EXPLORING A FRAMEWORK FOR ADVANCED ELECTRONIC BUSINESS TRANSACTIONS

Alex Norta[1]

Alexander.Norta@cs.helsinki.fi

Abstract

With the emergence of service-oriented computing technology, companies embrace new ways of carrying out business transactions electronically. Since the parties involved in an electronic business transaction (eBT) manage a heterogeneous information-systems infrastructure within their organizational domains, the collaboration complexity is considerable and safeguarding an inter-organizational collaboration with an eBT is difficult, but of high significance. This paper describes a conceptual framework that pays attention to the complexities of an eBT and its differentiating characteristics that go further than traditional database transactions. Since the eBT is a framework that comprises separate levels, pre-existing transaction concepts are explored for populating the respective levels. To show the feasibility of the described eBT framework, industry initiatives that are aspiring to become business-transaction standards, are checked for eBT compatible characteristics. Since realizing an eBT framework raises many tricky issues, the paper maps out important research areas that require scientific attention. Essentially, it is required to investigate how the business semantics influences the nature of an eBT throughout its lifecycle.

Keywords: electronic business transactions, inter-organizational business collaboration.

Dept. Computer Science

University of Helsinki

Technical Report C-2008-211

[1] Alex Norta is a member of the CINCO-group, Department of Computer Science, University of Helsinki

1 INTRODUCTION

The emergence of electronic business promises companies a sustainable market advantage that comprises an integration and coordination of information flow and product flow between heterogeneous information-system infrastructures. Such information flow that bridges different organisations, includes the linking of business elements into an integrated whole. For such electronic business collaboration (eBC), a loose coupling of those information systems is a requirement as a tight coupling of information systems results in too many agreement details and too much shared context has to be revealed to the business counterpart.

In eBC, the registration of business transactions is of major legal importance for organisations. A business transaction (Little & Freund 2003) is a consistent change in the state of a business relationship that is driven by a well-defined business function. Each party in a business collaboration holds its own business transaction. For eBC, a transaction concept is important to ensure reliability.

To facilitate a loose coupling and highly dynamic establishment of business collaboration, the service oriented computing (SOC) paradigm is increasingly important. Services are self-describing, logical manifestations of physical resources that are grouped as a process, i.e., as a set of actions (Webber & Parastatidis 2003) that an organisation is prepared to execute and expose to the web.

With the complexity involved in eBC, no single transaction model is able to meet all requirements. Instead it is necessary to inter-organizationally establish transaction frameworks in a way that does not force companies into disclosing an undesirable amount of business internals. In this paper a conceptual model of an electronic business transaction (eBT) is put forward, based on an investigation of features that also incorporates business aspects and in which collaborating organizations can safeguard their business internals. Note that an eBT needs to safeguard the legally binding contractual relationships between collaborating parties that dictate responsibilities and the consequences of behaviour. The importance of business semantics in an electronic collaboration also has consequences for the nature of the atomicity, consistency, isolation, and durability properties of an eBT.

The remainder of this document is structured as follows. Section 2 discuses the challenges and specific characteristics of business transactions, followed by the description of an electronic business transaction framework that takes the previously mentioned characteristics into account. Section 3 shows which pre-existing transaction concepts may be used for the earlier presented electronic business-transaction framework. Section 4 discusses industry initiatives for electronic business transactions and investigates to which extent they realize the framework. Finally, Section 6 concludes this document and maps out future research directions.

2 A FRAMEWORK FOR E-BUSINESS TRANSACTIONS

The complexity of transactions that span multiple organisations rises in loosely coupled distributed computer networks that are enabled by SOC (Norta 2007). Here, business processes that use database systems must be inter-organizationally integrated. The transactional parts of a business process are referred to as a business transaction. By linking business processes inter-organizationally with electronic means, collaborating parties hope to safe time and money during the setup, enactment, and post-enactment of supply chains.

For eBC that takes place in a highly dynamic environment, applying conventional transaction mechanisms is not sufficient, as data from resources that are at the back-end of web services, would need to be locked in order to assure atomicity and isolation. However, locking data for isolation in long running transactions for eBC is unrealistic as this might block resources that are consequently not available for others. For example, locking tables for selling a product blocks other potential customers, results in lower turn-over, and prevents other organisations from participating in the business process.

The synchronization of business processes between organisations must be part of a wider business coordination protocol that defines the publicly agreed business interactions between business parties, which is based on web services. Additionally, well founded possibilities are missing to compose eBTs out of several transaction models (Weikum & Schek 1992) to support long running transactions for heterogeneous systems that are integrated in a loosely coupled fashion.

2.1 Features of electronic business transactions

An eBT is a consistent change in the state of the business that is carried out with electronic means and that is driven by a well-defined business function. An eBT is automated, complex, long running and may involve multiple internal and external parties. Additionally, an eBT requires commitments to the transaction that need to be negotiated by the participating organisations (Little & Freund 2003). Further features of an eBT are: support for the formation of contracts, shipping and logistics, tracking, varied payment instruments and exception handling. Compared to traditional database transactions, eBTs have several distinguishing characteristics. Firstly, they extend the scope of traditional transaction processing as they may encompass classical transactions which they combine with non-transactional processes. Secondly, they group both classical transactions as well as non-transactional processes together into a unit of work that reflects the semantics and behavior of their underlying business task. Thirdly, they are governed by unconventional types of atomicity that go further then traditional atomicity know from the domain of database systems, e.g., payment atomicity, goods atomicity, delivery atomicity, contract atomicity. In (Yu et al. 1998) these unconventional atomicities are described in further detail.

Unconventional behavioral features (Papazoglou et al. 2000) of an eBT are specified as follows: Generic characteristics tackle issues like who is involved in the transaction, what is being transacted, the destination of payment and delivery, the transaction time frame of permissible operations. Examples for special purpose characteristics are links to other transactions, receipts and acknowledgments, identification of money transferred outside national boundaries. Furthermore, advanced characteristics are the ability to support reversible and repaired transactions, the ability to reconcile and link transactions with other transactions, to specify contractual agreements, liabilities and dispute resolution policies, transactions that guarantee the integrity of information, confidentiality and non-repudiation; the ability for transactions to be monitored, logged and recovered.

2.2 Managing electronic business transactions

The integrated heterogeneous systems of an eBC need to be loosely coupled because of different reliability requirements that exist within long running eBTs. Different reliability requirements result from the properties of an eBT such as the phase the transaction is in and the level in which the transaction is taking place.

In (Papazoglou 2003) a phased model is introduced that distinguishes between pre-transaction, main transaction and post-transaction phases in a collaborative business process. In (Grefen et al. 2002) the need for a three level process framework is identified as companies are not willing to directly connect their legacy system. Enabling interoperability between systems of different organizations is not just a matter of coupling systems, as this introduces interoperability problems, like semantic differences, autonomy, non-disclosures, company ’secrets’, etc. Presenting the backend systems applications as services, i.e., wrapping them as web services, is a valid solution for most of the problems mentioned.

Given the complex features of eBT as described in Section 2.2, the conceptual model of Figure 1 represents an integration of the separate solution concepts (Papazoglou 2003, Grefen et al. 2002) that permits a manageable separation of concerns. In Figure 1, the organizational domains of a service consumer and a service provider are bridged by an external level where companies inter-organizationally harmonize their business collaboration and, hence, their business transaction. Along a time line, the external-level phases of an eBT are visualized that need to be coordinated with the eBT phases on the conceptual level within an organization. Finally, the conceptual-level coordinates the legacy system of the internal level, e.g., ERP systems, workflow systems, database systems, and so on. The latter give technical feedback to the higher level to inform about the success or failure of a transaction. Likewise, the conceptual level releases coordination information to the external level for aligning an eBT with the domain of the collaborating counterpart.

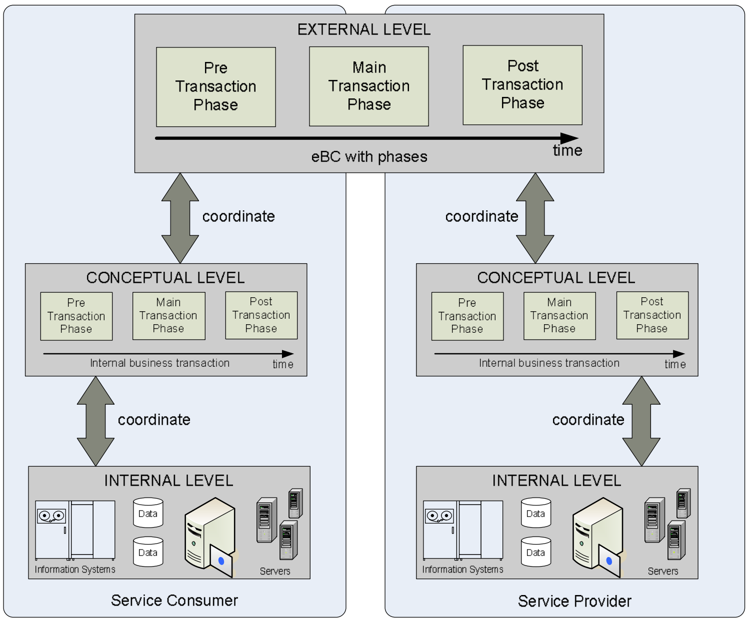

Figure 1 The levels and phases of the eBT framework

To control an eBT as depicted in Figure 1, spheres of control are a useful vehicle (Norta 2007) to demarcate process parts that are provided by a collaborating party. The theory of spheres of control (Bjork & Davis 1978) originates from the domain of traditional database transactions. So called workflow spheres (Leymann & Roller 2000) expand the transaction theory into the dynamic world of complex business processes. Those concepts are applied in (Heuvel & Artyshchev 2002) for analyzing atomicity criteria dependencies and atomicity spheres without relating the workflow concepts of highly dynamic inter-organization processes. In the work of (Tygar 1996) a substantial emphasis is put on the characteristic atomicity properties of e-business. These unconventional atomicities for spheres in electronic business transactions (eBT) are explored and related (Papazoglou 2003) to each other along the categories of system-level atomicity, business-interaction atomicity, and operational-level atomicity. The unconventional atomicities need to be part of a transaction model that pays attention to the business realities that form the context of eBC. In (Norta 2007) the concept called eSourcing uses spheres for enabling inter-organizational business collaboration.

Based on an elaborate historic survey (Wang & Grefen 205) that covers the development of business transactions up to the latest developments in the domain of SOC, the following Section 3 investigates what existing transaction concepts are suitable for populating which level of the eBT framework depicted in Figure 1. Furthermore, Section 4 uses the same historic transaction survey to explore if the eBT framework is realized in industry efforts that aspire to become a standard for electronic business transactions.

3 POPULATING eBT LEVELS WITH TRANSACTIONS

For the collaboration levels of Figure 1, pre-existing transaction concepts are available and assessed in this paper for eBT management on different levels of concern. The remainder of this section is structured as follows. First, Section 3.1 discusses transaction management for the internal level of an eBT, followed by Section 3.2 that comprises advanced transactions for the conceptual and external level. These advanced transaction concepts must be harmonized on the external level into transaction frameworks, which Section 3.3 covers.

3.1 Internal-Level Transactions

To traditionally manage data in a sound way, transactions must fulfil the following requirements. Atomicity states a transaction executes completely or not at all, consistency means a transaction preserves the internal consistency of the database, isolation infers a transaction executes as it were running alone with no other transactions, and finally, durability demands the results of a transaction are not be lost in a failure. These ACID properties of *flat transactions* are instrumental for handling exceptions in transaction management that are shielded from applications running on top of databases.

Flat transactions still dominate the database world because of their simple structures and they can be easily implemented Hence, ACID properties are suitable for the legacy systems of the internal level of an eBT as depicted in Figure 1. However, from a technical point of view, web-service composition faces the transactional challenges of relaxed atomicity, i.e., a situation where intermediate results may be kept without rollback despite the failure to complete the overall execution of a composite service.

3.2 Advanced Transactions for the Conceptual and External Level of an eBT

Advanced transaction models are extensions to flat transactions that release one or more ACID constraints to meet with specific requirements. Two strategies have been adopted for extension purposes to achieve different structures inside a transaction, namely the modularization of a complex transaction with hierarchies, and the decomposition of a long-lasting transaction into shorter sub-transactions. In the sequel the first type is referred to as distributed and nested transactions and the latter as chained transactions and Sagas.

A supporting concept for advanced transactions is the mechanism of savepoints (Astrahan et al. 1976) that enables a transaction rollback to an intermediate state for recovery. Savepoints are important for supporting recoveries in distributed and nested transactions. Furthermore, the use of checkpoints in transaction logs is instrumental in chained transactions to indicate a point until which a rollback does not result in an inconsistent database state. Details are explained in the sequel of this chapter.

3.2.1 Distributed and Nested Transactions

A *distributed transaction* is needed if an organization must integrate several database systems that reside on different servers in different geographic locations in a bottom-up way. As depicted in Figure 2, distributed transactions consist of sub-transactions, which may access multiple local database systems and comprise two types of transactions (Breitbart et al. 1992), namely local transactions and global ones. Local transactions are executed under the control of the local database management system (DBMS), while the multi-database system (MDBS) is in charge of global transactions. Hence, local and global integrity constraints must be aligned. Also the atomicity and isolation is managed globally when the whole transaction is aborted if any sub-transaction fails. The most influential example of distributed transactions is the X/Open Distributed Transaction Processing (X/Open DTP) software architecture (X/Open Company 1996) that is a standard for the two-phase commit (2PC) protocol. In combination with ACID transactions, a multiphase protocol like 2PC is used to ensure sound database-state changes.

A *nested transaction* is a generalization of savepoints (Gray & Reuter 1993), which is suitable for complex-structured applications and adopts a top-down method to decompose a complex transaction into sub-transactions or child transactions according to their functionalities (Moss 1981). In nested transactions, parts of a transaction may fail without aborting the entire transaction. Sub-transactions are composed in a hierarchical manner and only the leaf sub-transactions perform database operations while others function as coordinators. As Figure 2 shows, a sub-transaction is atomic and when it aborts, the parent may trigger another sub-transaction as an alternative without necessarily violating the database consistency.

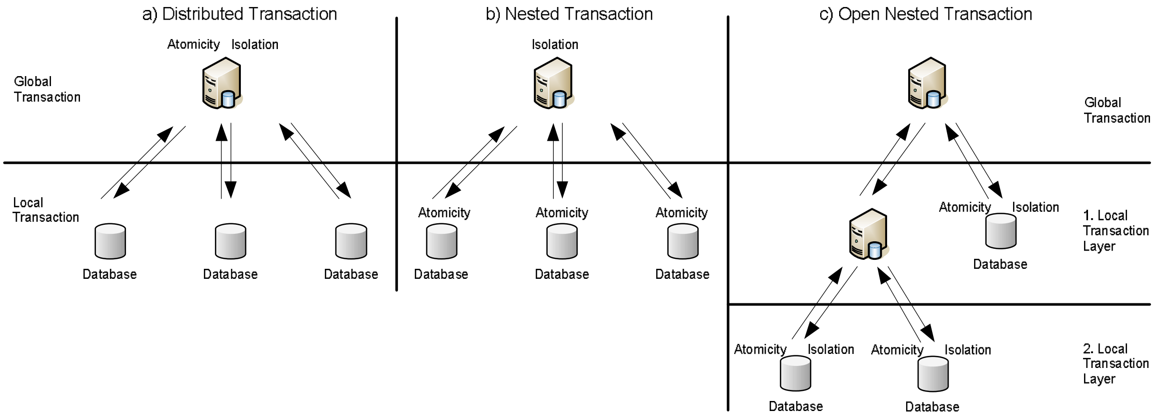

Figure 2 Simplified Informal Models of Distributed and Nested Transactions

Multilevel transactions are a variation of a nested transaction that are also called layered transactions (Weikum & Schek 1992) and their generalization is called open nested transactions (Lewis et al. 2001). In multilevel transactions a transaction tree has its layers corresponding to those of the underlying system architecture. Here, a pre-commit concept allows an early commitment of a sub-transaction before the root transaction actually commits, which requires a sub-transaction to semantically undo the committed one.

Multilevel transactions evolve to *open nested transactions* if the structure of the transaction tree is no longer restricted to layering, i.e., leaves in different layers are allowed (see Figure 2). Open nested transactions relax the ACID properties to achieve a higher level of concurrency. They guarantee global level isolation, which means in open nested transactions, the intermediate results of committed sub-transactions in nested transactions are invisible to other concurrently executing ones.

3.2.2 Chained Transactions and Sagas

Differently to a nested transaction and its extensions, a *chained transaction* is appropriate for a time-consuming application with long-lasting transaction processes. As depicted in Figure 3, a chained transaction is a variation of savepoints (Gray & Reuter 1993), where long running transactions are decomposed into small, sequentially executing sub-transactions that roughly correspond to savepoint intervals.

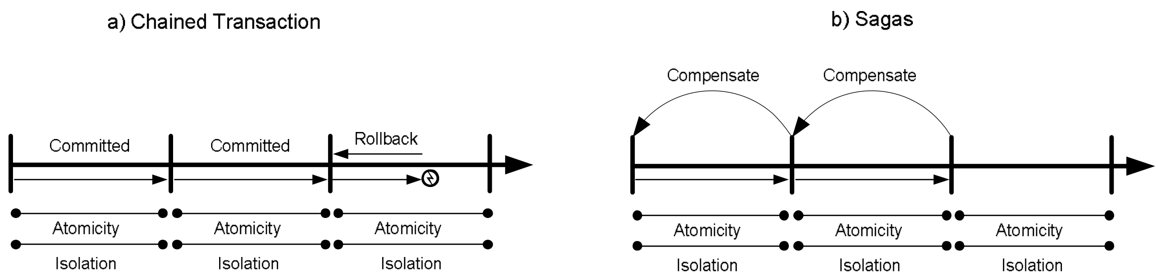

Figure 3 Simplified Informal Models of a Chained Transaction and Saga

The difference is that each sub-transaction is atomic, while each interval between every two save points is part of an atomic transaction. In the chain, a sub-transaction triggers the next upon commit until the whole chained transactions commit. When encountering a failure, the previously committed sub-transactions have already durably changed the database so that only the results of the currently executing sub-transaction are lost. This way the rollback only returns the system to the beginning of the most recently-executing sub-transaction.

Since the atomicity and isolation properties are relaxed in a chained transaction, this leads to aborting problems of the whole chain in the middle of execution as all the committed sub-transactions cannot be undone and other concurrent transactions see intermediate results generated during the execution of the chain.

Sagas (Garcia-Molina & Salem 1987, Chrysanthis & Ramamritham 1992) adopt the idea of chained transactions of including a compensation mechanism to roll back. As Figure 3 shows, Sagas divide a long lasting transaction into sequentially executed atomic sub-transactions with ACID properties and each sub-transaction, except the last one, has its own compensating sub-transaction. When any failure arises, the committed sub-transactions are undone by compensating sub-transactions. Unlike chained transactions, Sagas can return the whole transaction back to the very beginning with compensations.

3.3 Transaction Frameworks

In some transaction frameworks several transaction models are integrated, which is a prerequisite for external-level business-transaction harmonization. This section briefly introduces the two relevant frameworks, namely ACTA and BTF.

The comprehensive framework called ACTA (Chrysanthis & Ramamritham 1990) unifies existing models to capture the semantics and reason about the concurrency and recovery properties of complex transactions. More elaborate extensions to this ACTA model are described in (Chrysanthis & Ramamritham 1992, Chrysanthis & Ramamritham 1993, Chrysanthis & Ramamritham 1994). In the ACTA framework, interactions among transactions are expressed in terms of effects, i.e., effects of transactions on other transactions and effects of transactions on objects they access. ACTA captures the effects of transactions on objects by two objects sets and the concept of delegation. Every transaction is associated with several other objects that are contained in a view set or access set. The view set contains all the objects potentially accessible to the transaction while the access set contains the objects that have already been accessed by the transaction. Based on ACTA, the ASSET transaction model (Biliris et al. 1994) uses primitives at a programming language level based on ACTA building blocks such as ’history’, ’delegation’, ’dependency’, ’conflict set’ etc.

A more applicable approach than ACTA for the eBT framework of Figure 1, is explored in the XTC (eXecution of Transactional Contracted electronic services) project (Wang et al. 2006) that aims at laying a generic foundation to the transactional support for business processes in a service-oriented environment. Hence, a Business Transaction Framework (BTF) is developed to support contract-driven, inter-organizational business processes. A BTF is a transaction hierarchy composed of so-called Abstract Transaction Constructs (ATC) that comprises existing transaction models, which are stored in a library. The architecture of the BTF is a multi-level, multi-phase design (Kratz et al. 2005). Three phases exist along the BTF life-cycle, namely, definition phase, composition phase and execution phase. During the definition phase, the ATCs are abstracted from the classic and adopt the transaction models based on a taxonomy, which covers and classifies the existing work in the transaction management domain. After the design of an ATC library, these contained constructs are used to build a transaction plan for a complex process within the composition phase. The abstract plans resulting from the composition phase (Kratz 2005, Wang 2005) are instantiated to form real business transactions for the execution phase.

4 INDUSTRY INITIATIVES FOR E-BUSINESS TRANSACTIONS

For service-oriented eBC applications, supporting technologies and standards are needed to guarantee consistency and reliability in eBTs. While no transaction mechanism is widely accepted as a standard, there are three possible candidates, which realize the eBT framework of Section 2.2 to different degrees.

4.1 Business Transaction Protocol

The XML-based business transaction protocol BTP (Ceponkus et al. 2002) is not exclusively designed for web services. BTP is instrumental for representing and seamlessly managing complex, multi-step business-to-business (B2B) transactions over the Internet to ensure consistent outcomes of parties that use applications disparate in time, location and administration, and that participate in long running business transactions (Little 2002).

In a BTP compliant web-service environment, a transaction manager confirms or cancels the backend system a web service encapsulates. Hence, a direct communication exists between the transaction manager and the backend system, which contradicts the web-service philosophy. Opening up backend systems to play the role of participant within the transaction for external parties introduces security issues and bypasses the purpose of web services.

Every phase of a transaction within BTP (Dala et al. 2003) stands on its own and may be implemented in any way by a BTP compliant web service or application. To reflect the differences with the traditional 2PC protocol, the commands used in BTP are different and extended. Additionally, using business logic in BTP, the application also determines which participants to commit as a consensus group and which to cancel.

4.2 Web Services Transactions

The combined Web Services Transactions specifications that are indicated by WS-Tx, consist of WS-Coordination (WS-C) (Cabrera et al. 2005a), WS-AtomicTransaction (WS-AT) (Cabrera et al. 2005b), and WS-BusinessActivity (WS-BA) (Cabrera et al. 2005c). The specifications are aimed at the reliable and consistent execution of web based business transactions using different interconnected web services.

While in BTP the coordination of an eBC is interwoven with transaction management, WS-Coordination (WS-C) defines a framework that solely focuses on outcome determination and processing. This way WS-C provides a generic coordination infrastructure for web services, making it possible to plug in specific coordination protocols (Freund & Story 2002, Little & Webber 2003). Currently, the WS-Transaction specifications (WS-AT and WS-BA) are the first and only protocol specifications based on WS-Coordination.

The WS-AtomicTransaction (WS-AT) specification is focused on the existing transaction systems and protocols with strict ACID requirements. These systems are heterogeneous and coupling them together within one organization is the first step towards interoperability. The following protocols are specified in WS-AT: Completion, TwoPhase Commit (2PC) with two variants, Volatile 2PC, and Durable 2PC. Details of these protocols (Cabrera et al. 2005b) can be found in the WS-AT specification.

While the WS-AT specification resembles very much traditional 2PC ACID transactions with its problems, WS-BA is designed to support long running business transactions and uses atomic transactions to preserve the autonomy of participating organizations whilst at the same time providing mechanisms to reach overall agreement. The WS-BA specification defines two types for a coordinator, namely the atomic outcome type and mixed outcome type. The first type requires the coordinator to drive all participants to the same final state. The latter type allows a coordinator to choose which participants need to commit or compensate. The behaviour of the coordinator is determined by the application driving the activity. Besides the two coordination types, the following two coordination protocols (Cabrera et al. 2005c) are specified: BusinessAgreementWithParticipantCompletion and BusinessAgreementWithCoordinator-Completion. The reader is referred to the specification for details on these protocols.

4.3 Web Services Composite Application Framework

The purpose of WS-CAF (Bunting et al. 2003a) is to develop an interoperable, easy to use framework for composite web services applications. WS-CAF is composed of a series of specifications consisting of WS-CTX (Bunting et al. 2003b), WS Coordination Framework (Bunting et al. 2003c) and WS Transaction Management (Bunting et al. 2003d). Each specification covers a certain level of the overall architecture required to build reliable business applications that span multiple systems and use web service technology.

In contrast to BTP and WS-Tx, the WS-CTX specification (Bunting et al. 2003b) defines a generic context management mechanism for sharing common system data (i.e., context) across multiple web services (Newcomer 2004). Compared to WS-CTX, WS-Coordination combines both context and coordination, while in BTP, context, coordination as well as transaction management are combined. WS-CTX (Bunting et al. 2003b) makes it possible to connect multiple web services into one activity or scope for correlating them with specific context information that is not managed by a coordinator.

The second layer of WS-CAF is the Web Service Coordination Framework WS-CF which provides a coordination service that is plugged into WS-CTX. It manages and coordinates multiple web services that are grouped in one or more activities to perform some task together. The WS-CF architecture has three main components. The *Coordinator* at which *Participants* can register so that they receive the context and outcome of an activity, and the *Coordination Service*, which defines the behaviour for a specific coordination model.

On top of the coordination framework, Web Service Transaction Management WS-TXM (Bunting et al. 2003d) specifies three different transaction protocols. These protocols can be used to reach an agreement of outcome among the participants of a transaction in a consistent way. To do this, the transaction protocols can use context information and coordination protocols.

The WS-TXM protocols are plugged into WS-CF and can be used with a coordinator to negotiate a set of actions for all participants that need to be executed, based on the outcome of a series of related web services executions (Bunting et al. 2003a). The web service executions are linked together in scopes by the overall context and can be nested and executed concurrently (Little & Webber 2003). WS-TXM binds the scope of an activity to the lifetime of a transaction.

Three specific transaction models are defined in WS-TXM that can be used for different situations: *ACID Transaction* resembles the traditional ACID transactions that is applies to web services, enabling tightly-coupled network-based transactions, which are most suitable, just like WS-AT, to achieve interoperability between existing transaction systems within one organization. *Long Running Action (LRA)* is designed to cover transactions that have a long duration. In LRA, an activity is seen as a set of business interactions for which compensation is possible comparable to Sagas. *Business Process Transaction Model* tries to integrate different heterogeneous transaction systems, e.g., using ACID transactions and messaging, from different business domains into one overall business-to-business transaction (Little & Webber 2003).

4.4 Comparison of Industry Initiatives

In a comparison of BTP and WS-Tx (Little & Webber 2003), both specifications address the problems of running transactions with web services while the differences in critical areas are present, e.g., transaction interoperability. The main problem of BTP is that it needs to leverage the ACID transactions that underlie the strongly coupled internal information infrastructures instead of replacing them with new models to design transactions for loosely coupled web services.

In (Kratz 2004) all three specifications are compared with the conclusion that a need exists for one open standard to realize the interoperability both in web services and business areas, possibly by integrating the existing ones within the WS-CAF framework. According to (Jin & Goschnick 2003), BTP is the most promising standard candidate for transaction management in combination with agent technology.

With the exception of BTP, the WS-Tx and the WS-CAF initiative show that a development is pursued towards an eBT with ACID properties on a lower level and advanced business transactions on a higher level. Still, the latter two industry initiatives lack the ability to create elaborate transaction frameworks for inter-organizationally harmonizing heterogeneous transactions of collaborating business domains.

5 RELATED WORK

In the area of workflow-oriented research, inter-organizational business transaction concepts were developed. Hence, the two most relevant research projects are described below. In (Grefen et al. 1997), a two-layer transaction model, known as the WIDE transaction model, is presented. The model combines the concept of savepoints with Sagas (Garcia-Molina & Salem 1987) so that more flexibility is offered in compensation paths in case of exceptions. The bottom layer consists of local transactions with a nested structure that conform to the ACID properties (Boertjes et al. 1998). The upper layer is based on Sagas that roll back the completed sub-transactions using the compensation mechanism, thus, relaxing the requirement of atomicity (Vonk et al. 1999). The semantics of the upper layer is formalized using simple set and graph theory (Grefen et al. 2001)). The local transaction layer is designed to model low-level, short-living business processes, whilst the global transaction models high-level and long-living business processes.

The flexible approach of WIDE is adopted in the CrossFlow project (Vonk & Grefen 2003) and developed into the more comprehensive X-transaction model. The X-transaction model is a three-level, compensation-based transaction model to support cross-organizational workflow management, namely the outsourcing level, the contract level and the internal level, each with a different visibility to the consumer or the provider organization. The X-transaction model views an entire workflow process as a transaction. For intra-organizational processes, X-steps are divided into smaller I-steps that adhere to ACID properties. Each I-step has a compensating step in case of failure.

6 CONCLUSION

This paper specifies a conceptual framework for managing the complex issues of electronic business transactions where a separation of concerns is achieved by establishing different coordinated transactional levels within and across organizational domains. That way it is not required to directly link the respective legacy systems of business parties who simultaneously are able to protect their competitive advantage. The paper discusses which pre-existing transaction concepts are employable for which level of an eBT. Furthermore, industrial initiatives are investigated for the extent to which they realize the electronic business transaction framework.

For future research activities about electronic business transactions, several domains need to be investigated. For specifying the requirements of a suitable transactional model, the requirements of business transactions in business process must be explored. The semantics of business collaborations must be clarified as a foundation for developing an automated composition of several transaction concepts for a heterogeneous system environment. For such automated rules-based composition, existing transaction models must be analyzed and formalized into abstract transactional primitives.

Such formalization also needs to cover unconventional atomicities for electronic business transactions. Hence, the automated composition needs to integrate traditional technical transactions from the domain of database management with inter-organizational electronic business transactions that are long-lived open and nested, that have unconventional atomicities and are suitable for a service-oriented collaboration environment. If exceptional situations occur in an electronic business transaction, it must be ensured that a semantic rollback of already carried out real-world collaboration activities is supported. Besides atomicities, it also needs to be investigated how isolation, consistency, and durability are affected in an electronic business transaction when the semantics of a business collaboration is taken into account.

References

Astrahan, M. M. and M. W. Blasgen, D. D. Chamberlin, Kapali P. Eswaran, J. Gray, P. P. Griffiths, W. F. King III, R. A. Lorie, P. R. McJones, J. W. Mehl, G. R. Putzolu, I. L. Traiger, B. W. Wade, and V. Watson, (1976). System r: Relational approach to database management. *ACM Trans. Database Syst.*, 1(2):97–137.

Biliris, A. and S. Dar, N. H. Gehani, H. V. Jagadish, and K. Ramamritham, (1994). Asset: A system for supporting extended transactions. In Richard T. Snodgrass and Marianne Winslett, editors, *SIGMOD Conference*, pages 44–54. ACM Press.

Bjork, V. and C.T. Davis, (1978) Data processing spheres of controldata.

Box, D. and D. Ehnebuske, and G. Kakivaya et al., (2003) *Simple Object Access Protocol (SOAP) 1.1.* http://www.w3.org/TR/SOAP/.

Bunting, D. and M. Chapman, O. Hurley, M. Little, J. Mischkinsky, E. Newcomer, J. Webber, and K. Swenson, (2003a). *Web Services Composite Application Framework (WSCAF)*, July 2003.

Bunting, D. and M. Chapman, O. Hurley, M. Little, J. Mischkinsky, E. Newcomer, J. Webber, and K. Swenson, (2003b). *Web Service Context (WS-CTX)*, July 2003.

Bunting, D. and M. Chapman, O. Hurley, M. Little, J. Mischkinsky, E. Newcomer, J. Webber, and K. Swenson, (2003c). *Web Service Coordination Framework (WS-CF)*, July 2003.

Bunting, D. and M. Chapman, O. Hurley, M. Little, J. Mischkinsky, E. Newcomer, J. Webber, and Keith Swenson, (2003d). *Web Services Transaction Management (WS-TXM)*, July 2003.

Boertjes, E. and P.W.P.J. Grefen, J. Vonk, and P.M.G. Apers, (1998). An architecture for nested transaction support on standard database systems. In G. Quirchmayr, E. Schweighofer, and T.J.M. Bench-Capon, editors, (1998), *DEXA*, volume 1460 of *Lecture Notes in Computer Science*, pages 448–459. Springer.

Breitbart, Y. and H. Garcia-Molina, and A. Silberschatz, (1992). Overview of multidatabase transaction management. *VLDB J.*, 1(2):181–239.

Cabrera, L. F. and G. Copeland, M. Feingold, R. W. Freund, T. Freund, J. Johnson, S. Joyce, C. Kaler, J. Klein, D. Langworthy, M. Little, A. Nadalin, E. Newcomer, D. Orchard, I. Robinson, J. Shewchuk, and T. Storey, (2005a). Web service coordination (WS-coordination), August 2005.

Cabrera, L. F. and G. Copeland, M. Feingold, R. W. Freund, T. Freund, S. Joyce, J. Johnson, C. Kaler, J. Klein, D. Langworthy, M. Little, A. Nadalin, E. Newcomer, D. Orchard, I. Robinson, T. Storey, and S. Thatte, (2005b). *Web Services Atomic Transaction (WSAtomicTransaction)*, August 2005.

Cabrera, L. F. and G. Copeland, R. W. Freund, T. Freund, S. Joyce, J. Klein, D. Langworthy, M. Little, F. Leymann, E. Newcomer, D. Orchard, I. Robinson, T. Storey, and S. Thatte, (2005c). *Web Services Business Activity Framework (WS-BusinessActivity)*, August 2005.

Ceponkus, A. and S. Dalal, T. Fletcher, P. Furniss, A. Green, and B. Pope, (2002). Business transaction protocol version 1.0, June 2002.

Chrysanthis, P. K. and K. Ramamritham, (1990). Acta: A framework for specifying and reasoning about transaction structure and behavior. In Hector Garcia-Molina, H. V. Jagadish: Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ, May 23-25, 1990. ACM Press 1990, pages 194–203.

Chrysanthis, P. K. and K. Ramamritham (1992). Acta: The saga continues. In *Database Transaction Models for Advanced Applications*, pages 349–397. Morgan Kaufmann, 1992.
Chrysanthis, P. K. and K. Ramamritham. Delegation in acta to control sharing in extended transactions. IEEE Data Eng. Bull., 16(2):16–19, 1993.
Chrysanthis, P. K. and K. Ramamritham. Synthesis of extended transaction models using acta. *ACM Trans. Database Syst.*, 19(3):450–491, 1994.
Dalal, S. and S. Temel, M. Little, M. Potts, and J. Webber, (2003). Coordinating business transactions on the web. *Internet Computing, IEEE*, 7(1):30–39, January / February 2003.
Freund, T. and T. Story, (2002). Transactions in the world of web services, part 1. Web Article, 8.
Garcia-Molina, H. and K. Salem, (1987). Sagas. In U. Dayal and I.L. Traiger, editors. *SIGMOD Conference*, pages 249–259. ACM Press.
Gray, J. and A. Reuter, (1993). *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann.
Grefen, P.W.P.J. and H. Ludwig, and S. Angelov, (2002). A framework for e-services: a three-level approach towards process and data management. In 02-07, C. T. R., editor, University of Twente. (also appeared as: IBM Research Report RC 22378; IBM Research Division, 2002).
Grefen, P.W.P.J. and J. Vonk, and P.M.G. Apers, (2001). Global transaction support for workflow management systems: from formal specification to practical implementation. *VLDB J.*, 10(4):316–333.
Grefen, P.W.P.J. and J. Vonk, E. Boertjes, and P. M. G. Apers, (1997). Two-layer transaction management for workflow management applications. In Abdelkader Hameurlain and A. Min Tjoa, editors, *DEXA*, volume 1308 of *Lecture Notes in Computer Science*, pages 430–439. Springer.
Hanemann, A. editor, (2005). *IBM PhD Student Symposium at the 3rd International Conference on Service Oriented Computing (ICSOC 2005) Amsterdam, The Netherlands, December 12, 2005, Proceedings*, IBM Research Report RC23826. IBM Research Division.
Heuvel, W.J. van den and S. Artyshchev, (2002). Developing a three-dimensional transaction model for supporting atomicity spheres. Proceedings of NetObjectDays 2002 vol. 2.
Jin, T. and S. Goschnick, (2003). Utilizing web services in an agent-based transaction model (abt). In *International workshop on Web Services and Agent-based Engineering*.
Kratz, B. (2004). Protocols for long running business transactions. Technical report, Infolab, Tilburg University.
B. Kratz, (2005). A business aware transaction framework for service-oriented environments. In Hanemann (2005).
Kratz, B. and T.Wang, P.W.P.J. Grefen, and J. Vonk, (2005). Flexible business-transaction composition in service-oriented environments. Technical report, Eindhoven University of Technology.
Lewis, P.M. and A. J. Bernstein, and M. Kifer, (2001). *Databases and Transaction Processing: An Application-Oriented Approach*. Addison-Wesley.
Leymann, F. and D. Roller, (2000). *Production Workflow: Concepts and Techniques*. Prentice-Hall, N.J.
Little, M. (2002). Business transaction protocol: Transactions for a new age. *Web Services Journal*, 2(11):50–55.
Little, M. and T. J. Freund, (2003). A comparison of web services transaction protocols. http://www-106.ibm.com/developerworks/webservices/library/ws-comproto/.
Little, M. and J. Webber, (2003). Introducing WS-CAF - more than just transactions. *Web Services Journal*, 3(12):52–55, December2003.
Moss, J.E.B. (1981). *Nested Transactions: An Approach to Reliable Distributed Computing*. PhD thesis, M.I.T., April 1981. Also available as Technical Report MIT/LCS/TR-260.
Newcomer, E. (2004). Context, coordinators, and transactions – the importance of WS-CAF.
Norta, A. (2007), *Exploring Dynamic Inter-Organizational Business Process Collaboration*, PhD Thesis, Technology University Eindhoven, Department of Information Systems.
Papazoglou, M.P. (2003). Web Services and Business Transactions. *World Wide Web: Internet and Web Information Systems*, 6:49-91.

Papazoglou, M.P. and D. Georgakopoulos, (2003). Service-oriented computing. Communications of the ACM, 46(10):25–28.

Papazoglou, M.P. and P. Ribbers, A. Tsalgatidou (2000), Integrated value chains and their implications from a business and technology standpoint, *Decision Support Systems* (29)4, pp. 323-342

Tygar, J.D. (1996). Atomicity in electronic commerce. In *Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing (PODC'96)*, pages 8-26, New York.

Vonk, J. and P.W.P.J. Grefen, E. Boertjes, and P.M.G. Apers, (1999). Distributed global transaction support for workflow management applications. In T.J.M. Bench-Capon, G. Soda, and A. Min Tjoa, editors, *DEXA*, volume 1677 of *Lecture Notes in Computer Science*, pages 942–951. Springer.

Vonk, J. and P.W. P. J. Grefen, (2003). Cross-organizational transaction support for e-services in virtual enterprises. *Distributed and Parallel Databases*, 14(2):137–172.

Wang, T. (2005). Towards a transaction framework for contract-driven, service-oriented business processes. In Hanemann 2005.

Wang, T. and P. Grefen, (2005). *A historic survey of transaction management from flat to grid transactions.*, Beta Working Paper (Rep. No. WP-138), 22 pp.

Wang, T. and P. Grefen, and J. Vonk, (2006). Abstract transaction construct: Building a transaction framework for contract-driven, service-oriented business processes. In A. Dan and W. Lamersdorf, editors, *Service-Oriented Computing – ICSOC 2006, 4th International Conference*, Volume 4294 of *Lecture Notes in Computer Science*, pages 352.364, Chicago, USA. LNCS Springer.

Webber, J. and S. Parastatidis, (2003). Demystifying service-oriented architecture. *Web Services Journal*, 3(11):40–44.

Weikum, G. and H.-J. Schek, (1992). Concepts and applications of multilevel transactions and open nested transactions. In Elmagarmid, A., editor, Database transaction models for advanced applications, chapter 13, pages 515–553. Morgan Kaufmann Publishers, San Mateo, California, USA.

X/Open Company Ltd., (1996). Distributed transaction processing: Reference model, version 3.

Yu, E. et al., (1998). From Organization Models to System Requirements: a “Cooperating Agents” Approach. In M.P. Papazoglou and G. Schlageter, editors, *Cooperative Information Systems: Trends and Directions. Academic Press*.