# A Business Aware Transaction Framework for Service Oriented Environments*

Benedikt Kratz

Tilburg University, Infolab
P.O. Box 90153, 5000 LE Tilburg, The Netherlands
B.Kratz@uvt.nl

**Abstract.** Transaction support is a key ingredient for reliable inter-organizational business transactions between distributed, loosely coupled, heterogenous systems. In this paper we present the outline and basic concepts of the Business Transaction Framework research project. In this project we want to develop a flexible and integrated framework that supports the structured application of abstract transactional constructs on business transactions. The ultimate goal of the project is the creation of reliable business transactions in the context of service oriented environments.

## 1 Introduction & Background

Growing economic activity and technical developments have lead to ever increasing interactions between businesses, thereby necessitating the coupling of business processes. The internal business processes of organizations need to be aligned with the external interactions in order to allow an efficient and effective realization of business goals. In this context, a business transaction occurs when the business processes of two (or more) organizations are connected and information is exchanged that changes the internal state of the organizations (we abstract here from the flow of physical goods). The consistent execution of business processes and registration of occurring business transactions is of utmost importance for organizations as these determine the profitability of an organization. With the emergence of information systems that support business processes and business transactions, automatic execution and registration of these processes and business transactions is now possible.

Transaction support is a well-known mechanism to ensure consistency in a multitude of situations. Using transactions to make changes on (distributed) data to reflect real world state changes (the state of the data reflects the state of the outside world as modeled by the relationship between them) has been a mature application within the database realm for multiple decades now. Transactions are guided by transaction models, like the ACID model, that define the significant

* The research reported in this paper is part of the eXecution of Transactional Contracted Electronic Services (XTC) project (No. 612.063.305) funded by the Dutch Organization for Scientific Research (NWO).

events that pertain to transactions adhering to that model [1]. The transaction concept is also applied within applications as application level transactions, e.g., within Enterprise JavaBeans. Transaction support for workflows has also been high on the research agenda over the past few years (e.g., [2]).

In comparison with the above transactions, business transactions that occur in a business process are quite different. Business transactions are complex and multi-level, span many organizations, involve multiple parties with heterogenous systems, usually have a long duration and optional parts and are automatically executed. When developing transaction support for business transactions, these characteristics need to be addressed to ensure a consistent execution of business transactions. Therefore, the business driven business transactions and technology driven data and application level transactions and supporting transaction models must be aligned.

In this paper, we outline the Business Transaction Framework (BTF), which is the core element of our research project. The goal of our research project is the development of concepts, constructs and mechanisms as well as infrastructure to enable flexible transactions in the context of e-business. To achieve our research goal we focus on the following points:

– Specification of the BTF, in particular the required concepts and supporting architecture.
– Representation of business semantics in transaction support for business transactions.
– Emulation of other existing similar transaction frameworks within the BTF.
– Formalization of the above points.

Our focus here is solely on transaction support, we do not for example focus on security aspects as these are out of the scope of the research project.

The interest into business transactions has recently gained momentum as the Service Oriented Computing (SOC) paradigm recognizes the need for business transactions and emphasizes their importance in loosely coupled and dynamic service oriented environments. The BTF is connected to the SOC paradigm in two ways. First, as SOC is the enabler of interoperability between heterogenous systems accompanied with loose coupling and highly dynamic establishment of business relations, we leverage the SOC philosophy to implement the BTF with Web Service technology. Secondly, the BTF extends SOC transaction models to address business driven requirements (intra- and inter-organizational) on automated long running business transactions. Therefore we abstract from concepts (e.g., context), also allowing the integration of related concepts from fields like databases and workflows to achieve an integrated approach in heterogeneous environments.

To present our research project, the remainder of this paper is organized as follows. In Sect. 2, we present the motivation for this research together with an exploration of related work. We point out several issues that are not addressed by current research. In Sect. 3, we present a short overview of the BTF concepts together with a short illustration. The paper ends in Sect. 4 with a summary and an outlook of future work.

## 2 Motivation & Related Work

In the area of business transaction automation there are a variety of issues, both from a technical and a business perspective, that need to be addressed and that need to be aligned before an automated execution of business transactions and processes with web service technology becomes feasible.

Starting with the technical perspective, it is obvious that a business process with long running business transactions requires a broad support of applicable transaction models to allow a consistent execution. First of all, the (sub) process structure (e.g., nesting, routing, etc) limits the suitable transaction models as transaction models usually support only a particular structure (e.g., the open nested transaction model supports nesting). Also the participants of a business process require particular transaction models as these participants can be heterogenous (e.g., databases, workflow engines or web services) in nature.

Current approaches (e.g., [2–5]) to address these above technical issues however only use one or a limited combination of transaction models, applicable only in particular environments (e.g., databases, workflows, web services) in an ad hoc manner. The focus of these transaction models (e.g., sagas, open nested, etc.) is on application level transactions and not on business level transactions. Other research into transaction support for workflows, e.g., [2], business processes and web services, i.e., WS-Tx [4, 5] and WS-CAF [6], is geared towards the use of a restricted context and domain specific set of transaction models. The combination of transaction models is usually fixed, while there is a need for support of any (suitable and feasible) combination of transaction models.

Several approaches to integrate advanced transaction models exist. The ACTA framework [1] facilitates the formal specification, analysis and composition of advanced database transaction models. In [7], a meta-model for web service transaction models is proposed. The approach in [7] aims at flexible transaction support for web services using existing transaction models. However, the sole focus is on the modeling and representation of transaction models (selecting or creating one transaction model per application as opposed to the composition of different transaction models as required for complex sets of business transactions), without any execution framework with a detailed architecture and a clear coupling between business process and transactions. Both, [1] and [7], are limited in their applicability because of their from database technology driven development without taking into account business and workflow requirements.

Apart from the incomplete satisfaction of technical issues, business issues are not addressed by current transaction models and only very limited by current research. When looking at a business process from a business perspective, transaction support is required for the complete business protocol that defines the public, agreed business interactions between business parties. The interactions involve negotiations, commitments, contracts, shipping and logistics, tracking, varied payment instruments and exception handling. In addition to these concepts, the phase a business process is in as well as the current level of the business process determine the required transaction support. In [8], a phased model is introduced that distinguishes between pre-transaction, main transaction and post-transaction phases in a collaborative business process. Because of aggregation and abstraction requirements in inter-organizational settings, [9] identified the need for a three level (external, interface and internal level) process framework to enable clear process descriptions in e-business system integration.

There is also a need for a more well-founded and formal approach towards the use of traditional transaction processing concerns of e-business. In [10], several atomicity criteria for business to consumer micro transactions that play a role in e-commerce are defined. The goal of our research is to use the atomicity criteria (like payment or contract atomicity) in large-scale e-business interactions as these provide business semantics at the business level to business transactions as opposed to the current, above mentioned, transaction models. A clear interrelation of these high level criteria with lower-lever transaction models needs to be established. This also enables the creation of a precise model that could help business partners in conflict situations (e.g., if a trading partner supports different atomicity criteria than another business partner).

## 3 The Business Transaction Framework

The BTF currently provides an initial conceptual model that will be briefly presented in this section. Currently, the focus of the framework is on the abstraction and composition of transaction models to address the technical issues as mentioned in Sect. 2. At present, we focus on the transaction support of business transactions from one participants viewpoint (i.e., orchestration oriented), and related to that, their business processes that contain these business transactions.

The conceptual model of the BTF describes and relates the concepts used in the framework. The framework's main concepts are distributed across three distinct phases (i.e., definition, composition and execution) to address the logical workflow of adding transaction support to business transactions. The need for a broad range of transaction models in a business conversation makes it necessary to define a transaction plan. The transaction plan connects different transaction models with each other to reliably execute the business process. The transaction models are the building blocks of a transaction plan, which in fact is a composition of transaction models. The definition of the building blocks is situated in the first phase of the framework. The second phase is concerned with the composition of transaction models from the first phase into the transaction plans. To be able to compose transaction models, existing transaction models with their properties and behavior must be analyzed and then it must be investigated what and how transaction models can be connected together and what the semantics of such a connection is. These first two phases are design-time phases. The compositions of transaction models can then be executed during the final, run-time (i.e., execution) phase of the framework.

The concept used in the first phase are the atomic building blocks of the framework called Abstract Transaction Constructs (ATC). An ATC is an abstract representation of a transaction model with its specific properties and functionality comparable to a transaction model class in object orientation. ATCs are (if allowed by the particular transaction model) composable horizontally, allowing to create a choreographed ordering of ATCs, and vertically, allowing to nest ATCs. The behavior of an ATC is parameterizable through its structural (i.e., the process structure) and behavioral (i.e., the behavior during run-time) interfaces. To investigate and achieve interoperability between and reusability of transaction models we propose a taxonomy of transaction models in which we initially group existing transaction models by using the abstraction relation into three categories, i.e., flat (e.g., WS-AT [4]), choreographed (e.g., LRAs [6]) and nested (e.g., WS-BA [5]) transaction models.

Injecting transactional capabilities on business processes requires as inputs the business process specification, the collection of ATCs, ATC composition rules and contracts constraining the other inputs. When composing the ATCs together by instantiating their structural interface, Abstract Composite Business Transactions (ACBT) are created. An ACBT is a structural transaction specification for a business process comparable to a workflow specification.

If a business process is started, the accompanying ACBT is instantiated by parameterizing the behaviorial interface of the ATCs connected in the ACBT. The result will be a Composite Business Transaction (CBT) with fully parameterized ATCs called Transactional Constructs (TC). TCs are the concrete transactions executed at some point during the business process.

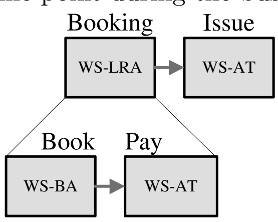


**Fig. 1.** ACBT example

In Fig. 1, a small example of an ACBT is presented. This ACBT is based on a simple (part of a) business process in which a travel can be arranged. The travel arrangement process contains one activity (*Booking*) with two subprocess steps and thereafter one further activity. Within the *Booking* activity, the subtasks of the *Book* subprocess (booking a hotel and a flight, both business transactions with other organizations; not shown) can be performed in parallel, after which the selected travel package is invoiced and the payment is registered and the tickets are electronically issued. Based on some process requirements for transaction support, ATCs can be composed into the presented ACBT. The WS-CAF LRA ATC is based on the SAGA transaction model and can be used to split the long-running *Booking* activity into smaller, shorter-running process steps, using compensation steps in case (part of) the process needs to rollback. A nested ATC (i.e., WS-BA) would suit the *Book* subprocess. A flat transaction model ATC (WS-AT) is used for the payment and issuing of the tickets. If the last activity (*Issue*) fails, it should be redone until it succeeds (forward recovery). Failing this activity should have no further impact on the rest of the already completed activities. Also, the payment task can be supported by a flat transaction model as its compensation is taken care of by the (higher level) WS-LRA.

## 4 Summary & Future Work

We explored the requirements for a business aware transaction framework (BTF) in service oriented environments. We introduced the concepts of the BTF to remedy current limitations of transaction support for business processes with business transactions that require an automated and consistent execution with clear semantics (in case of errors) in service oriented environments. Current approaches are not able to provide a platform-independent way of integrating required transaction models accommodating the various activities and their resources (databases, workflows, web services) throughout the business process into a cohesive whole. Representing transaction models by Abstract Transactional Constructs and composing these into Abstract Composite Business Transactions allows the flexible creation of reusable transaction plans to support business processes. The execution behavior of ACBTs is parameterizable at run-time in order to create Composite Business Transactions.

Future work for the BTF from a technical viewpoint will concentrate on the precise definition of the semantics of interactions between the transaction models. This will result in a set of composition rules for ATCs. Formal analysis is required to define the correctness criteria of resulting compositions. To support the BTF we are also currently developing a dynamic and flexible three-phased and three-level architecture. Once the technical requirements for the BTF are fulfilled, the business requirements will be addressed by incorporating unconventional atomicity criteria into ATCs. Next to the incorporation of existing standardization efforts from the business process arena (e.g., RosettaNet, EbXML), the BTF will make a shift from an orchestration oriented approach to a choreographed approach. We also plan to develop a proof of concept prototype.

## References

1. Chrysanthis, P.K., et al: Synthesis of extended transaction models using acta. ACM Transactions on Database Systems **19** (1994) 450–491
2. Grefen, P., et al: Database Support for Workflow Management: The WIDE Project. Kluwer Academic Publishers (1999)
3. Elmagarmid, A.K., ed.: Database Transaction Models for Advanced Applications. Morgan Kaufmann (1992)
4. Cabrera, L.F., et al: WS Atomic Transaction (WS-AT). (2004)
5. Cabrera, L.F., et al: WS Business Activity Framework (WS-BA). (2004)
6. Bunting, D., et al: WS Transaction Management (WS-TXM). (2003)
7. Hrastnik, P., et al: Using advanced transaction meta-models for creating transaction-aware web service environments. International Journal of Web Information Systems **1** (2005)
8. Papazoglou, M.: Web services and business transactions. World Wilde Web: Internet and Web Information Systems **6** (2003) 49–91
9. Grefen, P., et al: A framework for e-services: a three-level approach towards process and data management. IBM Research Rep. RC22378, IBM Res. Division (2002)
10. Tygar, J.D.: Atomicity in electronic commerce. Proc. of the 15th annual ACM symposium on principles of distributed computing, ACM Press (1996) 8–26