COMBINATORIAL BUSINESS TRANSACTIONS
A Way for Increasing the Flexibility and Reliability of Electronic Marketplaces

Juha Puustjärvi
Lappeenranta University of Technology, Skinnarilankatu 34, Lappeenranta, Finland

Keywords: e-Business, B2B, Web services, advanced transaction models.

Abstract: Electronic marketplaces are virtual places that reside somewhere in the Internet. Through electronic market places buyers and sellers are able to interact with each other inside an architecture that is easy to use and maintain. Recent research on electronic marketplaces and electronic auctions has focused on new formats such as on negotiation based business transactions, combinatorial business transactions and multi-attribute business transactions. These new formats of business transactions set new requirements on the software that runs the business transactions in the marketplace. Unfortunately due to the lack of appropriate transaction models and software that support them these new forms of business transactions cannot be processed in a transactional way. This is regrettable since only by guaranteeing transactional properties these business transactions can be processed in a reliable and correct way. In this article we introduce a new transaction model, called Combinatorial business transaction, which is tailored for supporting a variety of new formats of business transactions processed in electronic marketplaces. Within Combinatorial business transactions failure atomicity is achieved through splitting the Combinatorial transaction into two successive transactions, called reservation transaction and certification transaction, and execution atomicity is enforced through semantic locks.

1 INTRODUCTION

In electronic business buyers and sellers should be able to interact with each others inside an architecture that is easy to use and maintain (Wurman, 2005). Electronic market-places are an interesting approach to achieve this goal by bringing together business in the web. Further the greatest benefits lie in cost reduction for buyers and sellers.

Technically, an electronic marketplace is a virtual place that resides somewhere in the Internet. They can provide several types of business processes depending upon their target audience. Recent research on electronic marketplaces and auctions has focused on new formats such as on negotiation based business transactions, combinatorial business transactions and multi-attribute business transactions. In a negotiation based business transaction dealing is made by first requesting the offer(s) and then making the deal. In combinatorial auctions bidders are allowed to place offers on sets on items whereas in multi-attribute auction price is not the only negotiable parameter.

The new formats of business transactions set new requirements on the software that runs the business transactions in the marketplace. Unfortunately due to the lack of appropriate transaction models and software that support them these new forms of business transactions cannot be processed in a transactional way. This is regrettable since only by guaranteeing transactional properties these business transactions can be processed in a correct and reliable way. By the transactional properties (Bernstein et al., 1987) we (as well as many other articles of transaction processing) refer to failure atomicity and execution atomicity. Failure atomicity means that all or none of the tasks of the transaction are executed while execution atomicity means that concurrent transaction instances will not interfere with each others (Puustjärvi, 2001).

Traditional transactions (usually called ACID-transactions) (Lynch, 1983) are supported by database management systems. This transaction model has evolved over time to in-corporate more complex transaction structures and to selectively relax the failure atomicity and execution atomicity (isolation) properties (Puustjärvi, 2006). The failure

Puustjärvi J. (2008).
COMBINATORIAL BUSINESS TRANSACTIONS - A Way for Increasing the Flexibility and Reliability of Electronic Marketplaces.
In *Proceedings of the Fourth International Conference on Web Information Systems and Technologies*, pages 477-482
DOI: 10.5220/0001513504770482

atomicity property is usually enforced by some form of compensation (Garcia-Molina, 1983).

A problem with compensation is that other transactions may access dirty (i.e., uncommitted data). Accessing dirty data is acceptable in many applications (Puustjärvi, 2004). However, accessing dirty data in electronic marketplaces may cause serious consequences (e.g., selling the same products twice), and therefore advanced transactions models based on compensating transactions are not suitable for electronic marketplaces

In this article we introduce a new transaction model, called Combinatorial business transaction, which is tailored for supporting (in a transactional way) a variety of new formats of business transactions processed in electronic marketplaces. In particular, Combinatorial business transactions can be used for three purposes:

- achieving the atomicity of the purchases of many products,
- providing alternative products for the products that are not available, and
- for supporting competitive offers.

Within Combinatorial business transactions failure atomicity is achieved through splitting the Combinatorial transaction into two successive transactions, called reservation transaction and certification transaction, and execution atomicity (i.e., concurrency control) is enforced through semantic locks.

The rest of the paper is organized as follows. First, in Section 2, we present the operational model of the marketplace that we are developing. Then, in Section 3 we motivate the use of Combinatorial business transactions by a travel planning example. In Section 4, we focus on the transactional properties of the Combinatorial business transaction model. First, in Section 4.1, we present how the execution atomicity can be achieved by using semantic locks, and in Section 4.2, we present the protocol which ensures the failure atomicity of Combinatorial business transactions. Finally, Section 5 concludes the paper by discussing the advantages and limitations of the Combinatorial business transactions.

2 THE OPERATIONAL MODEL OF THE MARKETPLACE

Our model of electronic marketplace has three types of users: sellers, buyers and the system administrator (Figure 1). Sellers and buyers communicate with the market-place through Web service interfaces (Newcomer, 2002; Daconta et al., 2003). The function of the Buyer application and Seller application is to carry out the protocols needed to support Combinatorial transactions (discussed in Section 4). In addition, they provide user friendly interfaces foe accessing the marketplace.

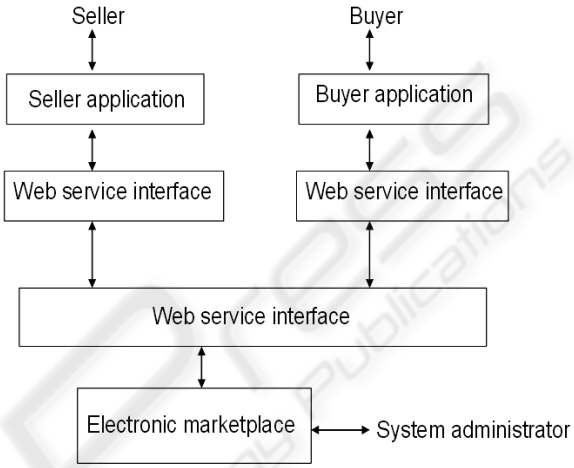

Figure 1: The architecture of the system.

The seller sets products for sale by sending *for-sale-message* to the marketplace's Web service interface. This message includes the following information:

- seller identification (denoted by SI),
- the product identifier (denoted by PI),
- the number of units to be sold (denoted by NU), and
- the price of the product (denoted by PR), which is a function of the number of products.

The price of the product may be static or dynamic and it may be dependent on the number of products to be sold and it may also be buyer specific.

The buyer purchases products by sending the purchase-request-message to the marketplace's Web service interface. The message includes:

- buyer identification (denoted by BI), and
- shopping list (denoted by SL)

Shopping list is a list of products and the number attached to each product. Formally a shopping list of k products is of the form:

{(n1, p1), (n2, p2), …., (nk, pk)}

The marketplace also has a database, which includes information about sellers, buyers and products. This information is collected and stored to marketplace's database when buyers and sellers have made their registration to the marketplace. Further, product information is classified according to the *product taxonomy*, which is maintained by the system administrator.

After receiving the *purchase-request-message* the marketplace analyses whether the requested products are available and after analyzing the message the marketplace sends the *availability-message* to the buyer. This message indicates which of the items of sellers shopping list are available and the prices of the available products. If a product of the shopping list is not available the number attached to the product is zero.

After the buyer has received the *availability-message* and analyzed the message, it sends the *certification-message* to the marketplace. In this message, the buyer certificates which and how many of the available products (none, some or all) it will buy. However, decreasing the number of products requires that the price given in the *availability-message* concerns one unit.

3 THE UTILIZATION OF COMBINATORIAL BUSINESS TRANSACTIONS

Basically there are four ways how the features of the Combinatorial business transactions can be utilized: to ensure the atomicity of a purchase, changing the shopping list, to provide replacing products, and to support competitive offers. We will illustrate the differences between these cases by a travel planning example. In this example the products that the buyer purchases are flight reservations and hotel reservations. For simplicity we assume that there are only two airlines, say KLM and SAS and two hotels, say Sheraton and Hilton.

- **Achieving Atomic Purchases:** the buyer sends the *purchase-request-message* concerning a specific two-way KLM-flight and a room reservation in Hilton. The buyer certifies the reservations (flight and room) only if they both are available.
- **Changing the Shopping List:** the buyer sends the *purchase-request-message* concerning ten rooms for a group of ten players, but due to the high prices the buyer decides to place two players for each room, and thus only certifies five rooms.
- **Providing Alternative Products:** the buyer sends the *purchase-request-message* concerning a KLM flight and a SAS flight and hotel reservation to Hilton and to Sheraton. Both flights are on the same date an on the same lane, and also the hotel reservation are alternatives for the buyer. The marketplace informs (through the *availability-message*) that all but the Hilton reservation is available. Then the buyer prefers and confirms the KLM flight (because the buyer has a bonus card on KLM but not on SAS) and the Sheraton reservation.
- **Supporting Competitive Offers:** Buyers *purchase-request-message* concerns both airlines and both hotels and the buyer s does not prefer any of the airlines or hotels. Then the marketplace informs that flights are available for both airlines and also rooms are available in both hotels. Then the buyer chooses a flight and a hotel based on a choice criterion (e.g., chooses the cheapest one).

The above illustrative examples belong to B2C and are taken for illustrative purposes. However, it is turned out that the most useful application areas of Combinatorial business transactions are in B2B.

4 MANAGING COMBINATORIAL BUSINESS TRANSACTIONS

A natural and sufficient criterion for transactions correctness is to support ACID-transactions (Bernstein, 1987). They have the following properties:

- *Atomicity* (failure atomicity) means that either all of a transaction be executed or none of it is.
- *Consistency* means that the state of a database satisfies the consistency constraints of the database.
- *Isolation* (execution atomicity) means that concurrently executed transactions do not interfere with each others.
- *Durability* means that if a transaction has completed its work, then its effect should not get lost even if the system fails.

Though ACID-properties are suitable for traditional short lasting transactions (Gray & Reuter, 1993) they would overly restrict the performance of long lasting activities such as Combinatorial business transactions. On the hand, it is well know that by using semantic information of specific transaction types it is possible to weaken the ACID-properties (Lynch, 1987; Garcia-Molina, 1987; Daconta et al., 2003). For example, with Combinatorial business transactions the requirements of execution atomicity and failure atomicity significantly deviates from traditional database transactions. These issues are considered in the following Sections 4.1 and 4.2.

4.1 Ensuring the Execution Atomicity of Combinatorial Business Transactions

A natural and sufficient criterion for transaction isolation correctness (execution atomicity) is that the executions of transactions are *serializable,* i.e., equivalent to a serial execution (Bernstein & Newcomer, 1997). This criterion is also intuitive and clear. However, though it is suitable for traditional transactions it would overly restrict the concurrency of long lasting activities such as Combinatorial business transactions.

Within Combinatorial business transactions concurrency control is only required to ensure that concurrent Combinatorial transactions will not interfere with each others. That is, if the marketplace has informed the availability of certain products, and if the buyer certifies the request, then the seller must still be able to realize the purchase.

In order to illustrate this let us assume that a seller has placed for sale120 units of product P. Then buyer A makes a request including 100 units of product P, and the marketplace sends the *availability message* to the buyer A. Then buyer B makes a request including 40 units of product P, and buyer C makes a request including 10 units of product P. What should the marketplace do?

The marketplace should now send the *availability message (*i.e., message in-forming that the 10 units of product P is available) to the buyer requesting 10 units of product P, but not with the buyer requesting 40 units of product P. Otherwise, if the buyers A and B would certify their requests on product P, then the marketplace does not have enough units of product P to deliver to both A and B, i.e., the execution atomicity would not be preserved. And so other concurrent activities jeopardized the execution atomicity of Combinatorial business transactions.

The actual problem here is that how correct actions can be determined automatically. If the execution is ensured by traditional data item locks, then the data item indicating the amount of requested products is locked in the marketplace before sending the *availability-message* to the buyer, and the lock is not released until the certification from the buyer is received. However, as the time for waiting the certification may be arbitrary long this kind of solution is not acceptable. Instead, according to our approach the execution of the Combinatorial business transactions is divided into two transactions (Figure 2).

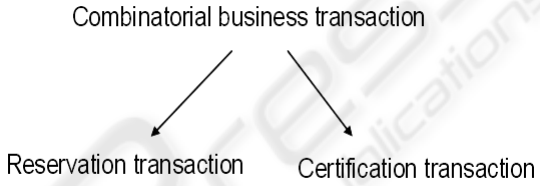

Figure 2: The structure of a Combinatorial business transaction.

- The first transaction, called *reservation transaction,* is executed after the availability of the products is checked and before the *availability-message* is sent to a buyer. It ensures (e.g., by making a preliminary reservation on the products) that the products that are available will also be available until the *certification-message* has been received, i.e., other concurrent activities cannot reserve or sell the product.
- The second transaction, called *certification transaction,* is executed after the marketplace has received the *certification-message*. It has two functions: it marks the products sold that the buyer certified (if any) and it cancels the reservations of the products that the buyer did not certified (if any).

Note that, the *reservation transaction* and the *certification transactions* are traditional database transactions, i.e., they are so called ACID-transactions, which either commit or abort. Also note that the *reservation transaction* ensures that the *certification transaction* cannot semantically fail. By a semantic failure we mean a case where a product cannot be sold as it is no more available. So within the Combinatorial transaction the reservation transaction behaves like the write–lock of a

traditional database transaction and the certification transaction releases the lock.

4.2 Ensuring the Failure Atomicity of Combinatorial Business Transactions

The failure atomicity of Combinatorial business transactions means that the execution tolerates system and communication failures. That is, if all existing failures are repaired and no new failures occur for a sufficiently long time period, then the Combinatorial business transaction will eventually be executed.

Before considering the failure atomicity protocol of the Combinatorial business transaction we will exactly state the correctness criterion of the Combinatorial business transaction.

Correctness Criterion. If the *reservation transaction* of the Combinatorial business transaction is successfully executed (i.e., the reservation transaction is committed and one or more reservation on products have been done), then the *certification transaction* is also successfully executed (i.e., all the reserved products are either sold or released).

Enforcing this constraint requires the coordination between the Buyer application and the Electronic marketplace (Figure 3). Technically this coordination is carried out through Web service interfaces (Newcomer, 2002; Pashtan, 2005) by sending SOAP-messages (SOA, 2007P).

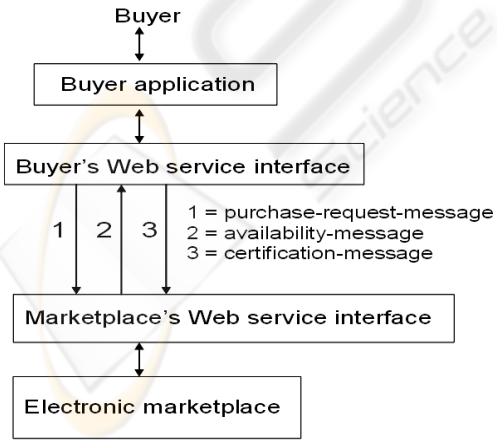

Figure 3: The architecture of the system.

For simplicity we first present the protocol assuming that there are no communication failures. In such a case the protocol goes as follows:

1. The transaction coordinator (a module in Buyer's application) writes a *purchase-request-record* to the log and then sends the *purchase-request-message* to marketplace's web service.
2. After the transaction participant (a module of the marketplace) has received the *purchase-request-message*, it executes the reservation transaction. If the execution failed, then the transaction participant sends to the transaction coordinator the *failure-message* and writes the *failure-record* to the log; otherwise it sends the *availability-message* and writes the *availability-record* to the log.
3. After the transaction coordinator has received the *availability-message*, it transmits the message to buyer's application which analyses the message. Then the transaction coordinator writes the *certification-message* to the log and sends the *certification-message* to the transaction participant

Note that, this protocol will terminate only if all messages are received. There are two places where a service is waiting for a message: in the beginning of steps 2 and 3. In the beginning of step 2, the buyer (transaction coordinator) waits for the *availability-message* from the marketplace (transaction participant). In step 3, the marketplace (transaction participant) is waiting for the *certification-message* from the buyer.

We say that when a service must await the repair of failures before proceeding, the service is blocked. Blocking is undesirable, since it can cause services to wait for an arbitrarily long period of time, and so also the reserved products cannot be sold or reserved for other buyers.

In order that a blocked service (transaction participant) can proceed it must communicate with the transaction coordinator. This kind of communication is carried out in the *termination protocol.* Marketplace activates the termination protocol when it has been waiting a predetermined time for a message. The termination protocol goes as follows:

1. Marketplace's termination coordinator sends *decision-request-message* to the transaction coordinator.

2. The transaction coordinator sends the *response-message* to the participant web service.

The marketplace's termination coordinator repeats the request if it has not received the response in a predetermined time period. Note that the transaction coordinator is always able to response to the request as it has no uncertainty period. *Uncertainty period* is the time period between the moment marketplace's transaction participant sent the *availability-message* to the transaction coordinator and the moment it has received the *certification-message*. During the uncertainty period the marketplace does not know whether the buyer will eventually buy any of the reserved products.

5 CONCLUSIONS

In this article we have considered electronic marketplaces from transaction's point of view. On the other hand, the automation of electronic marketplaces is another important goal of electronic marketplaces: ideally, by introducing appropriate ontologies (Gruber, 1993; Antoniou, & Harmelen, 2004) and business process models (BPEL4WS, 2007; White, 2007) it is possible to replace buyers or sellers or both by software modules, i.e., introduce software programs that place bids on behalf of the user.

The greatest benefits of this kind of automation lie in cost reduction for buyers and sellers. In addition, it shortens the time required for processing business transactions. And most importantly, from Combinatorial business transactions' point of view this means that the time the products are reserved for buyers will significantly decrease. This in turn will increase the throughput of the marketplaces supporting Combinatorial business transactions.

REFERENCES

Antoniou, G. & Harmelen F., 2004. A semantic web primer. The MIT Press.

Bernstein, P., Hadzilacos, V &. Goodman N., 1987. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley

Bernstein, P. & Newcomer E., 1997 *Principles of Transaction Processing.* Morgan Kaufmann Publisher.

BPEL4WS, 2007. Business Process Language for Web Sevices. http://www.w.ibm.com/developersworks/ webservices/library/ws-bpel/

Daconta, M.,. Obrst, L. & Smith K., 2003. *The semantic web.* Indianapolis: John Wiley & Sons.

Garcia-Molina, H., 1983. Using semantic knowledge for transaction processing in a distributed database. ACM Transactions on Database Systems, 8(2):186-313.

Gray, J. & Reuter A. 1993. *Trasaction Processing: Concepts and Techniques*. Morgan Kaufman.

Gruber, T. 1993. *Toward principles for the design of ontologies used for knowledge sharing*. Padua workshop on Formal Ontology..

Lynch N., 1983. *Multilevel atomicity – a new correctness criterion for database con-currency control*. ACM Transactions on Database Systems, 8(4):65-76.

Marinescu, D., 2002. *Internet-based workflow management.* John Wiley & Sons.

Newcomer, E., 2002. Understanding Web Services Addison-Wesley.

Pashtan, A., 2005. *Mobile Web Services*. Cambridge University Press. Cambridge.

Puustjärvi, J., 2001. Workflow concurrency control. The Computer Journal, 44(1)

Puustjärvi, J., 2004. Concurrency control of Internet-based workflows. *In Proc of the Sixth International Conference on Information Integration and Web-based Applications & Services (iiWAS2004),* pages 647-654.

Puustjärvi, J., 2006. CWS-transactions: a transaction model for composed web services. *In Proc. of the Second International Conference on Web Information Systems and Technologies (WEBIST2006)*. pages 69-74.

SOAP, 2007. – Simple Object Access Protocol. http://www.w3.org/TR/SOAP/

White, A., 2007, Introduction to BPMN, IBM Corporation http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf

Wurman, P., 2005. Online Auction Site Management. http://www.csc.ncsu.edu/faculty/wurman/Papers/Wurman-article.pdf