In: Proceedings of the 8^{th} International Workshop on Foundations of Models and Languages for Data and Objects: Transactions and Database Dynamics (TDD'99), pages 193-202. Schloss Dagstuhl, Germany, September 1999. Lecture Notes in Computer Science (LNCS), Volume 1773, Springer-Verlag.

Execution Guarantees in Electronic Commerce Payments

Heiko Schuldt, Andrei Popovici, and Hans-Jörg Schek

Database Research Group
Institute of Information Systems
ETH Zentrum, 8092 Zürich, Switzerland
{schuldt,popovici,schek}@inf.ethz.ch

Abstract. Electronic Commerce over the Internet is one of the most rapidly growing areas in todays business. However, considering the most important phase of Electronic Commerce, the *payment*, it has to be noted that in most currently exploited approaches support for at least one of the participants is limited. From a general point of view, a couple of requirements for correct payment interactions exist, namely different levels of atomicity in the exchange of money and goods of a single customer with different merchants. Furthermore, as fraudulent behavior of participants in Electronic Commerce has to be considered, the ability to legally prove the processing of a payment transaction is required. In this paper, we identify the different requirements participants demand on Electronic Commerce payment from the point of view of execution guarantees and present how payment interactions can be implemented by transactional processes. Finally, we show how the maximum level of execution guarantees can be provided for payment processes in a natural way by applying transactional process management to an Electronic Commerce *Payment Coordinator*.

1 Introduction

Along with the enormous proliferation of the Internet, Electronic Commerce is continuously gaining importance. The spectrum of applications that are subsumed under the term Electronic Commerce leads from rather simple orders performed by Email to the purchase of shopping baskets consisting of several goods originating from different merchants while electronic cash tokens are spent for payment purposes.

Remarkably, Electronic Commerce is a very interdisciplinary research area. As existing approaches are powered by different communities (i.e., cryptography, networking, etc.), they are very heterogeneous in nature and thus always focus on different special problems. From the point of view of the database community, atomicity properties have been identified as one key requirement for payment protocols in Electronic Commerce [15,16]. The more complex interactions with consumers and merchants become, the more dimensions of atomicity have to be addressed. In the simplest case, only money has to be transferred atomically from

the consumer to the merchant. However, considering complex shopping baskets filled with (electronic) goods from several merchants, atomicity may also be required for the purchase of all these goods originating from different possibly independent and autonomous sources, along with the atomic exchange of money and all goods.

Due to their distributed nature, protocols that have been suggested to support payment atomicity in Electronic Commerce impose high requirements on the participating instances (e.g., NetBill [4]). In these approaches, each participant not only has to implement a given set of interfaces. Since the application logic of these payment approaches is not centrally defined but distributed to all participants, they also impose high prerequisites to the participating instances.

However, with a centralized payment coordinator, the complex interactions of the various participants can be embedded within a payment process, thus reducing the prerequisites for merchants and customers to participate in Electronic Commerce. Transactional process management [13] can then be exploited in order to provide the necessary execution guarantees for Electronic Commerce payment processes in a natural way.

This paper is structured as follows: In Section 2, we provide a general framework for Electronic Commerce payment interactions. Based on this framework, we analyze the different atomicity requirements for Electronic Commerce payment (Section 3). Then, in Section 4, we shortly summarize transactional process management and describe how these ideas can be exploited in order to let Electronic Commerce payment process benefit from the execution guarantees provided by a payment process coordinator. Section 5 finally concludes the paper.

2 Schema for Payment Protocols in Electronic Commerce

The description for sales interactions in non-electronic markets [12] encompasses three phases: information, negotiation, and payment. During the information phase, a customer evaluates and compares the offers of several merchants. After selecting the best offer, she negotiates with the chosen merchant the conditions for the deal (negotiation). If they reach an agreement, the last step (the payment) involves the money transfer from customer to merchant and the service (the merchant fulfills his contract).

Most electronic payment systems only focus on the money transfer of the last phase. Our view of an *electronic payment scheme* also considers the systems and protocols for accomplishing both the money transfer and the service.

2.1 Participants

An electronic payment scheme involves participants originating from two distinct worlds: on the Internet side there are the customer, the merchant, and the payment server (also known as payment gateway) as a third entity which coordinates the former ones. The other side is represented by the financial world

with its proprietary network infrastructure and protocols. The participants are financial institutes and again the payment server, that has to consistently transform the data flow on the Internet side in corresponding "real world" money flow. The participants are depicted in Figure 1.

2.2 Steps of an Electronic Commerce Transaction

Prior to the payment transaction, the participants are involved in an *initialization* phase, depicted in Figure 1 by dashed arrows. Both customer and merchant have to establish accounts within the financial institutes "issuer" (or "acquirer", resp.). The transformation of the electronic money into real money is performed using these accounts. Also in this phase the customer receives from his bank a *customer secret* which enables him to perform electronic payments. The customer secret is visible only for the customer herself, for the issuing bank and (eventually) for the payment server. The most common form of the customer secret is the credit card number, in electronic cash schemes (such as eCash^{TM} [5]), the customer secret is an E-cash token. Because account operations are rather less often than payments, we can consider them as part of the initialization phase.

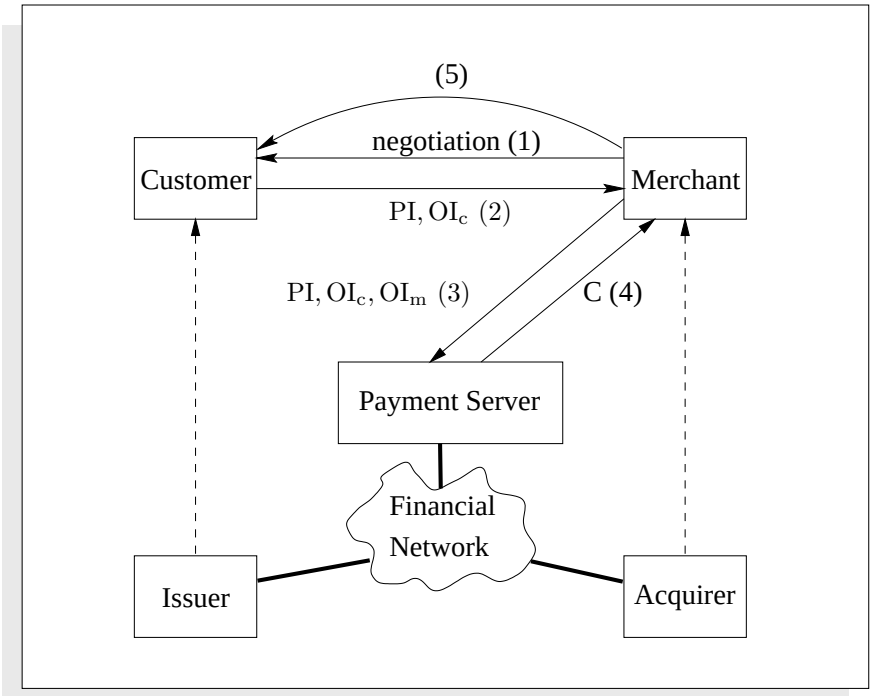

Fig. 1. Generic payment steps

Almost all the payment schemes contain the following steps, marked in Figure 1 with numbers 1 to 5:

- Negotiation (1): the customer selects the desired service or merchandise she wants from the merchant, and negotiates with the merchant the price of the service. The result of this step is the Order Information. The Order Information is a protocol of the negotiation phase, including service (merchandise) and price specification.
- Payment order (2): the customer sends Payment Information (PI) and Order Information (OI_c) to the merchant. The OI_c is the customer's view of the agreement with the merchant.
- Payment authorization (3): the merchant forwards PI, OI_c, OI_m and additional data to the payment server. OI_m is the merchant's view of the agreement with the customer.
  The payment server directly or indirectly verifies the validity of the payment information, the consistency of the payment using OI_c and OI_m. It eventually triggers the real world money transfer using its role on the non-Internet side. At the end of the payment authorization, the merchant receives a confirmation message C from the payment server (4).
- Purchase response (5): The merchant sends himself a confirmation to the customer. In case of electronic (non-tangible) goods, the purchase response can be immediately followed by the merchandise or the service itself.

In most existent payment protocols, the payment server is invoked by the merchant. This is not an intrinsic restriction, and communication between customer and payment server is also possible.

2.3 Characteristics of Payment Protocols

Several criteria serve as classification models of electronic payments schemes. Starting from the moment of transformation of real money into electronic money, payment protocols can be split in *pre-paid systems* and *pay-by-instruction* ones. *Atomicity* is another item, which will be discussed in detail later. Some protocols introduce the notion of *provability*, which is the ability of each party to prove their correct interactions. *Anonymity* is especially addressed by cash-based-systems. There are also implementation issues like *scalability*, *flexibility*, *efficiency*, *ease of use* and *off-line operation*, which are also important because of the large number of persons expected to use these systems.

3 Atomicity in Electronic Commerce

One key requirement in Electronic Commerce is to guarantee atomic interactions between the various participants in Electronic Commerce payment. As Electronic Commerce and thus also payment takes place in a highly distributed and heterogeneous environment, various aspects of atomicity can be identified: aside of money and goods atomicity [15,16], also the atomic interaction of a customer with multiple merchants is needed. In what follows, we analyze and classify these different atomicity requirements in detail.

3.1 Money Atomicity

The basic form of atomicity in Electronic Commerce is associated with the transfer of money from the customer to the merchant. This is denoted by the term *money atomicity* [15]. As no viable Electronic Commerce payment solution can exist without supporting this atomicity property, multiple solutions have been proposed or are already established [9,5]. However, the atomicity property is tightly coupled with the protocol architecture and design.

3.2 Certified Atomic Delivery

Aside of money, also goods have to be transferred. Therefore, a further requirement is that the delivery takes place atomically. This can even be reinforced in that both associated parties —customer and merchant— require the necessary information in order to prove that the goods sent (or received, resp.) are the ones both parties agreed to in the initial negotiation phase (*certified atomic delivery*, encompassing the goods atomicity and the certified delivery described in [15]). This strengthened requirement results from the fact that —in contrast to traditional distributed database transactions where only technical failures have to be addressed— in Electronic Commerce also fraudulent behavior of participants has to be coped with.

Especially when dealing with goods that can be transferred electronically, the combination of money atomicity and certified delivery is an important issue. In [3], this is realized by a customized Two-Phase-Commit protocol (2PC) [6]. To support both dimensions of atomic interactions and to avoid a payment coordinator to deal with the goods to be transferred, cryptographic mechanisms are applied. Prior to the payment process, the merchant sends the ordered goods in an encrypted way to the customer. On successful termination of the payment, the coordinator has both to transfer the money from the customer to the merchant and the key needed for the decryption of the previously received goods to the client in an atomic way.

3.3 Distributed Purchase Atomicity

In many Electronic Commerce applications, interaction of customers is not limited to a single merchant. Consider, for instance, a customer who wants to purchase specialized software from a merchant. In order run this software, she also needs an operating system which is, however, only available from a different merchant. As both goods individually are of no value for the customer, she needs the guarantee to perform the purchase transaction with the two different merchants atomically in order to get both products or none. *Distributed purchase atomicity* addresses the encompassment of interactions with different independent merchants into one single transaction.

This problem is in general reinforced by the fact that different heterogeneous interfaces are involved and different communication protocols are supported by

the participating merchants. To this end, in order to support distributed purchase atomicity in very heterogeneous environments with applications using communication protocols for which there are no transactional variants (such as, for instance, HTTP), the Transaction Internet Protocol (TIP) [8] has been proposed. TIP is based on the Two-Phase-Commit protocol (2PC). The main idea of this protocol is to separate communication between transaction managers from the application communication protocol (two-pipe-model). While communication at transaction manager level takes place by the TIP 2PC protocol, arbitrary protocols can independently be exploited at application communication level (such as, for instance, HTTP).

3.4 Summary of Atomicity Requirements

In Figure 2, the three dimensions of atomicity that can be identified in Electronic Commerce applications are depicted. Most currently deployed payment coordinators only support money atomicity while some advanced systems also address distributed purchase atomicity. However, to our best knowledge, all three dimensions are not provided by existing systems and protocols although the highest level of guarantees would be supported and although this is required by a set of real-world applications.

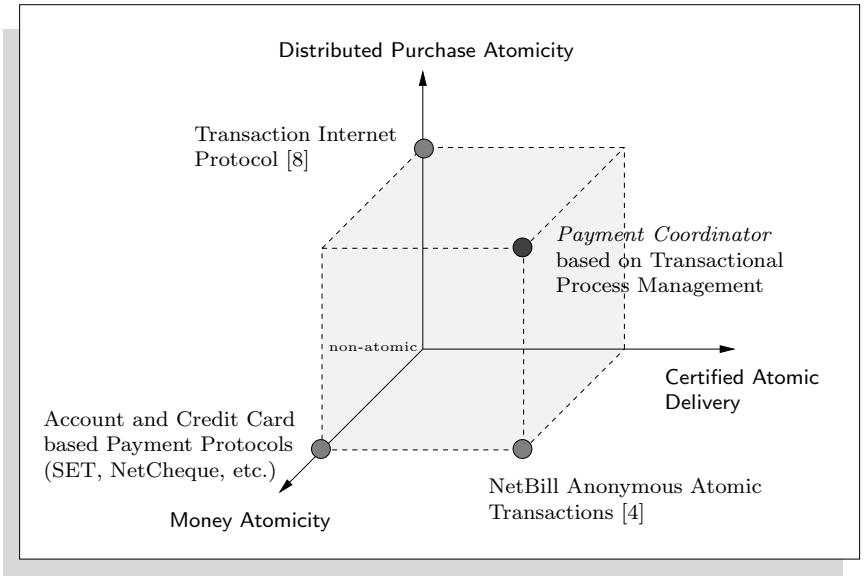

Fig. 2. Classification of Atomicity in Electronic Commerce

This lack of support for full atomicity in Electronic Commerce payment is addressed in this paper where we apply transactional process management (Section 4) to realize an Electronic Commerce payment coordinator.

4 Transactional Process Management

In this Section, we introduce the theory of transactional process management that provides a joint criterion for the correct execution of processes with respect to recovery (when failures of single processes have to be considered) and concurrency control (when multiple parallel processes access shared resources simultaneously) and we point out how this theory can be applied for payments in Electronic Commerce.

4.1 Overview

In conventional databases, concurrency control and recovery are well understood problems. Unfortunately, this is not the case when transactions are grouped into entities with higher level semantics, such as *transactional processes*. Since concurrent processes may access shared resources simultaneously, consistency has to be guaranteed for these executions.

Transactional process management [13] has to enforce consistency for concurrent executions and, at the same time, to cope with the added structure found in processes. In particular, and unlike in traditional transactions, processes introduce flow of control as one of the basic semantic elements. Thus, it has to be taken into consideration that processes already impose ordering constraints among their different operations and among their alternative executions. Similarly, processes integrate invocations to applications with different atomicity properties (e.g., activities may or may not be semantically compensatable).

The main components of transactional process management consist of a coordinator acting as top level scheduler and several transactional coordination agents [14] —one for each subsystem participating in transactional processes— acting as lower level schedulers. Processes encompass *activities* which are invocations in subsystems scheduled by the coordinator. Firstly, the execution guarantees to be provided by the coordinator include guaranteed termination, a more general notion of atomicity than the standard all or nothing semantics which is realized by partial compensation and alternative executions. Secondly, the correct parallelization of concurrent processes is required and thirdly, by applying the ideas of the composite systems theory [1], a high degree of parallelism for concurrent processes is to be provided.

The key aspects of transactional process management can briefly be summarized as follows: The coordinator acts as a kind of transaction scheduler that is more general than a traditional database scheduler in that it

i.) knows about semantic commutativity of activities,
ii.) knows about properties of activities (compensatable, retriable, or pivot, taken from the flex transaction model [10,17]), and
iii.) knows about alternative executions paths in case of failures.

Based on this information, the coordinator ensures global correctness but only under the assumption that the activities within the processes to be scheduled themselves provide transactional functionality (such as atomicity, compensatability, order-preservation, etc.).

4.2 Application of Transactional Process Management in Electronic Commerce

According to [11], trade interactions between customers and merchants can be classified in three phases: pre-sales, sales, and post-sales. While the sales phase has a well-defined structure (especially the payment processing, see Section 2), this is in general not the case for the pre-sales and the post-sales phase.

Due to this well-defined structure, processes are a highly appropriate means to implement the interactions that have to be performed for payment purposes. Furthermore, the atomicity requirements for payments in Electronic Commerce can be realized in an elegant way by applying the ideas of transactional process management in an *Electronic Commerce Payment Coordinator*.

Based on the NetBill protocol guaranteeing both money atomicity and certified atomic delivery, payment processes can be enhanced to additionally provide distributed payment atomicity. To this end, and in contrast to the currently applied payment schemes, the payment has to be initiated by the customer. She has to invoke a payment process at the Payment Coordinator by specifying the payment information PI and all n bilaterally agreed Order Information (and thus also all different merchants) that have to be encompassed within one single payment transaction. Therefore, a tuple $(OI_c, M)_j$ with Order Information OI_c and Merchant Identifier M for each product j with $1 \leq j \leq n$ has to be sent to the Payment Coordinator. Within the payment process invoked, the necessary steps are taken to guarantee all three dimensions of atomicity. The Payment Coordinator first contacts all merchants involved and collects the merchant's views on the Order Information $(OI_m)_j$. Then, in order to determine the success of the payment transactions, $(OI_m)_j$ and $(OI_c)_j$ are compared for each product j. In the case of success, the Payment Coordinator collects all keys from all merchants participating in the transaction, checks the validity and the value of the E-cash token received and atomically delivers all keys to the customer while at the same time initiating the money transfer to the merchants and sends a confirmation C_j to all merchants. In case that $(OI_m)_j$ and $(OI_c)_j$ do not match for some j, some keys are not available, or the E-cash token is not correct, the Payment Coordinator aborts the payment transaction and no exchange will take place.

Electronic Commerce payment can benefit from a Payment Coordinator based on transactional process management ideas in several ways. Firstly, as application logic is centrally defined, the prerequisites for the participants of Electronic Commerce trade (customers, merchants, banks) are minimized. Secondly, with the inherent structure of payment processes invoked by a customer, it is possible to provide all dimensions of atomicity identified as requirements in Electronic Commerce. This is thirdly enhanced by additional properties as, for instance, the possibility to legally prove correct execution of a payment process by persistently logging process execution. This process log is part of the transactional process management and thus, provided by the Payment Coordinator in an elegant and straightforward way. Finally, by executing payment processes by a Payment Coordinator, the monitoring of the state of a payment interaction is facilitated compared with the distribution found in current payment protocols.

5 Conclusion

This paper provides a detailed analysis of requirements participants in Electronic Commerce payment impose with respect to atomicity issues. Different levels of atomicity can be identified which, however, are not simultaneously provided by existing approaches. Using the notion of processes, it has been shown that all payment interactions can be embedded into a single payment process where all possible levels of execution guarantees can be provided while at the same time the prerequisites of the participants are reduced. Finally, by applying the ideas of transactional process management, it has been shown how a Payment Coordinator supporting atomic and provable payment processes can be developed.

This process-based Payment Coordinator is currently being implemented within the WISE system [2]. Based on this implementation, we will in our future work extend the analysis of payment processes to further properties (such as, for instance, anonymity, scalability, or flexibility). Our goal is to decouple these properties, to identify the building blocks needed to realize them and to flexibly generate payment processes with user-defined properties by plugging together the building blocks needed.

References

1. G. Alonso, S. Blott, A. Feßler, and H.-J. Schek. Correctness and Parallelism in Composite Systems. In *Proceedings of the* 16^{th} *ACM Symposium on Principles of Database Systems (PODS'97)*, pages 197–208, Tucson, Arizona, May 1997. ACM Press.
2. G. Alonso, U. Fiedler, C. Hagen, A. Lazcano, H. Schuldt, and N. Weiler. WISE: Business to Business E-Commerce. In *Proceedings of the* 9^{th} *International Workshop on Research Issues in Data Engineering. Information Technology for Virtual Enterprises (RIDE-VE'99)*, pages 132–139, Sydney, Australia, March 1999. IEEE Computer Society.
3. J. Camp, M. Harkavy, D. Tygar, and B. Yee. Anonymous Atomic Transactions. In *Proceedings of the* 2^{nd} *Usenix Workshop on Electronic Commerce*, pages 123–133, November 1996.
4. B. Cox, D. Tygar, and M. Sirbu. NetBill Security and Transaction Protocol. In *Proceedings of the* 1^{st} *USENIX Workshop on Electronic Commerce*, pages 77–88, July 1995.
5. DigiCash, 1999. `http://www.digicash.com/`.
6. J. Gray and A. Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, 1993.
7. A. Hermanns and M. Sauter, editors. *Management–Handbuch Electronic Commerce*. Vahlen, 1998. In German.
8. J. Lyon, K. Evans, and J. Klein. Transaction Internet Protocol Version 3.0. Network Working Group, Request for Comments (RFC 2371), July 1998. `http://www.ietf.org/html.charters/tip-charter.html`.
9. MasterCard and Visa. *Secure Electronic Transaction Specification*. MasterCard and Visa, draft edition, June 1996. Book 1: Business Description, Book 2: Programmer's Guide, Book 3: Formal Protocol Specification (Slightly revised version of Book 3 appeared August 1, 1997).

10. S. Mehrotra, R. Rastogi, A. Silberschatz, and H. Korth. A Transaction Model for Multidatabase Systems. *Bulletin of the Technical Committee on Data Engineering*, 16(2), June 1993.
11. P. Muth, J. Weissenfels, and G. Weikum. What Workflow Technology can do for Electronic Commerce. In *Proceedings of the EURO-MED NET Conference*, Nicosia, Cyprus, March 1998.
12. B. Schmidt. *Elektronische Märkte – Merkmale, Organisation und Potentiale.* In: [7]. 1998. In German.
13. H. Schuldt, G. Alonso, and H.-J. Schek. Concurrency Control and Recovery in Transactional Process Management. In *Proceedings of the* 18^{th} *ACM Symposium on Principles of Database Systems (PODS'99)*, pages 316–326, Philadelphia, Pennsylvania, USA, May 31-June 2 1999. ACM Press.
14. H. Schuldt, H.-J. Schek, and G. Alonso. Transactional Coordination Agents for Composite Systems. In *Proceedings of the* 3^{rd} *International Database Engineering and Applications Symposium (IDEAS'99)*, pages 321–331, Montréal, Canada, August 1999. IEEE Computer Society.
15. D. Tygar. Atomicity in Electronic Commerce. In *Proceedings of the* 15^{th} *Annual ACM Symposium on Principles of Distributed Computing*, pages 8–26, May 1996.
16. D. Tygar. Atomicity versus Anonymity: Distributed Transactions for Electronic Commerce. In *Proceedings of the* 24^{th} *Conference on Very Large Databases (VLDB'98)*, New York, USA, August 1998.
17. A. Zhang, M. Nodine, B. Bhargava, and O. Bukhres. Ensuring Relaxed Atomicity for Flexible Transactions in Multidatabase Systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'94)*, pages 67–78, Minneapolis, Minnesota, May 1994. ACM Press.