Coordinating Variable Collaboration Processes in Logistics

Meijler, Theo Dirk; Stollberg, Michael; Winkler, Matthias; Erler, Kevin

SAP Research

Chemnitzer Strasse 48, 01167, Dresden, Germany

theo.dirk.meijler@sap.com

Abstract:

In current trade and logistics, "agility" that is reacting quickly to changing circumstances, needs and offerings is essential. This paper describes a Collaboration Hub that partners can use on-demand to coordinate their collaboration. Especially, it focuses on the what and how of advanced collaboration support, including active coordination and data sharing, and ad-hoc business user flexibility, allowing the overall collaboration process to be ad-hoc defined. Our contribution lies in improvement in advanced collaboration support as well as process flexibility which are both essential to achieve agility. Our approach is based on the so-called "entity-centric" process modeling and execution, which integrates data and process.

Keywords:

Collaboration, ICT-based coordination, Data sharing, Entity-centric, ad-hoc variability

INTRODUCTION

In current supply chains, movement of goods and therefore logistics plays an essential role. Complex products may require combining components from different origins all over the globe. In logistics, agility is essential: logistic service provider and (other) logistics partners must be able to flexibly react to changing circumstances. This is due to the need to react to new needs and offers on the transportation market, but more relevant, as logistics is known to have high risks (Hamadi and Leitz 2008), due to the need to handle issues during the transport.

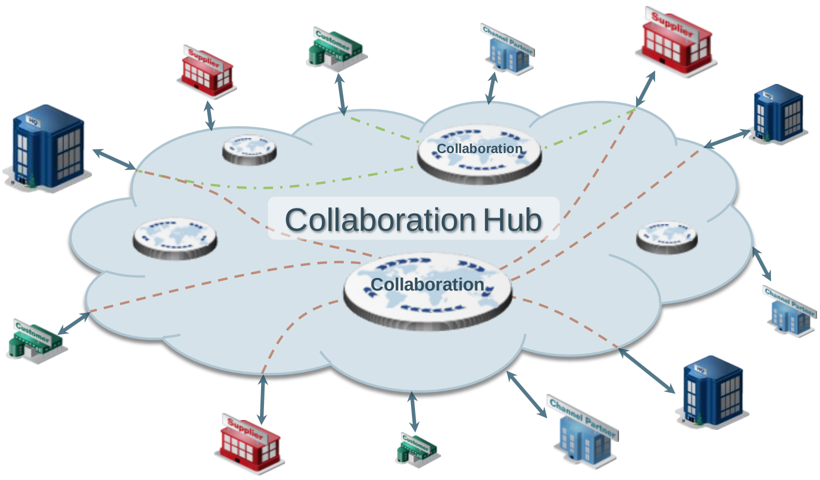

Figure 1 A Collaboration Hub that can be used on-demand

An IT-based collaboration platform can play an important role in realizing such agility. Firstly, it can support collaboration as such, in enabling partners to communicate, enabling partners to have insight in the status of the process and know what they can or have to do next. Thus, the collaboration process is visible to the partners, and partners can (re)act as needed.

Secondly, it can support *adapting* the collaboration process. In logistics, the process how to fulfill an order strongly depends on the order itself and circumstances. A logistics process *variant* can be expressed amongst others, in terms of the route and mode of transport for the transportation steps (so-called "legs"), as depending on the location of origin and destination, the type of the goods to be transported, the required speed, costs etc. Ideally, furthermore, the partners that are involved may be selected as needed, thus collaboration support should not tightly couple partners.

While most current IT solutions to collaboration either require some central partner (e.g., the logistics service provider) to set up and maintain the platform or need locally distributed gateways (Medjahed, et al. 2003), it is our vision to build a "*Collaboration Hub*" (CH) that supports large, but also smaller organization to collaborate in a relative ad-hoc manner, allowing organizations to use the hub "on-demand" (see Figure 1). We present here our first results to realize this vision.

Generic technologies for inter-organizational collaboration systems are often situated in the area of workflow management, examples are WISE (Lazcano, Alonso, Schuldt and Schuler 2000) and eFlow (Casati, Ilnicki, Jin and Wangler 2000). Wende et al. (2006) provide an overview. These technologies however, have shortcomings for our purposes. Firstly, workflow technologies are mainly used for coordination, not for sharing data as well, leading to a lack of visibility in the supply chain, vd Aalst, Weske and Grünbauer (2005) call this the "context tunneling" problem. Secondly, even though quite some work has been done on supporting workflow flexibility (Burkhard and Loos 2010), ad-hoc process variant definition is either not possible, as pre-defined templates must exist e.g., (Adams, ter Hofstede, Edmond and vd Aalst 2006), or expect users to completely define the new collaboration process schema e.g., (Casati et al. 2000). Collaboration environments such as CARAMBA (Dustdar 2004) and CTM (Stoitsev, Scheidl, Flentge, and Mühlhäuser 2008) that do support ad-hoc composition of activities have the disadvantage that the substructure of such activities is completely free. Thus, a business user who wants to focus on ad-hoc specification of the optimal logistic process variant in terms of its legs would also need to define the complex collaboration processes between partners around the individual transportation legs, such as reserving, tracking and tracing the actual transport, handling issues etc.

Following Hull et al. (2010), we turn to the so-called "entity-centric[1]" approach for realizing collaboration support. The contribution of this paper is firstly, that it introduces a more extended vision of an on-demand collaboration hub, including (logistics) domain based requirements such as ad-hoc composition. Secondly, in contrast to previous papers on the entity-centricity, we represent *collaborative activities*, such as transport and transport leg as business entities. We therefore come up with two process abstraction levels: The higher abstraction at which the business user views and composes a logistics process in terms of collaborative activities such as transport and transport legs, and the lower abstraction level where the complex collaboration processes and data sharing around these collaborative activities are expressed through the definition of the corresponding business entity types.

Our overall contribution therefore lies in answering the following research question: How can improved collaboration support, integrating active process coordination with data sharing, be combined with the possibility to let a business user ad-hoc compose the overall process variant in terms of Business-user relevant "collaborative activities" (e.g. transportation legs)?

[1] Also called „artifact-centric" in previous papers

Our research methodology is based on the so-called design methodology (van Aken 2004), where the result is a so-called technological rule[2]: A problem is described in terms of requirements, and based on real-world scenario(s). We intend to show that our solution indeed provides technology rules that solve the problem by showing how it fulfills the requirements. Our way of showing this in this paper is mainly explanatory.

The rest of this paper is subdivided as follows: In the next section we describe a motivating scenario, on basis of which we describe user requirements for a collaboration hub. After introducing "business entities" following the entity-centric approach, we subsequently build on top of that and introduce our main concepts. We provide some details how our Collaboration Hub is implemented, and subsequently explain how our user requirements are fulfilled. In our conclusion we also give an outlook of further work.

MOTIVATING SCENARIO

The motivating scenario is based on input from SAP domain experts in the area of logistics and supply chain management, thus based on years of experience of working with customers of SAP using this kind of software.

There is an exhibition of cars in Detroit, starting at June 21st. At this event, the company "Detroit Cars" and a car manufacturer intend to exhibit and sell cars from the manufacturer, specifically cars produced in Dresden. The cars to be exhibited in Detroit will be directly sold there, against a reduced price by "Detroit Cars", thus Detroit Cars will buy the cars from the manufacturer. The cars will therefore be transported from the car manufacturer, who is the *consignor* (aka "shipper") of the transport, to Detroit Cars, who is the *consignee* of the transport.

The car manufacturer will organize and pay for the transport of the cars. This transport is relative non-routine, with a limited number of cars –three– to be shipped, and all aspects of the transport need to be uniquely organized for this event. The manufacturer therefore decides to ask a freight forwarder (FF) called "German Forwarder" to organize the transport.

The FF –in agreement with the manufacturer– organizes the transport in terms of four so-called transportation "legs" using four so-called *carrier* organizations: Firstly the cars will be moved by tram from the factory to the manufacturers' logistics center of the manufacturer in Dresden Friedrichstadt, this is organized by the manufacturer, the local public transport organization will be the carrier. From there, the cars will be transported by train to Bremerhaven, the train company is the carrier for this leg and from Bremerhaven by Ship to New York by a company called "ShipCars". From New York, the cars will be moved by truck to Detroit by a carrier called "New England Logistics". In such a scenario other partners are involved as well, such as banks, insurance companies, customs, harbor terminals etc.

Through this scenario, the following user requirements can be motivated:

1. *Partners need to have access to information about the process*, for example information about the goods (cars in this case) where they come from, who owns them, where they are supposed to go etc. Both Volkswagen and Detroit Cars need to have access to the status of the transport, especially where the cars are located at a certain point in time, and what the expected time of arrival is.
2. *Partners need to be actively coordinated*. This is to take care that the process continues and, that the process status is maximally up to date. One example of such

[2] a chunk of general knowledge, linking an intervention or artefact with a desired outcome or performance in a certain field of application.

active coordination is requesting a partner to respond to an invitation to take part in the process.

3. *Multiple partners must potentially be involved in a collaboration, with a potentially complex collaboration process*. For example for the transport as a whole Volkswagen and the FF are involved, also an insurance company and customs of both Germany and the USA.
4. Partners must be able to invite partners, who can again invite other partners to take over responsibilities. Thus *"viral" collaboration must be supported*.
5. *A (business) partner must be able to ad-hoc define the overall process variant*. The FF needs to be able to handle the transport process based upon the origin, destination, the kind of goods, duration and cost requirements etc. and define the best process variant in terms of logistic legs. Note that ad-hoc process adaptation due to issues such as delay lies outside the scope of this paper.
6. *Collaboration protocols between partners about the handling of essential parts of the process must be fixed*. This is for example the case for the handling of a single transportation leg. This allows for fixed private processes of the partners (Wende et.al. 2006) and corresponding fixed links with back-end systems. Further technical details about this aspect also lie outside the scope of this paper.

BACKGROUND

This paper builds on top of work that has been done by Hull et al. (2010) in the domain of entity-centric modeling and execution. This approach has also been called: "Artifact-centric" modeling and execution in previous papers. The Entity-centric approach is a relatively new data-centric approach to modeling and executing business processes, where the so-called business entities (the term business entity will be abbreviated with "BE") are the main data objects around which organizations revolve. A BE has a lifecycle that describes its behavioral model (process), including the stages it can be in during business operation. The most useful feature of the entity-centric approach lies in the natural combination of data and process.

In the paper (Hull et al. 2010) a modeling language called GSM[3] has been formally defined that is more powerful[4] then final state transition machines for describing the lifecycle of a BE. They also introduce a record like notation for describing the data contents of a BE, however, for simplicity reasons, we stick here to the standard UML class notation. As we will use the GSM notation throughout this paper, it will be briefly introduced along a relative simple example.

The example concerns a so-called "Role Fulfillment", see Figure 2. A role-fulfillment concerns the data and (simplified) process for inviting partners to a collaboration. For more extensive, realistic BE types for partner invitation, we refer to (Hull et al. 2010). The example already illustrates in its most primitive form how in our work BEs are used to actively coordinate processes.

As shown in Figure 2b for a Role Fulfillment BE two partner roles are defined: an Inviter, the partner that invites another partner, and an invitee. In its (other) data, the role fulfillment refers back to the BE for which that partner will overtake the role, it furthermore describes which role must be fulfilled.

[3] Guard-Stage-Milestone

[4] In the paper it is described as follows: "GSM lifecycles are substantially more declarative than the finite state machine variants, and support hierarchy and parallelism within a single entity instance."

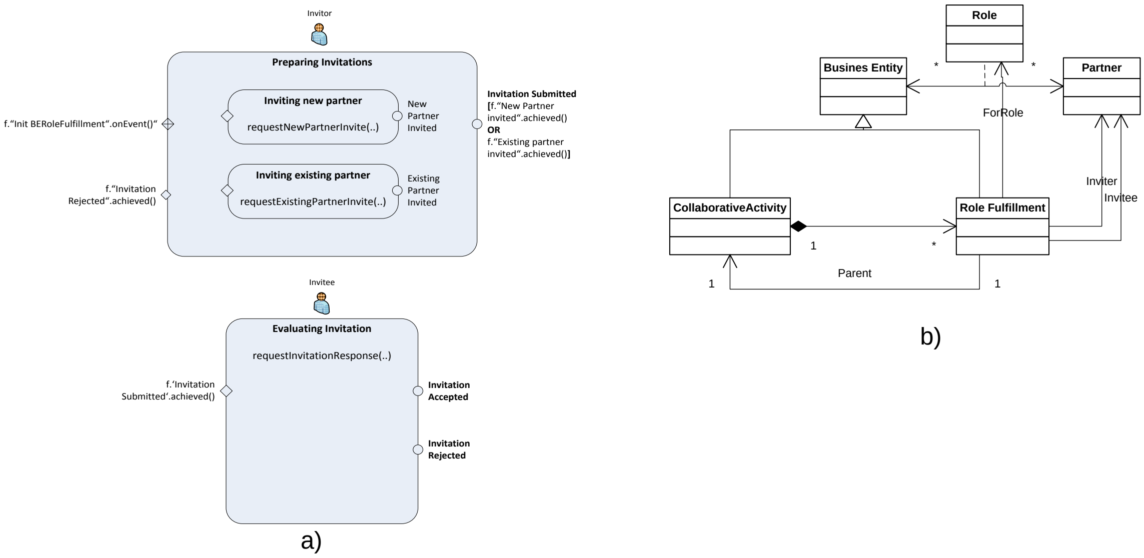

Figure 2 a) GSM model of a „Role Fulfillment“ business entity. b) Its UML data definition

In a GSM model as presented in Figure 2a, the main element is that of a so-called *stage*. The Role-fulfillment has two main stages “Preparing Invitation” and “Evaluating Invitation”, in the first stage the Inviter can invite an Invitee, in the second stage the Invitee can evaluate the invitation. A stage has a *milestone*, which represents a business relevant objective of the stage. For example the stage “Evaluating Invitation” has two possible milestones: “Invitation Accepted” and “Invitation Rejected”. A milestone is expressed using a condition over the information model and possibly a triggering event (Hull et al. 2010). A stage has a *stage body*. The stage body can be realized in terms of a service call, as for the Stage “Evaluating Invitation”, which calls the service “requestInvitationResponse”. Through this service call the partner receives a so-called action item that (s)he must respond to, (s)he must either accept or reject the invitation. A stage body can also exist of *sub-stages*, for example the “Preparing Invitation” has two (alternative in this case) sub-stages “Inviting new partner” and “Inviting existing partner”. A stage has a guard i.e., a condition and possibly a triggering event that, when achieved, enables entry into the stage body (Hull et al. 2010). In our example the stage “Preparing Invitation” has two guards: The first guard (indicated by an extra cross), is the one connected to the instantiation of a new Role-fulfillment BE. The second guard achieves repetition, as it triggers when the Invitee has rejected the invitation.

MAIN PRINCIPLES

In the following we describe the main relevant and innovative principles of our work.

1) Collaborative Activities

Figure 3 and Figure 4 represent the GSM models for the “Transport leg” and “Transport” BEs. Both are so-called “collaborative activities”. Figure 5 presents the corresponding data model showing how a Transport is expressed in terms of Transport legs, the various roles of partners collaborating through these BEs, and (a relevant subset of) shared data. The GSM model describes how partners collaborate on these collaborative activities. For example:

For Transport:

1. First the consignor invites an organizer to overtake the responsibility of handling the transport. In the scenario this is done by the FF “German Forwarder”.
2. Subsequently, the organizer plans the transportation legs. Note that consignee and consignor can have access to the Transport BE data, and thus get insight in the plan.
3. Subsequently, the legs will be created and initialized (init event on Transport leg).

4. The transport is finished when the last transport leg has arrived.

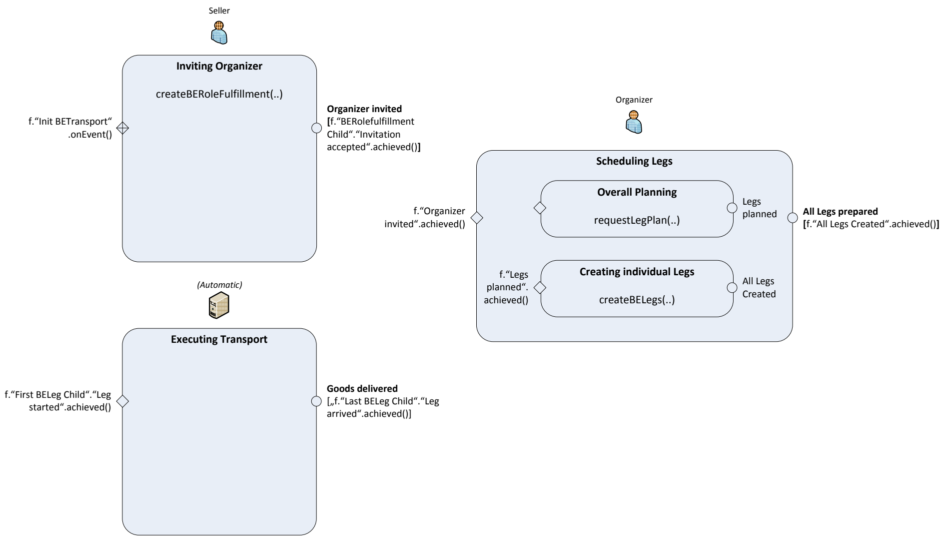

Figure 3 GSM Model for Transport

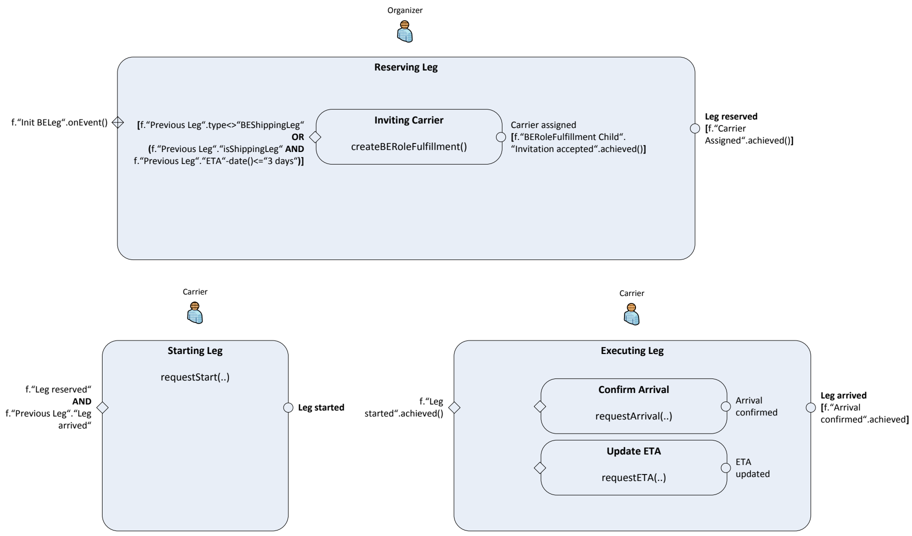

Figure 4 GSM Model for Transport leg

For Transport leg:

1. The organizer firstly invites a carrier who overtakes the responsibility for the transportation leg.
2. Subsequently the leg is executed by the carrier. The hub:
    a. Waits for the start of the actual transportation on this leg.
    b. Once the leg is really executing, the leg is tracked, i.e. the expected time of arrival and its final arrival is tracked.
3. The leg is finished executing when the arrival is signaled by the carrier.

Collaborative activities are simplified, partners such as customs and insurance have been left out.

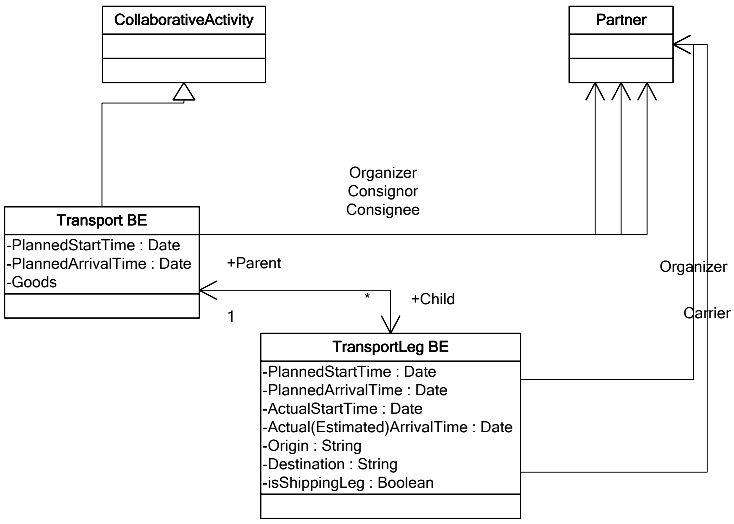

Figure 5 UML Definition of Transport and TransportLeg

2) Active Coordination

A typical aspect for this approach is that *action items* (AIs) are sent to partners. Partners are therefore actively asked to respond to that action item.

Examples of such sending of action items are:

1. For the BE: Role Fulfillment, through the “requestInvitationResponse” service call the partner is sent an AI to either accept or reject an invitation
2. For the BE: Transport, through the “requestLegPlan” service call the partner is sent an AI to create a plan

3) Two process abstraction levels

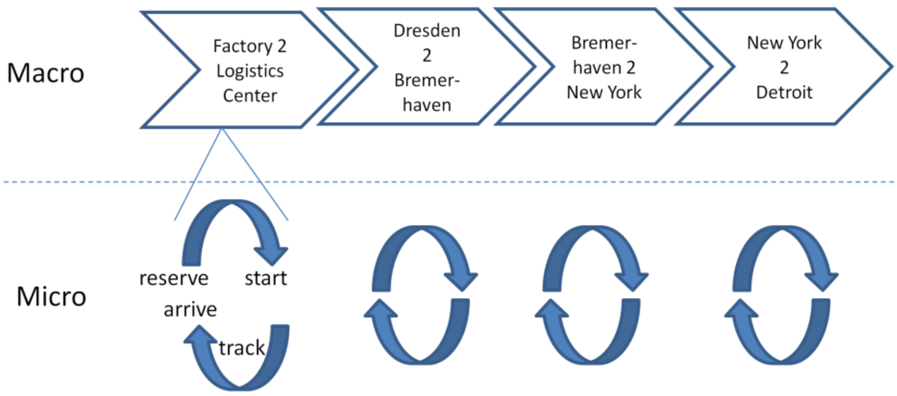

Figure 6 Two process abstraction levels, the macro level (top) and the micro level (bottom)

A BE has two (process) abstraction levels as illustrated in Figure 6:

1. The lifecycle as presented before, and shown for example in Figure 3 and Figure 4. These represent the lower or “micro” process abstraction level of the BEs Transport and Transport leg respectively.
2. The BE structure. The structure of the Transport BE in terms of the transport legs as based on the data model (Figure 5) represents the higher “macro” process abstraction level. Note that the sequence of transport legs indeed represents a (executable) process as will be discussed below.

In the Planning stage of the Transport BE lifecycle, the business user creates a planning for the transport. As a result the process at the macro abstraction level is created, and can subsequently be executed.

4) Synchronization points

Execution of the transport process at the macro level means for our example that one transportation leg is coordinated after the other. This process flow *across* transportation leg BEs is achieved through so-called synchronization points between the lifecycle of subsequent transportation legs. Thus, for instance (see Figure 4), the transportation leg waits in its ”executing Leg” stage for the “Leg arrived” event of the previous leg.

IMPLEMENTATION ASPECTS

Business partners interact with the Collaboration Hub through a portal, or using links (e.g., web services) with back-end systems. We focus here on implementation aspects regarding the interaction of these external components with the CH. Details around a possible implementation of the CH realizing BEs can be found in (Hull et al. 2010). Requirements for the interfacing with the CH are as follows:

1) Partners must respond to action items (AIs). In general this is done by receiving an AI in a portal and responding to it manually, but this may also be done through a web service connection, or on basis of an email
2) Partners must share data
    a. They must have read access to BE data. This includes read access to AIs ordered per BE that they must respond to. In this way, a user can login at a portal at any time, and get an up-to-date view on the BE data plus relevant AIs
    b. They must also be kept up-to-date on changes of the data, for example when there is a new ETA (Expected Time of Arrival) or when goods have arrived

To achieve these requirements the communication with the CH is based on the following:

- The CH provides a read access web service for reading BE data (including AIs)
- Through an MVC principle, change events are sent out from the CH when any changes are available on BE data. The portal can subsequently read the corresponding updated BE data from the CH
- AIs are not only available through read-access. AIs are also sent as payload of the corresponding change events, so that these AIs can directly (and not through an extra read on the BE data) be sent on to any of the external components
- A communication layer (middleware) takes care of routing events to the right external components
- The communication layer allows various mechanisms for feeding events to the external components:
    - It allows external components to subscribe or poll for new events. In fact the portal polls on new events
    - It also allows pushing events with payload to the corresponding external component, e.g. to a web service or to an email server
    - It translates events plus AI payload in format needed for the external component

FULFILLMENT OF THE REQUIREMENTS

The above principles have indeed been implemented in the CH and tested on our motivating scenario. On basis of the main principles described above and our tests, our requirements are fulfilled in the following way:

- Requirement 1: Since relevant information about the collaborative activities is stored in the corresponding BEs (Hull et al. 2010) [5]. Partners in the collaboration can view this information as based on their access rights, and receive update events.
- Requirement 2: Since partners are explicitly requested for a response due to the sending of an action item.
- Requirement 3: Since a BE allows any number of partners, and the lifecycle can be complex as based on the GSM modeling approach (Hull et al. 2010).
- Requirement 4: As shown in our example, a seller can invite an organizer for executing a transport, who can again invite carriers for the transportation leg. In each case, as based on the "Role fulfillment" BE, either completely new partners can be invited, or partners that are already known to the CH.
- Requirement 5: Variability can indeed easily be used by business users as they specify the process variant at the macro abstraction level by defining the set of legs to be executed. The business user therefore does not need to care about the lifecycle and data definition of the BE. Principally, they can even use their favorite planning tool for this[6]. The results are stored in an XML format, and this format is parsed to create the corresponding Transport leg BEs.
  Transport leg BEs ad-hoc composed in this manner can indeed be used to coordinate the overall transport process on basis of the individual lifecycles of each of the legs and their synchronization points. We have not proven, just tested that the specific lifecycles plus corresponding synchronization points indeed lead to a correct coordination of the overall process. Principally, correctness should be provable using Object-oriented predicate/transition nets (OPTNS) (Dong and Chen 2001).
- Requirement 6: The use of pre-defined lifecycle of different types of BEs leads to the fulfillment of this requirement. The use of pre-defined connections with back-end systems is foreseen but has not yet been tested.

CONCLUSIONS

This paper explains requirements for organizational collaboration, motivated by a (simplified) logistics scenario. It also explains how focusing on two major aspects, namely advanced collaboration support and compositional flexibility, the corresponding requirements are fulfilled. The advantage of the approach with respect to existing approaches lies in the improved collaboration support, integrating active process coordination with data sharing, together with the possibility to let business users ad-hoc compose the collaboration process in terms of abstractions (so-called "business entities") such as transportation legs.

Future work will focus in three domains: Linking pre-existing systems, allowing for issue handling (e.g., delays, cancellations etc.), and providing formalized proofs that the lifecycles of the composed business entities will indeed result in a correct execution of the overall process.

---

[5] "The entitycentric approach enables business insights and improves communication among diverse stakeholders about the operations and processes of a business, in ways that activityflow-based and document-based approaches have not"

[6] Creating a link with some planning tool has not yet been implemented through.

REFERENCES

van der Aalst, W.M.P., Weske and M. Grünbauer, D., 2005. Case handling: a new paradigm for business process support. *Data & Knowledge Engineering* 53. pp. 129-162.

van Aken, J.E., 2004. Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules. *Journal of Management Studies,* 41 (2)

Adams, M., ter Hofstede, A.H.M., Edmond, D. and van der Aalst, W. M.P., 2006. Worklets: A Service-Oriented Implementation of Dynamic Flexibility in Workflows. In Meersman, R. and Tari, Z., Eds. 2006. *Proceedings of the 14th International Conference on Cooperative Information Systems (CoopIS'06)* LNCS 4275. pp. 291-308

Burkhart, Th. And Loos, P., 2010. Flexible Business Processes – Evaluation of Current Approaches. In Schumann, M., Kolbe, L. M., Breitner, M. H. And Frerichs A., 2010 *Proceedings of the Multikonferenz Wirtschaftsinformatik MKWI 2010.* pp. 1217-1228

Casati, F., Ilnicki, S., Jin, L.J., Krishnamoorthy, V. and Shan, M.C., 2000. Adaptive and Dynamic Service Composition in eFlow. In Wangler, B. and Bergman L. eds. 2000. *Proceedings of CAiSE 2000, LNCS 1789*. Berlin Heidelberg: Springer-Verlag. pp. 13-31

Dong, M., Chen, F.F, 2001. Process modeling and analysis of manufacturing supply chain networks using object-oriented Petri nets. In *Journal of Robotics and Computer Integrated Manufacturing*. 17 121-129

Dustdar S., 2004. Caramba—A Process-Aware Collaboration System Supporting Ad hoc and Collaborative Processes in Virtual Teams. In *Journal Distributed and Parallel Databases*, 15, pp. 45–66

Hamadi, M. and Leitz, *A.,* 2008. *Supplier Collaboration with SAP SNC*. SAP Press ISBN 978-1-59229-194-6

Hull, R. et al. Introducing the Guard-Stage-Milestone Approach for Specifying Business Entity Lifecycles. In *Proceedings of 7th Intl. Workshop on Web Services and Formal Methods (WS-FM 2010), Revised Selected Papers, Lecture Notes in Computer Science 6551*. Springer-Verlag

Lazcano A., Alonso G., Schuldt H. and Schuler C., 2000. The WISE approach to electronic commerce. In *Int J Comput Syst Sci Eng,* 15(5) pp. 343-355

Medjahed, B., Benatallah, B., Bouguettaya, A., Ngu, A., and Elmagarmid, A., 2003. Business-to-business interactions: issues and enabling technologies. In *The VLDB Journal of Computing and Information Technology, 12*, 59-85

Stoitsev, T., Scheidl, S., Flentge, F. and Mühlhäuser, M., 2008. From Personal Task Management to End-User Driven Business Process Modeling. In Dumas, M. Reichert, M. and Shan M.-C. eds. *BPM 2008, LNCS 5240*, pp. 84-99

Wende, K. et al., 2006. Überbetriebliches Prozessdesign mit Public Processes - Konzept und Anwendung. In: (Report in German) St.Gallen: Universität St. Gallen - Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften (HSG).

ACKNOWLEDGEMENT

This work has been supported by the projects: ECSIT, ADiWa (German: BMBF), GinSeng and FInest (EU)