

Towards A Transaction Framework for Contract-Driven, Service-Oriented Business Processes

Ting Wang

Information Systems Subdepartment, Department of Technology Management,
Eindhoven University of Technology,
P.O. Box 513, 5600 MB Eindhoven, the Netherlands
t.wang@tm.tue.nl

Abstract. Transaction support is vital for reliability of business processes which nowadays can involve dynamically composed services across the Internet. However, no single transaction model is comprehensive enough to accommodate various transactional properties demanded by those processes. Therefore we intend to develop a Business Transaction Framework (BTF) that utilizes existing ingredients in the context of contractual business relations. The proposed BTF has flexibility by using a library containing abstracted transaction models, validity based on formalization and business trustworthiness guaranteed by contractual specifications of transactional qualities.

1 Introduction

E-business has moved from the simple automation of business applications to the complex integration and coordination of business processes that belong to different organizations. Business processes have grown to be very intricate to involve miscellaneous activities and resources, and often need to invoke services provided by others. In such an environment, various business relationships need to be regulated, which result in a large amount of contracts along the processes. In addition, the dynamically composed activities usually have complex inter-related dependencies which make exceptions and errors prone to occur during the process execution. Suppose there is a booking process in a travel agency which needs to invoke three Web services, hotel booking, car rental and flight booking respectively. Besides there are activities like billing, payment check etc. which happen internally. These activities exhibit various transactional semantics and the process in a whole may demand different transactional qualities during execution. For instance, when a customer cancels the booking before payment, the previous completed activities have to be rolled back as if the process has never been initiated. However in case of a cancel after payment, the process probably can not be returned to the exact same state when it started. Therefore the process needs comprehensive and flexible transactional support to guarantee reliability.

To address the above concern, the XTC (eXecution of Transactional Contracted electronic services) project was proposed which aims at laying a generic

foundation to the transactional support for processes in service-oriented environment. Within this project, my PhD research focuses on the development of a Business Transaction Framework (BTF) for contract-driven, inter-organizational business processes and in particular, on contractual agreements specifying transactional qualities of those business processes.

The rest of the paper is organized as follows. We describe the research background that motivates our work and review the related work in Sect.2. We introduce the BTF and discuss our research approach in Sect.3. We end this paper in Sect.4 with the summary and the discussion of future work.

2 Research background and related work

The challenge of e-business nowadays is to integrate distributed applications into a corporate process which enables free and secure data-flow and control-flow across the organizational boundaries. The former way of tightly coupled integration like EAI (Enterprise Application Integration) can no longer meet today's requirements. To loosely couple the applications, one approach is wrapping them as ready-to-use services. The paradigm of Service-Oriented Computing (SOC) and its underlying Service-Oriented Architecture (SOA) [1, 2] allow distributed services ready to be utilized by application developers despite their locations, platforms and internal implementations. While the services may refer to any piece of functional software over the network (the Internet, Intranet or the Grid), Web services have been considered by the academia and the industry a fundamental technology to implement SOA [2–6]. Having moved beyond the basic layer of the extended SOA [2] focusing on service publishing, discovering and binding, current endeavors of Web services are made towards the higher layers of service composition and management.

Transaction management in Web services-based SOC paradigm is an intriguing topic in the research of service composition and management. First appeared in the mid 1970s to address the multi-user, concurrent-program concern in databases, transaction management has become a prevalent mechanism to guarantee reliability to running applications. Earlier work like [7–10] on ACID transactions and advanced transaction models is more or less database-oriented thus not suitable for complex business processes. Later on, research (e.g. [11, 12]) flourished to address the transactional support for business process automation and integration. Meanwhile, driven by the growing need of service composition, Web services transaction specifications are proposed by different standardization bodies (e.g. [13]). However, the above mentioned process-oriented transactions are either predefined in structure before process execution (e.g. workflow transactions) or not suitable for other types of applications (e.g. Web service transactions). A possible way to realize transaction support for a complex business process, which usually consists more than one Web services, is to orchestrate loosely coupled services into a single business transaction by guaranteeing coordinated, predictable outcomes for the participating partners [14]. Therefore we target the development of a high-level transaction framework that is more

flexible and comprehensive to accommodate various transactional semantics and applications.

Another field of our interest is e-contracting, which ensures business processes with legal liability and cost-efficiency. To sustain a certain level of process reliability, we propose to use contracts as service level agreements to specify the agreed transactional qualities of the whole business process as well as each component service. For example, we may leverage the WSLA (Web Service Level Agreement) framework [15] to enclose transactional semantics in the SLAs that serve as contracts to bond the service providers and consumers together. Work on the general Quality of Service (QoS) for processes such as [16] can serve as the starting point. We suppose that the desired Transactional Quality of Service (Tx-QoS) is a subset of the QoS for Web services composition. For instance, cost is a very common non-functional Web service QoS but will be excluded in our case. Also work on contractual support for Web services like [17, 18] can serve as a basis for us to develop a contractual approach to specify Tx-QoS. With regard to the specification language for such contracts, we consider e-contracting languages and the WSLA language as candidates. Currently there are some proposals about e-contracting languages such as [19, 20], but none is widely adopted to our best knowledge. The WSLA language may well define normal Web services QoS but seems too constraint for the expression of transactional semantics. Therefore we intend to develop a specification language for our purpose of contractual support for Tx-QoS.

Today's business environment, which involves multiple partners, diverse processes and heterogeneous systems, motivates us to develop a BTF with its flexibility by using the ATC (Abstract Transactional Construct) library, validity based on BTF/ATC formalization and business trustworthiness guaranteed by the contractual specification of Tx-QoS.

3 Business Transaction Framework (BTF)

So far we have elaborated a conceptual framework with an underlying architecture that is shown in Fig.1. The basic idea is to utilize existing transactional models and abstract them into ATCs so that they can be selected and composed on demand. The ATC Editor creates, deletes and modifies ATCs through the ATC Library Manager in the definition phase. Fed by a process specification, the ACBT (Abstract Composite Business Transaction) is created by the ACBT Composer and stored in the ACBT library in the composition phase and leverages ATCs as its building blocks. During the execution phase, the CBTs (Composite Business Transaction) are instantiated from the ACBT by the CBT Creator and consist of TCs (Transaction Constructs) instantiated from ATCs. Each CBT Manager monitors and manages one CBT. At the highest level there is a BTF Manager that coordinates the activities of other function components and manages transactions through all three phases of the process life cycle. x

To ensure a flexible and valid transaction framework, we employ classic and widely adopted transaction models and abstract them to shape the ATC library.

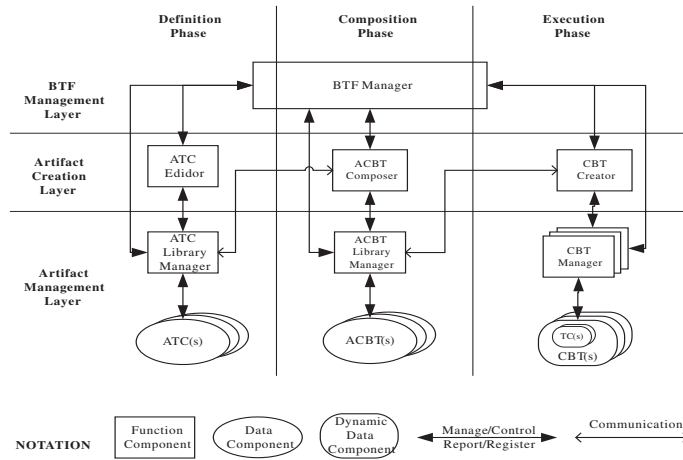


Fig. 1. BTF Architecture

For each ATC, we define a set of parameters, among which some are assigned in the definition phase to specify its internal structure while the others are assigned in the composition phase to specify its position in an ACBT. We design the framework based on the Web services standards and place it on top of the composition layer. To implement the BTF, we develop a reference architecture delivering the necessary functionalities by following the rules and techniques suggested in [21], with the focus on the qualities like modifiability, reusability and portability. This architecture serves as a blueprint to design a prototype system (e.g. on WebSphere platform) running some business processes based on real cases.

One advantage of the proposed BTF lies in the quality control by means of contractual agreements between the service providers and consumers. So far there is no available research on contractual transaction specifications. Besides, there is no clear definition about what are the measurable transactional qualities of business processes. As a business process may contain activities exhibiting different transactional semantics, the supporting transaction framework should accommodate the required Tx-QoS for each component activity as well as the whole process flexibly. Again we take the travel booking process of Sect.1 for example, which includes multiple parties like a customer, a travel agency and candidate hotels, airlines and car-rental companies. Each company involved in this process has its own internal automated processes and offers business functions through Web services (here we assume each component activity is automated and wrapped as a Web service). The whole process can be viewed as a hybrid transaction consisting of various sub-transactions, each dealing with a transactional Web service. It is very likely that these component services require different transactional qualities such as the unconventional payment atomicity [14] for the payment activity. Meanwhile the composite process demands a cer-

tain level of qualities for the whole transaction and the demand may vary over time. We propose to extend the contractual agreements (for example, the WSLA files) to specify the expected or agreed transactional properties for each component service. Derived from these agreements, we can specify the Tx-QoS clauses in the process contract. Therefore the trustworthiness is guaranteed by means of legal contracts.

4 Summary and future work

Our preliminary design of the conceptual and architectural BTF/ATC described in Sect.3 is the first step towards a comprehensive and flexible transaction framework for contract-driven, service-oriented business processes. The idea is to extract and group existing transaction models into an ATC library and pick up the needed ones to compose a transaction hierarchy on demand. The design work is far from complete, with several directions for the future elaboration:

The first direction is to achieve a complete and sound BTF/ATC specification. To guarantee the correctness of the proposed BTF, we need to develop a method to extract and connect ATCs into an ACBT. One approach is to encapsulate the transactional semantics with parameterizable interfaces based on an ATC taxonomy. This way an ACBT can be composed through these interfaces by predefined composition rules.

Secondly, a refined architecture with a complete set of components delivering proper functionalities is required as the blueprint to implement a prototype system. Bearing the unique features (e.g. flexibility, extensibility, compatibility) of the BTF in mind, we plan to decompose some of the function components shown in Fig.1 to get a deeper level of architectural overview. For example, the BTF manager can be decomposed into a second-level architecture that may contain a messenger, a contractor and a monitor and the contractor can be further refined if needed.

Last but not least, we are going to develop a Tx-QoS specification language. To achieve this, we need to investigate the existing e-contracting languages. In addition, a clear description of intended QoS for transactions is needed. Afterwards we expect to combine the contractual Tx-QoS with the BTF specification.

Acknowledgement

The XTC project is funded by the Dutch Organization for Scientific Research (NWO No. 612.063.305). Thanks go to Paul Grefen, Benedikt Kratz and Jochem Vonk whom I work with in this project.

References

1. Papazoglou, M.: Service -oriented computing: Concepts, characteristics and directions. In: Procs. of the 4th Int. Conf. on Web Information Systems Engineering (WIDE'03). (2003) 3–12

2. Papazoglou, M., Georgakopoulos, D.: Service-oriented computing. *Communications of the ACM* **46** (2003) 25–28
3. Brown, A., Johnston, S., Kely, K.: Using service-oriented architecture and component-based development to build web service applications. White paper, Rational Software (2002)
4. Lehmann, M.: Deploying large-scale interoperable web services infrastructures. *Web Services Journal* **5** (2005)
5. Mahmoud, Q.H.: Service-oriented architecture (SOA) and web services: The road to Enterprise Application Integration (EAI). Technical article, Sun Developer Network (2005)
6. Ferguson, D.F., Lovering, B., Storey, T., Shewchuk, J.: Secure, reliable, transacted web services: architecture and composition. Technical article, Microsoft Corporation and IBM Corporation (2003)
7. Haerder, T., Reuter, A.: Principles of transaction-oriented database recovery. *ACM Comput. Surv.* **15** (1983) 287–317
8. Moss, J.E.B.: Nested transactions: an approach to reliable distributed computing. PhD thesis, EECS Department, M. I. T. (1981)
9. Garcia-Molina, H., Salem, K.: Sagas. In: *Procs. of the 1987 ACM SIGMOD Int. Conf. on Management of data (SIGMOD'87)*. (1987) 249–259
10. Chrysanthis, P.K., Ramamritham, K.: ACTA: a framework for specifying and reasoning about transaction structure and behavior. In: *Procs. of the 1990 ACM SIGMOD Int. Conf. on Management of data (SIGMOD'90)*. (1990) 194–203
11. Alonso, G., Agrawal, D., Abadi, A.E., Kamath, M., Gunthor, R., Mohan, C.: Advanced transaction models in workflow contexts. In: *Procs. of the 12th Int. Conf. on Data Engineering (ICDE '96)*. (1996) 574–581
12. Grefen, P., Vonk, J., Apers, P.: Global transaction support for workflow management systems: from formal specification to practical implementation. *The VLDB Journal* **10** (2001) 316–333
13. Cabrera, L.F., et al.: Web Services Transactions. Available at <http://www-128.ibm.com/developerworks/library/specification/ws-tx/> (2005) Including WS-Coordination, WS-AtomicTransaction and WS-BusinessActivity specifications.
14. Papazoglou, M.: Web services and business transactions. *World Wide Web: Internet and Web Information Systems* **6** (2003) 49–91
15. IBM Corp.: Web Service Level Agreements (WSLA) Project. Available at <http://www.research.ibm.com/wsla/> (2004)
16. Cardoso, J., Sheth, A., Miller, J., Arnold, J., Kochut, K.: Quality of services for workflows and web service processes. *J. of Web Semantics* **1** (2004) 281–308
17. Tasic, V., Pagurek, B.: On comprehensive contractual descriptions of web services. In: *Procs. of the 2005 IEEE Int. Conf. on e-Technology, e-Commerce and e-Service (EEE'05)*. (2005) 444–449
18. Milanovic, N.: Contract-based web service composition framework with correctness guarantees, *The 2nd Int. Service Availability Symposium* (2005)
19. Angelov, S., Grefen, P.: Requirements on a B2B e-contract language. Beta working paper WP 140, Eindhoven University of Technology (2005)
20. Berry, A., Milosevic, Z.: Extending choreography with business contract constraints. *Int. J. Cooperative Inf. Syst.* **14** (2005) 131–179
21. Bass, L., Clements, P., Kazman, R.: *Software Architecture in Practice*. Addison-Wesley Longman (1998)