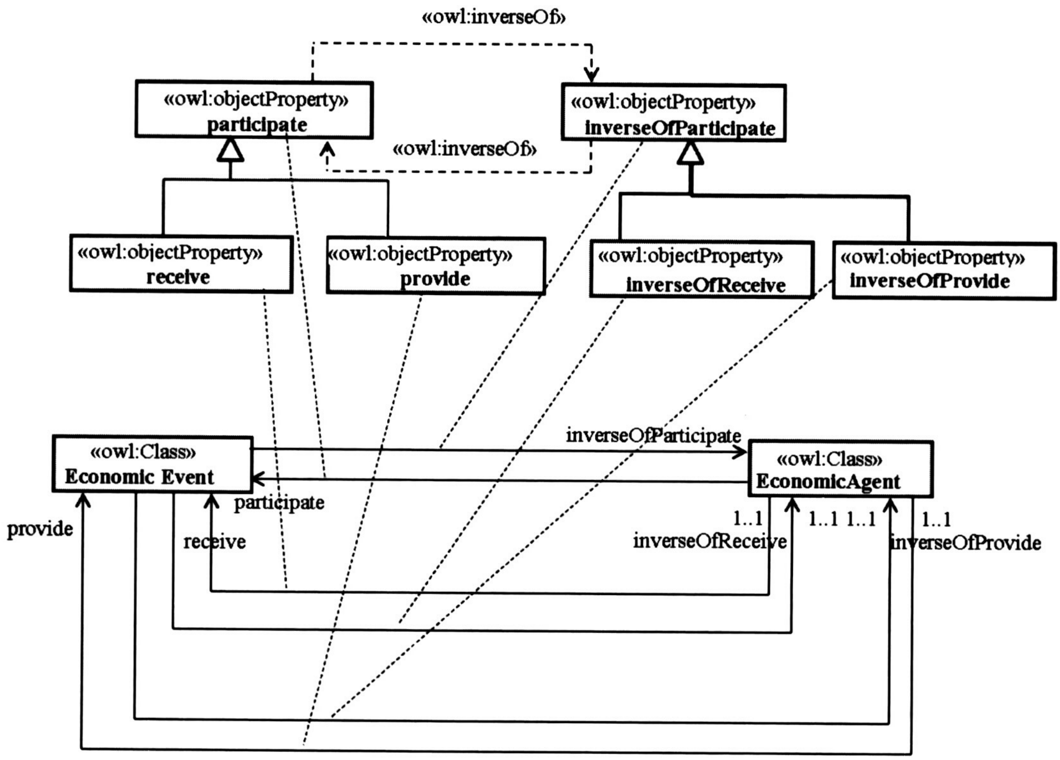
The other REA axioms are also captured in this new representation by adding multiplicity constraints to the objectProperties. A thorough discussion of the choices that were made would take us too far here and can be found in Gailly and Poels (2007b). The main difference with previous REA ontology specifications is that we clearly distinguish between axioms that are defined at the operational level and those that are defined at the knowledge level. For example, the duality axiom that defines that "all events effectuating an outflow must be eventually paired in duality relationship with events effectuating an inflow and vice-versa" (see REA Ontology Axioms in Appendix A), can only be modeled at the knowledge level (i.e., by specifying minimum cardinalities of 1 on the uni-directional dualityType and inverseOfDualityType associations between the IncrementEconomicEventType and DecrementEconomicEventType classes). The axiom implies that a type of increment economic event cannot exist without there being a dual type of decrement economic event (and vice versa), but does not imply that there cannot occur an increment (or decrement) event without there occurring a decrement (or increment) event. For instance, an enterprise might have received cash (increment economic event) without having delivered a product or service yet (decrement economic event). However at the policy level, the receive cash type of events only exists because there also exists a dual decrement type of economic events.

The same approach can be followed for the participate relation between an Economic Event and an Economic Agent. We decided not to specialize Economic Agent into Provider and Receiver sub-classes because provider and receiver are roles assumed by economic agents when participating in events and roles should not be modeled as sub-classes (Parsons and Li 2007). Depending on which event an agent participates in, the agent may be a provider or a receiver. For instance, a cashier is a provider for a cash disbursement event but a receiver for a cash receipt event. This is in contrast with the increment/decrement specialization of economic events, because the trading partner view that we assume dictates that an event is either one of these, but not both. Instead of specializing the Economic Agent class, we specialized the participate relation into provide and receive relations. Figure 6 represents these specializations using the UML profile for OWL.

IV. REA AS A BUSINESS MODELING ONTOLOGY

Figure 3 gives an overview of the transformations that we have already executed during our formalization of the REA ontology. The bottom of the figure contains the graphical representation of the proposed formal REA ontology specification as was presented in the previous section. At the left side of the figure, it is shown how a formal REA ontology specification can be used by knowledge representation tools for different purposes (e.g., ontology engineering, reasoning, visualization, etc.). This use will be illustrated as a proof of concept application for a formal, machine-readable REA ontology specification. The application scenario assumed for this proof of concept is ontology-driven business modeling.

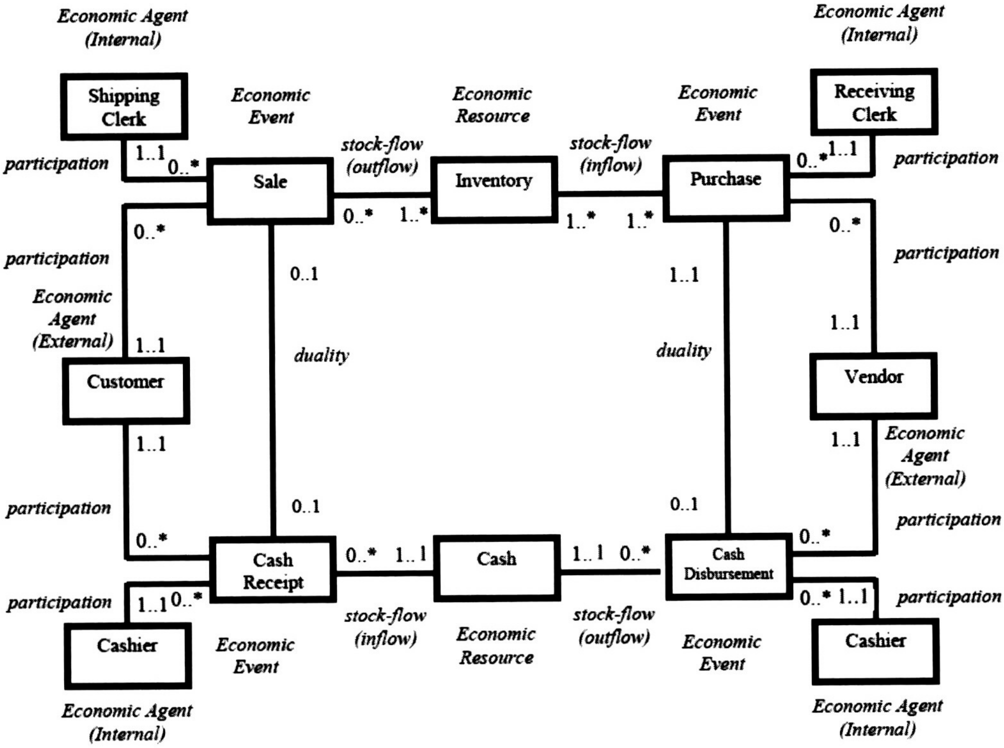
FIGURE 6
Participate Specialization Representation with OWL UML Profile



As already mentioned in Section II, 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 education, there are few documented accounts of REA ontology-driven business modeling in practice. In this section, we show that a machine-readable representation of the REA ontology can be used for the tool-supported modeling of a simple business process. To illustrate our ideas, we use the well-known knowledge representation tool Protégé, the ontology language OWL, and a simple REA-structured Enterprise schema which was taken from Geerts (2004) and is shown in Figure 7.

The first step was introducing in Protégé the OWL specification of the REA ontology. This formal representation of the REA ontology (http://www.managementinformation.ugent.be/REAontology/REAontology.owl; see also the Related Work sub-section in Section III for a discussion on its development) was the basis for and is consistent with the graphical representation (using the UML OWL profile) described in the previous section. At this stage, a Protégé plug-in that uses the UML OWL profile for ontology modeling is not yet available, but this seems to be a very useful and easy to realize addition to the ODM project in the future. The availability of such a plug-in would allow to directly maintain the graphical representation within the Protégé environment, whereas now an intermediate translation step via OWL is needed.

FIGURE 7
REA-Structured Enterprise Schema
(Geerts 2004)

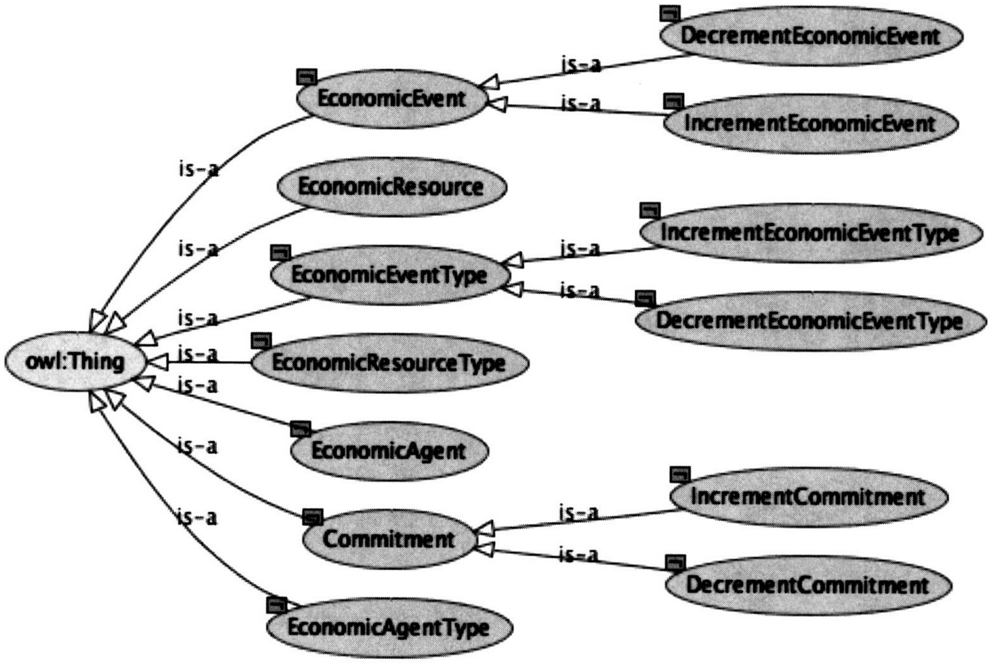


An overview of the resulting OWL classes and OWL objectProperties in the Protégé environment is shown in Figure 7 by means of diagrams produced by the owlViz and ontoViz visualization tools. The owlViz and ontoViz tools are popular ontology visualization plugins for Protégé. The owlViz visualization gives an overview of the classification of the REA ontology concepts in OWL and also indicates the disjointness constraints (by means of a ⟶|). The ontoViz visualization gives a graphical overview of the OWL objectProperties. In order to keep the visual representation conveniently arranged, the ontology visualizations shown in Figure 8 offer only a partial view and do for instance not include the inverseOf OWL objectproperties.

The next step consisted in using the OWL specification of the REA ontology to model an enterprise (we take the example of Figure 7). The availability of the semantic description of the REA ontology concepts, relations, and axioms (as a result of the first step) guides the development of a specific enterprise schema. Protégé can not only be used as an ontology editor but also to populate an ontology. For this purpose, the Protégé OWL individual editor provides a specific, customizable form for every REA ontology concept which makes sure that when instantiating a concept, the mandatory and optional relations with other concept instantiations are included. For instance, Figure 9 shows the Protégé form for an increment economic event. For the Increment Economic Event *purchase*, the Provider (*vendor*), the

FIGURE 8
OwlViz and OntoViz Visualization REA Ontology
(Partial View)

Panel A: OwlViz



Panel B: OntoViz

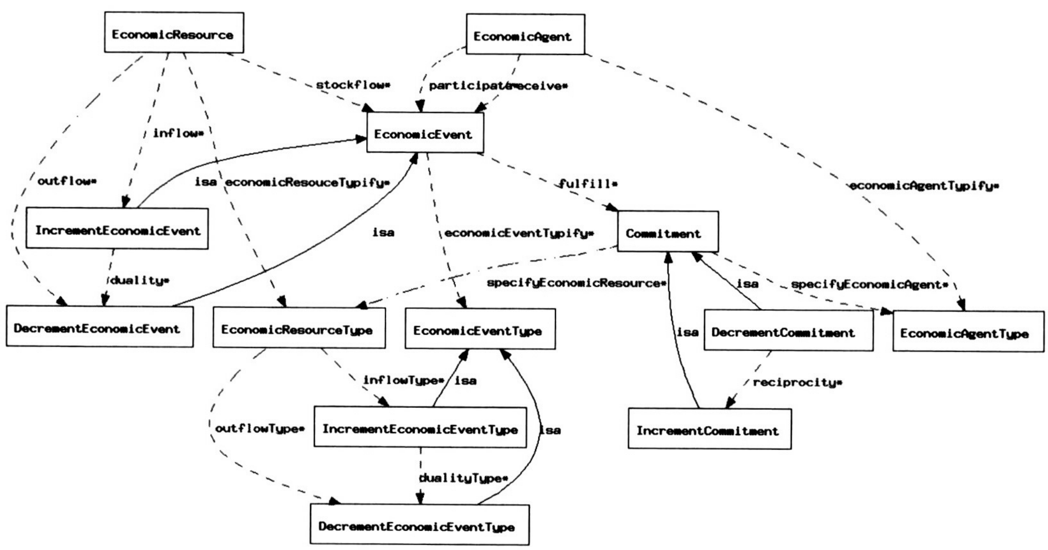
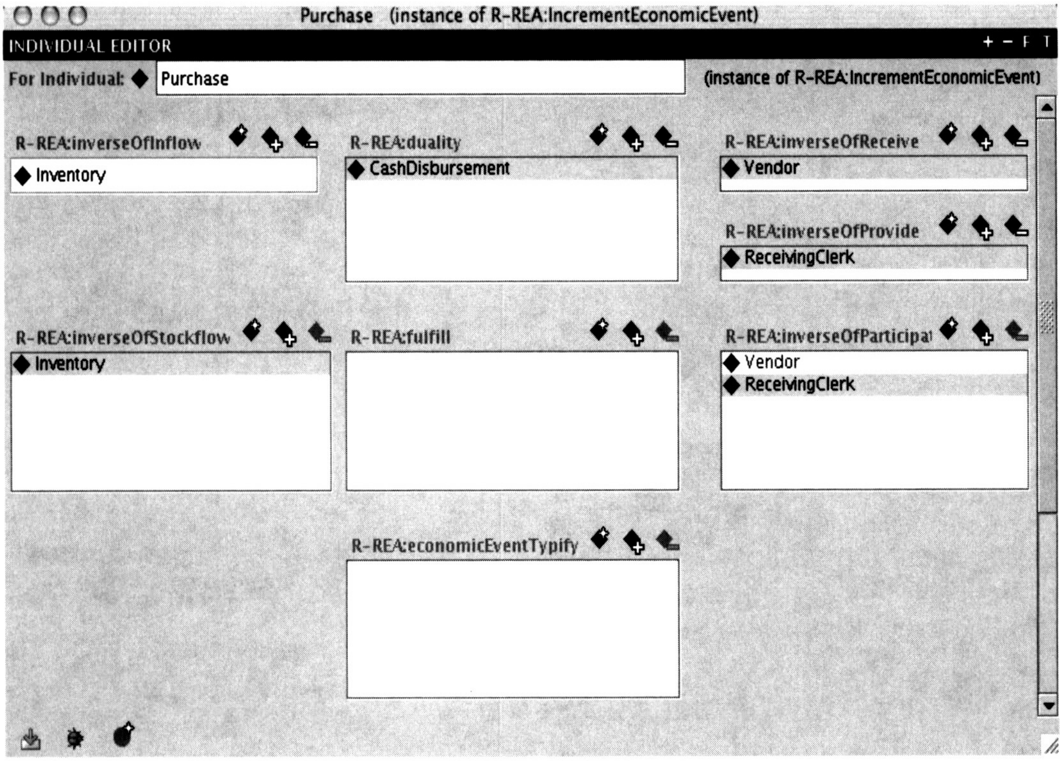


FIGURE 9
Protégé Form for an Increment Economic Event



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. Appendix C contains the complete OWL specification of the example enterprise schema.

It is clear that Protégé is not the best tool to use for business modeling because a more specific user interface is needed. For example, the form presented in Figure 9 in some cases uses the inverse-of relationships because it displays all the objectproperties which have as domain an increment economic event. Instead of Protégé, an ontology-driven business modeling tool should be implemented which uses the machine-readable REA ontology at runtime and provides an easy to use user interface for business modeling.

V. CONCLUSIONS AND FUTURE RESEARCH

After 25 years of REA research, one of the most challenging research directions is improving the usability of the REA ontology. This paper groups three related research efforts that all use existing ontology research outcomes in order to make the REA ontology more applicable. Together they all contribute to a better positioning of the REA ontology as a business domain ontology.

Research Contributions

First, the current state of the REA ontology and the REA ontology applications was analyzed by using existing ontology classification frameworks. The classification based on the subject of the conceptualization is very useful for characterizing the REA ontology as a domain ontology. The further evaluation and development of the REA ontology should be performed in the light of this positioning. 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 semantically rich business domain ontology that needs to be further formalized to make more of the inherent semantics explicit.

After this conceptual analysis, a number of REA ontology applications were reviewed using specific ontology applications frameworks. Different applications of the REA ontology have been proposed and at 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 contribution was the development of a formal REA ontology specification in a graphical representation format which is more complete and precise than the currently available REA specifications in UML or ER diagramming formats without adding considerable complexity for business experts. A formal specification using a knowledge representation or ontology language (or a representational ontology) has as main advantage that it can be processed automatically (and thus opens up the way to many interesting ontology applications) but has as major drawback that it is hard to understand by humans. 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. The use of this specification and profile allows for smooth back and forth translations between mutually consistent formal, machine-readable and graphical, human-readable representations of the REA ontology.

Third, this paper also contains a preliminary proof of concept to persuade the REA research community of the usefulness of having a formal specification of the ontology. A formal specification in OWL of the REA ontology was imported in the Protégé knowledge representation environment and afterward used for developing a simple REA-structured enterprise model. It was further shown how the Protégé tool uses the REA OWL specification to instantiate the ontology concepts and relations and to impose the ontology axioms. The final result is a machine-readable business model that is consistent with the REA ontology and that can subsequently be used for various knowledge representation applications and ontology-driven information systems engineering.

David et al. (2002) question whether a research effort is truly novel, given the current state of the field, whether the problem addressed difficult or easy and whether there is a proof of concept or of feasibility already in order to evaluate the research contribution. In our opinion, the research in this paper is novel in both the problem addressed and the method followed. Our research specifically addresses the form of the current REA ontology specification to make REA easier to operationalize for applications, but without fundamentally altering its content or questioning its theoretical basis. The bulk of REA design science research to date (see e.g., David et al. [2002]), with some exceptions mentioned in the paper, was aimed at improving REA's content and theoretical background, but has to a considerable extent ignored representational, formalization, and operationalization issues. Although we do not question nor criticize the relevance and significance of this work, we believe that equal attention should be paid to ontology engineering aspects that aim at improving the explicitness, preciseness, completeness, and formality of REA's specification. Further, our research is the first effort that applies a set of proven ontology engineering principles and techniques to the REA ontology in a systematic way, following a carefully designed ontology reengineering methodology (which was previously published after thorough peer-review in Gailly and Poels [2007a, 2007b]), in order to position it properly as a domain ontology for enterprises and to restructure and formalize its specification. We believe that the result, which is a formal specification of the REA ontology in an ontology language (OWL) that is both machine-readable and transformable into a human-readable and standardized graphic format (UML), to be a valuable contribution to the development of the REA ontology.

Regarding the second question—we acknowledge that the problem addressed is relatively easy in the sense that the different methodologies and technologies that we used to address it are already available and do not need to be developed first. However, applying these methodologies and technologies was not straightforward and required a great amount of knowledge about ontology engineering, conceptual modeling, and the REA ontology in order to make justifiable modeling decisions. Therefore, although ontology engineering has produced the necessary toolset to reengineer domain ontologies, applying this toolset was not experienced as a routine procedure and often required research on its own.

The business modeling application presented in Section IV is an attempt to answer the third question. This application shows some of the benefits of having a machine-readable specification of the REA ontology. Although the specific application developed is original and could only be realized because we formally specified the REA ontology in an ontology language (i.e., a proof of feasibility), we admit that the application presented is only a skeleton that needs further development to really show the practical benefits on a realistic scale. Clearly, additional research is required to lift our illustrative application (similar to related REA applications discussed in Section II) to the level of a convincing proof of concept that is implemented in practice.

Future Research

First of all, research is required to develop specific proofs of concept for the viability of having a machine-readable REA ontology specification available for the different REA ontology application domains. In all different REA ontology application domains, major issues need to be resolved. For instance, a major issue when using ontologies for modeling is the difference in interpretation of some basic language characteristics between ontology languages and information or software modeling languages. In future research we plan to further investigate these differences and critically evaluate the appropriateness of using formal languages like OWL for business modeling and software engineering purposes. At this stage, business-modeling research is heavily investigating how UML tools and UML extension mechanisms can be used for domain-specific modeling. Future research should also investigate how business domain ontologies like REA specified in OWL or UML OWL profile can be used by model-driven development environments for REA ontology-driven modeling or software engineering.

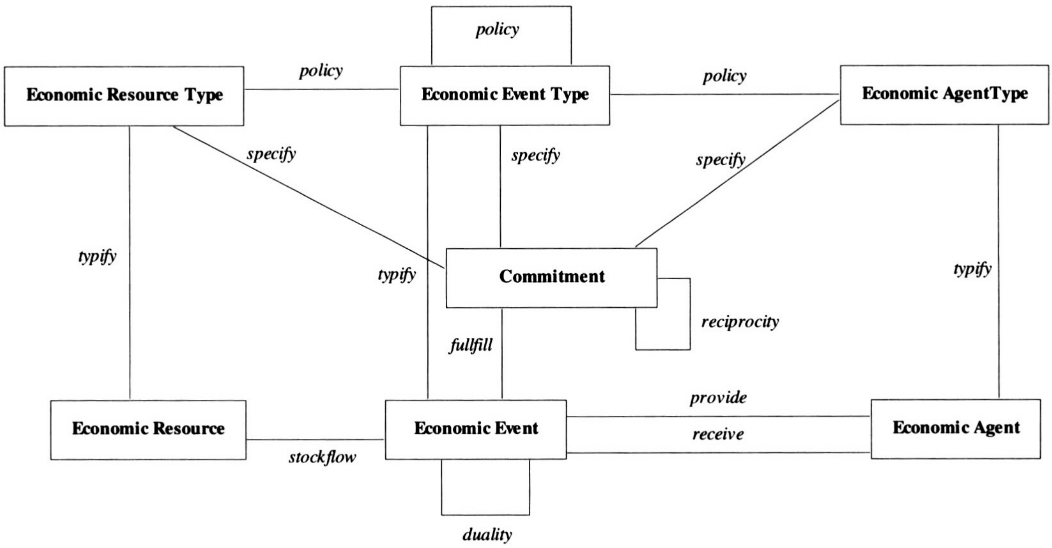
Future research should also look for empirical results that motivate an REA ontology-driven modeling approach for business models. Additionally, more research is needed to prove that the formal REA ontology can be used at run-time to create interoperability between different enterprise applications and for knowledge representation applications.

The success of these applications will depend largely on the soundness and sharedness of the formal specification of the REA ontology. As a result, we also plan to further improve the REA ontology by repeating the ontology reengineering process. A major concern will be resolving the different views by different researchers resulting in a shared understanding of the ontology. 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 the REA enterprise ontology.

APPENDIX A
Definitions of Concepts Operational REA Ontology—Business Process Level

Concept	Definition
Economic Resource	A thing that is scarce and has utility for economic agents and is something users of business applications want to plan, monitor and control.
Economic Agent	Is an individual or organization capable of having control over economic resources, and transferring or receiving the control to or from other individuals or organizations.
Economic Event	Represent either an increment or a decrement in the value of economic resources that are under control of the enterprise.
Commitment	Is a promise or obligation of economic agents to perform an economic event in the future.
Contract	Is a collection of increment and decrement commitments and terms. Under the commitments specified by the terms, a contract can create additional commitments. Thus, the contract can specify what should happen if the commitments are not fulfilled.

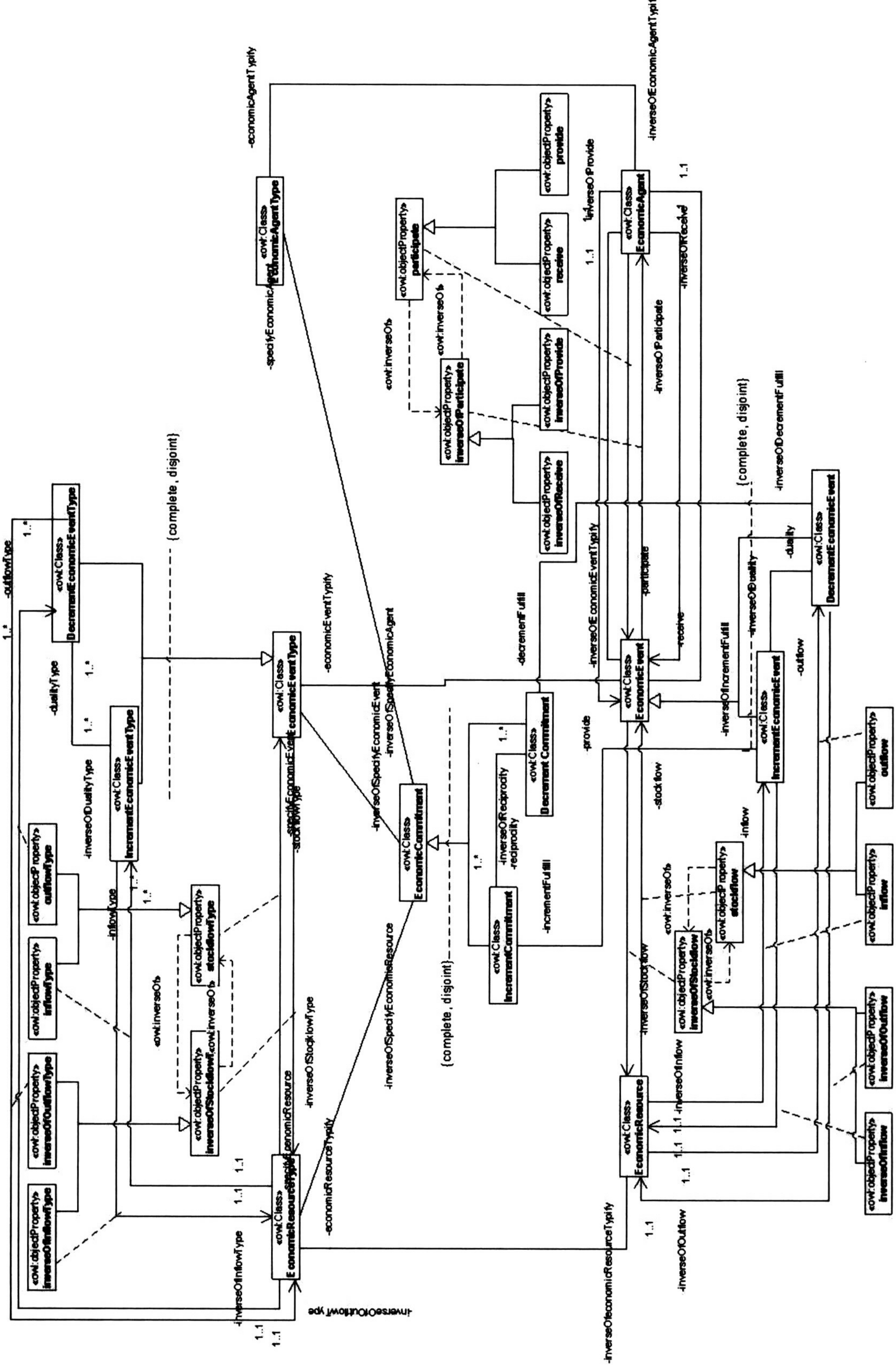
REA Ontology Concept Relations at the Business Process Level



REA Ontology Axioms

Axiom Name	Definition
Stockflow axiom	At least one inflow event and one outflow event exist for each economic resource; conversely, inflow and outflow events must affect identifiable resources.
Duality axiom	All events effecting an outflow must be eventually paired in duality relationships with events effecting an inflow and vice versa.
Participation axiom	Each exchange needs an instance of both the inside and outside subsets.

APPENDIX B


«owl:Class» EconomicAgentType
«owl:Class» EconomicAgent
«owl:Class» DecrementEconomicEventType
«owl:Class» IncrementEconomicEventType
«owl:Class» EconomicEventType
«owl:Class» EconomicCommitment
«owl:Class» Decrement Commitment
«owl:Class» IncrementCommitment
«owl:Class» EconomicEvent
«owl:Class» IncrementEconomicEvent
«owl:Class» DecrementEconomicEvent
«owl:Class» EconomicResource
«owl:Class» EconomicResourceType
«owl:objectProperty» participate
«owl:objectProperty» inverseOfParticipate
«owl:objectProperty» provide
«owl:objectProperty» receive
«owl:objectProperty» inverseOfProvide
«owl:objectProperty» inverseOfReceive
«owl:objectProperty» stockflow
«owl:objectProperty» inverseOfStockflow
«owl:objectProperty» inflow
«owl:objectProperty» outflow
«owl:objectProperty» inverseOfInflow
«owl:objectProperty» inverseOfOutflow
«owl:objectProperty» stockflowType
«owl:objectProperty» inverseOfStockflowType
«owl:objectProperty» inflowType
«owl:objectProperty» outflowType
«owl:objectProperty» inverseOfInflowType
«owl:objectProperty» inverseOfOutflowType
{complete, disjoint}

APPENDIX C


```
<?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 protege “http://protege.stanford.edu/plugins/owl/protege#” >
<!ENTITY    R-REA    “http://www.managementinformation.ugent.be/REAontology/
    REAontology.owl#” >
]>
<rdf:RDF xmlns=“http://users.ugent.be/~fgailly/REAontology/BusinessModelling#″
xml:base=“http://users.ugent.be/~fgailly/REAontology/BusinessModelling”
xmlns:xsd=“http://www.w3.org/2001/XMLSchema#″
xmlns:protege=“http://protege.stanford.edu/plugins/owl/protege#″
xmlns:rdfs=“http://www.w3.org/2000/01/rdf-schema#″
xmlns:R-REA=“http://www.managementinformation.ugent.be/REAontology/
    REAontology.owl #″
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://www.managementinformation.ugent.be/REAontology/
    REAontology.owl “/>
</owl:Ontology>
<R-REA:EconomicResource rdf:ID=“Cash”>
<R-REA:inflow rdf:resource=″#CashReceipt”/>
<R-REA:outflow rdf:resource=″#CashDisbursement”/>
</R-REA:EconomicResource>
<R-REA:DecrementEconomicEvent rdf:ID=“CashDisbursement”>
<R-REA:inverseOfReceive rdf:resource=″#Vendor”/>
<R-REA:inverseOfProvide rdf:resource=″#Cashier”/>
<R-REA:inverseOfOutflow rdf:resource=″#Cash”/>
<R-REA:inverseOfDuality rdf:resource=″#Purchase”/>
</R-REA:DecrementEconomicEvent>
<R-REA:EconomicAgent rdf:ID=“Cashier”>
<R-REA:receive rdf:resource=″#CashReceipt”/>
</R-REA:EconomicAgent>
<R-REA:IncrementEconomicEvent rdf:ID=“CashReceipt”>
<R-REA:inverseOfReceive rdf:resource=″#Cashier”/>
<R-REA:inverseOfProvide rdf:resource=″#Customer”/>
<R-REA:inverseOfInflow rdf:resource=″#Cash”/>
<R-REA:duality rdf:resource=″#sale”/>
</R-REA:IncrementEconomicEvent>
<R-REA:EconomicAgent rdf:ID=“Customer”>
<R-REA:receive rdf:resource=″#sale”/>
</R-REA:EconomicAgent>
<R-REA:EconomicResource rdf:ID=“Inventory”>
<R-REA:inflow rdf:resource=″#Purchase”/>
<R-REA:outflow rdf:resource=″#sale”/>
```

```
</R-REA:EconomicResource>
<R-REA:IncrementEconomicEvent rdf:ID="Purchase">
<R-REA:inverseOfReceive rdf:resource="#Vendor"/>
<R-REA:inverseOfProvide rdf:resource="#ReceivingClerk"/>
<R-REA:inverseOfInflow rdf:resource="#Inventory"/>
<R-REA:duality rdf:resource="#CashDisbursement"/>
</R-REA:IncrementEconomicEvent>
<R-REA:EconomicAgent rdf:ID="ReceivingClerk"/>
<R-REA:DecrementEconomicEvent rdf:ID="sale">
<R-REA:inverseOfReceive rdf:resource="#Customer"/>
<R-REA:inverseOfProvide rdf:resource="#ShippingClerk"/>
<R-REA:inverseOfOutflow rdf:resource="#Inventory"/>
<R-REA:inverseOfDuality rdf:resource="#CashReceipt"/>
</R-REA:DecrementEconomicEvent>
<R-REA:EconomicAgent rdf:ID="ShippingClerk"/>
<R-REA:EconomicAgent rdf:ID="Vendor">
<R-REA:receive rdf:resource="#Purchase"/>
<R-REA:receive rdf:resource="#CashDisbursement"/>
</R-REA:EconomicAgent>
</rdf:RDF>
```

REFERENCES

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. Berlin, Germany: Springer.

Bialecki, A. 2001. REA ontology. Available at: 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 systems. Paper presented at Workshop on Model Driven Architecture: Foundations and Applications, Enshede, The Netherlands.

Borst, W. N. 1997. *Construction of Engineering Ontologies*. Enschede, The Netherlands: Centre for Telematica and Information Technology.

Burton-Jones, A., V. C. Storey, V. Sugumaran, and P. Ahluwalia. 2005. A semiotic metrics suite for assessing the quality of ontologies. *Data & Knowledge Engineering* 55 (1): 84–102.

Chou, C.-C. 2006. Using ontological methodology in building the accounting knowledge model—REAP. Paper presented at 2006 AAA Mid-Year Meeting AI/ET Workshop.

Cognitive Science Laboratory. 2006. *WordNet*. Princeton, NJ: Princeton University.

Cranefield, S., and M. Purvis. 1999. UML as an Ontology Modeling Language. Paper presented at Workshop on Intelligent Information Integration, July, Stockholm, Sweden.

David, J. S., G. J. Gerard, and W. E. McCarthy. 2002. Design science: An REA perspective on the future of AIS. In *Researching Accounting as an Information Systems Discipline*, edited by V. Arnold, and S. G. Sutton, 35–64. Sarasota, FL: American Accounting Association Information Systems Section.

de Bruijn, J., R. Lara, A. Polleres, and D. Fensel. 2005. OWL DL vs. OWL flight: Conceptual modeling and reasoning for the semantic web. Paper presented at 2005 World Wide Web Conference, May 10–14, Chiba, Japan.

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

Fernández-López, M., A. Gómez-Pérez, and N. Juristo. 1997. METHONTOLOGY: From ontological art towards ontological engineering. Paper presented at Working Notes of the AAAI Spring Symposium on Ontological Engineering, Stanford, Stanford, CA.

Gailly, F., and G. Poels. 2007a. Ontology-driven business modeling: Improving the conceptual representation of the REA-ontology. In *Conceptual Modeling—ER 2007—Lecture Notes in Computer Science 4801*, edited by C. Parent, K.-D. Schewe, V. C. Storey, and B. Thalheim, 407–422. Auckland, New Zealand: Springer.

———, and ———. 2007b. Towards ontology-driven information systems: Redesign and formalization of the REA ontology. In *Lecture Notes in Computer Science 4439*, edited by W. Abramowicz, 245–259. Poznan, Poland: Springer.

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

Geerts, G., W. E. McCarthy, and S. R. Rockwell. 1996. Automated integration of enterprise accounting models throughout the systems development life cycle. *The International Journal of Intelligent Systems in Accounting, Finance & Management* (September): 113–128.

———, and ———. 1999. An accounting object infrastructure for knowledge based enterprise models. *IEEE Intelligent Systems and Their Applications* 14 (4): 89–94.

———. 2000. Augmented intensional reasoning in knowledge-based accounting systems. *Journal of Information Systems* 14 (2): 127.

———, and W. McCarthy. 2002. An ontological analysis of the economic primitives of the extended-REA enterprise information architecture. *International Journal of Accounting Information Systems* 3 (1): 1–16.

———. 2004. An XML architecture for operational enterprise ontologies. *Journal of Emerging Technologies in Accounting* 1: 73–90.

———. 2006. Policy-level specification in REA enterprise information systems. *Journal of Information Systems* 20 (2): 37–63.

Gómez-Pérez, A., and M. D. Rojas. 1999. Ontological reengineering and reuse. Paper presented at the 11th European Workshop on Knowledge Acquisition, Modeling and Management, Dagstuhl Castle, Germany. Berlin, Germany.

———, Fernández-López, and O. Corcho. 2004. *Ontological Engineering*. Springer-Verlag.

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

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, 25–32. Amsterdam, The Netherlands: IOS press.

———, and R. Poli. 1995. Editorial: The role of formal ontology in the information technology. *International Journal of Human-Computer Studies* 43 (5–6): 623–624.

———. 1998. Formal Ontology and Information Systems. Paper presented at International Conference on Formal ontology in Information Systems (FOIS '98), Trento, Italy.

Guizzardi, G., and G. Wagner. 2004. A unified foundational ontology and some applications of it in business modeling. Paper presented at Proceedings of the Enterprise Modeling and Ontologies for Interoperability workshop (EMOI-INTEROP workshop CAISE 2004), Riga, Latvia.

Haugen, R., and W. E. McCarthy. 2000. REA: A semantic model for Internet supply chain collaboration. Paper presented at Proceedings of the Business Objects and Component Design and Implementation Workshop VI: Enterprise Application Integration.

Hepp, M., and J. de Bruijn. 2007. GenTax: A generic methodology for deriving OWL and RDF-S ontologies from hierarchical classifications, thesauri, and inconsistent taxonomies. Paper presented at Fourth European Semantic Web Conference, Innsbruck, Austria.

Hruby, P. 2005. Role of domain ontologies in software factories. Paper presented at OOPSLA Workshop on Software Factories, San Diego, California.

———, J. Kiehn, and C. V. Scheller. 2006. *Model-Driven Design Using Business Patterns*. New York, NY: Springer.

Jarrar, M. 2005. *Towards Methodological Principles for Ontology Engineering, STARLAB*. Brussels, Belgium: Vrije Universiteit Brussel.

Kogut, P., S. Cranefield, L. Hart, M. Dutra, K. Baclawski, M. K. Kokar, and J. Smith. 2002. UML for ontology development. *Knowledge Engineering Review* 17 (1): 61–64.

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. Berlin, Germany: 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.

Lindland, O. I., G. Sindre, and A. Solvberg. 1994. Understanding quality in conceptual modeling. *IEEE Software* 11 (2): 42–49.

McCarthy, W. E. 2003. The REA modeling approach to teaching accounting information systems. *Issues in Accounting Education* 18 (4): 427–441.

Mylopoulos, J. 1998. Information modeling in the time of the revolution. *Information Systems* 23: 127–155.

Nakamura, H., and E. R. Johnson. 1998. Adaptive framework for the REA accounting model. In *OOPSLA '98 Workshop on Business Object Design and Implementation IV.* Vancouver, Canada: OOPSLA.

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

———. 2008. MDA guide version 1.0.1. Available at: http://www.omg.org/cgi-bin/doc?omg/03-06-01.

Parsons, J., and X. Li. 2007. An ontological metamodel of classifiers and its application to conceptual modeling and database design. *Lecture Notes in Computer Science 4801*: 214–228.

Sedbrook, T., R. Newmark, and C. Cullom. 2007. Applying semantic web technologies that automate extended REA typifications, groupings and associations. In *REA-25.* Newark, Delaware: University of Delaware.

Spaccapietra, S., C. Parent, C. Vangenot, and N. Cullot. 2004. On using conceptual modeling for ontologies. Paper presented at 2004 Web Information Systems Workshops (WISE 2004), *Lecture Notes in Computer Science 3307.*

Spyns, P., R. Meersman, and M. Jarrar. 2002. Data modeling versus ontology engineering. *SIGMOD record: Special Issue on Semantic Web and Data Management* 31 (4): 12–17.

———. 2005. Object role modeling for ontology engineering in the DOGMA framework. Paper presented at 2005 On the Move to Meaningful Internet Systems (OTM 2005), Agia Napa, Cyprus.

Stamper, R., K. C. Liu, M. Hafkamp, and Y. Ades. 2000. Understanding the roles of signs and norms in organizations—A semiotic approach to information systems design. *Behaviour & Information Technology* 19 (1): 15–27.

Topi, H., and V. Ramesh. 2002. Human factors research on data modeling: A review of prior research, an extended framework and future research directions. *Journal of Database Management* 13 (2): 3–19.

Ushold, M., and R. Jasper. 1999. A framework for understanding and classifying ontology applications. Paper presented at IJCAI Workshop on Ontologies and Problem-Solving Methods.

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.

Wand, Y., and R. Weber. 2002. Research commentary: Information systems and conceptual modeling—A research agenda. *Information Systems Research* 13 (4): 363–377.

Welty, C., F. Lehmann, M. S. Fox, D. L. McGuiness, M. Ushold, and M. Grunninger. 1999. Invited panel: Ontologies: Expert systems all over again? Paper presented at Sixteenth National Conference on Artificial Intelligence (AAAI-99), Orlando, Florida.