Compensating Transaction Modelling for Web Services

Darsana C.S., Dr. Surekha Mariam Varghese

Abstract— Typical e-business transactions are Long transactions, or long-running transactions in nature. It always costs a long time, maybe one hour, one day, or longer, hereby the relative resources may be locked for a long time. An E-business transaction involves a number of partners, and comprises many failure points. In the web service environment, long transactions always take a long time to finish, which demands more resources that often keep the database item locked for a long time. This would bring down the performance of the transaction processing system. To deal with this issue the transactions are divided into sub transactions and executed independently. If one sub transaction failed and others are committed, the system needs to roll back other sub transactions. However rollbacks of committed transactions are not possible. The solution is to be a compensating transaction with reverse effects. This will improve the system efficiency by avoiding resources being locked for a long duration. When building a business process, most of the time spent in dealing failure points. No guidelines are available to deal with this issue. The objective of this project is to design compensation activities for an online shopping website, which is implemented as a web service. A resource-event-agent (REA)-based value modeling, is used for this purpose. It focuses on the resources, exchanged or transformed during a business activity, providing a useful metaphor to think about compensation. A three-step compensation design approach, implemented with the help of triggers is followed.

Index Terms— Compensating Transaction, e-business transactions, Long Transactions, REA modeling, Sub transaction, Value Modelling,, Web Service

1 INTRODUCTION

A long lived transaction is a transaction whose execution, takes a substantial amount of time, possibly on the order of hours and more. A long lived transaction has a longer duration compared to the majority of other transactions either because it accesses many object of database, it has lengthy computations, it pauses for inputs from the users, or a combination of all these factors. A typical e-business transaction involves a number of partners, and comprises a great number of failure points. Each failure point may involve undoing some steps or reversing their effects fully or partially. Many database researches are running on the field of long running transactions. Relaxed atomicity to the global transaction is the solution defined for this problem. For ensuring that either the process completes successfully as a whole or to have its effects reversed. Due to the long running nature of business processes, it is unthinkable to lock the resources to ensure the ACID properties. Approaches like Sagas consist of slicing the business process into a set of activities treated as ACID transactions. If one Saga transaction fails at runtime, then the whole process should stop and the running Saga should be treated by a regular rollback. However, cannot be rolled back previously committed Sagas and their semantic effects must be reversed. This preserves system consistency. This is called a compensation process. As stated in the BPMN standard, compensation is concerned with undoing steps that were already successfully completed. Because their results and possible side effects are no longer desired and need to be reversed. The designer of a service orchestration (i.e. the consumer of web services) has the responsibility to account for the many exception errors that may occur during the execution of a business process with little or no guidance [1].

For example, making a travel plan, including reserving a flight ticket and reserving a hotel, could be a long transaction, reservation process may needs some time to the available tickets, modify the database, and make the customer confirm. At this time, if another process tries to access the same database, it will be blocked.

Web services provide interoperability between applications. It allows easy integration of new applications to the existing software functions with platform independent fashion [2]. Implementation of the web service affects the transactional behavior of the function, which is accessed as a web service.

For the implementation of a business process, most time spent for handling possible failures or exceptions. Studies report that nearly 80% of the time spent for this [3]. However the main difficulty is that there appears to be numerous ways of compensating for a single activity, and process designers and business analysts, alike, are left with no assistance, and little guidelines, if any, to design compensation activities. Another issue is to make problems worse, process designers are often expected to investigate how to compensate for activities taking place within their business mates. Objective of this paper is to design compensation activity for an online shopping web site; in this case compensation is taken as a business issue.

A value based process modeling is the right level of abstraction for portraying the business decisions that underlie compensation. More clearly, a resource-event-agent (REA)-based value modeling, which focuses on the resources exchanged or transformed during a business activity, contributes useful metaphor to think about compensation.

2 REA MODELING

On the implementation of business process, major time consuming area is to deal failures and exceptions. Compensation can be performed on different ways for a single activity. Proper assistance is not available for the process designers and system analysts. A few guidelines are there for designing compensation activity.

Resource-Event-Agent is an abstract modeling of business process. REA was proposed as a generalized accounting model by William E. McCarthy. REA contains the concepts of resources, events and agents. These concepts are adapted to the computer age. McCarthy defines them as being "objects that (1) are scarce and have utility and (2) are under the control of an enterprise" [4].

REA treats the accounting system as a virtual representation of the actual business [5]. That is, it creates computer objects that directly represent the real world business objects. REA is an ontology in computer science terms. In the real objects included in the REA models are

- Money, goods or services, i.e., resources
- Business transactions or agreements that affect resources, i.e., events
- People or other organizations, i.e., agents

For papers accepted for publication, it is essential that the electronic version of the manuscript and artwork match the hardcopy exactly! The quality and accuracy of the content of the electronic material submitted is crucial since the content is not recreated, but rather converted into the final published version.

A value chain based method is the appropriate representation of business process which requires business compensation activities. Specifically, REA based value modeling, which focuses on the resources exchanged or transformed during a business activity, provides a useful metaphor to think about compensation.

Economic Resources are scarce. And it has utility for economic agents, and is something users of business applications want to plan, monitor, and control. Examples of such resources are products and services, raw materials, money, labor, tools, and services the enterprise uses. An individual or organization capable of having control over economic resources, and transferring or receiving the control to or from other individuals or organizations is called economic agent. Customers, vendors, employees, and enterprises are examples of economic agents. From the perspective of an enterprise we create REA model, which is an economic agent. Event controls the value of resources. Event represents either a decrement or an increment in the value of resources that are under the control of the enterprise. Some economic events occur simultaneously, such as sales of goods. Some events occur over time, such as rentals, labor acquisition, and provision and use of services. A high level REA model of shopping website ABC is shown in Fig.1.

- *Darsana C.S., is currently pursuing masters degree program in computer science and engineering in Mar Athanasius College of Engineering,Kerala,India. E-mail: darsanacs23@gmail.com*
- *Dr.SurekhaMariam Varghese is currently heading the department of computer science ang engineering,Mar Athanasius College of Engineering,Kerala,India.. E-mail: surekha.laju@gmail.com*

Resources are exchanged between agents on specific events. When exchanging resource a duality is maintained in the system. In Fig.1 Book and money are the resources. ABC shopping site and customer are the agents. ABC site sell a book to the customer. Customer got ownership right of the book and ABC got money. ABC follows a decrements event by giving ownership right of the book to the customer. At the same time, ABC got money from the customer. This is an increment event. Here, duality is maintained. In the same sense opposite actions taken place for the customer. He got ownership right, i.e., is an increment event and bank balance will be deducted, i.e., is a decrements event.

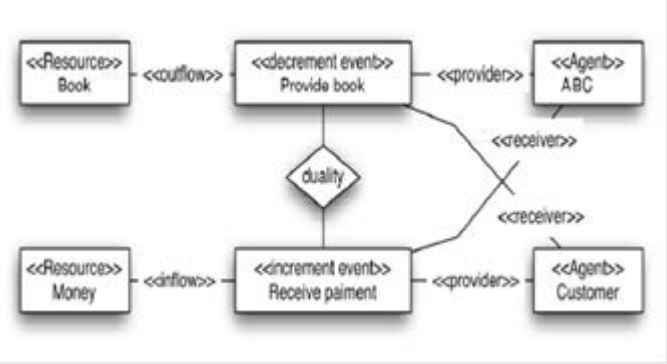

Fig. 1. High level REA model of a shopping website ABC

3 DESIGN

Resource-Event-Agent (REA) framework as an accounting framework aims to record economic phenomena in a shared data environment [1]. It has since been used as an approach to conceptualize and record business activities within an information system and its foundation as a business ontology has been established. REA framework enables us to model business activities in pure business terms using a small set of concepts and the relationships between them; abstracting away from the dynamic aspects of the process. The REA framework permits us to model the business process in terms of business assets - i.e. the Resources- that are controlled by process participants - i.e. the Agents - and exchanged within economic events.

This model follows a three step methodology to analyze and infer business process compensation activities. Compensation design approach involves following steps:

Details of these steps are explained below.

3.1 Value Chain Design

To find out compensation activities, first of all define and encode the business processes. REA ontology provides us with a relevant abstraction enabling us to answer the following three questions:

First find out the entities involved in the process. Resource concept related to it. Different treatments are taken for these entities. Check whether they are exchanging them or converting them. Different participants are involved in the process. Find out who are the actors involved in those treatments.

3.2 Compensating Factors

Two types of compensation factors are consider when designing a compensation process. Class factors and Instance factors.

- Class factors: A REA process may contain transactional exchange and conversional exchange. For compensation design we must find out type of exchange for a particular process. Set of resource rights are exchanged in transactional exchange. Selling a book is a transactional exchange. Customer will get ownership right of the book and company got ownership of money as cost of the book. But pure transactional exchanges are not enough to deal business processes. Some transformational activities are involved in the process, which use/alter or consume some resources. This is performed to gain new or enhanced resources having an added value praised by the customer. These are called conversional exchanges. Resources involved are defined by a set of properties being altered by the economic events. Resources have different characteristics such as reversibility, perishability, appreciation, depreciation and discreteness. Business processes can be categorized based on its reversibility. A process may be reversible or irreversible or semi reversible. Some product values depend on time. A product has increase or decreases its value over time. It depends on the characteristics of the product. These characteristics should consider when designing compensation. Some events are gradual. For a compensation process find out whether it is a gradual event or not. Some events have cost. A company has its own compensation policies. Company may claim money from customers due to cancellation of product. Or company may ready to bear the loss. Based on company rules and policies will vary.
- Instance factors: Find out Time of interruption and who the accountable agents are. In some cases it is customer, another case company itself.

These are the factors considered for designing compensation process.

3.3 Modeling Compensation

A set of compensation rules based on the factors identified. Objective is to provide the analyst with guidance on how to compensate for a given potential failure point. Thus, we must determine for each failure point how to instantiate the compensation value chain and produce a failure point specific value-chain1. To this end, establish a correspondence between the dynamic view of the business process and the REA-value chain. A tagging approach by asking the analyst to identify the activity - or the set thereof - corresponding to each REA event. Identifying set of <Activity, Event> couples. And determine which REA events occurred for each failure point and instantiate the compensation value-chain respectively.

Compensating algorithm follows Saga approach which consists of reversing the value chain graph where, for each REA process. Generate an inverse REA exchange with inflow (respective outflow) resources becoming outflow (respective inflow) resource. Property alterations as well as right transfers are also reversed at this stage. Properties recover their initial values and rights are affected to their initial owner agent. The resulting saga returned will be a syntactically valid REA value chain. Then, iterate through every exchange of the saga and consider each conversional exchanges in order to determine how to compensate for each altered resource. To this end, analyze each property alteration and infer how to compensate for the resource relying on a set of rules. Classify each property against two dimensions: its reversibility and its primary value.

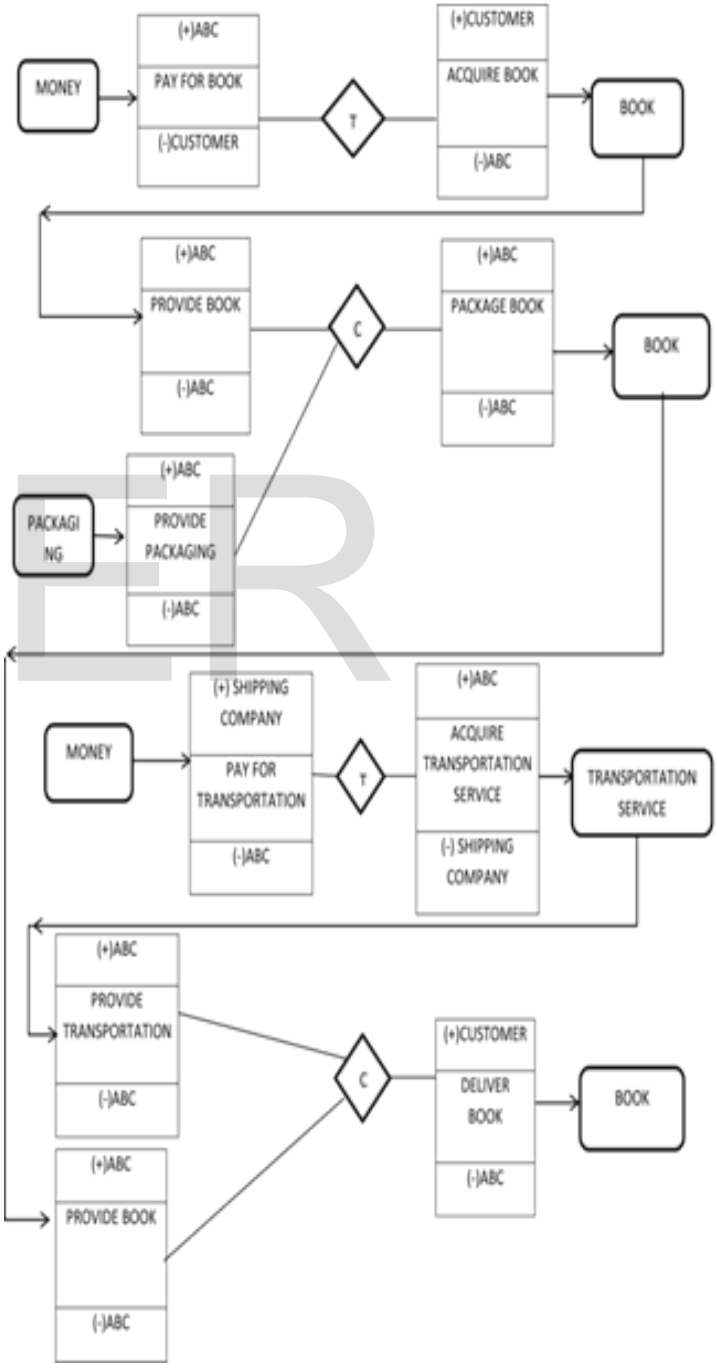

Fig. 2. Value Chain Modeling

First identify resources, events and agents. Based on this

data a value chain model is designed. It is shown in the figure below Fig. 2 shows value chain modeling of shopping website ABC. It shows the resources, events and agents, and how it will be transformed. In the payment event E1 customer buys a book. ABC provides book to customer and customer pay for it. Customer gets ownership of the book by providing money. Then, this book is passed to the packaging section. Package is considered a resource because it includes labor and packaging objects. These are resources. Packaging service and book together converted into packaged state. Here, conversional exchange will be performed. E3 is the transportation event. ABC company pay to the transportation company for their delivery service. Money is transformed into transportation service. Transportation company and ABC company are the agents involved in this exchange. This is a transactional exchange. E4 is the delivery event. With the help of transportation service book is delivered to the customer. It is a conversional event. This is the value chain of a typical book delivery process. From this value chain finds out compensation processes for each event. From this design compensation for each process separately encoded.

4 Implementation

Proposed system provides trigger based semi-automatic mechanism implemented inside the DBMS. Compensating transactions are performed through three steps. [2] First, the rules for generating compensating operations are specified by the web service designer based on the application logic. Next, the database system generates compensating operations, according to the rules, during the execution of the sub transaction. At last when the sub transaction commits, compensating operations for the sub transaction are combined into a compensating transaction when the sub transaction commits. The generated operations of the compensating transaction are stored in the database for later execution according to the rules of the recovery protocol. Fig.3 shows trigger based compensation. For an insert event corresponding delete operations will be set on triggers. Specific situations this trigger will work.

In the web service environment online shopping website is created. In the front end client programs and interfaces are implemented. In the back end server program installed as a web service. To support server Database is implemented. Fig.4 shows screenshots of shopping website ABC.

Customer gets online and checks products. Then, customer order products and retailer checks for the availability of the ordered items. At the same time, the customer's payment information is analyzed and checks balance and other details. In case the debit authorization fails, the process aborts and the order is cancelled. Otherwise, the ordered items are packed and shipped to the customer, using transportation services provided by a shipping company. In the meantime, the order amount is charged to customer's credit card by the financial institution. The process ends when the customer takes possession of the goods he/she ordered.

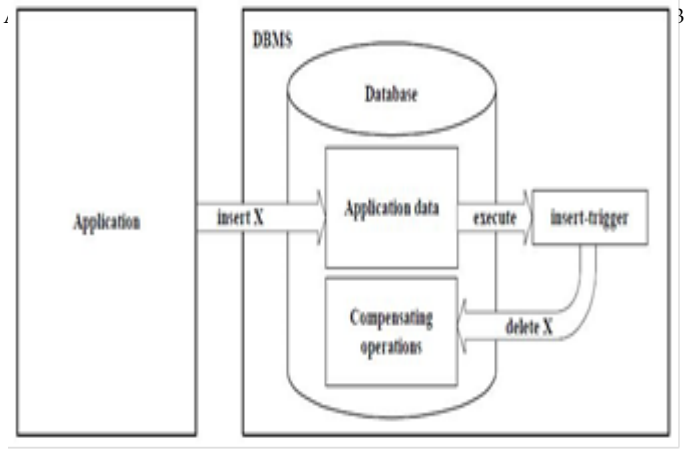

Fig. 3. Trigger Based Compensation [2]

Fig. 4. Online shopping Website

5 Conclusion

The Internet has become a major resource in modern business, thus electronic shopping has gained significance not only from the entrepreneur's but also from the customer's point of view. For the entrepreneur, electronic shopping generates new business opportunities and for the customer, it makes comparative shopping possible. According to survey, most consumers of online stores are impulsive and usually make a decision to stay on a site within the first few seconds. Web service environments, with their loosely-coupled nature and autonomy requirements, need to base transaction execution on extended transaction models. In particular, these environments need the concept of compensating transactions in order to preserve autonomy of participating component transactions. Using a trigger-based mechanism implement compensating transactions. This will avoid the situation that resources are locked for a long time. So it increases the availability to users. To design compensating transactions follow a business oriented approach. A transaction can be compensated on different ways. It is a complex task to select the appropriate action. Compensation is taken as a business problem. The work aims to find out and implement the decision factors for the compensation. Solve the compensation activity in a systematic way, by considering these factors.

References

[1] Anis Boubaker, Hafedh Mili, Yasmine Charif and Abderrahmane Leshob,

"Methodology and Tool for Business Process Compensation Design", 17th IEEE International Enterprise Distributed Object Computing Conference, 2013, pp. 148 -157.

[2] Thomas Strandenæs, Randi Karlsen, "Transaction Compensation in Web Services", Tromsø Univ.,2006.

[3] P. Greenfield, A. Fekete, J. Jang, and D. Kuo. Compensation is not enough. In Seventh IEEE International Enterprise Distributed Object Computing Conference, pages 232–239. IEEE, 2006.

[4] W. McCarthy. The REA Accounting Model - A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review, 57(3):554–578, 1982.

[5] Resources, events, agents (accounting model) – Wikipedia: "https://en.wikipedia.org/wiki/Resources,_events,_agents_(accounting_model)".

[6] Schaefer, Michael, Dolog, Peter, Nejdl, Wolfgang, "An Environment for Flexible Advanced Compensations of Web Service Transactions", ACM,. Aalborg Univ., ACM Transactions on the Web, Vol. V, No. N, January 2008, Pages 1–35.

[7] Hector Garcia-Molina, Kenneth Salem ,"Sagas", In Proc. of the international conference on Management of data, ACM., Princeton Univ., October 1987 pp 249–259.

[8] Christian Colombo, Gordon J. Pace, "A Compensating Transaction Example in Twelve Notations", February 2011.

[9] C. Sonnenberg, C. Huemer, B. Hofreiter, D. Mayrhofer, A. Braccini, "The REA-DSL: A Domain Specific Modeling Language for Business Models".

[10] Fang-Fang Chua, Syahrul Amri Bin Ngazizan, Musa Bin Hassan, "Design and Implementation of Airline Reservation Web Services Using Service-oriented Architecture", Proceedings of the World Congress on Engineering 2010 Vol I,WCE 2010, June 30 - July 2, 2010, London, U.K.

[11] Ramez Elmasri, Shamkant B. Navathe,"Fundamentals of database Systems", Fourth Edition, Pearson.

[12] [12] Debmalya Biswas, "Compensation in the World of Web Services Composition", Springer-Verlag Berlin Heidelberg 2005, pp. 69 – 80

[13] Philip A. Bernstein, Eric Newcomer, "Principles of transaction processing", Second Edition, Morgan Kaufmann Publishers.

[14] Sun Meng and Farhad Arbab, "A Model for Web Service Coordination in Long-Running Transactions", Fifth IEEE International Symposium on Service Oriented System Engineering, 2010, pp.121-128.