Enterprise Interoperability Ontology for SOC applied to Logistics

Wout Hofman

TNO, P.O. Box 5050, 2600 GB, Delft, The Netherlands

Abstract. The Service Oriented Architecture (SOA, [1]) can be applied to enterprise integration. It creates an Internet of services for enterprises. Service science [2] defines services in term of value propositions of enterprises to customers. Both service science and the Internet of services require a form of mediation between customer requirements and service capabilities like for instance identified in [3]. However, mediation is not yet automated, thus can not be applied real time. Furthermore, many business areas already have agreement on semantics and interaction sequencing based on existing business documents, thus, mediation is not always required for a certain business area. This paper presents an ontology to support business services (Ontology Web Language (OWL), [18]). The concepts shared at business level are based on existing approaches like Resource, Event, Agent used for auditing and control [4] and builds upon service frameworks like OWL-S [13]. The ontology will be specialized to logistics and compared with other approaches that might be applied to enterprise integration.

1 Introduction

Over the past decades, services have become the most important part of economies [5]. Basically, the service economy refers to the service sector. It leads to more sophisticated forms of cooperation, or what is called value co-creation [2]. As Spohrer also points out, service systems are dynamic configurations of resources that interact via value-proposition-based interactions with governance mechanisms.

Each value proposition can be supported by a service. One could distinguish business services that directly relate to a value proposition, and IT services, mostly known as web services [6]. From an IT perspective, IT services can be defined by each individual service provider with its own semantics specified as ontology, thus requiring mediation to match user requirements and leading to mash ups for enterprise integration [3]. Mediation not only requires the matching of customer requirements to provider's capabilities, but also semantic mappings, and mapping of interaction sequencing (the latter is also called 'choreography', [7]). An alternative is to develop a shared ontology at business level for a particular class of business services common to more than one actor, e.g. logistics, customs, etc. The basic research question of this paper is on the feasibility of enterprise interoperability ontology common to all business services or phrased otherwise: do all enterprises use identical concepts in their business. To answer this question, we apply concepts from accountancy [4] to

Hofman W. (2010).
Enterprise Interoperability Ontology for SOC applied to Logistics.
In *Proceedings of the 4th International Workshop on Architectures, Concepts and Technologies for Service Oriented Computing*, pages 66-78
DOI: 10.5220/0003050400660078

interoperability, since these concepts already focus on value propositions and value exchange. This ontology will then be the basis for defining application area specific ontology. Firstly, we define the interoperability ontology, and secondly specialize it to logistics. Finally, we discuss the relation between our interoperability ontology and other approaches.

2 Concepts for Enterprise Interoperability

This section defines enterprise interoperability ontology. The concepts of this ontology are based on value propositions, their supporting electronic business documents and accountancy concepts. The concept ‘actor’ is used in this section to represent ‘enterprise’, ‘organizational unit’, or ‘government organization’. Although we discuss class - and data constraints, these are not formally shown in this paper. Furthermore, we did not have a tool to visualize classes and their properties, we have drawn such a visualization where classes are visualized by (blue) ellipses; the instances as (grey) rectangles. Class constraints are visualized by arrows between a domain and a range. First, the classes are defined and, second, briefly explained. Finally, some guidance on the construction of a business system semantic model is given.

2.1 Classes Representing Interoperability Concepts

The following classes represent enterprise interoperability concepts at business level:

- *Business System Interoperability Model*: a model of all business activities and a business system semantic model for modelling interoperability in a given business system or application area, e.g. logistics, customs, and supply chains.
- *Business Activity* (or *activity*): a generic activity provided by one or more actors that is able to change the state of a business system for customers. ‘*Capability’,* ‘*event’* and ‘*state transition*’ are synonyms for business activity as they express state change that can be induced by an actor to the outside world.
- *Business System Semantic Model*: semantic model of all classes and their constraints that are common to all business activities in a business system.
- *Value proposition*: a value offered by a particular actor based on a business activity of that actor [2].
- *Business Service*: the constraint between an actor and a value proposition of that actor. This definition implies that value propositions are the main concepts from an actor’s viewpoint.
- *Business Process*: the internal process of an actor to provide a business activity.
- *Business Activity Choreography*: an ordered set of event types for the execution of a business activity. A synonym is *business activity protocol.* A business activity choreography can be common to more than one business activity (see lateron).
- *Event Type*: a means to exchange data between a customer and a service provider within the context of a business activity, see. REA [4]. *Interaction type* is a synonym.

- *Business Transaction*: information exchange between a customer and provider for actual value exchange referring to a particular business activity or value proposition. A business transaction behaves according the business activity choreography of the referred business activity.
- *Customer*: an actor that consumes a business activity (or business service) by initiating a business transaction.
- *Service Provider*: an actor providing a particular business activity with related value propositions.
- *Resources*: two types of resources are distinguished, namely operand and operant resources [2]:
  - *Operant resources (also called t-resources)*: the persons and/or means to execute a business process.
  - *Operand resources (also called d-resources)*: the actual resources that are exchanged between a customer and a service provider. Three types of resources are distinguished: goods, services, and rights [4].
- *Resource Allocation*: the mechanism to assign operant resources to one or more business transactions.
- *Bound resources*: those operant resources that are assigned to one or more business transactions.
- *Business Transaction Phases*: the different phases in the initiation and execution of a business transaction. Together, these phases compose a business activity choreography.

We will discuss these classes in more detail in the next section. The classes and their class constraints are shown in the following figure; an OWL representation of this model is constructed with Protégé. The concepts all comprise a business system interoperability model. The business system semantic model is not visualized, see later.

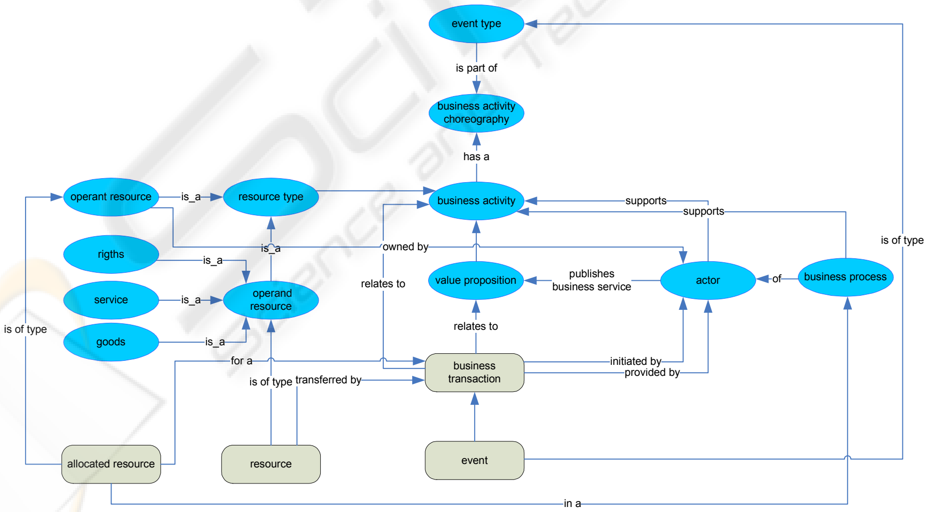

Fig. 1. Interoperability classes and their class properties.

2.2 Explaining the Classes

The classes need further explanation. This section elaborates on the difference between classes representing dynamic and static concepts, class constraints, and clarifies the core concept 'business activity'.

Firstly, we distinguish between static and dynamic concepts, meaning that dynamic concepts change more frequent than static ones. Static concepts are shown in blue; dynamic ones in grey. It implies for instance that a business activity of an actor will change less frequent than business transactions referring to those business activities, i.e. the core business activity of an enterprise will probably not change during years.

Second, we will define class constraints. Business activity, value proposition, business transaction, business activity choreography, event type, and event are the central concepts. A business activity has choreography of event types. Choreography of event types is not modelled in this figure; we will discuss this aspect later on. A business transaction consists of a specific sequence of events that are of a type and need to be exchanged according to the business transaction choreography. The right part of the figure shows actors and their support of business activities. An actor supports a business activity and can define value propositions for those business activities. These value propositions are published as business services. An actor has a business process that supports one or more activities and their value propositions. As a business process supports a business activity, it must also support the related business transaction protocol and its choreography. The left part of the figure shows the resource types, which can be the operand and operant resource types. The operant resources are owned by an actor and can be allocated at a specific time and location to a business process for the execution of one or more business transactions. These business transactions control the exchange of the so-called operand resources: goods, service or rights. It implies that all information exchanged in a business transaction must be sufficient to exchange the goods, service or right.

Thirdly, the core concept 'business activity' needs clarification. We have defined a business system interoperability model that defines all real world aspects shared by actors and modelled by business activities provided by those actors, e.g. the transportation of containers from one place to another, financial risk management, and insurances against risk. These real world aspects, that either have a physical or a virtual nature, are considered as business activities that are provided by actors. The semantics that those business activities have in common are given by the business system semantic model. A business activity, or in short activity, is able to change the state of the real world by exchanging a resource (an operand resource defined according to [4]) according to a particular value proposition. Thus a state transition of a business system can be induced by executing a business activity. Thus, a business system interoperability model consists of a state space, modelled by a business system semantic model, and state transitions on that state space, modelled by business activities.

State transitions, and thus activities, in general consist of [8]:

- *Pre-conditions*: a set of conditions (or predicates) that must always be true before the execution of an activity.
- *Post-condition*: the actual result of the execution of an activity. The result can be defined as the state in case an activity is executed successfully.

- *Firing rule*: the (ordered) set of rules that are executed when all pre-conditions are met and result in the post-condition. A firing rule actually changes the state of the world. In our interoperability framework, a firing rule behaves according business activity choreography.

Pre- and post-conditions can be expressed in terms of the state space, i.e. the real world effects that can be changed by their activity. These pre- and post-conditions can be expressed at different abstraction levels that require different knowledge. An abstract specification by logical expressions can for instance not be understood by business persons, whereas they are the ones that specify the activities and thus the pre- and post-condition.

Finally, we have stated that firing rules between pre- and post-conditions are specified by business activity choreography. The latter can consist of different transaction phases as for instance [11] and [12] distinguish a negotiation and an execution phase after activity discovery. In the negotiation phase, all data needs to be exchanged to allow that pre-conditions can be met, whereas in the execution phase all required data for actual firing an activity is required. One could state that an activity can logically be decomposed into two activities that reflect transaction phases. Each of these phases consists of ordered set of event types exchanged between customer and service provider. The phases need to be completed by a rollback phase, see [11] and [12]. One could argue that business activity choreography always consists of the same set of phases. However, one could also distinguish between choreography for public and commercial services. The main difference would be that in public services there is no negotiation on prices and conditions of a value proposition, but a public service provider has to state officially that pre-conditions are met and a firing rule can be executed (in Dutch: ‘ontvankelijkheidverklaring’). Execution results in a decision (e.g. an ordinance or a grant; Dutch ‘beschikking’), e.g. a building permission, permission for transportation of goods, and the reception of unemployment benefit. A decision can always be followed by an official objection or a different viewpoint. A decision is always according to the request, negative, or partly positive.

2.3 Refining Pre- and Post-conditions for an Application Area

In this section of the paper, we try to combine physical business systems and more abstract systems by introducing the concept of ‘place’. This latter concept represents either a physical location or a state. A place is always connected to a business activity and a place can contain zero, one or more resources at a given time. In our framework, a business activity equals a state transition with pre- and post-conditions. Pre-conditions consume so-called operant resources that are produced as a post-condition according to firing rules and their choreography utilising operand resources. For instance, in case a business activity is the assembly of cars as output resources, the input resources are its parts and assembly machines and personnel the operand resources. This example illustrates that all its parts need to be present at a given time to assemble a car at a given time.

We take the approach formulated in [12] and [14] and state that:

- A pre-condition of a business activity specifies the types and number of operand resources that are required for its execution and the types and number of operant resources that are input to the activity. Pre-conditions are connected to a place by an input connector.

- A post-condition of a business activity specifies the types and number of operand resources that are produced as a result of the business activity, including the duration between consumption of the input operand resources and the output operand resources. Post-conditions are connected to a place by an output connector.
- Between two actors, a business transaction shares the availability of resources of type operant resources in a place connected to a pre-condition and the availability of resources, also of type operant resources, that are produced by a post-condition in a place connected to that post-condition via an output connector.

The concepts 'place', 'connector' and 'availability' to our ontology (see next figure) support the ability to express availability of resources in a given place at a given time. Time is expressed as 'availability' that refers to a place (or state) in a Petrinet and resources. At any given time, a place can contain various resources belonging to different business transactions for a business activity. In this approach, an input connector relates to a pre-condition and an output-connector to a post-condition. This extension expresses that a business transaction must always consider 'time', 'resource' and 'place' to be shared by two actors that participate in that business transaction. Place can represent a physical location or a state in a state space shared by two (or more) actors.

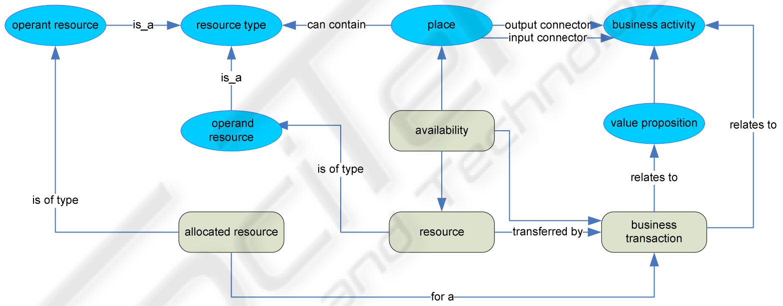

Fig. 2. Adding place and time to ontology.

According to Abstract State Machines [8], the relation between pre- and post-conditions is not expressed, which is however the case according to [14] by places and connectors. The latter means that a Petrinet of processors, places, and connectors exists and each place can contain object types. In our approach, these object types are resources and a business activity is equal to a processor. In some application areas, it might be worthwhile to specify such a network explicitly with a Petrinet, e.g. in transport a network of ports exists between which sea transport is possible. In other application areas like government, the network can become very complex due to the large number of business activities, each of which changes part of the state space, and an approach based on ASM will suit better. In the latter case, a Petrinet has to be constructed to validate that state transitions are really executable (although it can be complex to visualize due to the large number of states, see [15]).

3 Specialization to Logistics and Transport

Logistics and transport consists of several actors that utilize each others resources to transport goods from one physical location to another. The transportation process can be quite complex, involving a variety of actors. International container transport by sea for instances consists of transport by truck to a port (pre-carriage), loading a container in a vessel, sea transport, discharge of the container in the port of destination, short sea shipment from that port to another port, and final transport by truck to the final destination. In other occasions, large shippers have regional warehouse in close to a port, in which the goods are stored for regional distribution. This paper only presents ‘transport’ as a business activity, whereas others like ‘transshipment’ and ‘storage’ are other business activities. First of all, we define ‘transport’ as activity and secondly we present a semantic model for this particular activity.

3.1 The activity ‘transport’

‘Transport’ is a specialization of ‘business activity’. Quite a number of actors offer ‘transport’ as a value proposition, e.g. a number of shipping lines offer transport of containers between for instance specific ports in Europe and the Asian Pacific. One organization can also offer a variety of transport services, e.g. express delivery (the same day), air cargo and an international parcel service. Although each transport service will have a different price, is offered for a different type of cargo, and has different durations, all value propositions for ‘transport’ have common elements. A specialization of ‘business activity’ called ‘transport’ defines those common elements: the physical movement of cargo between two places with duration, cost, and a specific transport means as an allocated resource. Specific transport means have a so-called transport mode and will give restrictions to the type of goods that can be transported, e.g. a container vessel can for instance only carry 20 or 40 feet containers and an airplane requires the particular containers that fit that type of plane. Thus a particular transport activity can be expressed by the number and type of operant resources that can be transported using a particular operand resource. The relation between operand and operant resources is expressed by the business activity, although it can also be expressed as a constraint between operand and operant resources. Duration also gives restrictions to type of cargo, e.g. heavy cargo needs more attention and will take more time. The pre- and post-conditions for ‘transport’ are derived from the above mentioned parameters and can be expressed quite simply:

- Pre-conditions are defined as in terms of the operant resources that can be consumed via input connectors:
  - The cargo offered by a customer is part of the set of cargo types defined by a service provider.
  - The weight of the cargo offered must be between the minimal and maximal weight defined by a service provider.
  - The physical dimensions of the cargo must be between a minimal and maximal dimension defined by a service provider.

- Post-condition: the cargo offered by a customer is transported by a service provider to a place of delivery according to the agreed duration and costs. It is said to be produced in a place connected to the post-condition by an output connecter.

Although the values of pre-conditions differ per value proposition, type and number of cargo are the only concepts requiring a value. Considering that 'transport' activity can be this simple, we still have to connect an activity to places with resource types. This connection can be specific for each value proposition of each service provider, although of course different service providers can use the same places (e.g. in terms of ports or other types of hubs for transshipment). By connecting places to 'transport' activities, a so-called transport network is defined. Such a network can be defined in various ways, e.g.:

- A transport activity is within a certain country, implying it is only national transport. Transport can in principle take place between any physical location within that country.
- A transport activity is in a region covering various countries or part of countries, e.g. between the Netherlands and northern part of Italy via Austria and Germany or between the EU and the Asian Pacific. The EU can also be an example of a region.
- A transport activity is defined from any place in a region and a particular location, e.g. a port. This transport activity is normally offered as inland transport in combination with sea-transport (see next example).
- A transport activity is defined between two physical locations, e.g. between two ports. By combining transport activities between various ports, a so-called voyage is defined. In this particular case, various value propositions between different ports can be defined on the same voyage[1].
- A similar approach to the last is to construct a so-called hub network. Between the hubs, that are physical locations, a transport activity is defined at a regular basis (e.g. daily between those hubs). Thus, a distribution network can be defined.

Additionally, a transport activity utilizes particular resources with a particular transport mode, e.g. vessels for sea transport and trucks for road transport. Whilst value propositions may differ for each transport mode, a customer may request a particular transport mode to be used. Considering a transport network, a business transaction has to contain the following information:

- The place of acceptance and delivery as indicated by a customer are in the geographical area that is connected to a pre-condition by an input connector and a post-condition by an output connector.
- The required transport mode is in the set offered by a service provider.
- The difference between the expected date and time of delivery and acceptance must be greater than or equal to the duration of the requested business activity.

[1] In sea-transport, a voyage most often has a port at which it starts and a port at which it ends, related to a particular time of start and end. In the meantime, the voyage passes different ports. A voyage is independent of the vessel used as an operand resource; that vessel can pass a port with different voyages in a given time period.

Thus, one could basically state that pre- and post-conditions can be simple, whereas complexity is in the network of places, connectors, activities, and resource types in those places. The structure is for all actors providing a ‘transport’ activity identical, but values will differ. The firing rule of ‘transport’ activity requires exact data of the cargo, e.g. the number and type of packages, container numbers, and other actors involved.

3.2 Transport Interoperability Ontology

Transport interoperability ontology is the specialization of the concepts introduced in section 2. Basic classes are already discussed in the previous part. The OWL document of the business system interoperability model is imported in an OWL document for transport and extended with the following concepts and class constraints for transport (see also the next figure):

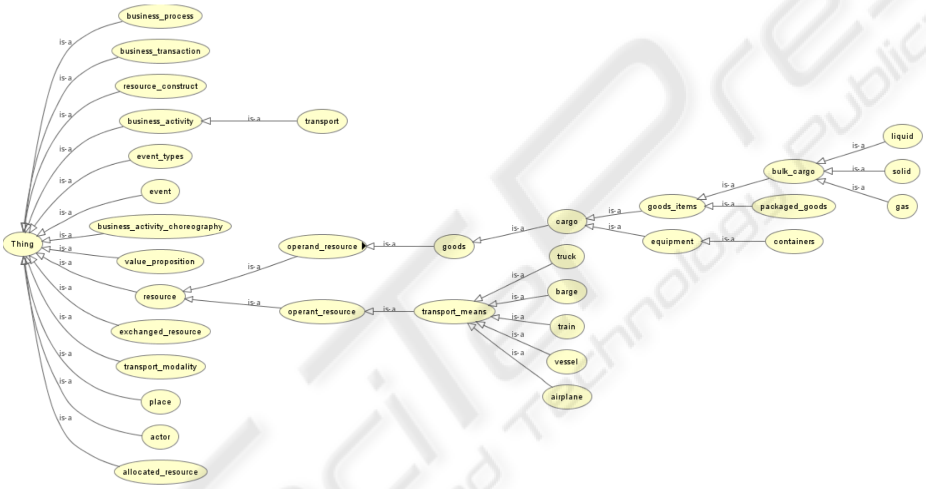

Fig. 3. Classes of transport ontology.

- Specialization of operant resources to transport means and packaging types. Transport means are further specialized to vessel, truck, barge, airplane, and train. Each of these transport means has a specific transport modality and its own characteristics, e.g. a truck has a license plate and a vessel a Radio Call Sign.
  Packaging types can be further specialized in those that are used once (e.g. cartons and boxes) and those that are used several times (e.g. containers and pallets). Note that these operant resources can also take the role of operand resources in which case the use is given to another actor (see for instance REA that defines ‘use’ [4]).
- Specialization of operand resources ‘goods’ to cargo. Cargo can be further specialized to ‘goods items’ and ‘equipment’. Goods items are further specialized to packaged goods and bulk cargo (either liquid or solid). Equipment, e.g. containers, can be both operant and operand resources: they can be used to facilitate transport and have to be allocated, and they are cargo.

- At the highest level, the concept transport modality representing a particular transport mode (sea, air, road, inland waterways) is introduced. Sea can be further decomposed into deep-sea and short sea.
- A number of specific properties is introduced like a property that a transport means has a modality (a specialized transport means like a truck should have modality 'road'). Additional properties can be defined like the fact that a certain transport means can only be used for container transport.

Fig. 3 shows the specialization of the overall model to transport. The figure only shows the classes, not the required class and data properties. It needs further extension for the support of for instance dangerous and temperature controlled cargo. Introducing these specializations also gives a number of additional properties like stowage restrictions to dangerous cargo. Further work needs to be done to complete this model.

4 Reference to Other Approaches

We have combined various concepts of other approaches to construct our ontology: Resource, Event, Agent (REA, [4]), Semantic Markup for Web Services (OWL-S, [13]), Web Service Modeling Ontology [3], and timed, colored Petrinets [14]. We will briefly discuss their relation to our framework.

REA is an accountancy framework to describe exchange of goods, money and rights between two organizations. The latter are called agents. Goods, money, and rights are called resources: a resource is a collection of rights associated with it: ownership, usage, and copy rights. An economic event is the transfer of rights associated with a resource from one economic agent to another. Resources, agents, and events can be of a type. Resources and their types are part of our model since they are exchanged between any two agents. An agent type is modeled as an actor and an event type as a business activity. Whereas REA distinguishes different event types like 'produce', 'consume', and 'use', those are part of pre- and post-conditions of 'business activity', e.g. a pre-condition consumes one or more resources and a post-condition produces them. In case there is no post-condition, a resource is said to be 'consumed'. The term 'event' in our model is restricted to information exchanged within the context of a business transaction. 'Business transaction' in our ontology seems to be identical to 'event' in REA. The approach taken in REA is to specialize these basic concepts to an application area, e.g. the production and consumption of pizza's or lending of books as illustrated in [4].

Semantic Markup for Web Services, OWL-S, defines a semantic model of all concepts required for a web service, or more generic, a service. A service has a profile (what it does), a grounding (how to access it), and a model (how it works). In our model, we do not describe grounding. A service profile defines aspects like pre- and post-conditions, input and output and a process. The process describes how to interact with a service; it can be constructed with a number of constructs like 'split' and 'join'. One of the core concepts in our ontology is 'business activity' with pre- and post-conditions, input and outputs, and a choreography. In this approach, a business activity can be compared with 'service'. However, we have taken the approach that many actors can have identical business activities, whereas each actor can have its particular value proposition. The combination of ‘business activity’ and ‘value proposition’ seems to be identical to ‘service’. Inputs and outputs are in our model defined as resource types that can be consumed or produced in places, and choreography describes the way to interact with a business activity.

Web Service Modeling Ontology is based on Abstract State Machines (ASM [8]). The core concepts are ‘goal’ to represent a customer requirement and ‘capability’ to represent a service, including their grounding to technical standards like XML Schema. Whereas OWL-S does not give guidance on the specification of input and output semantics, WSMO does by applying OWL. Goal and capability are expressed by pre- and post-conditions of an ontology. Furthermore, goal and capability have interfaces with choreography. Mediation is meant to match a goal and a capability. Thus, WSMO uses similar concepts as defined in OWL-S, but slightly different. We will not discuss differences, but one could state that similar to OWL-S our concepts ‘business activity’ and ‘value proposition’ are identical to ‘capability’. They all reflect the concept of ‘state transition’ within ASM.

Finally, we have introduced concepts from timed colored Petrinets. We define actors in terms of the business activities they can provide according to a value proposition. A business activity is identical to a Petrinet ‘process’ that has connectors to places. These places can contain tokens with a color that can be modeled by ontology. The concept ‘state’ is expressed by all tokens in all places at a given time. Thus, the concept ‘state’ is defined different from that in Abstract or Finite State Machines, where it refers to a specific object (e.g. the ‘state’ of a container). Whereas in Petrinets all processes need to be connected to places, Abstract State Machines specify the individual state transitions (the ‘processes’) on states which are arbitrary data structures. In ASM, ‘state’ can be derived from any pre- or post-condition, whereas in timed colored Petrinets, this part of state is reflected by the color of tokens that can be consumed or produced. Whereas Petrinets present (graphically) a complete model connected processes via places, ASM needs chaining of state transitions based on logical expressions defining pre- and post-conditions.

5 Conclusions and further Research

We have presented enterprise interoperability ontology for Service Oriented Computing at business level and applied it to logistics. The ontology is based on an accountancy model called REA, extended with concepts from existing service frameworks like WSMO, and defines concepts that can be shared amongst different classes of business services. It seems obvious that business transactions and their events contain all relevant information for executing a business activity, e.g. the availability of resources in a particular place. We have not discussed business activity or value proposition discovery, which differs from service discovery. It is described in more detail for a particular case [10].

There are still a number of issues for further research:

1. We have taken the basic concepts of timed colored Petrinets and applied them to enterprise interoperability for transport. However, other cases are better modeled with Abstract State Machines, see for instance [8]. Both approaches

offer identical functionality and a combination of approaches needs to be considered.

2. The concepts consider both process and semantic aspects, e.g. it models choreography as a concept and semantics of exchanged information. Choreography should be further refined, e.g. by applying process modeling approaches as defined for instance in OWL-S [13] or YAWL [16].
3. Relevant aspects like organizational issues and grounding need to be considered further. As grounding is clear, organizational issues consider the maintenance of models, service directories, etc. Furthermore, integration of semantic models of different application areas has to be considered. This is especially the case in for instance government interoperability where each government organization is responsible for (part of the) government data. Is a common model of enterprise interoperability meeting these requirements? Ontology import might offer some solution. Is it also allowed for actors to define their specific models based on a generic model? What is the impact of actor-refined models on an execution environment?
4. An execution environment should combine dynamic service mediation supported by user interaction, with automatic service mediation based on rules. What is the role of enterprise interoperability concepts in an execution environment? What would be the architecture and functionality of such an execution platform? Can this functionality be offered by for instance the DynamiCoS framework [17] or the SESA execution environment [3]?
5. There are some practical issues to consider when defining semantic models for an application area. One can for instance define a complete model for international logistics or a separate model for each modality. Another approach is to define separate models for each activity. Again, practical issues are the integration of different models to create different views on logistics, e.g. a customs view may differ from a transporters view.
6. Currently, practical applications still need the specification of the information exchanged in particular events that are of a type. Each event type only contains that specific information that is required by a particular value proposition. Further research supported by a Proof-of-Concept is required to investigate the impact in practice regarding interoperability without specific semantic models for each event type.
7. We have only applied the model to logistics in this particular paper. However, we also have examples of the application of our proposed enterprise interoperability ontology to government services [10].

References

1. T. Erl, Service-Oriented Architecture – concepts, technology, and design, Prentice Hall, 2005.
2. J. Spohrer. and S.K. Kwam, Service Science, Management, Engineering and Design (SSMED) – An emerging Discipline – Outline and references, International Journal on Information Systems in the Service Sector, May 2009.
3. D. Fensel, M. Kerrigan, M. Zaremba (eds.), Implementing Semantic Web Services – the SESA framework, Springer-Verlag, 2008.

4. P. Hruby, Model-Driven Design using Business Patterns, Springer-Verlag, 2006.
5. J. Heineke, M. Davis, The emergence of service operations management as an academic discipline, Journal of Operations Management 25 (2007) 364–374.
6. Y-H Tan, W.J. Hofman, J. Gordijn, J. Hulstijn, A framework for the design of service systems, Springer-Verlag (to appear in Service Sciences).
7. Business Process Modelling Notation (BPMN), version 1.2, january 2009.
8. E. Börger: "High Level System Design and Analysis Using Abstract State Machines", Proceedings of the International Workshop on Current Trends in Applied Formal Method: Applied Formal Methods, p.1-43, October 07-09, 1998
9. M. Holtkamp, W.J. Hofman, Semantic Web Technology – state of the art report, Internal TNO report, 2009.
10. W.J. Hofman, M. van Staalduinen, Dynamic Public Service Mediation, eGov2010.
11. J.L.G. Dietz, Enterprise Ontology, Theory and methodology, Springer-Verlag, 2006.
12. W.J. Hofman, A conceptual model of a business transaction system, Uitgeverij Tutein Nolthenius, 1994.
13. OWL-S, Semantic Markup for Web Services, W3C member submission, 2004.
14. K.M. van Hee, Information systems engineering: a formal approach, Cambridge University Press, 1994
15. H.M.W. Verbeek, A.J. Pretorius, W.M.P. van der Aalst ... [et al.], Visualizing state spaces with Petri Nets, Eindhoven : Technische Universiteit Eindhoven, 2007.
16. A.H.M. ter Hofstede, W.M.P. van der Aalst, M. Adams, N. Russel, Modern business process automation: YAWL and its support environmnet, Springer-Verlag, 2009.
17. E. Silva, L. Ferreira Pires, M. van Sinderen, Supporting Dynamic Service Composition at Runtime based on End-user Requirements, 1st International Workshop on User-Generated Services, 2009.
18. P. Hitzler, M. Krotzsch, S. Rudolph, Foundation of semantic web technologies, CRC Press, 2009.