In: Proceedings of the International Workshop on Technologies for E-Services (TES'00), Cairo, Egypt, Sept. 2000.

Generation and Verification of Heterogeneous Purchase Processes*

Andrei Popovici Heiko Schuldt Hans-Jörg Schek

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

Abstract

Complex purchase protocols allow several participants (merchants and clients) to be involved within a single purchase transaction. Moreover, they provide the possibility to merge different heterogeneous payments into one atomic transaction. Although such complex purchases are urgently required, they are, so far, neither provided by existing protocols nor by available products or research prototypes.

In this paper we show how to dynamically build such complex heterogeneous purchase processes in a reliable way, by modeling complex interactions between several merchants and clients as transactional processes. Apart from the distribution aspect, we focus on the heterogeneity of the electronic purchase, by allowing different existing protocols for payment or goods transfer to be merged incrementally, as the user performs a distributed purchase on the Internet. Before executing such a complex purchase, we provide a (technical) analysis of the resulting process which yields the properties of the compound transaction like atomicity, debit/credit character, or anonymity. We have implemented a *Purchase Coordinator* prototype which allows for an incremental analysis of compound purchase transactions and which supports the execution of the corresponding purchase processes.

1 Introduction

The proliferation of electronic commerce (E-Commerce) has led to a diversification of the interaction patterns between service providers and consumers. In most cases, however, this view is limited in that a one-to-one interaction between suppliers and providers of services is assumed. The purpose of this paper is to consider more complex patterns by allowing one consumer to deploy services from *several* providers within a single transaction. This is applied to a concrete scenario, the interaction between customers and merchants in electronic purchases.

1.1 Motivation

In business to customer (B2C) E-Commerce, the most structured phase —the electronic purchase— is characterized by various protocols which allow customers and merchants to exchange goods and money. The purpose of this paper is to consider complex interactions in

*Part of this work has been funded by the Swiss National Science Foundation under the project INVENT.

B2C E-Commerce by allowing one customer to purchase goods from *several* merchants within a single transaction albeit *different* payment protocols (e.g., DigiCash [Cha], SET [MV96], or NetBill [CTS95]) are used. The fact that different payment protocols are combined increases the complexity of such compound purchases. This affects the way the different steps of each single protocol have to be plugged together in order to maximize the properties of the compound purchase (for instance, with respect to atomicity in the transfer of money and/or goods) and, as a consequence, it also impacts the complexity of the analysis of such a compound purchase. The latter is a very important feature in practical applications since a customer will very likely not invoke a compound purchase whose characteristics are not known to him or her, even when all single components are well known.

However, all currently implemented commercial protocols lack support for integration. They have been defined for three party transactions (one customer, one merchant, and a bank/credit card company) but do themselves not consider the possibility to encompass several merchants within one transaction, thus they cannot be generalized. Furthermore, they impose high requirements on the participating instances (e.g., NetBill, SET, or MilliCent [Mil99]) in that special applications have to be installed for each of them. Considering the nature of Internet, the chance for a customer to use the same protocol when atomically purchasing goods from several merchants is minimal. Therefore, we consider in this paper the aspect of *heterogeneity* — the ability to simultaneously use several payment protocols in the same distributed purchase transaction.

Note that, although we focus here on the deployment in B2C E-Commerce, our approach is not limited to this kind of application. It can also be used for other business models, for instance to support supply chain management in business to business (B2B) E-Commerce where interactions with various suppliers have to be coordinated.

1.2 Heterogeneous Distributed Purchases: A Running Example

In what follows, we present a sample distributed purchase, illustrated in Figure 1, which we will refer to throughout the paper.

Consider the following scenario: a customer, Bob, wants to simultaneously purchase hardware and software within one single transaction. He first goes to the Migros Online Store and chooses a personal computer which is worth 1500.- CHF (purchase A). This is definitively no electronic good, so the PC will be shipped to Bob by a delivery company. The payment has macro-payment character — Bob has to use his credit card since Migros exploits a credit-card-based payment instrument (e.g., Cyber Cash). Next, Bob wants to add an application for Internet phone to his purchase. Since this is not available from Migros, he goes to the COOP Web Shop and chooses an appropriate application for 12,5 CHF (purchase B). Its delivery will be performed electronically; COOP accepts NetBill as payment instrument. Third, Bob still needs a calling card for the newly purchased phone. The best offer is provided again by Migros. So he returns to the Migros Online Store and purchases the calling card anonymously using E-cash (purchase C). Its shipment is performed via the web. Although different merchants are involved and different payment protocols used, Bob wants to combine all purchases in order to get all three goods or none of them (illustrated in Figure 1 by means of the metaphorical cloud encompassing all three independent purchases).

The combination of these purchases in the presence of heterogeneous protocols imposes various problems in different respects: since the protocols may differ in key properties such as atomicity, it is not immediately clear to Bob which property the compound purchase has.

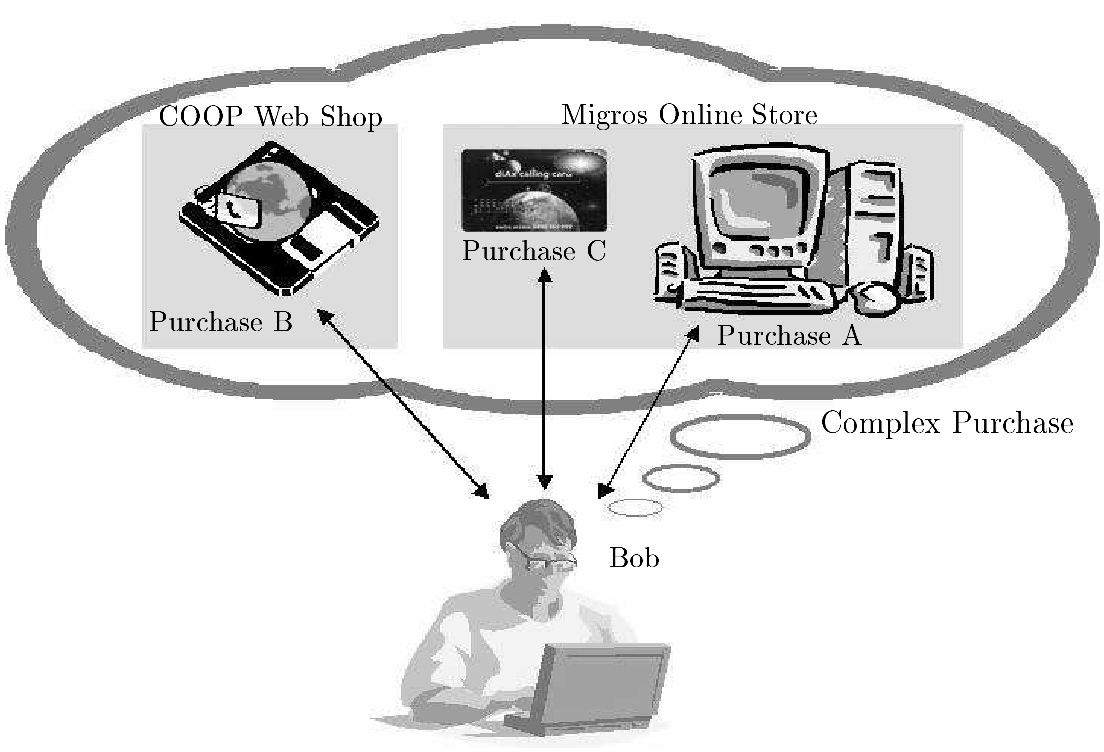

Figure 1: Sample Distributed Purchase

This is also true for other properties like anonymity (note that purchase C is performed anonymously while this is not the case for purchase A).

1.3 The Problem

Generating heterogeneous interactions renders the compound transaction difficult to understand. The generation of such a transaction has to take into account the particularities of each payment and goods transfer in part. Special care has to be taken to ensure atomicity, given the different implementations of payments. Anonymity of such a compound purchase transaction can be compromised if only one part of it reveals sensible user data to a participant. In general, before executing such a transaction, the user has to be able to analyze and influence its contents and execution.

This paper describes an approach for dynamic generation of complex and heterogeneous purchase transactions and their analysis. A main feature of our approach is to provide certain essential properties, such as atomicity, even in the presence of heterogeneity and distribution. It is based on a process management system acting as centralized purchase coordinator. The complex interactions of the various participants can be embedded within a single purchase process. A flexible number of participants is supported by generating process descriptions for each complex purchase to be coordinated. The necessary execution guarantees for these purchases –with respect to several dimensions of atomicity and concurrency control– are provided by exploiting transactional process management. In here, we rely on the framework established in [SAS99]; related approaches combining transactional guarantees and process executions can also be found in [CD96] or [WR92, RSS97].

The paper extends previous work on the generation of reliable payment processes [SPS00] in several ways: First, we address purchase interactions in a broader context in that we jointly consider payments and the associated transfer of goods. Second and most important, we deal with heterogeneity in all these aspects, i.e., we allow the combination of different payment and goods types (electronic and non-electronic) into one purchase process. Third, we have implemented a dedicated component of our purchase coordinator which allows for an analysis of the properties of a complex purchase process prior to its execution.

The paper is structured as follows: Section 2 summarizes related work. In Section 3, the most important features of an electronic purchase are outlined. These requirements serve as guidelines in the process of building complex purchase transactions. The purchases themselves are formalized as transactional processes. The generation of transactional processes implementing complex electronic transactions is explained in Section 4. The prototype we have implemented is described in Section 5; Section 6 concludes the paper.

2 Related Work

According to [MWW98], trade interactions between customers and merchants can be classified in three phases: pre-sales, sales and post-sales. In this paper, we will concentrate on the sales phase of an E-Commerce transaction. This phase is well defined [Pop99] and is actually the common denominator for every electronic trade scenario.

Considering electronic purchases, initial work by Doug Tygar provides a first and general overview of requirements of E-Commerce from a customers' perspective [Tyg96, CHTY96, Tyg98]. Following this classification, the electronic purchase is considered as consisting of a payment and a goods transfer phase. The commonly accepted scenario is that one merchant and one customer are exchanging goods and money. Efforts to improve the reliability of this scenario [Tan96], the security [Tan95], and the atomicity of distributed E-Commerce transactions [SPS99] have been done. A detailed discussion of the requirements imposed on B2C E-Commerce interactions both from a technical and from a legal perspective can be found in [Vei99].

Several agent-based approaches to E-Commerce purchases exist, e.g., [KB99, WYL+99]. However, all these approaches either lack support for distribution (which is in general present in the negotiation phase that takes place prior to the actual purchase but not with respect to the purchase itself [WYL+99]) or they do not provide appropriate support for transactional execution guarantees in distributed environments (the Payment Agents in the sense of [KB99], for instance, only address funds transfer but do not consider the combination of the latter with goods transfer). In the case of transactional execution guarantees, Papazoglou [Pap99] even proposes to exploit process-centric approaches (as we consider in this paper) since they exceed agent-based approaches in terms of support for long-lived distributed business transactions, i.e., distributed purchases.

3 Requirements and Properties of E-Commerce Purchase Protocols

Payment protocols can be mapped into a common reference protocol, which can then be exploited as classification instrument. The reference payment schema is an abstract model of payments facilitating the mapping of the interactions between a merchant and a customer in a more abstract framework. Based on this reference model, we were able to uniformly characterize existing payment protocols and extract several requirements classified in security aspects (anonymity, atomicity, provability and cryptography), technical aspects (scalability, efficiency, flexibility) and finally economical aspects like transaction costs and ease of deployment. We will extend the discussion of payment protocols to a broader context, the analysis of purchase protocols in which payments are embedded. In this section we will first briefly describe a simple purchase model that is restricted to the standard three party case, then broaden this view by extending and generalizing the simple model to an n party case (with one customer, one payment coordinator, and multiple merchants). Furthermore, we explain why heterogeneity, distribution and dynamics are of central importance. We will also present a formal notation for the common properties of a simple purchase protocol and provide mechanisms to derive the properties of a compound purchase given the basic properties of each part.

3.1 Model of a Purchase Protocol

A typical purchase interaction commonly involves a customer, a merchant, and a trusted instance (which, in turn, interacts with one or more banks). Most of the existing commercial protocols in which one customer purchases goods from one merchant are well approximated by the abstract model presented in Figure 2. In here, we consider such simple purchase interactions in which, however, multiple goods (encompassed within a shopping cart) can be involved. These simple purchases will then serve as the basis for complex, i.e., distributed and heterogeneous purchases.

In the first step (*order*) the client provides a description of the goods to be purchased, as well as some data representing a payment information (e.g., e-cash token, credit-card number, or account information). It also incorporates the initiation of the transaction, of the means to contact the participating instances (client, merchant, banks), etc.

In the second step (*money check*) the validity of the payment information specified in the order (existence of account, correctness of credit-card information, etc.) is checked by

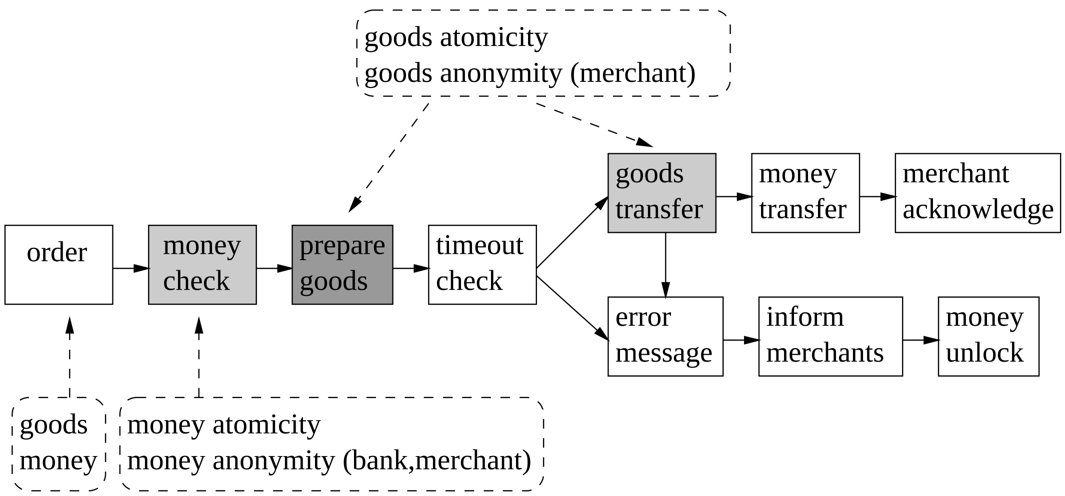

Figure 2: The Reference Purchase Protocol

the third party involved, i.e., a bank or a credit card company. If electronic cash token are used as means of payment, they will be "marked" after positive validation in order to prevent double-spending. In general, this verification is payment protocol-specific and may encompass other checks than just the verification of the data provided (if, for instance, payment is done by credit card, this step also includes the verification of constraints limiting the expense to a certain amount per day). After this step, the financial institution guarantees that subsequent money transfer can be enforced.

The third step (*prepare goods*) represents the preparatory phase for goods delivery. This step and its predecessor (money check and prepare goods) are related in that preparation is not executed until subsequent payment is guaranteed by the financial institution. However, money transfer is independent from goods transfer; the latter directly relates merchant and customer while money transfer requires the additional financial institution. If the goods to be delivered are electronic, they are usually sent to the customer in an encrypted form, already in this step. For non-electronic goods, a similar approach has to be taken by using electronic contracts instead of the real goods. It is important to mention that electronic contracts are legally binding for both participants, especially for merchants.

While establishing such a contract, a merchant can specify a timeout before which the customer has the right to cancel the purchase. The *timeout check* is the point-of-no-return for a purchase transaction. It is followed by two branches, reflecting either the success of the purchase or its abort. In both cases, subsequent actions have to be taken to ensure a correct termination of the purchase transaction.

If successful, the goods delivery has to be accomplished (*goods transfer*). Again, this step depends on the type of goods (electronic or real) to be delivered. The transfer of goods is closely related to the actions performed in the *prepare goods* phase where the details of this transfer have been arranged. In the case of electronic goods that have already been shipped in an encrypted way, this step corresponds to the transfer of the keys needed for decryption. Analogously, when encrypted electronic contracts have been transferred previously, the appropriate keys are delivered, and, in the case of non-electronic goods, shipment is initiated subsequently. Failures in the transfer of these keys can be immediately handled by executing the abort branch, without money transfer. In the case of business models requiring payment after successful delivery, shipment failures for non-electronic goods can be handled the same way. However, shipment failures require special treatment in business models allowing money transfer to be performed in parallel to the delivery of non-electronic goods. To this end, claims can be raised based on the electronic contracts (which are legally binding since the corresponding keys have been successfully transfered previously). While payment must not succeed the provision of all services that are part of a complex purchase, cases where, for instance, service contracts of long duration are dealt can be handled correctly, without unreasonably delaying the payment.

The purchase is financially settled after successful termination of the *goods transfer* step by accomplishing the *money transfer* (which is again payment-protocol dependent) and acknowledging the merchant about the success of the purchase. Note that this step is guaranteed to succeed due to the promises of the financial institution given in the *check money* step.

The abort branch is similar, and takes the necessary steps to release the money and goods locks (*inform merchants* and *money unlock*).

It is important to notice that the grey-shaded actions are the ones which depend on the payment protocol implementation or goods delivery type. Most of the existing purchases can be modeled in such a way that their specific parts are within the *money check*, *prepare* *goods*, or *goods transfer* actions. The simple reference protocol is characterized by the type of delivery (real or digital goods) and by the payment character (debit or credit). The most important properties are as follows.

3.2 Atomicity

One key requirement in E-Commerce is to guarantee atomic interactions between the various participants in purchases. As E–Commerce and thus also purchases take place in a highly distributed and heterogeneous environment, three independent aspects of atomicity have been identified: money atomicity, goods atomicity [Tyg96, Tyg98] and the atomic interaction of a customer with multiple merchants — called distributed atomicity [SPS99]. Although they are all called atomicity and are seemingly related, they are orthogonal concepts. Consider again Figure 2. Some steps like *money check*, or *prepare goods* may have inherent atomicity properties which are not related to each other. In the presence of heterogeneity — i.e., a non-atomic SET payment combined with an atomic NetBill one — a transaction may loose the money atomicity while still maintaining its goods-atomic character.

3.3 Anonymity

Anonymity is one of the most controversely discussed requirements for purchase processes. Without anonymity, the user's identity may be misused and sensible data becomes available to unauthorized third-parties. But anonymity itself may also be misused. Anonymity and provability (the ability to legally prove one's correct behavior) are conflicting. Although anonymity may be an important issue in certain applications, this is definitely not the case in all E-Commerce interactions. Yet, there are a lot of scenarios where anonymity cannot be achieved since the customers identity is closely linked to the goods to be purchased. Therefore, although we consider the analysis of the degree of atomicity that can be achieved in a complex purchase transaction, we are aware that this might not be required in all possible cases.

Unlike in other contributions which consider a purchase to be either anonymous or non-anonymous, we have a fine-grained definition of anonymity. This differentiation of anonymity considers the problem against which participant the customer's identity can be hidden (or cannot be hidden), leading to the following notions: bank-anonymous, merchant-anonymous, and non-anonymous. Payment information (e.g., credit card information) may disclose the identity of the user both to the bank and the merchant which reads the data, while a non-anonymous goods transfer discloses the information about the user's address only to merchants involved in the shipping. Similar to atomicity, the *money check*, *prepare goods*, or *goods transfer* actions have inherent anonymity properties.

3.4 Composition of Properties of a Purchase Protocol

The individual actions forming a simple purchase protocol define the properties of the different blocks in Figure 2. For each such action (e.g., *money check* or *goods transfer*) we can define specific properties which affect the overall characteristics of the purchase transaction. The most important ones, together with other general payment properties are summarized in Table 1. A simple payment may have debit or credit character. This fact is reflected in the first row of Table 1, which specifies that the Payment type is a singleton set consisting either of { debit } or { credit }. Analogously, the character of the payment (granularity) which is specified in the *order* block of a simple transaction (Figure 2) is also a singleton set. Note that

Property Name		Property Definition
Payment type	$\subset$	{ debit, credit }
Money atomicity	::=	{ atomic } ‖ { non-atomic }
Money anonymity	$\subset$	{ non-anonymous, merchant-anonymous, bank-anonymous }
Goods type	$\subseteq$	{ real, electronic }
Goods atomicity	::=	{ non-atomic } ‖ { goods-atomic }
Goods anonymity	$\subset$	{ non-anonymous, merchant-anonymous }
Timeout	::=	*real*
Latency	::=	*real*
Granularity	$\subset$	{ macro, micro, pico }

Table 1: Properties of a Simple Purchase Transaction

unlike payment type and granularity, the *Goods Type* for a simple purchase also allows {real, electronic} as a possible value. The intuition behind this is that current Internet merchants allow the building of shopping carts, which may consist of both electronic and real goods. The contents of a shopping cart, as a compound, define the goods type of a simple purchase.

Anonymity may be lost through goods transfer in the blocks *prepare goods* or *goods transfer* against the merchant. Both blocks therefore affect the *goods anonymity* property. Moreover, the payment information may disclose the identity of the user in the *money check* to the bank, too. Therefore, the *money anonymity* definition in Table 1 contains the three possible values for anonymity.

3.5 Distributed Purchase Protocols

The possibility of a purchase protocol to encompass several participants of the same type is referred to as the distribution aspect of an electronic purchase. The distribution aspect becomes hard to enforce if properties like heterogeneity of payment protocols and the atomicity of transactions are taken into account.

Two common scenarios motivate the need for distribution. In the first case, we consider the possibility of a customer to atomically purchase goods from different merchants, i.e., the sample purchase of Bob presented in Section 1.2 with the associated merchants Migros and COOP. If so, every merchant offers merchandise types separately, but the customer needs a combination of them, for instance to optimize costs or because of quality considerations.

A second scenario is that several customers aim to get a price reduction if purchasing a large number of similar items from a given merchant. In this case, distribution and atomicity also have to be harmonized.

From a technical point of view, distribution means to combine several individual purchases as the one described in Figure 2 into one whole while preserving the important features of each member-purchase in part (see also Sections 3.2 and 3.3).

3.6 Heterogeneous Purchases

The need for the support of heterogeneous purchases can be discussed only in the context of distribution. Consider again the sample purchase of Section 1.2. The probability for all merchants to use the same payment protocol (ECash, CyberCash, MilliCent, etc.) decreases with the number of merchants involved in the transaction. Moreover, we cannot rely on the assumption that all goods can be send over the Internet (such as programs or pieces of data), so we have to cope with different means of goods delivery, too.

For a distributed purchase transaction it is therefore important to encompass several payment protocols or goods transfer modes. A source of heterogeneity may also arise when two payment instruments with the same character –e.g., credit, like SET or CyberCash– have different implementations. The duration of the individual purchases may vary, because of the goods transfer or because debit and credit payment usually do not have a common payment time. Timeouts for different purchases may also differ. The degree of anonymity of the involved protocols is predefined and is independent of the complex transaction.

4 Dynamic Generation and Analysis of Transactional Purchase Processes

As shown in the previous section, purchases have a well defined structure. This fact will be exploited since we implement purchases as processes which encompass the flow of control and data between the different steps. Using processes, we can even embed several purchase transactions within one purchase process where the process description is generated dynamically. In this section, we will concentrate on the algorithms and steps to be performed when creating purchase processes involving different payment protocols and various goods types.

4.1 Purchases as Transactional Processes

Due to the well-defined structure and the repeated execution patterns, transactional processes [SAS99] are a highly appropriate means to implement the interactions which have to be performed for purchase purposes. To this end, each step of a purchase process is mapped to an activity that is performed at one of the participant's sites. Transactional processes not only encompass flow of control and flow of data between different activities (representing the building blocks of a purchase transaction) but include also very sophisticated failure handling strategies by alternative executions.

An important feature of purchase interactions is the provision of execution guarantees for the corresponding purchase processes. These execution guarantees are twofold.

First, they include the possibility to check whether a single process is correctly defined. This validation considers both the alternative executions that are part of the process model and the different termination properties of single activities. In a generalization of traditional transaction management, transactional processes consider activities which can be either compensatable (**c**) or non-compensatable, called pivot (**p**). Such pivot activities are allowed to be committed in the middle of a process execution given that alternative executions are available, consisting only of retriable activities (**r**), i.e., such activities that are guaranteed to succeed, following the ideas of the flex transaction model [MRSK92, ZNBB94]. Processes having this structure are called *processes with guaranteed termination*, which provide a more general notion of atomicity than the traditional "all-or-nothing" one.

Second, starting with the correct specification of single processes having guaranteed termination property, the purchase process coordinator's task is to enforce the correct execution of transactional processes with respect to concurrency control and recovery. The correct execution of these transactional purchase processes then encompasses some of the previously identified requirements imposed on purchase interactions such as the atomic exchange of money and goods.

Here, it has to be noted that both aspects, the correctness of single processes and the enforcement of correct executions with respect to concurrency control and recovery, are essential for this kind of application. Hence, these constraints limit the applicability of commercial workflow management systems since most of these systems do not provide the required execution guarantees in a satisfactory way.

In addition, this process-based approach facilitates the dynamic generation of compound purchase processes and especially the correctness validation of these processes which would be much more complex in distributed, i.e., agent-based, approaches.

Aside of the termination properties of activities, we can enrich the specification by additionally considering the E-Commerce specific properties described in Section 3.4. Because of the inherent modularity of a purchase description, merging of transactional processes is facilitated.

4.2 Composition of Purchase Protocols

We merge several of the simple purchase transactions previously described into a more complex one. Consider again the initial example presented in Section 1.2 in which Bob purchases a personal computer, an application for Internet phone, and a calling card from the Migros Online Store and the COOP Web Shop, respectively, within one single compound transaction while using different payment protocols for each single purchase.

Figure 3 depicts a template for the resulting purchase process. We use circles for those parts which are frozen and are not payment type specific or goods transfer specific. Every circle denotes an action to be executed by one of the participants. The rectangular boxes are to be filled in with concrete actions. Every rectangle corresponds to an activity (or to a set of activities) of the purchase process. All dark grey shaded rectangles belong to purchase A of the initial example, the other ones to B and C, respectively. When the compound purchase is built, the generic purchase process has to be filled with concrete transactional process activities for each of the boxes. Then, when instantiating the so-generated process with parameters provided at run-time, the characteristics determined during the initial negotiation phase will be enforced.

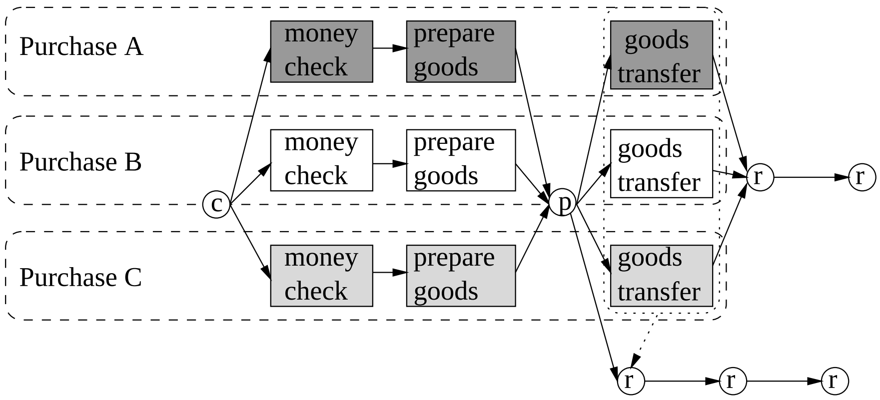

Figure 3: Merging Several Reference Purchase Transactions into a Compound Transaction

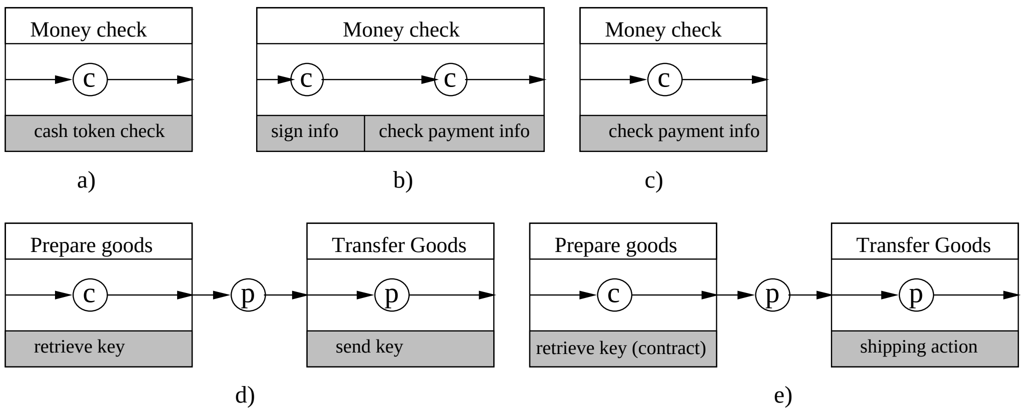

Figure 4: The Building Blocks to be used in the Compound Transaction.

Figure 4 depicts some concrete realization for the grey-shaded goods transfer step of Figure 3. The first row contains payment blocks for account-based payments (a), for credit-card payments (b) and for e-cash tokens (c). The second row contains blocks for the preparation and the shipping of electronic (d) and real goods (e). Case (d), for instance, contains the implementation of the atomic goods transfer model proposed by Tygar, where a merchant sends encrypted electronic goods to a customer, and then sends the keys to a trusted log [CHTY96]. In our case, the (trusted) purchase coordinator plays the role of the trusted log by collecting keys from the merchant and distributing them after verification of the transaction fairness. The same is true for the digital contracts in the case of non-electronic goods.

For every goods transfer type and for each payment protocol, such building blocks can be supplied. In our prototype we created building blocks for electronic and real goods as well as for three common payment mechanisms (SET, NetBill, an account-based payments). The individual activities shown in Figure 4 form together a payment and shipment library from which single activities can be taken and then plugged together dynamically into the template of a compound purchase process like the one shown in Figure 3. When this process-based approach to complex purchases is applied to other scenarios, i.e., to B2B supply chain processes, then, aside of the basic model of the interaction pattern (similar to the model of a simple purchase protocol presented in Section 3.1), only appropriate building blocks have to be identified and to be made available.

4.3 Analyzing a Compound Purchase Protocol

In the previous section we have shown how to model several purchases within one compound process, which will have to be executed atomically. Due to the use of building blocks for payments and goods transfer, we are able to generate heterogeneous purchase protocols.

Such protocols induce an increase of the complexity of the purchase. For simple cases (such as purchase of electronic goods using e-cash tokens) most of the commonly accepted requirements (atomicity, anonymity) can be fulfilled [CHTY96, Tyg96]. For more complex ones it is sometimes impossible to have all of them fulfilled simultaneously. Prior to generating the process template for a compound process, we have to analyze its behavior with respect to atomicity, anonymity, and other important aspects described in Section 3. Since every building block is tagged using the properties described in Table 1, we can compute the overall character of the payment. Hence, aside of the implementation of the corresponding step, each building blocks requires its characterizations with respect to the properties identified in Section 3.4. For the following example, we consider n as being the number of parallel blocks in a compound purchase (CP) which influence a given property. Thus, we aim in analyzing the combination of n independent purchases.

Goods Type The goods type of the compound purchase is the union of the individual goods transfers. Table 1 contains the possible values for $GoodsType_i$

$$GoodsType_{CP} = \bigcup_{i=1}^{n} GoodsType_i$$

Payment Type The compound money or payment mode is the union of individual payment modes. For the compound payment to have, e.g., only credit-character, all payment protocols used for purchases must be credit-based. $PaymentType_i$ denotes the payment character of the i-th individual purchase, the possible values being defined in Table 1.

$$PaymentType_{CP} = \bigcup_{i=1}^{n} PaymentType_i$$

Payment Character As with the other payment information, when combining several payments we cannot rely on a uniform macro, micro, or pico-payment. As with the other purchase properties, the possible values may be found in Table 1.

$$Granularity_{CP} = \bigcup_{i=1}^{n} Granularity_i$$

Anonymity The most natural way to compute the anonymity character of a purchase transaction would be to take the minimum anonymity level (i.e. non-anonymous interactions) of all member purchases. However, this does not reflect the fact that loosing anonymity against one merchant does not corrupt the anonymity status against other participating instances (banks and other merchants). A pessimistic customer may compute the resulting anonymity $Anonymity_{CP}$ according to the following rule:

$$Anonymity_{CP} = \begin{cases} \text{non-anonymous} & : \text{if non-anonymous} \in \\ & \quad \bigcup_{i=1}^{n} (GoodsAnonymity_i \\ & \quad \cup\ MoneyAnonymity_i) \\ \bigcap_{i=1}^{n} (GoodsAnonymity_i & : \text{otherwise} \\ \quad \cap\ MoneyAnonymity_i) & \end{cases}$$

In contrast, if the customer trusts the merchant with the minimal level of anonymity (non-anonymous), the subjective degree of anonymity may even be higher. In order to allow user preferences (like subjective perception of anonymity) we permit the overriding of individual characteristics by user preferences in our prototype (Section 5).

Latency and Timeout Every purchase has a timeout check, after which a transaction abort is initiated. Since even the slowest purchase process must have a chance to complete, the overall latency is computed as being the maximum of waiting times for the individual processes.

$$Latency_{CP} = \max_{i=1}^{n} Latency_i$$

$$Timeout_{CP} = \min_{i=1}^{n} Timeout_i$$

The latency of a protocol corresponds to the delay induced by a merchant. For a transaction to succeed, it must be ensured that the maximal latency is smaller than the minimal timeout of the individual transactions.

4.4 Exploiting the Analysis Results

This approach allows the customer a way to derive the properties of his/her compound purchase process prior to its execution, i.e. its instantiation.

Therefore, modifications are possible, e.g. by exchanging purchases or by removing single purchases in order to increase certain properties of the compound purchase.

5 The INVENT Purchase Coordinator

A purchase model (Figure 2) can be implemented either in a distributed manner (each participating instance containing one part of the application logic) or using a centralized component. A coordinator-based architecture, relying on a process support system, offers many advantages over the distributed architecture, especially in terms of execution guarantees for the purchase processes to be executed.

The trusted status of the coordinator is often considered as being the major drawback of this approach. However, this prerequisite is far from being unrealistic since all current commercial systems implicitly contain trusted instances (such as verificator instances in SET [VM]). In the Internet age, this does not impose unsolvable restrictions since such trusted institutions are already established, for instance in the form of certification authorities or clearing houses. Such coordinators then offer added-value services, namely the coordinated execution of several purchases within one compound transaction. Hence, their incentive is similar to that of other service providers or information brokers in that they request a share in the form of a predefined percentage of the total amount of each purchase transaction they coordinate.

5.1 Coordinated Purchase Processes

A *Purchase Coordinator* controlling the execution of purchase processes has been implemented within the INVENT project. The prototype extends basic functionality of a transactional process support system in that it additionally supports the special requirements of purchase processes, i.e., the functionality needed for the generation of heterogeneous purchase processes using the aforementioned techniques. The E-Commerce scheme supported consists of two phases: The first part is Internet-based and coincides with the negotiation phase while

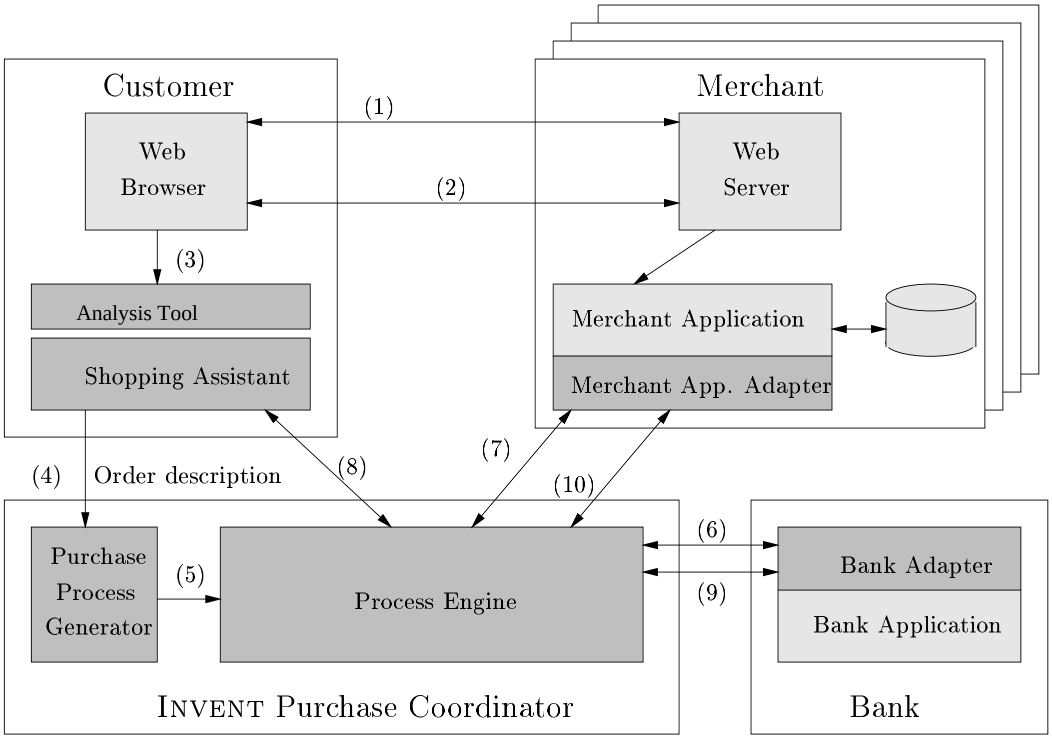

Figure 5: Overview of the Purchase Coordinator Architecture

the second part is the actual payment controlled by the Purchase Coordinator. The main components of the whole purchase coordinator together with the interactions between the various participants are depicted in Figure 5 (the parts belonging to the Purchase Coordinator are colored in dark grey).

The customer has to use a web browser to choose the items from merchants which are offering their services or goods through their electronic store fronts (step 1 in Figure 5). After the negotiation step, the customer can download electronic goods or e-contracts for the real goods from the associated merchants (steps 2). The merchant uses any arbitrary application (*Merchant Application*) to manage the data related to the electronic goods needed or real goods (e.g., encryption keys and shipping certificates). A realistic assumption is that this data is stored in a database on the merchant's site. After the customer has received the goods related data, he/she can decide which ones to encompass into one E-Commerce transaction (step 3). To this end, the customer uses the *Analysis Tool* [Bra00] in order to analyze the current properties of the transaction which would be generated out of the current negotiation state. The Analysis Tool uses the analysis techniques presented in Section 4.3 to verify the degree of anonymity and atomicity as well as the expected latency of the compound payment and of other purchase properties presented in Section 3.4. It has to be noted that this analysis is performed automatically, without user intervention, based on the characterization of the diverse steps of all participating simple purchases (which is part of the negotiation phase). But, it serves as additional information to a customer on which he or she may base the decision to actually initiate a complex purchase transaction.

The Analysis Tool is connected to a *Shopping Assistant* [Bac99], both being plugged into the customers web browser. They collect information about all interactions the customer performs with different merchants during the negotiation phase and use a multi-wallet to manage different types of payment information (e.g., credit, debit, or account information).

When all interactions are combined within the Shopping Assistant, the negotiation phase is accomplished. The following steps describe the purchase transaction (step 4). The Shopping Assistant invokes a purchase process at the Purchase Coordinator by specifying the customer's view of the order information. The order information describes the type of each purchase protocol to be used, a merchant identifier, the goods transfer type of each delivery and all other properties which characterize each individual purchase.

In order to support processes with arbitrary numbers of merchants involved and heterogeneous character, a process description has to be *generated* dynamically according to the information gathered by the customer during the negotiation phase (Order Information). To this end, a *Purchase Process Generator* has been implemented which follows the techniques for the combination of purchases discussed in Section 4.2. By adding this component to the process engine, a customer is able to run flexible and heterogeneous purchase processes, whose structure is derived from the initial web based negotiation phase (steps 1 - 3). This dynamically generated process description is instantiated (step 5) and enacted by the process engine. The concrete actions retrieved from the purchase library of activities —corresponding to the specific payment or goods transfer actions— are now inserted in the process description.

Within the purchase process invoked, the bank application is accessed via its associated adapter (steps 6 and 9). Similarly, invocations of services within the merchant applications (e.g., for a comparison of the order information) are performed via the adapters that have to be provided for these applications (steps 7 and 10). The adapters which perform the integration of all external components are implemented by so-called *Transactional Coordination Agents (TCAs)* [SSA99] acting as advanced wrappers by allowing the invocation of services in the merchant and bank applications. Although some parts of these TCAs are of generic nature, they have to be tailored to the interfaces and APIs of the corresponding applications they are plugged to. Hence, once a new participant joins the purchase coordinator architecture, an appropriate TCA has to be provided and installed locally (similar to the software that has to be installed for each payment instrument that is to be used) such that it can be used for all subsequent purchase processes coordinated by the INVENT system.

5.2 Analysis in the Prototype

The implementation of the analysis tool in the INVENT purchase coordinator provides for a considerable degree of flexibility. As mentioned, the analysis is performed prior to sending the order information to the purchase template generator. The results of the analysis are expressed in a pleasant, understandable way. Consider again the initial example, in which the three types of goods are purchased from two online merchants. Figure 6 shows the properties of the compound purchase, as they appear in the Analysis Tool embedded in the client's browser.

The purchase analysis dialog informs the user, e.g., that the purchase protocols used are heterogeneous, thus no common point for the termination of the transaction can be specified. The analysis of the overall degree of anonymity is also performed. The screen-shot is a good example of heterogeneous purchases: no property (anonymity, debit/credit character) is to be found in a pure form such that one has to give up some of the requirements.

Notably, the analysis is performed *incrementally*, while the user is selecting goods from different web-stores. Before triggering the generation of the purchase process (step 4), the

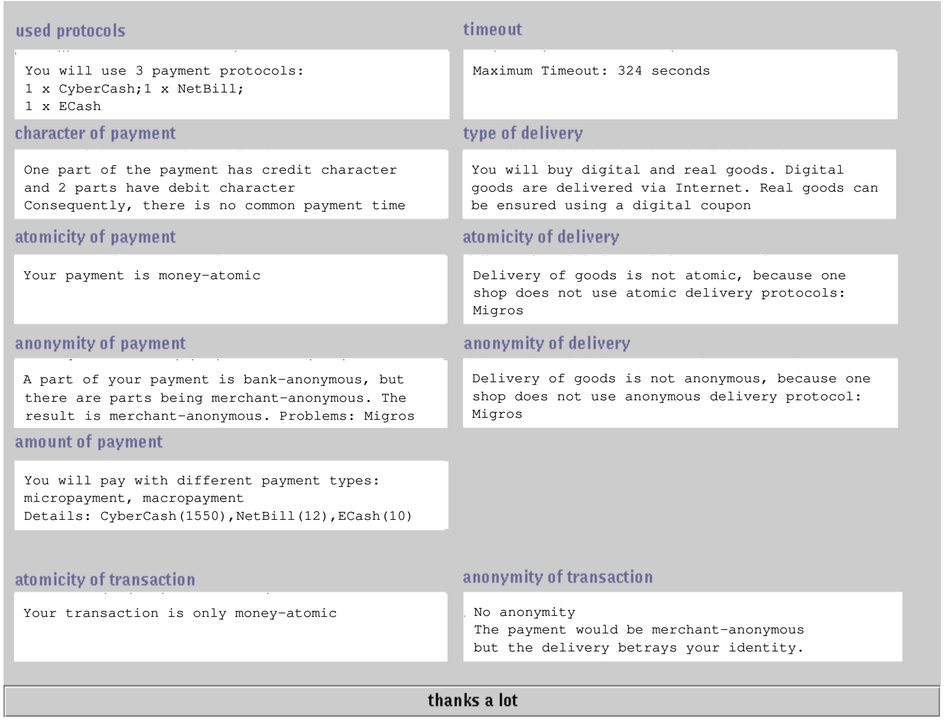

Figure 6: A Screen–Shot of the Analysis Tool

user may still remove one or several of the individual purchases from the compound in order to improve one specific property of the compound transaction. In the example above, if we would remove the purchase from Migros (Purchase A) before sending the purchase description to the purchase template generator, we could achieve goods atomicity and merchant anonymity for the resulting compound process then containing two individual purchases. It is also possible to override the degree of anonymity with a personalized value. Above, misuse of personal data by the Migros Online Store store data may happen. However, it may happen that the customer trusts Migros for personal reasons (a good history, personal contacts, etc). In this case, the customer may override the degree of anonymity derived from a specific purchase protocol with a personal value. As a result, the purchase may be considered anonymous because the customer trusts in a specific merchant that he will not disclose or misuse personal information.

6 Conclusions

The necessity to merge several purchase protocols into one atomic transaction is very important, along with the diversification of business models in the E-Commerce area. Heterogeneity arises not only because of the several payment types, but also due to the goods transfer, atomicity and anonymity requirements, or duration and timeout specification of each purchase. Faced with this complex purchase transaction, a user can no longer rely on the *pure* properties of each protocol in part. He/she therefore has to have the ability to analyze the description of the compound purchase and eventually add or remove certain parts according to personal preferences prior to the execution of a compound purchase in which the different properties are enforced.

In this paper we have shown how compound purchase processes can be generated. Moreover, we have presented a prototype system, the INVENT purchase coordinator, which executes these processes and enforces their correctness. An *Analysis Tool* is interposed between the user and the component in charge with generating process descriptions (*Purchase Process Generator*), thus allowing the interactive control of the contents of the compound transaction. The analysis of the transaction is user-friendly and user preferences can be specified on several levels.

Based on this prototype, we will in our future work extend the framework established and make process-based distributed purchase coordination available in B2B processes. The goal is to plug the purchase coordinator as a distinguished component to such processes (e.g., in the context of Virtual Enterprise processes) in oder to provide a fair and reliable per-service charge of the different services consumed.

References

[Bac99] D. Bacher. Distributed Internet Transactions and their Application to Atomic Payment Processes in Electronic Commerce. Diploma thesis, Database Research Group, Institute of Information Systems, ETH Zürich, 1999. In German.

[Bra00] P. Brantschen. Generation of Flexible, Distributed and Heterogenous Payments and their Correctness Validation. Diploma thesis, Database Research Group, Institute of Information Systems, ETH Zürich, February 2000. In German.

[CD96] Q. Chen and U. Dayal. A Transactional Nested Process Management System. In *Proceedings of the 12^{th} Int. Conference on Data Engineering (ICDE'96)*, pages 566–573, 1996.

[Cha] D. Chaum. Digicash/Solutions for Security and Privacy. http://www.digicash.com/.

[CHTY96] J. Camp, M. Harkavy, D. Tygar, and B. Yee. Anonymous Atomic Transactions. In *Proc. of the 2^{nd} Usenix Workshop on Electronic Commerce*, pages 123–133, November 1996.

[CTS95] 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.

[Elm92] A. Elmagarmid, editor. *Database Transaction Models for Advanced Applications.* Morgan Kaufmann, 1992.

[JK97] S. Jajodia and L. Kerschberg, editors. *Advanced Transaction Models and Architectures.* Kluwer Academic Publishers, 1997.

[KB99] L. Kerschberg and S. Banerjee. An Agency-Based Framework for Electronic Business. In *Proceedings of the 3^{rd} International Workshop on Cooperative Information Agents (CIA'99)*, pages 265–290, Stockholm, Sweden, July 1999. Springer LNCS Vol. 1652.

[Mil99] MilliCent, 1999. `http://www.millicent.digital.com`.

[MRSK92] S. Mehrotra, R. Rastogi, A. Silberschatz, and H. Korth. A Transaction Model for Multidatabase Systems. In *Proceedings of the 12^{th} International Conference on Distributed Computing Systems (ICDCS'92)*, pages 56–63, Yokohama, Japan, June 1992.

[MV96] MasterCard and Visa. *Secure Electronic Transaction (SET) 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).

[MWW98] P. Muth, J. Weissenfels, and G. Weikum. What Workflow Technology can do for Electronic Commerce. In *Proceedings of the EURO-MED NET Conference*, 1998.

[Pap99] M. Papazoglou. The Role of Agent Technology in Business to Business Electronic Commerce. In *Proceedings of the 3^{rd} Int. Workshop on Cooperative Information Agents (CIA'99)*, pages 245–264, Stockholm, Sweden, July 1999. Springer LNCS Vol. 1652.

[Pop99] S. Popp. Overview of Electronic Payment Methods. *Informatik – Informatique. Journal of the Swiss Computer Science Society*, April 1999.

[RSS97] A. Reuter, K. Schneider, and F. Schwenkreis. *ConTracts Revisited*, chapter 5, pages 127–151. In: [JK97]. Kluwer Academic Publishers, 1997.

[SAS99] H. Schuldt, G. Alonso, and H.-J. Schek. Concurrency Control and Recovery in Transactional Process Management. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS'99)*, Philadelphia, Pennsylvania, USA, May 31-June 2 1999.

[SPS99] H. Schuldt, A. Popovici, and H.-J. Schek. Execution Guarantees in Electronic Commerce Payments. 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, Dagstuhl Castle, Germany, September 1999. Springer LNCS, Vol. 1773.

[SPS00] H. Schuldt, A. Popovici, and H.-J. Schek. Automatic Generation of Reliable E-Commerce Payment Processes. In *Proceedings of the 1^{st} International Conference on Web Information Systems Engineering (WISE'00)*, Hong Kong, China, June 2000. IEEE Computer Society Press.

[SSA99] H. Schuldt, H.-J. Schek, and G. Alonso. Transactional Coordination Agents for Composite Systems. In *Proceedings of the International Database Engineering and Applications Symposium (IDEAS'99)*, Montreal, Canada, August 1999.

[Tan95] L. Tang. A Set of Protocols for Micropayments in Distributed Systems. In *Proceedings of the first USENIX Workshop of Electronic Commerce*, pages 107–116, New York, USA, July 1995.

[Tan96] L. Tang. Verifiable Transaction Atomicity for Electronic Payment Protocols. In *Proceedings of the 16th International Conference on Distributed Computing Systems (ICDCS'96)*, pages 261–269, Hong Kong, May 1996. IEEE.

[Tyg96] 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.

[Tyg98] 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.

[Vei99] J. Veijalainen. Transactions in Mobile Electronic Commerce. 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 203–224, Dagstuhl Castle, Germany, September 1999. Springer LNCS, Vol. 1773.

[VM] Visa and Mastercard. Secure Electronic Transaction (SET) Specification Book1. http://www.mastercard.com/shoponline/set/.

[WR92] H. Wächter and A. Reuter. *The ConTract Model*, chapter 7, pages 219–263. In: [Elm92]. Morgan Kaufmann Publishers, 1992.

[WYL+99] X. Wang, X. Yi, K. Lam, C. Zhang, and E. Okamoto. Secure Agent-Mediated Auctionlike Negotiation Protocol for Internet Retail Commerce. In *Proceedings of the 3^{rd} International Workshop on Cooperative Information Agents (CIA'99)*, pages 291–302, Stockholm, Sweden, July 1999. Springer LNCS Vol. 1652.

[ZNBB94] A. Zhang, M. Nodine, B. Bhargava, and O. Bukhres. Ensuring Relaxed Atomicity for Flexible Transactions in Multidatabase Systems. In *Proceedings of the ACM SIGMOD Conference*, pages 67–78, 1994.