Overview on Decentralized Establishment of Multi-lateral Collaborations

Andreas Wombacher

Department of Computer Science, University of Twente, The Netherlands
a.wombacher@ewi.utwente.nl

Abstract. Service oriented architectures facilitate loosely coupled collaborations, which are established in a decentralized way. One challenge for such collaborations is to guarantee consistency, that is, fulfillment of all constraints of individual services and deadlock-freeness. This paper presents an overview on a decentralized approach to consistency checking, which utilizes only bilateral views of the collaboration.

1 Introduction

A multi-lateral collaboration is the act of several parties working jointly [1]. Several forms of collaboration exist covering almost all areas of people's life like for example at work, where people are employed by a company being involved in producing a good. Within this example, the employees have the same goal of producing a good. However, there exist collaborations where each party has its own goal that can only be achieved by interacting with other people, like for example, people going shopping at a market or companies having joint ventures.

These two generic types of collaborations exist also in Information Technology. The execution of such a collaboration usually involves several parties, each providing different tasks, that is, a "logical unit of work that is carried out as a single whole by one" party [2]. A collaboration can be characterized by a set of tasks that have to be performed and the causal and temporal dependencies between the different tasks. A model describing the coordination of tasks, that is, managing dependencies between these tasks [3], dependencies between activities" within a collaboration is known as a workflow.

Workflows have been studied for several years. Initially, workflows have been carried out completely by humans manipulating physical objects [4]. Later, with the introduction of Information Technology, processes are partially or totally be automated by information systems, which are controlling the execution of tasks and the performance of the tasks themselves. However, the main goal of workflow management systems is not the complete automation of workflows but the separation of control logic and logic contained in the tasks, where a task is either performed by a information system or by a human. Based on this separation reuse of tasks in different workflows is supported [5]. Electronic data interchange (EDI) performed over the Internet and the extendable markup language (XML) standard family are key factors for the emergence of Web-based workflows. Due

M. Hemmje et al. (Eds.): E.J. Neuhold Festschrift, LNCS 3379, pp. 11–20, 2005.

to this improved and simplified communication and coordination mechanisms, inter-organizational cooperations and virtual organization structures are evolving, where the interaction and work performed by several parties forming a multi-lateral collaboration have to be coordinated and controlled. Establishing such a multi-lateral collaboration is a major challenge, while the effort increases with the number of parties that have to agree on

- the connectivity, that is, the supported communication protocols (like for example FTP, HTTP, SMTP,...) as well as the communication languages, which are message formats in case of electronic data interchange (EDI).
- the tasks to be used within the collaboration, that is, the combination of tasks taken from different parties forming a successful collaboration.
- the coordination of the selected tasks, that is, the order in which the different tasks have to be executed guaranteeing a successful collaboration.

To achieve connectivity the transformation of protocols and messages might be required. An automated transformation of messages requires an "understanding" of the meaning of a message's content, which is expressed in terms of an ontology and is addressed by the Semantic Web community. The second and third aspect, that is, deriving the set of used tasks and the coordination of those tasks represents an establishment of a commonly agreed workflow specification. The approach sketched in this paper focus on these workflow aspects required to establish a multi-lateral collaboration.

2 Centralized Establishment of Multi-lateral Collaborations

Nowadays, multi-lateral collaborations are usually set up by a group of people representing the different parties involved in the collaboration. In particular, these people meet, discuss the different options, and finally decide on the definition of a multi-lateral collaboration: which communication protocols and messages are going to be used, and what are the workflow options that have to be supported by a collaboration.[1] The agreement on the multi-lateral collaboration derived by this group specifies a multi-lateral collaboration from a global point of view. Based on this specification a multi-lateral collaboration can be checked for consistency, that is deadlock-freeness: the specification of a multi-lateral collaboration has to ensure that all potential execution sequences of the collaboration guarantee a successful termination of the collaboration, that is, there exist no execution sequence resulting in a non-final state where no further interaction with any party is possible.

The global view specification of a multi-lateral collaboration can be used to derive a specification of the collaboration from a local point of view, that is, the

[1] As a basis for this discussion the different parties try to ensure that the integration effort needed to adapt the local infrastructure and processes to the multi-lateral collaboration is minimized.

view of an individual party. Approaches exist, which allow to derive the local point of view from the global one ensuring that the interaction of the local views implements the global view of a multi-lateral collaboration [6]. Further, it can be guaranteed that the interaction of the local views is consistent if the global specification of a multi-lateral collaboration has been consistent.

The approach described above starting with a global specification of a multi-lateral collaboration, deriving the local views of it, and finally implementing the local views at each party guaranteeing the properties of the global view is also known as top-down approach of a collaboration establishment. This way of collaboration establishment is quite expensive, because people have to come to an agreement and the implementation of the local views of a collaboration afterwards requires a considerable implementation effort. Further, changes of the collaboration require to go through the whole process again making changes also very expensive. As a consequence, the top-down approach of collaboration establishment works fine for well established and quite static multi-lateral collaborations. However, current development in IT technology supports more flexible structures like for example Service Oriented Architectures (SOA), which are used to realize loosely coupled systems inherently providing a high potential of establishing collaborations between parties in a quite flexible and dynamic way.

3 Web Services

A Service Oriented Architecture (SOA) is defined as "a set of components which can be invoked, and whose interface descriptions can be published and discovered" [7]. Where a component is a "software object interacting with other components, encapsulating certain functionality or a set of functionalities" [8] and maintaining an internal state [9]. Thus, a SOA consists of components accessible as services, where each service provides a certain functionality, an internal state, and an interface to publish the provided functionality to potential service requesters. Opposed to component based architectures, where components are combined during the development phase, in SOAs services are combined after the deployment of services, that is, at run-time [10]. This change can be characterized as a step from supply-driven collaborations to demand-driven ones [11].

A concrete technology implementing the SOA are Web Services. In particular, the W3C Web Service Architecture Working Group defines a Web Service as "a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards." [7]

From the definition of SOA the following properties can be derived: i) the services are distributed, since each service can be provided by a different party, and ii) the services are autonomous, because state changes within a service are independent of other service's states. As a consequence, a stateless service representing a certain functionality is comparable to a single task within a workflow, while a stateful service represents a set of tasks and its inherent dependencies represent a local workflow. The interaction of several services results in a multi-lateral collaboration being constructed from a set of pre-existing local workflows provided by services, resulting in a global workflow. This approach is further called bottom-up approach opposed to the top-down approach introduced in Section 2.

4 Decentralized Establishment of Multi-lateral Collaborations

Bottom-up establishment of multi-lateral collaborations should always result in a consistent collaboration, that is, a consistent global workflow. As a consequence, the bottom-up approach has to guarantee that the resulting global workflow is consistent. Because the global workflow is never instantiated, the decision on the consistency of the global workflow has to be made by local decisions of the involved parties. Consistency of a multi-lateral collaboration can be decided locally by a single party in case of a hierarchical structure of services, where a single service requester centrally coordinates the services, which interact only with the service requester and are provided by the remaining parties of the multi-lateral collaboration. Due to the limitation of services to interact with the service requester only, a single party exists knowing all complete local workflows, which is able to derive the global workflow of the multi-lateral collaboration and to decide on the global workflow's consistency [12].

Opposed to this very specific case, in all other cases no party knows the global workflow, thus, the decision on consistency of the global workflow has to be made in a decentralized way based on partial knowledge of the global workflow. From decentralized systems it is known that this kind of decision can not be made directly from local decisions based on a bilateral comparison [13].

This paper addresses this issue and provides an overview on an approach to decide consistency of a multi-lateral collaboration in a decentralized way, that is, without instantiating the corresponding global workflow. The decision can be made in a decentralized way by deriving some additional consistency properties.

5 Workflow and Communciation Models

Before discussing the concrete approach, a sufficient workflow modeling approach has to be selected providing means to represent bilateral consistency as well as a centralized version of multi-lateral consistency as a starting point. Multi-lateral collaborations may rely on different communication infrastructure, which can be generally classified as synchronous, that is, a message sent by a party must be received by anotehr party immediately, and asynchronous, that is, a message sent by a party has to be received by another party later on, but latest before completion of the local workflow. While there exist several approaches for the asynchronous communication model, the synchronous communication model has not been addressed so far. In case of asynchronous communication Workflow Nets [2] have been used besides other approaches like for example [14,15]. For the synchronous case, an extension of Finite State Automata [16] called annotated Finite State Automata (aFSA) [17] has been proposed. In particular, standard Finite State Automata are extended by a notion of mandatory and optional transitions, that is, all messages sent by a party that must be supported by a recipient party are called mandatory messages, while messages received by a party are called optional messages, because they are not necessarily sent by another party.

Evaluation of Workflow Net properties are defined on the derived occurrence graph, which has the same expressiveness as aFSA. Thus, a mapping from Workflow Nets to aFSA can be defined, while it can be shown that the definitions of multi-lateral consistency consider the equivalent set of collaborations to be consistent. As a consequence, further discussion of decentralized consistency checking can be focused on the notion of aFSA.

6 Decentralized Consistency Checking

As discussed in detail in [18], the trivial approach of basing a decentralized multilater consistency decision on bilateral consistency turned out to be incorrect, because of information loss introduced by having only a partial view on the multilateral collaboration. Especially, information loss has been observed on message parameter constraints, that is, parameter values are considered which have been excluded by other parts of the global workflow already, and occurrence graph constraints, that is, message sequences are considered although the execution of a message contained in this message sequence has been excluded already by another part of the global workflow.

Due to the loss of information, decentralized consistency checking requires to make use of transitivity properties on parameter and occurrence graph constraints. Thus, deciding consistency of a multi-lateral collaboration in a decentralized way proceeds in three steps:

1. Resolving Cycles:
   Local workflow models of parties are made acyclic by representing cycles as iterations of at most N steps.
2. Propagation:
   Parameter and occurrence graph constraints on already performed transitions are made available to all parties involved in the multi-lateral collaboration. This comprises:
   (a) Propagation of parameter constraints within local workflows, as well as between bilateral interactions until a fixed point has been reached.
   (b) Propagation of occurrence graph constraints within local workflows, as well as between bilateral interactions until a fixed point has been reached.

3. Decentralized Consistency Checking:
   Each party checks consistency of its bilateral interactions and local workflow. If they are all consistent, then the party considers the multi-lateral collaboration to be consistent until any other party falsifies this decision by considering the multi-lateral collaboration to be inconsistent.
4. Consensus Making
   A protocol is required to decentrally check whether all parties consider their bilateral interactions and local workflows as consistent, and to inform all parties about the final consensus. This kind of problem is known in distributed systems as consensus making problem [13].

A decentralized decision requires to make use of transitivity properties of parameter and occurrence graph constraints, which requires the underlying workflow model to support parameter constraint transitivity. Since cyclic graph structures are not transitive used workflow models have to be acyclic. As a consequence, cycles have to be resolved in step 1. In particular, cycles are resolved by explicitly representing all potential execution sequences of the cycle with at most N steps of a cycle within a single execution sequence of the local workflow.

Step 2 is required because bilateral workflows hide all parameter and occurrence graph constraints that is not immediately seen by the two involved parties. The goal of parameter constraint propagation is to make sure that all parameter constraints can be met, even though they may not immediately be visible in a bilateral workflow. The parameters of transitions are assumed to be immutable, that is, after they have been set initially they can not be changed. As a consequence, a parameter constraint holds for all transitions following the transition at which it has been specified. On these grounds parameter constraints can be propagated to all following transitions within a workflow as well as to the workflow of the partner. The goal of propagating occurrence graph constraints is to discard all those transitions, which cause a deadlock in a bilateral workflow but can never be executed due to constraints imposed by the invisible part of the global workflow.

Step 3 is the consistency checking itself, that is, making a local decision on local consistency of a local party. This step is performed by every party independently and the consensus making protocol is applied next.

Finally, step 4 aims to make an agreement between a set of parties having reached a fixed point with regard to parameter and occurrence graph constraint propagation, and forming a multi-lateral collaboration. Since no party knows all parties involved in the collaboration none can act as a coordinator of the collaboration. In particular, the following tasks must be performed:

– collect the local consistency decision of each party,
– check whether all parties consider the collaboration to be consistent, and finally
– inform all parties being involved on the final decision.

This generic consensus making problem is addressed by the distributed systems and algorithms community (see for example [13]). However, due to the fact that a fixed point on constraint propagation of constraints is required anyway, the aim is to define multi-lateral consistency as a kind of propagation to overcome the consensus making problem. The underlying idea is to reflect mandatory and optional messages as structural aspects of a workflow model effecting the occurrence graph, thus, being propagated via the corresponding propagation mechanism as discussed above. However, the modification of the occurrence graph with respect to mandatory and optional messages has to be performed via an explicit operation. As a consequence, a fixed point can be reached, where either non or all local workflows of the collaboration are consistent.

7 Evaluation

The evaluation of the sketched approach based on a synchronous communication model in the Web Services domain is two-fold: First, the applicability and expressiveness of the introduced annotated Finite State Automata (aFSA) as a workflow model is evaluated. In particular, a mapping of the Business Process Execution Language for Web Services (BPEL) [19] to aFSA is specified and realized. A detailed description of this mapping is contained in [20]. However, there exist some BPEL language constructs, which are not mapped to the aFSA workflow model, because they are irrelevant for consistency checking or can be added later on to the mapping by an additional processing step potentially increasing the complexity of the aFSA model significantly. Thus, the mapping works fine, which has been further evaluated by representing all potential workflows derived from the Internet Open Trading Protocol (IOTP) [21] specification in terms of aFSA. It turned out that all potential workflows covered by IOTP can be modeled as aFSAs.

Second, the process of establishing consistent multi-lateral collaborations has to be based on the propagation of constraints and consensus making as sketched above. To be able to set up these multi-lateral collaborations, a service discovery is required to find potential trading partners, which considers only thos service providers, which do have a consistent bilateral workflow with the own local workflow. In particular, this requires an implementation of a bilateral matchmaking based on the mapping from BPEL to aFSA as mentioned aboved. Such a service discovery based on the aFSA workflow model has been implemented and described in [22], where the supported queries are described as an extension of classical UDDI, the architecture is introduced, and a performance measurement based on the constructed IOTP data set is provided. The implementation is based on the assumption that receiving a message is always unconstraint, that is, the receiving party has to be able to handle all potential parameter values. As a consequence, the implementation does not consider the handling of parameters, hence, parameter constraint propagation can be neglected. Based on the derived service providers a multi-lateral collaboration can be established in a decentralized way requiring a unique collaboration identification to handle concurrent involvement of parties in several collaborations. In particular, a protocol is required to derive multi-lateral collaborations from bilateral collaborations in a decentralized way.

Thus, based on these two evaluation steps the sufficient expressiveness of aFSA to represent real world workflows is illustrated.

8 Summary and Outlook

The paper provides a high level description of an approach to establish consistent multi-lateral collaborations in a decentralized way. At various points references to further readings are provided. In particular, the aim of this paper is to sketch the basic ideas, which are the requirement of constraint propagation resulting in a fixed point allowing local consistency checking, and the consensus making on the multi-lateral consistency based on propagated structural workflow properties.

As an outlook, additional application scenarios are briefly sketched like for example ebXML, GRID, or Peer-to-Peer (P2P) environments, which all have in common a necessity of forming multi-lateral collaborations.

The electronic business XML initiative (ebXML) [23] specification provides a framework supporting XML based exchange of business data [11]. In particular, the Collaboration Partner Profiles (CPP) and the Collaboration Partner Agreements (CPA) are part of this framework, which are reflecting the description of a party and the subset of this description to be used within a concrete collaboration. Although, Patil and Newcomer [24] consider ebXML as a top-down collaboration establishment approach, this is not enforced by the framework since no centralized coordinator of a collaboration has to be specified and the assignment of potential trading partners supports late binding similar to the service discovery phase in Web Service.

Another technology is the GRID infrastructure. Foster defines in [25] the GRID as "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations". In particular, different organizations provide resources and request capacities for solving problems. However, the different parties are independent of each other although they agreed to participate in the GRID, which is right now a quite static relationship with high availability of the different participants. Due to this structure, there is no need for a more flexible handling of relationships. But, the GRID community started to think about more flexible relationships, where the availability of different parties is lower and more flexible and short to mid-term relationships have to be established and managed as for example addressed by the Diligent project [26].

A further example of a potential technology are Peer-to-Peer (P2P) systems. One definition of P2P considered suitable for this discussion is provided by the Intel P2P working group: "P2P is the sharing of computer resources and services by direct exchange between systems" [27]. In P2P environments every party (peer in P2P terminology) are considered to be independent. This means that a peer offers services or resources to a community, but at the same time, it can consume services/resources from others in the community. An important property of P2P systems is the lack of a central administration, the flexibility of the set of peers forming the community, and the decentralized organization of the community. As illustrated by Risse et.al. [28] P2P systems are on the move from well known file sharing to large scale decentralized and reliable systems relying on decentrally coordinated and established multi-lateral collaborations.

References

1. WordNet. (http://wordnet.princeton.edu)
2. Aalst, W., Hee, K.: Workflow Management - Models, Methods, and Systems. MIT Press (2002)
3. Malone, T.W., Crowston, K.: The interdisciplinary study of coordination. ACM Computing Surveys (CSUR) **26** (1994) 87–119
4. Georgakopoulos, D., Hornick, M., Sheth, A.: An Overview of Workflow Management: From Process Modelling to Workflow Automation Infrastructure. Distributed and Parallel Databases **3** (1995) 119–153
5. Mohan, C.: Workflow management in the internet age. In Litwin, W., Morzy, T., Vossen, G., eds.: Proceedings of the Second East European Symposium on Advances in Databases and Information Systems (ADBIS), Springer LNCS 1475 (1998) 26–34
6. Aalst, W.: Interorganizational workflows: An approach based on message sequence charts and petri nets. Systems Analysis - Modelling - Simulation **34** (1999) 335–367
7. Haas, H., Brown, A.: Web services gloassary. http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/ (2004)
8. Forum, C., Keahey, K.: CCA terms and definitions. http://www.cca-forum.org/glossary.shtml (2004)
9. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, University of Calivornia, Irvine (2000)
10. Kaye, D.: Loosely Coupled - The Missing Pieces of Web Services. RDS Press (2003)
11. Bussler, C.: B2B Integration - Concepts and Architecture. Springer (2003)
12. Wombacher, A., Mahleko, B., Risse, T.: Classification of ad hoc multi-lateral collaborations based on workflow models. In: Proceedings of Symposium on Applied Computing (ACM-SAC). (2003) 1185–1190
13. Lynch, N.A.: Distributed Algorithms. Morgan Kaufmann (1996)
14. Fu, X.: Formal Specification and Verification of Asynchronously Communicating Web Services. PhD thesis, University of California Santa Barbara (2004)
15. Kindler, E., Martens, A., Reisig, W.: Inter-operability of workflow applications: Local criteria for global soundness. In: Business Process Management, Models, Techniques, and Empirical Studies, Springer-Verlag (2000) 235–253
16. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison Wesley (2001)
17. Wombacher, A., Fankhauser, P., Mahleko, B., Neuhold, E.: Matchmaking for business processes based on choreographies. In: Proceedings of International Conference on e-Technology, e-Commerce and e-Service (EEE), IEEE Computer Society (2004) 28–31
18. Wombacher, A., Aberer, K.: Requirements for workflow modeling in P2P-workflows derived from collaboration establishment. In: Proceedings of International Workshop on Business Process Integration and Management (BPIM). (2004) 1036–1041

19. Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S.: Business process execution language for web services, version 1.1 (2003)
20. Wombacher, A., Fankhauser, P., Neuhold, E.: Transforming BPEL into annotated deterministic finite state automata enabling process annotated service discovery. In: Proceedings of International Conference on Web Services (ICWS). (2004) 316–323
21. Burdett, D.: Internet open trading protocol - IOTP - version 1.0. http://www.ietf.org/rfc/rfc2801.txt (2000)
22. Wombacher, A., Mahleko, B., Neuhold, E.: IPSI-PF: A business process matchmaking engine. In: Proceedings of Conference on Electronic Commerce (CEC). (2004) 137–145
23. ebXML: ebXML home page. (http://www.ebxml.org/)
24. Patil, S., Newcomer, E.: ebxml and web services. IEEE Internet Computing **7** (2003) 74–82
25. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the Grid: Enabling scalable virtual organization. The International Journal of High Performance Computing Applications **15** (2001) 200–222
26. diligent consortium: A digital library infrastructure on grid enabled technology. http://diligentproject.org/ (2004)
27. Milojičić, D., Kalogeraki, V., Lukose, R., Nagaraja, K., Pruyne, J., Richard, B., Rollins, S., Xu, Z.: Peer-to-peer computing. Technical report (2002)
28. Risse, T., Knezevic, P., Wombacher:, A.: P2P evolution: From file-sharing to decentralized workflows. it-Information Technology (2004) 193–199