Describing Coordination Services with REA

Hans Weigand[1], Birger Andersson[2] , Paul Johannesson[2], Maria Bergholtz[2],
Jeewanie Jayasinghe Arachchige[1]

[1] Tilburg University, P.O.Box 90153,
5000 LE Tilburg, The Netherlands
H.Weigand@uvt.nl, J.JayasingheArachchig@uvt.nl
[2] Royal Institute of Technology
Department of Computer and Systems Sciences, Sweden
ba,pajo,maria@dsv.su.se

Coordination services are services, possibly implemented as web services, that support the coordination of (real-world) services that a consumer would like to take. To support users of a future Internet of Services, the effect of the coordination services must be described in such a way that users are not only able to discover services but also to detect and prevent possible conflicts in their composition. In this paper, REA is applied as a solution approach to this requirement. The REA business ontology has proven to be a good foundation for the description of services, but we argue that its conceptualization of commitments can be improved.

Keywords: REA, Internet of Services, coordination

1. Introduction

In spite of considerable progress that has been made in the area of Service Oriented Computing, the impact on society has still been limited. There is not yet such a thing as an Internet of Services that would allow users to integrate the services they want to use easily and seamlessly. It should be noted that for users, web services are merely interfaces to "real" services such as traveling, meeting support, or child care. So, as argued also by [16], research on an Internet of Services should focus on real services and the key assets these services relate to. Real services must be distinguished clearly from software services [15, 20]. [4] argues that much work on automatic discovery of services fails to provide a viable solution as it mixes up the two concepts, and assumes wrongly that complete and correct descriptions of (real) services are available.

Fig. 1 depicts a user-centric service coordination cycle: users (or service composers addressing a particular user segment) compose mashups and interact with the widgets in them to access web services. The web service typically supports the coordination with a service provider who provides a real-world service as part of a service bundle. The service affects a resource that concerns the user (the resource could be the user himself, for instance in the case of a hotel reservation). That web services themselves

M. Petit, G. Gal, A. Castiaux, J. Ralyté, and P. Plebani (Eds.):
CAiSE 2010 Workshop BUSITAL'10, Hammamet, Tunisia, pp. 120-134, 2010.

may be composite software entities is left out of this figure as being less relevant to the user, but is of course relevant to the software developer.

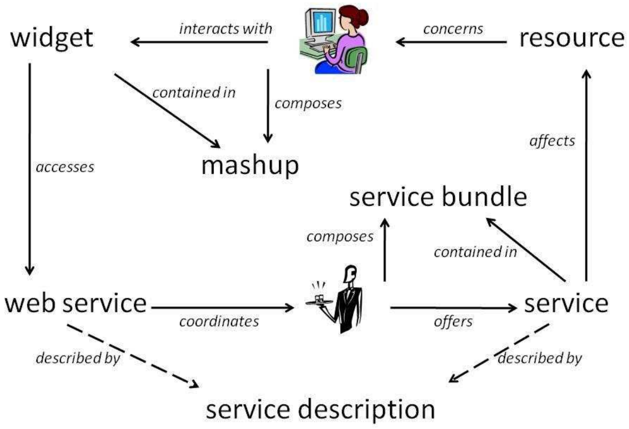

Fig. 1 User-centric service coordination cycle

Both web services and services need a description, but what should be in this description? In composing web services, a major challenge is to reconcile incompatible data representations. The *availability* of the data is not a problem, as data can be copied without virtually any limit. In composing services in the real world, a major challenge is to meet the constraints imposed by the fact that resources are scarce, can only be in one place at a time and often cannot be shared. For that reason, [16] argues convincingly that "asset-driven" service modeling will be a central concern in developing an Internet of Services and claims that "novel methodologies and tools are needed to support the modeling of the key assets of services". In our view, this modeling should be guided by at least two objectives: conflict prevention and conflict detection.

In order to make conflict prevention and conflict detection possible at all, we need a generic language to describe services, the resources they use, as well as planned and actual events on the type level, Web services can use this language to represent the preconditions and effects of the real services they connect to as well as their own semantics. A mashup environment can collect and combine this information, integrate it with other sources such as the user's agenda (that should be represented in the same format) in order to provide the user with the conflict prevention and conflict detection functionality described above.

In this paper, we propose to ground the service description language in the REA ontology [11] where we concentrate on coordination services. A coordination service is a service that supports an exchange of a good or a service [WJ09]. The use of REA has two advantages: first of all, we believe it can be a basis for user-based service composition as it is "as simple as possible, but not simpler than that". There is evidence that REA is easy to understand both for the users and for consultants and application developers [10]. Secondly, resources ("assets") are its core concept. In earlier work, it has been shown that REA can provide a unified view of services [20] - both real services and web services.

This paper is structured as follows; in section 2 we will consider the OASIS reference model for SOA and derive that describing web services is centered around the management of commitments. In section 3 we introduce the REA business ontology. We continue with a discussion about commitments in REA and what fulfillment of a commitment amounts to, and we explain our view of the notion of service in REA. In section 4 we introduce the coordination service as a service that supports an exchange process. Two common patterns of coordination services are worked out. Finally, section 5 concludes the paper with a summarization and directions for future research.

2. Coordination as a Service

According to the OASIS reference architecture foundation for SOA, it is essential that participants can use a SOA-based system to realize actual effects in the world [13]. However, when talking about the real world, OASIS makes a sharp distinction between the social world and the physical world (note that this is fully in line with the Language/Action Perspective tradition and the communicative theory of Habermas adopted there [3]). It is said that many, if not most, effects that are desired in the use of SOA-based systems are actually social effects rather than physical ones. For example, opening a bank account is primarily about the relationship between a customer and a bank – the effect of the opened account is a change in the relationship between the customer and the bank. For that reason, OASIS talks about social actions that result in social facts. “A social fact is an element of the state of a social structure that is sanctioned by that social structure”. Social facts include policies and commitments where “a commitment is a social fact about the future: in the future some fact will be true and a participant has the current responsibility of ensuring that that fact will indeed be true”. A completed business transaction establishes a set of social facts relating to the exchange; typically to the changes of ownerships of the resources being exchanged.

The OASIS model describes the relationship between communicative actions performed by means of information systems and their social effects as follows. “When we state that a communicative action counts as a service action, we are relating a system of communication to a system of action against services. Since a participant cannot (normally) act directly on a service it must use some means of mediating the action. However, from the perspective of all the participants involved, when a participant uses a communicative action appropriately, the participants are *expected* to understand the communication *as though* a service action were actually performed. When a customer ‘tells’ an airline service that it ‘confirms’ the purchase of the ticket it is simultaneously a communication and a service action – two ways of understanding the same event, both actions, one layered on top of the other, but with independent semantics”. (p.32)

OASIS is right that in most cases information systems (SOA based or not) produce social effects. Business transactions are not the result of a causal chain of instrumental actions, but of a coordination process made up by communicative actions, and information systems are well-suited to support the latter. What remains a bit out of the OASIS picture is that these social facts refer to physical world events, such as the delivery of a product. For a full account of service effects, this relationship between social facts and the real world must be made explicit (Fig. 2).

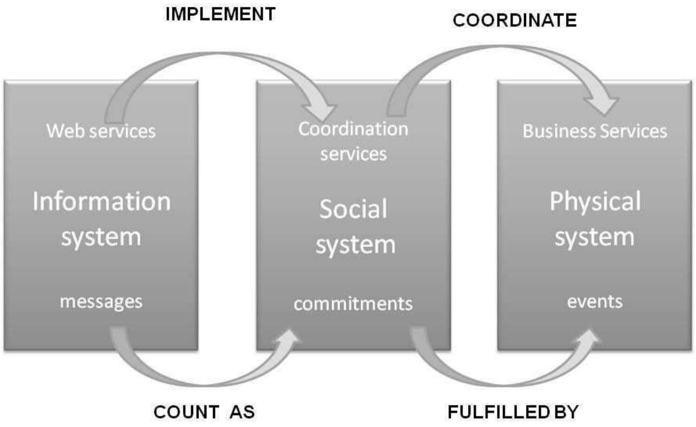

Fig. 2 Coordination services are the glue between web services and business services

In two kinds of situations, the “social effect semantics” are not relevant, or less so. First, in so-called pervasive environments, the system may contain devices that have a direct physical effect such as switching on the light [18]. This interaction is a case of instrumental rather than communicative action. Secondly, certain business transactions are fully automated, such as electronic payment, or digital music provisioning. They do involve a coordination process including a contract between service provider and customer, but the normal execution does not involve human or other non-digital resources anymore. In this case, the social effect semantics are still relevant for explaining what happens on the business level, but the social facts do not play a coordinating role operationally.

It is widely recognized that input and output descriptions of web services, or its operations, are not sufficient for capturing the semantics that users need. Precondition and Effect descriptions have been added. Although WSDL-S provides a mechanism to include these attributes, it does not give guidance on how to do specify their contents. The OASIS reference model views web services as coordination mechanisms and emphasizes the social effects. How these are to be represented, and how these social facts relate to real-world business events is still to be worked out. In the following, we address this research gap by proposing the REA ontology for coordination service description.

3. REA-based Service Description

3.1 REA background

The Resource-Event-Agent (REA) ontology was formulated originally in [11] and has been developed further, e.g. in [7, 10, 17]. It was originally intended as a basis for accounting information systems and focused on representing increases and decreases of value in an organization. REA has been extended with patterns to form a foundation for enterprise information systems architectures [10], and it has also been applied to e-commerce frameworks [17]. The following is a short overview of the core concepts of the REA ontology based abbreviated from [20].

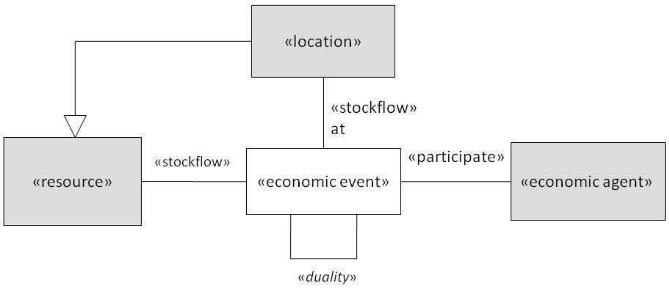

Fig. 3 REA basic categories including location. Events are rendered in white, the other objects in grey

A *resource* is any object that is under control of an agent and regarded as valuable by some agent. The value can be monetary or of an intangible nature, such as status, health state, and security. Resources are modified or exchanged in processes. A *conversion process* uses some input resources to produce new or modify existing resources, like in manufacturing. An *exchange process* occurs as two agents exchange resources. To acquire a resource an agent has to give up some other resource. An *agent* is an individual or organization capable of having control over economic resources, and transferring or receiving the control to or from other agents.

The constituents of processes are called *economic events*. An economic event is carried out by an agent and affects a resource. The notion of stockflow is used to specify in what way an economic event affects a resource. REA identifies five stockflows: produce, use, consume, take and give, where the first three occur in conversion processes and the latter two in exchange processes. REA recognizes two kinds of duality between events: conversion duality and exchange duality. Events can be assigned to a *location;* there is not a direct resource-location relationship, only via economic event. Sometimes the acronym REAL is used for REA plus location [14]. Locations can be considered as a special kind of resource, as in Fig. 3. They can be used and they may have a maximum capacity for use.

3.2 Commitments in REA

Commitments were added to the REA ontology in [7] as “important economic phenomena”, and modeled as the pair-wise connection of requited commitments. The pair-wise connection is similar to the *duality* relationship between actual exchanges or conversions but as it is not between events, REA calls it a *reciprocal* relationship. In the following, we refine and extend the commitment concept of REA by adding explicit commitment events and by rethinking the “reserve” relationship. Starting point is that we consider a commitment as a special type of resource, so that it can be handled in the same way, that is, by manipulated and used in exchange and conversion events using stockflow relationships. One consequence of this approach is that the “resource” becomes equivalent to the FASB notion of “asset” and it is not necessary to make the exception anymore that McCarthy made in [11:562].

As discussed in section 2, a commitment is a promise regarding the future. Commitments are formalized as clauses in contracts and those commitments are subsequently fulfilled through economic events. A distinction can be made between increment commitments (assets in the agent’s perspective) and decrement commitments (liabilities in the agent’s perspective) [10].

Depending on the commitment type (decrement vs. increment) the relationship of the provider to the commitment is characterized as a give or take stockflow (Fig.4). When the customer promises to pay, this means that the provider receives an i-commitment. A customer can, in a *decommit* event, take a d-commitment that is received by a provider in the same event. This represents an absolving of a commitment. Similarly, the provider can give back a previously received i-commitment.

A structure involving increment commitments can be constructed as well (not illustrated here) for the customer’s part of the contract, but still from the provider's point of view. In a commit event a provider becomes the receiver of an i-commitment (increment) through a take stockflow. The customer owes the provider. The provider can, in a decommit event, give the customer an i-commitment back, thereby cancelling the debt. Note that the customer cannot cancel this debt himself, but he can request for it. The exchange reciprocity between commitments reflects an exchange duality between commitment events.

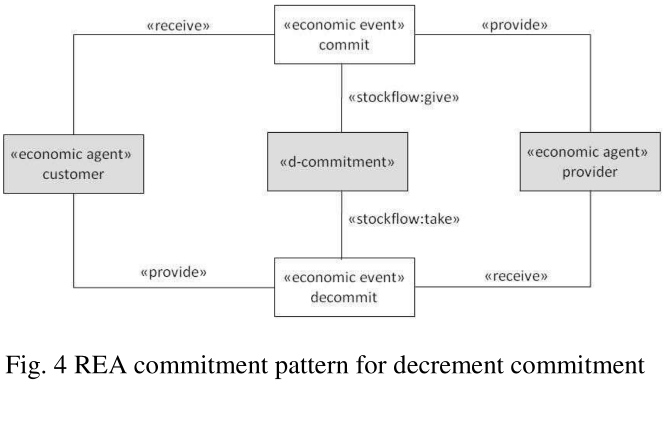

Fig. 4 REA commitment pattern for decrement commitment

Thus contract formation can be thought of as giving and taking corresponding d-commitments and i-commitments. Committing is modeled as an economic event. For standard REA, the creation and deletion of commitments are not economic events; they can be thought of as "system events" to be positioned at the implementation level. In our view, this approach has several disadvantages: first, the notion of provide/receive is ambiguous, as it is used to characterize both event/agent relationships and commitment/agent relationships. In our conceptualization, the definition is univocal. Secondly, in REA an economic event represents an "increment or decrement in the value of economic resources of the enterprise". This definition also applies to taking or giving a commitment, e.g. the commitment to a future payment. Thirdly, it can be remarked that already in [11], "claims" were discussed extensively. In the outline of the framework, resources were materialized as base objects while claims were not, but further on in the paper, it is remarked that "in practice this disparity is not always warranted". Interestingly, the paper continues by saying: "Should the accountant and database designer decide together to maintain certain claims as separate base objects, they also would have to include two additional events sets (inflow and outflow) for each one". Although commitments have been included in the REA ontology later, these "additional events sets" have not been recognized so far.

Commitments are returned in a *decommit* event. Two main types of decommit can be distinguished that maintain the duality axioms of REA. In the case of *canceling*, the commitment is returned in exchange with the reciprocal commitment being returned. For instance, a purchase order is cancelled and the payment is cancelled at the same time (of course, the contract may specify a penalty for the one who requests cancelation or even forbid cancelations altogether). In the case of *fulfillment*, the commitment is returned in exchange with some other economic event being provided, being the content of the commitment. For instance, when a delivery is made, the purchase order commitment is returned. In the following, we will adhere to the standard REA fulfill relationship as an abbreviation for this duality.

There is a second issue on which we have refined the standard REA ontology. Commitments are most often about resource *types* (e.g. a non-smoking hotel room, or a certain book title to be delivered), whereas the business transaction itself is about a resource instance, that is, a specific hotel room or a specific copy of the book. In some cases, the commitment is about the resource instance itself, e.g. in the sales contract of a house. Within the REA community, the reservation is handled in different ways. According to [7] *"Reserves* is a special kind of stockflow relationship that describes the scheduled inflow and outflow of resources". A sales order results in a reservation of the finished goods to be delivered. Gailly and colleagues [6] don't include this relationship between commitment and resource in their REA ontology; instead, they define a *specify* relationship between commitment and *resource type*. Yet another approach is followed by Hruby [10] when describing a commitment pattern. Here it is said that a commitment can be related to a resource type *and* to a resource. The commitment pattern expresses that the commitment (reservation) can be related to a resource type first, and related ("allocated") to an actual resource before the economic event starts.

Is it possible to do justice to the various positions and still have a univocal definition of reservation? To answer this question without resorting to complex intensional logics, we propose to draw on the notion of resource group [8]. Let the object of the commitment be a resource group of a certain resource type. Cardinality of the set/quantity of the resource is the most important attribute of resource group, and additional constraints can be specified. The relationship between resource group and resource type is a *policy* relationship [8]. It specifies the type of resources that may go into the resource group. In the case that a particular resource is to be reserved, the *grouping* relationship is already made at commitment time. In all other cases, it is specified later when the purchase contract is being executed. Fig. 5 presents the revised "reserve" relationship.

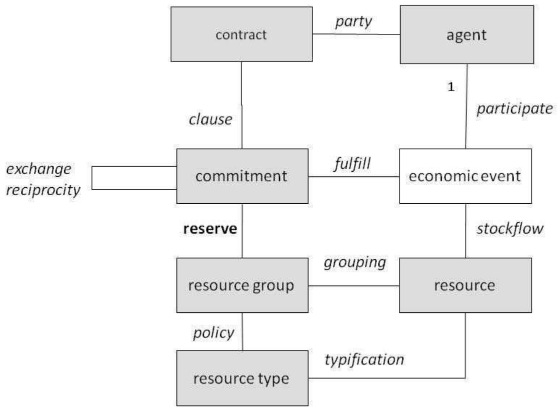

Fig. 5 REA relationship "reserve" revisited

For example, a reservation of two hotel rooms is formalized as a commitment being a clause in a contract. The object of the commitment is a group. The policy for this group says that it must contain "double non-smoking rooms", and 2 of these. In the economic event that fulfills the commitment two double rooms (resources of the specified type) are allocated to this group.

3.3 Services in REA

In REA, a service is a resource as it is viewed as valuable by some agent and can be transferred between agents [20]. As such, it inherits all features of resources, in particular that it can be exchanged between agents, that it is governed by a contract and that it is part of a conversion process chain. Although a service cannot be owned – the customer cannot resell it, only the right on a service may be resalable, as in the case of a hotel coupon – he does have a certain control over it and has a right to make use of it for some time. A service is produced by one agent for another agent using certain resources or capabilities. The production and consumption of the service are not independent events, as in the case of goods, but occur simultaneously. As a consequence, the customer participates in the service execution and a service is a typical example of co-creation of value.

As depicted in Fig. 6, the service is exchanged between agents in return for money (top right cluster). All the coordination services that can be used within an exchange process apply to service exchanges as well; we will use this feature below. At the same time, the service is a resource produced in a conversion process by the provider (top left cluster), and consumed in a conversion process by the customer (bottom cluster). REA usually renders only one agent perspective, but for the understanding of the service interfacing between the provider and customer, we have included both perspectives (indicated by dotted rectangle) in one figure. Note that Fig. 6 is simplified in order to reduce clutter. In particular, the usual agent boxes are not present in top left or bottom.

The economic increment event for creating the service stands in conversion duality to one or more resource use events. For example, a hotel service is realized by *using* the hotel room resources. At the customer side, we distinguish between service use and service consumption. Both can be used to add value (production event) to some target resource, typically in combination of some effort of the customer himself – that is why we also include a resource use event here. However, in the case of service consumption, the service is no longer available after the event, whereas service use draws on the existence of the service without changing its status. The service consumption may be conceived of as an atomic event or as a process over time. The latter is especially the case when the service is offered for a certain period. Then for economic purposes, the amount of service consumption is typically linear on the time having passed by.

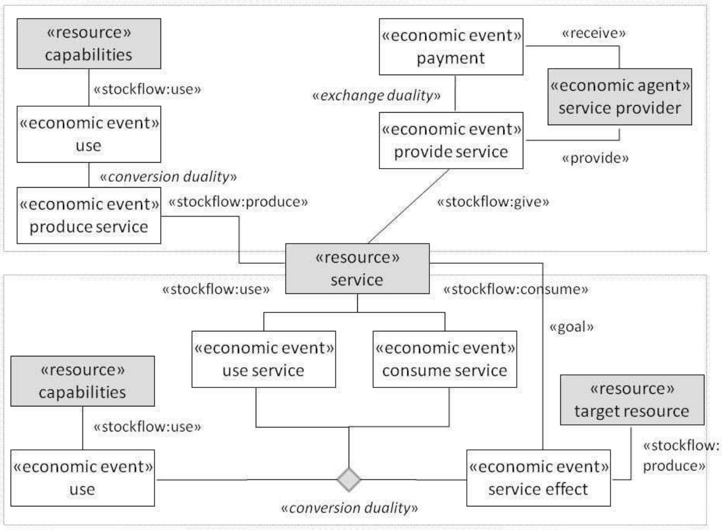

Fig. 6 REA application pattern for Service Exchange.
It is assumed that the «goal» stereotype is defined in the REA meta-model

The difference between service consumption and service use can be illustrated by the example of [5] of a fire brigade. This could be a service hired by a municipality for a certain period. Service consumption is here a matter of time: at the end of the period, it is completely consumed. During the period, the fire brigade may become active in the case of an emergency, as stipulated in the service contract. This is service use. The effect of service use is a particular house (resource) being rescued, whereas the effect of the service consumption is the increased security of all houses in town (resource). Security of the resource in this case does not mean that no accident could happen but that the damage will be limited.

If we want the rescuing to be modeled as a service (e.g. because it can occur independently from the overall fire brigade service), this can be represented by coupling the service use to an event that creates a right on such a rescuing service. So then the service goal is no longer "rescue", but "create right on a rescuing service".

For the user-centric description of a service, the "*goal"* is important [WJ09]. A service aims to produce an effect on resources of the customer in such a way that the value increases. If the effect is not reached, this may cause the transaction to fail. Formally, the goal relationship can be seen as an extension of the REA meta-model. However, as it can also be seen as a derived relationship, since it is defined as "the production events at the customer' side that stands in conversion duality with the service use and consumption". When also the *consumption* events at the customer and provider side are relevant, we could add a "source" relationship, analogous to the "goal" relationship. Together, source and goal provide a reference to all resources affected by the service execution. As the description of all kinds of failures and exceptions is never exhaustive, we refrain from including that in the effect. It can be specified in the contract.

For web services and similar software artifacts to deserve the label "service", the service model elements should be clear. What is the goal of the web service, that is, what resources does it create or affect that have value to the client? Who are the actors involved in the exchange process? In the next section, we will consider coordination services as one important subclass of web services.

4. Coordination services

Coordination services are defined in [20] as services supporting an exchange process (a set of events) for a good or a service alike. Processes like identification, negotiation, order execution and after-sales take place in both cases. We introduce the notion of coordination object for the object of these processes: what *is* negotiated and executed? Well, the central coordination object in an exchange process is the purchase order, not in the sense of a document, but as the commitment to deliver, to be fulfilled by the exchange event. Complex processes can include more coordination objects. Reservation and appointment are two coordination objects that reoccur often, especially when services are concerned. The reason for that is simply that the delivery of a service affecting resources from both the provider and customer to be present at the same time and place requires more coordination than the delivery of a good.

In terms of REA all coordination objects can be specified in terms of commitments. Therefore, another way of characterizing coordination services is to say that these services manipulate commitments (their goal is to give, take and fulfill commitments). We assume that for all coordination objects there is a negotiation and contracting process first followed by an execution and evaluation process, that is, the coordination process per coordination object takes the form of a “Conversation for Action” [22, 3, 9]. The message exchange in these conversations is not in the scope of this paper, but what is important is the effect of these conversations, since that is directly relevant for a user composing and using a certain mashup application.

Strictly speaking, a reservation in REA is any relationship between a commitment and a resource group. In the following, we use the term “reservation” more specifically for a commitment that *precedes* the purchase order (this is how the term is used in common speech when we talk about hotel reservation, for example). From an economic point of view, the main objective of this kind of reservations is to reduce uncertainty about the business transaction – to mitigate the risks involved, such as items being out of stock or functionality not available, or to reduce the need for slack [19]. So although the reservation has some costs in the form of less operational discretion, it increases the total value for both customer and provider.

The model in Fig. 7 contains and relates two coordination objects: reservation and purchase order. The reservation is a relationship between the reserve commitment and a resource set, being the resource set specified (purchased) in the intended purchase contract.

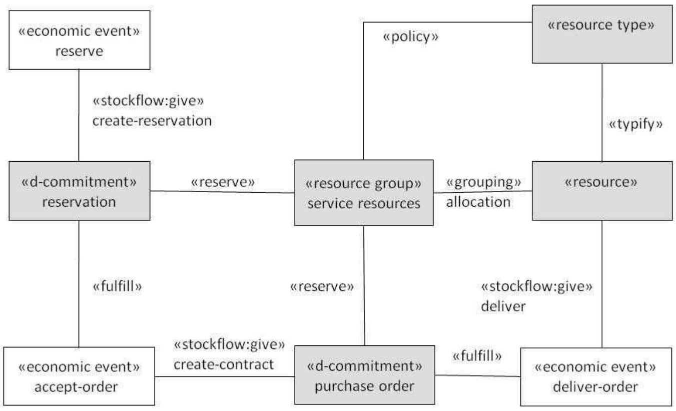

Fig. 7 REA Application Model linking “reservation” and “purchase order” coordination objects

A reservation is fulfilled by one or more economic event, but which events? It is consistent to model the fulfillment as the creation of the purchase contract (usually one, but could be more). In other words, the reservation commitment is fulfilled and ends at the moment of ordering – when the two agents engage in a purchase contract. There may be some time between this engagement and the actual realization (the good being delivered, the service being fully consumed), but in this period the coordinating role of the reservation is taken over by the purchase contract, and the reservation is not relevant anymore. If a delivery problem arises, the other party will fall back on the purchase contract, not on the reservation.

It should be noted that although the meaning of reservation and purchase order is quite stable over different domains, these two coordination objects are not always applied in the same way. In the case of a hotel, the purchase order is made when the customer checks in. At that moment, the reservation, if any, is fulfilled. In the case of a flight ticket, the purchase order is made when the ticket is sold, typically long before the check-in at the airport. What happens at the check-in is the allocation of a specific resource (a chair with a number). Sometimes, it is possible to take an option on a flight ticket for a few days before buying it. That is a case of reservation.

The complete reservation pattern is represented in Fig. 8 It shows the reciprocity relationships with other commitments that are grouped together in a contract.

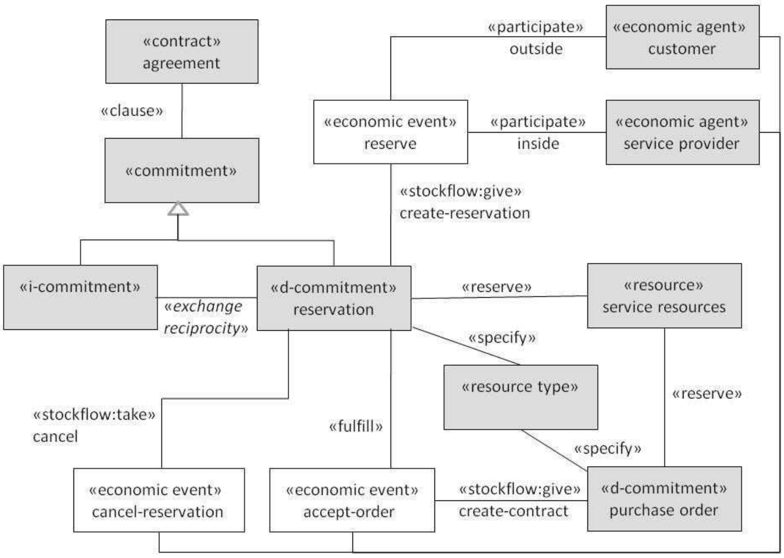

Fig. 8 REA Application Model for Reservation

An *appointment pattern* is used when two or more agents want to meet at a specific location. Appointments can be made for their own sake, but can also be part of a purchase contract, for example, when customer and provider have to agree on where to deliver the service or good. Fig. 9 shows an application model for show-up appointments where the commitment is from the side of the customer (so it is an i-commitment), typically reciprocal to an appointment of the other party to be there as well. Since the appointment includes at least a resource (the customer himself, or some resource related to the customer; and the location) there are two “reserve” links. In accordance with our “reserve” ontology, these links point to groups that specify the reservation on an abstract level and that are populated at some time with specific instances.

We have described three coordination objects, corresponding to three coordination services. Although they capture perhaps 80% of business coordination, the question whether more coordination services exist remains open and should be addressed by both formal and empirical research. For instance, according to [1] it may be important for the business in certain cases to reserve (lock) not only resources, but also functionality or agents, in the sense that it will be guaranteed that the agent remains in existence or the functionality being offered. In the same paper, it is argued that agents not only have a need to commit but also to check. For instance, does the service that is described in this registry still exist? A check service does not change the social world, but it does change the cognitive status (subject world) of participants. A fully comprehensive set of coordination services is still to be determined.

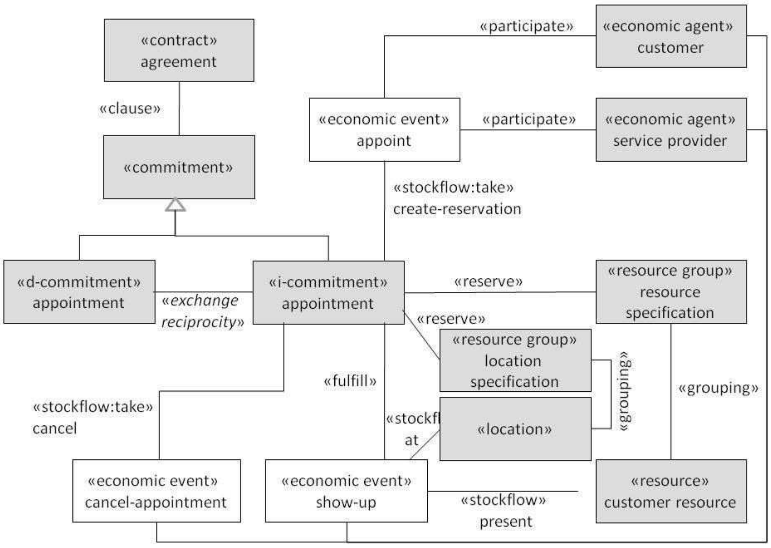
«contract» agreement
«clause»
«commitment»
«participate»
«economic agent» customer
«economic event» appoint
«participate»
«economic agent» service provider
«stockflow:take» create-reservation
«d-commitment» appointment
«exchange reciprocity»
«i-commitment» appointment
«reserve»
«resource group» resource specification
«reserve»
«resource group» location specification
«grouping»
«grouping»
«fulfill»
«stockflow:take» cancel
«stockfl at
«location»
«economic event» cancel-appointment
«economic event» show-up
«stockflow» present
«resource» customer resource

Fig. 8 REA Application Model for (show-up) Appointment

5. Conclusion

From an end-user perspective, coordination services form an important service class (cf. [2]), as these services allow the user to manage real services that matter to him/her. When using these services, he should be aware of the real-world effects, to detect and prevent possible conflicts with his own agenda (already existing commitments) or the agenda of other resources involved. In this paper, we have explored how REA can be used to describe the effect of services in general and of a representative set of coordination services. What this means for the IOPE (input-output-precondition-effect) parameters of a service description [23] is worked out in detail in a companion paper [21].

The focus of this paper has been on the description of coordination services which are the services that support an exchange process (a set of events) of a resource. Creating, executing and evaluating commitments is done in a combination of informational and material processes (in the sense of [12]). The Language/Action Perspective ([3, 9]) has explored a couple of standard micro-patterns on which the informational processes can be based. However, in this paper we have focused on the essential business level that abstracts from process and implementation aspects.

Although we adhere to the REA way of denoting the "fulfill" relationship, we have argued that it can be interpreted in terms of stockflow relationships when we distinguish a "take" commitment event that stands in exchange duality with the economic event that executes the commitment. At first sight, this event may seem superfluous, but what it expresses is that for the fulfillment of a commitment it is not only necessary that the commitment is executed, but also that the customer accepts this as execution of the fulfillment – and therefore absolves the claim. At this point we do not agree with [11] who interprets claims in terms of imbalances in the economic exchange only, for instance, when money has been received but the goods have not been shipped. When there is a contract, the other party has a claim independent of the time of payment. It is possible that a party balances one duality (e.g. shipping a good for which a corresponding payment was made) without balancing the other (e.g. because the shipping is not according to contract).

We see at least two directions for future research; the first line involves a deeper investigation of our proposed extension of the REA ontology regarding the 'resource-resource group-resource type' construct. The main rationale for this construct was to understand and model the notion of reservation while reconciling some previous proposals of solutions for the same problem.

The second line of research concerns the relationship between coordination objects and rights. The business scenario that we have described in section 4 assumes that first a commitment to reserve some resource is created (a hotel room, say). At the same time, a commitment to show-up is created. Subsequently, upon arrival the reserve commitment is returned and a new commitment, involving a specific hotel room is created. This second commitment is returned when the guest is satisfied with the hotel service having being delivered. What is interesting to explore is the interpretation of the commitments in terms of rights. When an agent commits (d-commitment), he gives away some right on the resources involved, which assumes that he did hold that right before. REA posits a "control" relationship between agents and resources. This control can be made more precise in terms of rights (ownership, custody, discretion). Viewed in this way, an economic exchange event represents not so much a change in the value of the resources but a change in the rights of the enterprise on the resources.

References

1. Artyshchev S, Weigand, H. Interoperable transactions for E-Business. Paper presented at 16th IFAC World Congress, Prague (2005).
2. Benatallah, B., Casati, F., and Toumani, F. Web Service Conversation Modeling: A Cornerstone for E-Business Automation. *IEEE Internet Computing* 8, 1(2004).

3. Dietz. J. Enterprise Ontology - Theory and Methodology. Springer, Berlin (2006).
4. Fensel D. et al. WWW or What is Wrong with Web Service Discovery. In: W3C Workshop on Frameworks for Semantics in Web Services, Innsbruck, Austria (2005).
5. Ferrario, R, Guarino, N., Fernandeze Barrera, M. Towards an Ontological Foundation for Service Science: the Legal Perspective (2009).
6. Gailly, F., Laurier, W., Poels, G. Positioning and Formalizing the REA enterprise ontology. Journal of Information Systems 22, 219-248 (2008).
7. Geerts, G., McCarthy, W. The Ontological Foundation of REA Enterprise Information Paper presented at the Annual Meeting of the American Accounting Association, Philadelphia, PA (2000).
8. Geerts, G., McCarthy, W. Policy-Level Specifications in REA Enterprise Information Systems. Journal of Information Systems Vol. 20 Issue 2, p. 37-63 (2006)
9. Goldkuhl, G. Action and media in interorganizational interaction. Comm.. ACM 49, 5 pp.53-57 (2006).
10. Hruby, P. Model-Driven Design of Software Applications with Business Patterns. Springer Verlag (2006).
11. McCarthy W. E., The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review (1982).
12. Mika, P., et al. Foundations for Service Ontologies: Aligning OWL-S to DOLCE. In S. Feldman et al. Proc. WW'04, ACM Press, pp.563-572 (2004).
13. OASIS Reference Architecture Foundation for Service Oriented Architecture 1.0. Available at http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-cd-02.pdf (2009)
14. O'Leary, D., REAL-D: A Schema for Data Warehouses, Journal of Information Systems, Volume 13, Number 1, pp. 49-62 (1999).
15. Preist. C., A Conceptual Architecture for Semantic Web Services. In F. van Harmelen S.A. McIlraith, D. Plexousakis (eds.), The Semantic Web ISWC 2004: Third International Semantic Web Conference, Springer, LNCS 3298, (2004).
16. Pistore, M., Traverso, P., Paolucci M., Wagner, M. From Software Services to a Future Internet of Services. In: G. Tselentis et al (eds), Towards the Future Internet. IOS Press, (2009).
17. UN/CEFACT Modelling Methodology (UMM) User Guide. Available at http://www.unece.org/cefact/umm/UMM_userguide_220606.pdf (2003)
18. Urbieta, A, et al. Towards Effect-based Service Description And Integration In Pervasive Environments. Proc. SIPE'08, ACM (2008).
19. Weigand, H., & Heuvel, W.J.A.M. van den. A conceptual architecture for pragmatic web services. In M. Schoop, A. de Moor, & J. Dietz (Eds.), Proc. 1st Int. Conf. on the Pragmatic Web (pp. 53-66). Heidelberg: Springer-Verlag (2006).
20. Weigand H., Johannesson, P., Andersson, B., Bergholtz Value-based Service Modeling and Design: Toward a Unified View of Services. Proc. CAiSE'09, Springer, pp.410-424 (2009).
21. Weigand H., Johannesson, P., Andersson, B., Bergholtz, Jayasinghe Arachchig, J. Closing the User-Centric Service Coordination Cycle. Proc. CAiSE 2010 Forum. (2010).
22. Winograd, T, Flores, F. Understanding Computers and Cognition – A New Foundation for Design. Ablex (1986).
23. WSDL 2.0 (W3C Recommendation 2007). http://www.w3.org/TR/wsdl20-primer/