# OWL-formalization of the Resource Event Agent Business Ontology

Frederik Gailly* and Geert Poels

Ghent University
Management Informatics Research Unit
Hoveniersberg 24, 9000 Gent
Frederik.Gailly@Ugent.be, Geert.Poels@Ugent.be

**Abstract.** Business domain ontologies offer great opportunities for facilitating communication between people in business, for improving the enterprise system engineering processes and for creating interoperability between enterprise systems. However despite these opportunities, their use in practice is still limited. This can be partly attributed to the lack of formal representation of these ontologies.
This paper proposes a formal representation of the REA business domain ontology. For the development a structured approach is used which uses conceptual models as intermediary representation for formalizing business domain ontologies.

## 1 Problem and Context

Business domain ontologies offer great opportunities for creating interoperability between enterprise systems. The successful application of these ontologies partly depends on the quality and proper formalization of the ontology. The last decade different business domain ontologies (Tove[1], Enterprise Ontology[2], REA Business Ontology[3], E3 value Ontology[4] and Business Model Ontology[5]) have been proposed. Most of these ontologies have a strong theoretical basis but lack a formal description.

A formalization of the application domain helps the developers to analyse their proposed ontology by comparing it with reality and can more easily expose critical issues. Formal languages describe the meaning of knowledge precisely and this supports automatic reasoning, which can be used for checking the consistency of the ontology[6]. A formal description also makes it easier to map different ontologies and find similarities between ontologies. Generally a formal description will improve the acceptance of the business domain ontology and will facilitate the development of applications.

From the previous mentioned ontologies the Business Model Ontology[5] is the only one with a representation in a formal language [7]. Based on the conceptual model of the Business Model Ontology and the description of the ontology a formal representation was developed in Protègè OWL. This leads to a more

* Corresponding Author

precise specification of the business domain constructs and relations, and makes it easier to use the Business Model Ontology as business model analysis tool.

The Resource Event Agent business ontology (REA-ontology)[3] is another business ontology that offers great possibilities because of his strong theoretical basis and its unique structural characteristics. Its foundation in strategic concepts, such as Porter's Value Chain[8] and basic accounting principles holds great promises for enterprise system design and creating interoperability between enterprise systems. Additionally the REA-ontology exists of multiple duality relations. This means that each increment event in the business's resources is linked with a corresponding decrement event. This constraint thus establishes expectations for relationships between events[9].

Unfortunately unlike the Business Model Ontology, the REA-ontology is underspecified as an ontology. The terminology of the constructs of the REA-ontology is sometimes inconsistent and confusing [10]. The proposed definitions, relations and constraints need to be converted into a formal language to discover inconsistencies and to analyze operational use[11]. A more formal conceptualization in an ontology language will improve the definition of the constructs and will make the ontology also readable for machines, which in turn will lead to new applications.

In this paper we want to develop a formal representation of the REA-ontology in OWL. In the next section a description of the REA-ontology is given and an overview of the current degree of formalization is given. In the third section a formal representation is developed in OWL and the limitations of our approach are formulated. Finally, the last section outlines the conclusions and future research avenues.

## 2 Existing Resources and Past Work

Recently the REA framework has also been integrated with newer information technologies. The framework has, for instance, been applied in UN/CEFACT Modeling Technology (UMM for business process modeling)[12] which is used in the context of ebXML. Furthermore , the use of REA modeling has been accelerated by the advent of ERP systems, with which the REA framework shares the process-oriented foundation of integrated enterprise value chains [13]. According to [14], the REA-model can be used to establish a theoretical basis for SAP and other ERP models.

### 2.1 Description REA business ontology

The original REA-model was developed by McCarthy in 1982 and presented as a semantic model for the development accounting systems. Over the years, many 'design science' efforts have resulted in extensions to the basic REA model. These include the modelling of accounting phenomena at different levels of abstraction (value chain, process and task) and additional ontological primitives and axioms(commitment, events, type images)[3].

For the description of the REA-ontology the recent work of Geerts and McCarthy can be used[15, 3, 11]. These papers offer an enumeration of the definitions of REA-ontology constructs and presents the relations between the different constructs by means of ER-diagrams or UML class diagrams. Table 1 summarises the most important definitions.

**Table 1.** REA-ontology concepts definitions[15, 3, 11]

| REA-construct | Definition |
|---|---|
| Business Process | a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer [16]. |
| Economic Event | a transaction where an internal agent (an economic unit or agent) gives something of value (an economic resource to an outside economic agent. |
| Business Event | a significant occurrence in time that enterprise management would like to plan, control or evaluate. |

The REA business ontology exists of a three-level architecture for presenting economic activities. Based on the work of [8], [17] and [18], an enterprise is presented by a value chain, which is seen as an network of business processes. A business process is in turn an aggregate of Economic Events. Figure 1 summarizes these relations in an UML class diagram.

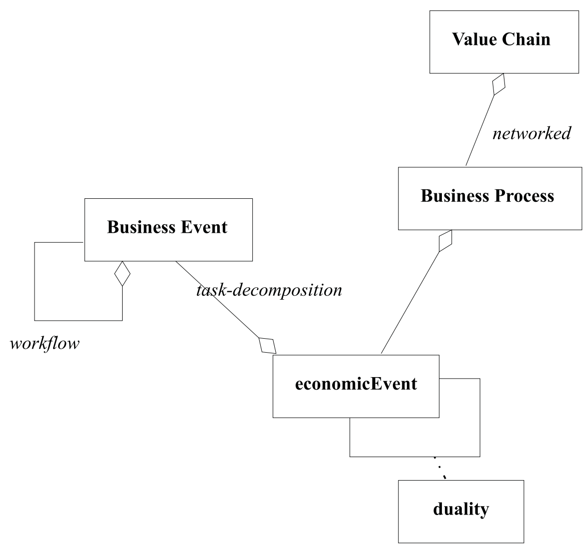



**Fig. 1.** Three-Level Architecture Model based on Geerts and McCarthy[11]

According to the REA-ontology each economic event is part of a reusable pattern of relations between three kind of objects that can be identified in every economic exchange or conversion process: economic resources, economic agents and economic events. Each economic resource is linked with an economic event that causes its inflow or outflow. Furthermore every economic event that results in a resource inflow (e.g. a purchase) must be paired by an event that results in a resource outflow (e.g. a cash disbursement), and vice versa. This pattern forms the basis of the REA-ontology and is derived from McCarthy's original REA model(figure 2)[19].

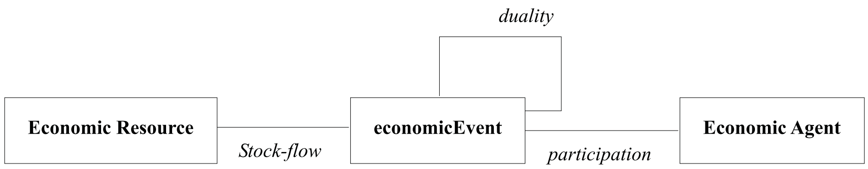



**Fig. 2.** REA model [19]

However to accomplish the economic events (business transformation or transfer) an ordered sequence of actions are needed. Originally Geerts and McCarthy labeled these activities as tasks [15], but in their last working paper the name Business Events was used [11]. "Business Events illustrate the task-decomposition structure of the workflow needed to accomplish the paired Economic Events ([11], p.8)." The REA ontology uses the ISO Open-edi (ISO 2002) as a foundation for the conceptualization of the business processes. However as this is still preliminary work, the formalization of the business process specification is not subject of this paper and will be added in future work.

Furthermore the basic REA-model is also extended horizontally with additional concepts and axioms. One of the extensions is the commitment concept. Based on Ijiri the REA-ontology defines a commitment "as an agreement to execute an economic event in a well defined future that will result in either an increase of resources or a decrease of resources"([20], p. 130). The commitment concept is linked with economic event by an 'fulfill' association. Similar to economic events a reciprocal relationship is defined between commitments. A commitment establishes an economic agreement which can be of two type: a schedule or a contract [11].

So far the presented business domain ontology only consists of actual economic phenomena both current and future. This operational infrastructure is extended with a knowledge infrastructure that conceptualizes the abstract phenomena that characterize the actual economic phenomena. The abstraction used in the REA business ontology is typification, which captures descriptions that apply to a group of actual phenomena. This abstraction adds a knowledge layer for planning and control above the operational infrastructure.

The knowledge infrastructure contains the following types of abstract classes: Economic Resource Type, Economic Event Type and Economic Agent Type. These classes can be associated with the type image relations: 'policies' and 'standards'. In the REA business ontology 'policies' are defined as abstractions that restrict the legal configurations of the actual phenomena and 'standards' define blueprints for the actual phenomena [11].

The knowledge and operational infrastructure interact by the economic commitment concept as this is both an operational and abstract phenomena. Commitments promise to execute economic events in the future and their specification is abstract because usually commitments consists of type-level designation of expected behaviour [11].

Figure 3 gives a general overview of the operational and knowledge infrastructue of the REA ontology at the business process level in a UML class diagram. It is based on the different class diagrams presented in the Geerts and McCarhty papers and it will be used for the development of the formal representation of the REA-ontology. Figure 3 contains also the different specialisations of the basic REA construct. Important to notice is that not only the different classes are further specialized but also the association classes. This must be taken into account during the formalization.

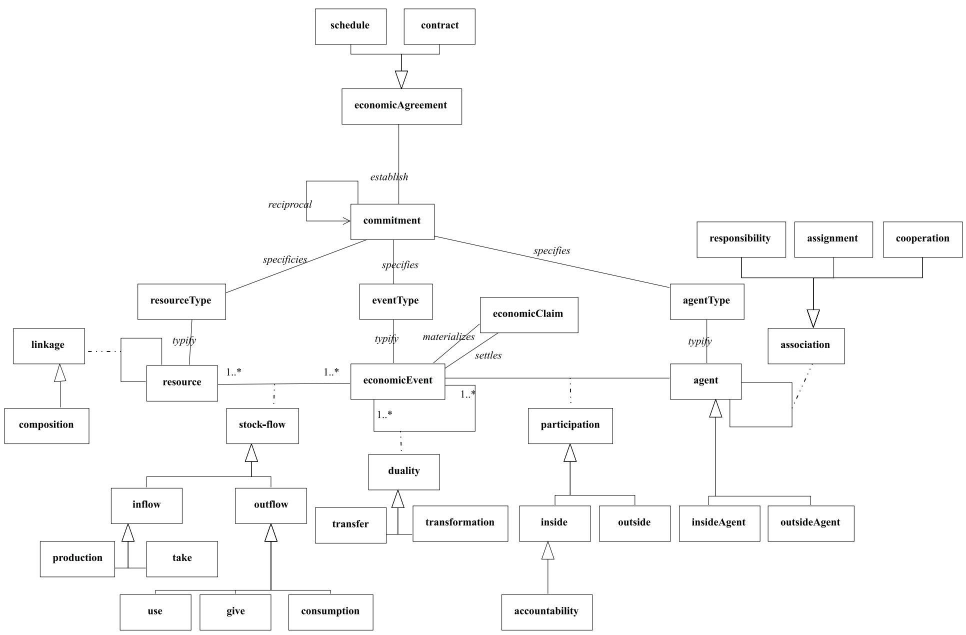



**Fig. 3.** UML Class-diagram REA-ontology [15, 3, 11]

### 2.2 Current formalization REA business ontology

Currently the formalization of the REA business domain ontology is very limited. In the papers of Geerts and McCarthy UML class diagrams are used for the description of the relations between the concepts. The definitions and axioms are only described in text but are not formalized in a formal language.

Geerts explored how XML technologies can be used for the operationalization of the REA enterprise ontology [21]. XML schema is used to formalize parts of the REA-ontology. However Geerts chooses XML-schema because of the wide acceptance of this language but it is our opinion that this will limit the use in practice of the REA-ontology and that the use of ontology language like RDF(S) and OWL offer far more opportunities.

However there were some efforts from other researchers to formalize the REA ontology. Like already mentioned business domain ontology can facilitate the software engineering process. The REA model can be used as an computation independent model (CIM) in the Model Driven Architecture (MDA) approach to software engineering. It is for this purpose that Borch et al. [22] an XML-schema of the basic REA-model which is transformed into a Java Representation Model.

Bialecki formalized the REA-ontology with Protègè [23] and exported it to RDF and OWL. However his latest updates date from 2001 and the REA-ontology has been subject to different changes since 2001. It is our intention to work further on his work with a strong focus on the ontology development process. Different firms may require different ontologies [9] and therefore a structured approach in the development and formalization of the business domain ontology is essential.

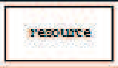
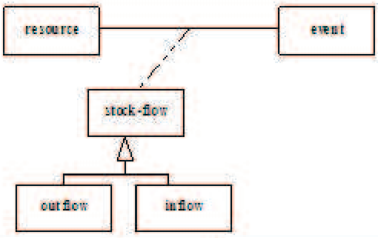
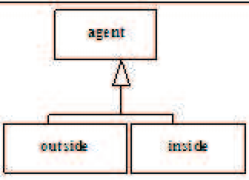
## 3 Formalization in OWL

In the ontology engineering field many different approaches are used for the formalization of an ontology. However, many authors have recently recognised the opportunities that the conceptual modelling and database field can offer for ontology engineering. Conceptual modelling approaches have been designed to give a semantically rich description of the universe of discourse and "could, at least to some extent, handle the description of the conceptualisation that is the subject of some ontology" ([24], p. 25).

In our approach we wish to use the UML class diagram presented in the previous section as an intermediate for the formal representation of the ontology. The mapping between UML and OWL is based on the work of [25] which compare UML with DAML and give some rules for the mapping between UML and DAML. In our transformation we will use these rules as guidelines for the mapping between the UML class diagram of the operational infrastructure of the REA-ontology and OWL. At this stage other RDF(S) could also be used, but we expect that in a later stage when additional constraints will incorporated, OWL will offer the best solution.

Table 2 gives some examples of the applied transformations. The UML classes were transformed in OWL classes, associations were represented in OWL as 'objectproperties'. The generalizations in the UML class were transformed in two ways depending on either if it were generalisations of classes or association classes. In the case of generalization of a normal class the OWL 'subTypeOf' construct was used, in the other case the 'subPropertyOf' construct was used. Another approach could have been reifying the association classes and using the 'subTypeOf' in every case. In appendix A the OWL representation of the REA business domain ontology is presented.

**Table 2.** Transformation examples between REA ontology UML class and OWL

| REA UML class diagram elements | OWL representation |
|---|---|
|  | <owl:Class rdf:ID="resource" /> |
|  | <owl:ObjectProperty rdf:ID="stock-flow"><br><rdfs:domain rdf:resource="#event" /><br><rdfs:range rdf:resource="#resource" /><br></owl:ObjectProperty> |
|  | <owl:ObjectProperty rdf:ID="outflow"><br><rdfs:subPropertyOf<br>rdf:resource="#stock-flow" /><br></owl:ObjectProperty><br><owl:ObjectProperty rdf:ID="inflow"><br><rdfs:subPropertyOf<br>rdf:resource="#stock-flow" /><br></owl:ObjectProperty> |
|  | <owl:Class rdf:ID="outside"><br><rdfs:subClassOf rdf:resource="#agent" /><br></owl:Class><br><owl:Class rdf:ID="inside"><br><rdfs:subClassOf rdf:resource="#agent" /><br></owl:Class> |

The mapping rules used at this stage were very straightforward and logical. However, the graphical representation will become more complex when more constructs and constraints are added and more complex mapping rules will be needed. In future research we will further evaluate existing mapping rules and how they can be used for the development of a formal representation of the REA-ontology. Finally, the mapping rules can be translated into an XSLT stylesheet which can be used for transforming the XMI representation of the UML diagrams into a representation in the target ontology representation language.

## 4 Conclusion and Future Research

A correct formal representation of the REA-ontology offers great opportunities and will facilitate the operationalization of the REA-ontology. In this paper an OWL representation of the REA-ontology is developed with as key characteristic the use of an UML class diagram as an intermediate step for this formalization.

In future research the UML class diagram will be further expanded with the business event specification of the REA-ontology. The REA-ontology also contains some constraints that are not incorporated in this paper and must be added in future research. Probably OCL will be needed to model the constraints in the UML class diagram and therefore it must be investigated how these constraints can be translated in OWL.

However it could be that bassic UML and OCL will not satisfy needs for representation of ontology concepts that are borrowed from Descriptive Logic. Maybe the use of UML profiles can offer a solution or maybe basic UML needs to be extended with additional elements[25, 26].

In future research the OWL representation of the REA business ontology will be used to evaluate the domain ontology, compare it with other business ontologies and investigate how it can be used in practice.

## References


1. Fox, M., Gruninger, M.: Enterprise modeling. AI Magazine **19** (1998) 109–121
2. Ushold, M., King, M., Moralee, S., Zorgio, Y.: The enterprise ontology. The Knowledge Engineering Review **13** (1998) 31–89
3. Geerts, G.L., McCarthy, W.E.: An ontological analysis of the economic primitives of the extended-rea enterprise information architecture. International Joural of Accounting Information Systems **3** (2002) 1–16
4. Gordijn, J.: Value based requirements engineering: Exploring innovative e-commerce ideas. PhD thesis, Vrije Universiteit Amsterdam (2002)
5. Osterwalder, A., Pigneur, Y.: An e-business model ontology for modeling e-business (2002)
6. Antoniou, G., van Harmelen, F.: Chapter 4. International Handbooks on Information Systems. In: Web Ontology Language: OWL. Springer (2004) 666
7. Pigneur, Y.: E-business modell ontology for improving business/it alignment. In FEUP, ed.: In Proceedings of the CAISE'05 Workshops Vol. 02. (2005)
8. Porter, M.E.: Competitive Advantage: Creating and Sustaining Superioir Performance. Free Press (1985)
9. O'Leary, D.: Different firms, different ontologies, and no one best ontology. IEEE Inteligent Systems (2000) 72–78
10. Lampe, J.: Discussion of an ontological analysis of the economic primitives of the extended-rea enterprise information achitecture. International Journal of Accounting Information Systems **3** (2002) 17–34
11. Geerts, G., McCarthy, W.E.: The ontological foundation of rea enterprise information systems (2005) Working Paper.
12. Bergholtz, M., Layaweera, M., Johannesson, P., Wohed, P.: Reconciling physical communicative, and social/institutional domains in agent oriented information systems - a unified framework. In: Lecture Notes in Computer SCience 2814. Spinger (2003) 180–194

13. McCarthy, W.E.: The rea modelling approach to teaching accounting information systems. Issues in Accounting Education **18** (2003) 427–441
14. O'Leary, D.E.: On the relationshsip between rea and sap. International Journal of Accounting Information Systems **5** (2004) 65–81
15. Geerts, G., McCarthy, W.: An ontological analysis of the primitives of the extended rea enterprise information achitecture. Working Paper (2000)
16. Hammer, M., Champy, J.: Reengineering the Corporation: A Manifesto for Business Revolution. Harper Business (1993)
17. Ijiri, Y.: The Foundations of Accounting Measurement. Prentice-Hall (1967)
18. Geerts, G., McCarthy, W.E.: Modeling business enterprises as value-added process hierarchies with resource-event-agent object templates. In: Business Object Design and Implementation, Springer-Verslag (1997) pp 94–113
19. McCarthy, W.E.: The rea accounting model: A generalized framework for accounting systems in a shared data environment. The Accounting Review (1982) 554–78
20. Ijiri, Y.: Theory af Accounting Measurement. American Accounting Association (1975)
21. Geerts, G.L.: An xml architecture for operational enterprise ontologies. Journal of Emerging Technologies in Accounting **1** (2004) 73–90
22. Borch, S., Jespersen, J., Linvald, J., Osterbye, K.: A model driven achitecture for rea based systems. In: Proceedings of the Workshop on Model Driven Achitecture: Foundations and Applications, Enshede, The Netherlands, University of Twente (2003)
23. Bialecki, A.: Rea ontology. http://www.getopt.org/ecimf/contrib/onto/REA/ (2001)
24. Spaccapietra, S., Parent, C., Vangenot, C., Cullot, N.: On using conceptual modeling for ontologies. In Sprnger, ed.: Proceedings of the Web Information Systems Workshops (WISE 2004 workshops). Volume 3307 of Lecture Notes in Computer SCience. (2004) 22–23
25. Baclawski, K., M.K. Kokar, P.K., Hart, L., Smith, J., Holmes, W., Letkowski, J., Aronson, M.: Extending uml to support ontology engineering for the semantic web. In: Proceedings of the Fourth International Conference on the Unified Modeling Language (UML 2001). Volume 2185 of Lecture Notes in Computer Science. (2001)
26. Guizzardi, G., Herr, H., Wagner, G.: Towards ontological foundations for uml conceptual models. In: 1st Int. Conference on Ontologies, Databases and Applications of Semantics (ODBASE 2002) LNCS, Springer (2002)

## A OWL formalization REA-ontology

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns="http://www.owl-ontologies.com/REA.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.owl-ontologies.com/REA.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Commitment"/>
  <owl:Class rdf:ID="Contract">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="EconomicAgreement"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Schedule">
    <rdfs:subClassOf rdf:resource="#EconomicAgreement"/>
  </owl:Class>
  <owl:Class rdf:ID="EventType"/>
  <owl:Class rdf:ID="ResourceType"/>
  <owl:Class rdf:ID="InsideAgent">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Agent"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="EconomicClaim"/>
  <owl:Class rdf:ID="EconomicEvent"/>
  <owl:Class rdf:ID="Resource"/>
  <owl:Class rdf:ID="OutsideAgent">
    <rdfs:subClassOf rdf:resource="#Agent"/>
  </owl:Class>
  <owl:Class rdf:ID="AgentType"/>
  <owl:ObjectProperty rdf:ID="establish">
    <rdfs:range rdf:resource="#EconomicAgreement"/>
    <rdfs:domain rdf:resource="#Commitment"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="stockflow">
    <rdfs:range rdf:resource="#Resource"/>
    <rdfs:domain rdf:resource="#EconomicEvent"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="consumption">
    <rdfs:subPropertyOf>
      <owl:ObjectProperty rdf:ID="outflow"/>
    </rdfs:subPropertyOf>
```

```
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="specifyEvent">
  <owl:equivalentProperty>
    <owl:ObjectProperty rdf:ID="specifyResource"/>
  </owl:equivalentProperty>
  <rdfs:range rdf:resource="#Commitment"/>
  <rdfs:domain rdf:resource="#EventType"/>
  <owl:equivalentProperty>
    <owl:ObjectProperty rdf:ID="specifyAgent"/>
  </owl:equivalentProperty>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="assignement">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:ID="association"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="materialize">
  <rdfs:domain rdf:resource="#EconomicClaim"/>
  <rdfs:range rdf:resource="#EconomicEvent"/>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:ID="settle"/>
  </owl:inverseOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="insideParticipation">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:ID="participation"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="reciprocal">
  <rdfs:range rdf:resource="#Commitment"/>
  <rdfs:domain rdf:resource="#Commitment"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="typifyEvent">
  <rdfs:range rdf:resource="#EconomicEvent"/>
  <owl:equivalentProperty>
    <owl:ObjectProperty rdf:ID="typifyAgent"/>
  </owl:equivalentProperty>
  <rdfs:domain rdf:resource="#EventType"/>
  <owl:equivalentProperty>
    <owl:ObjectProperty rdf:ID="typifyResource"/>
  </owl:equivalentProperty>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="transfer">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:ID="duality"/>
```

```
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#specifyAgent">
  <owl:equivalentProperty rdf:resource="#specifyEvent"/>
  <rdfs:range rdf:resource="#Commitment"/>
  <rdfs:domain rdf:resource="#AgentType"/>
  <owl:equivalentProperty>
    <owl:ObjectProperty rdf:about="#specifyResource"/>
  </owl:equivalentProperty>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#participation">
  <rdfs:range rdf:resource="#EconomicEvent"/>
  <rdfs:domain rdf:resource="#Agent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="accountability">
  <rdfs:subPropertyOf rdf:resource="#insideParticipation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="cooperation">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#association"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#typifyResource">
  <rdfs:range rdf:resource="#Resource"/>
  <owl:equivalentProperty>
    <owl:ObjectProperty rdf:about="#typifyAgent"/>
  </owl:equivalentProperty>
  <rdfs:domain rdf:resource="#ResourceType"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  <owl:equivalentProperty rdf:resource="#typifyEvent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#typifyAgent">
  <rdfs:range rdf:resource="#Agent"/>
  <owl:equivalentProperty rdf:resource="#typifyEvent"/>
  <rdfs:domain rdf:resource="#AgentType"/>
  <owl:equivalentProperty rdf:resource="#typifyResource"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="production">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:ID="inflow"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="take">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#inflow"/>
```

```
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="transformation">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#duality"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#association">
  <rdfs:range rdf:resource="#Agent"/>
  <rdfs:domain rdf:resource="#Agent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="responsibility">
  <rdfs:subPropertyOf rdf:resource="#association"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#settle">
  <rdfs:domain rdf:resource="#EconomicEvent"/>
  <rdfs:range rdf:resource="#EconomicClaim"/>
  <owl:inverseOf rdf:resource="#materialize"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#specifyResource">
  <rdfs:range rdf:resource="#Commitment"/>
  <owl:equivalentProperty rdf:resource="#specifyEvent"/>
  <rdfs:domain rdf:resource="#ResourceType"/>
  <owl:equivalentProperty rdf:resource="#specifyAgent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#duality">
  <rdfs:range rdf:resource="#EconomicEvent"/>
  <rdfs:domain rdf:resource="#EconomicEvent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#inflow">
  <rdfs:subPropertyOf rdf:resource="#stockflow"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="outsideParticipation">
  <rdfs:subPropertyOf rdf:resource="#participation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="linkage">
  <rdfs:domain rdf:resource="#Resource"/>
  <rdfs:range rdf:resource="#Resource"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="use">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#outflow"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="give">
```

```
    <rdfs:subPropertyOf>
      <owl:ObjectProperty rdf:about="#outflow"/>
    </rdfs:subPropertyOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#outflow">
    <rdfs:subPropertyOf rdf:resource="#stockflow"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="composition">
    <rdfs:subPropertyOf rdf:resource="#linkage"/>
  </owl:ObjectProperty>
</rdf:RDF>
```