Towards a Framework for Modeling Business Compensation Processes

Anis Boubaker[1], Hafedh Mili[1], Yasmine Charif[2], and Abderrahmane Leshob[1]

[1] University of Quebec at Montreal (UQÀM), LATECE Laboratory
[2] Xerox Research Center at Webster

Abstract. A typical e-business transaction takes hours or days to complete, involves a number of partners, and comprises many failure points[8]. With short-lived transactions, database systems ensure atomicity by either committing all of the elements of the transaction, or by canceling all of them in case of a failure. With typical e-business transactions, strict atomicity is not practical, and we need a way of reversing the effects of those activities that cannot be rolled back: that is *compensation*. For a given business process, identifying the various failure points, and designing the appropriate compensation processes represents the bulk of process design effort[8]. Yet, business analysts have little or no guidance, as for a given failure point, there appears to be an infinite variety of ways to compensate for it. We recognize that compensation is a business issue, but we argue that it can be explained in terms of a handful of parameters within the context of REA ontology [20], including things such as the type of activity, the type of resource, and organizational policies. We propose a three-step process compensation design approach that 1) starts by abstracting a business process to focus on those activities that create/modify value, 2) compensates for those activities, individually, based on values of the compensation parameters, and 3) composes those compensations using a Saga-like approach [10]. In this paper, we present our approach, and discuss issues for future research.

1 Introduction

Consider an order and delivery process used by an e-retailer such as *Amazon*. The process starts when the customer gets online and places an order. The ordered items (say, books) are then checked for availability by the *Amazon* warehouse. At the same time, the customer's payment information is ran through a verification process by a financial institution, releasing a debit authorization if it succeeds. In case debit authorization fails, the process aborts and the order is canceled. Otherwise, the order is packaged 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 she ordered.

This process description establishes the "*happy path*". However, every robust business process must address alternate paths handling situations where something could go wrong. In our example, we could envision situations where, say,

S. Nurcan et al. (Eds.): BPMDS 2013 and EMMSAD 2013, LNBIP 147, pp. 139–153, 2013.

the book is out of stock, or the payment is rejected by the customer's financial institution, or the wrong book is shipped, or the order is delivered to a wrong address or, more simply, the order was cancelled by the customer at any moment within the process.

A typical e-business transaction takes hours or days to complete, involves a number of partners, and may comprise a great number of failure points[8]. Each failure point may involve undoing some steps (pretending they never happened) or reversing their effects fully or partially. Database research has thoroughly addressed the problems raised by long running transactions (LRT), aiming to achieve relaxed atomicity to the global transaction by ensuring that either the process completes successfully as a whole or to have its effects reversed (e.g. [10,5]). Due to the long running nature of business processes, it is unthinkable to lock the resources to ensure ACID properties. Approaches like Sagas[10] consist of slicing the business process into a set of activities treated as ACID transactions (i.e. Sagas). If one Saga fails at runtime, then the whole process should stop and the running Saga should be treated by a regular rollback. However, previously committed Sagas cannot be rolled back [21] and their semantic effects must be reversed in order to preserve system's consistency. This is what 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 possibly side effects are no longer desired and need to be reversed[23].

Some studies report that nearly 80% of the time spent on implementing a business process is spent on handling possible exceptions/failures (as mentioned in [18]). An overriding issue seems to be that there appears to be numerous ways of compensating for a single activity, and business analysts and process designers, alike, are left with no assistance, and few guidelines, if any, to design compensation activities. To make problems worse, process designers are often expected to figure out how to compensate for activities taking place within business partners. Our objective is to develop tools and techniques to help business analysts design compensation activities.

We recognize that compensation is primarily a business issue. However, that does not mean that it cannot be explained or rationalized. To the contrary, we argue that the major business decisions that underlie a compensation process can be explained in terms of a handful of parameters that capture the (business) essence of the products and services being manipulated by the process, and the nature of those manipulations. To get to this level of analysis, we need to abstract away from the low-level implementation details of the process (e.g. the interfaces/APIs of the services invoked), and focus on the underlying *business transactions.*

Many authors argued that the essence of a business process resides in the creation of *value* by consuming or transforming a set of *resources* in order to obtain another set of resources perceived by the customer as having a greater overall value (e.g. [16]). We share this view and we believe that a value-based process modeling is the right level of abstraction for representing the business decisions that underlie compensation. More specifically, a *resource-event-agent* (REA)-based value modeling [11], which focuses on the resources exchanged or transformed during a business activity, provides a useful metaphor to think about compensation. In particular, within the context of REA, we have been able to identify *seven* parameters whose values determine compensation. We propose a three-step process compensation design approach that 1) starts by abstracting a business process to focus on those activities that create/modify value, 2) compensates for those activities, individually, based on values of the compensation parameters, and 3) composes those compensations using a Saga-like approach [10].

The remainder of this paper goes as follows. Section 2 briefly describes the REA framework. In section 3 we present the compensation decision factors and illustrate our approach through our *Amazon* running example. We then relate our work to previous research and discuss outstanding issues in section 4, before concluding in section 5.

2 The REA Framework

In an early work, McCarthy[20] introduced the Resource-Event-Agent (REA) framework as an accounting framework aiming to record economic phenomena in a shared data environment. 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 (see [12,13]). Essentially, the REA framework enables us to model business activities in pure business terms using a small set of concepts and the relationships between them.

To illustrate the main concepts of the REA framework, let's consider the *Amazon* example introduced earlier. Taken from a high level, the entire process is concerned with providing a customer with a book in exchange for a money payment (see Fig. 1). Both *Amazon* and the customer undertake actions in order to achieve this business objective. Intuitively, 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*. In the following, we describe each of these concepts.

A business process is a set of activities carried on in order to achieve some business objective. These activities utilize valuable assets as raw material, employee labor, money, and so on. We call these assets *Economic Resources.* McCarthy defines them as being "objects that (1) are scarce and have utility and (2) are under the control of an enterprise" [20].

Activities of the business process are performed by physical persons (e.g. our customer) or moral entities - i.e. organizations or organizational units - (e.g. *Amazon*) called *Economic Agents.* Agents have control over the involved economic resources and are granted the ability to relinquish or acquire them. We associate each economic resource to two economic agents: the one who *provides* the economic resource and the one who *receives* it.

Finally, the REA framework conjectures that each economic phenomenon involves complementary activities called *economic events*: one that increases some

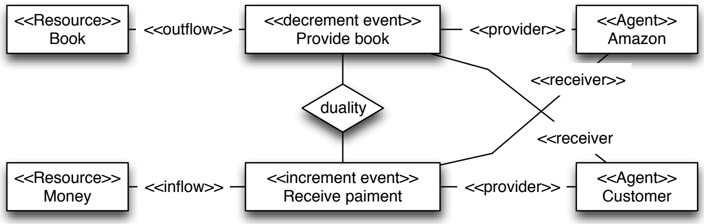

Fig. 1. A high level REA model of *Amazon*'s process

resources on hand - the *increment event* - and one that relinquishes some other resources under the control of the company - the *decrement* event. McCarthy adopted Yu's definition of an economic event[28] as being "a class of phenomena which reflect changes in scarce means resulting from production, exchange, consumption and distribution". Increment and decrement events are linked to their corresponding resources through a *stock-flow* association - i.e. *inflow* and *outflow*, respectively.

Increment and decrement events of a given economic phenomenon are also linked through a *duality* relationship exhibiting the *exchange* between involved agents. Thus, the exchange shows the essence of value creation and aims to rationalize the business process by exhibiting why resources have been relinquished. Note that REA events differ from the traditional metaphor of events as seen in programming languages (e.g.: WS-BPEL) by being long lasting rather than instantaneous.

The discussion above introduces two aspects the REA ontology permits us to represent. First, REA enables us to exhibit the transactional nature of a business process by exposing the sequence of what is relinquished by the process and what is gained in exchange. Thus the concept of an exchange is the cornerstone of the REA ontology and establishes the rationale behind business activities by recording *why* the business engages in such activities. Second, the REA framework aims at exhibiting *how* value is created through business activities by modeling a value chain as introduced by Porter[25].

In the next section we describe our approach to modeling compensation processes relying on the REA abstraction.

3 Value-Oriented Compensation - Our Approach

We argued in[21] that current implementations of compensation in service-oriented architectures have inherent problems in regards to language constructs. The designer of a service orchestration (i.e. the consumer of web services) has the responsibility to account for the many exceptions errors that may occur during the execution of a business process with little or no guidance. In other words, the designer has to know what services he should invoke to compensate a given service or has to implement his own compensation activities – thus impeding service reusability.

Notwithstanding the technical issues, we share the view that compensation is first and foremost a business problem. However, we argue that behind the seemingly infinite variety of compensation responses that organizations can deploy to a given failure, there lay a handful of principles that we should be able to codify [21]. This, in turn, relies on our ability to analyze the activities within a typical business process in pure economic and business terms, abstracting away—for the time being—the idiosyncrasies of the corresponding record-keeping by the IT system. To this end, we propose, as a first step to our approach, to use the REA ontology for analyzing business processes. The second step defines the requirements of compensation activities relying on a set of factors expressed in pure business terms. The last step will construct the compensation process value chain intended to provide the process designer with a desirable insight on which compensation activities should be considered and what each activity should be concerned about.

In the following subsections, we will build on the *Amazon* example introduced earlier in order to explain each step of our approach.

3.1 REA Value-Chain Design

In Figure 1 we have modeled our *Amazon* process at a level that does not account for all the resources that are consumed in the process of delivering the book to the customer– and that may need to be compensated for in the case of a process failure. For example, the exchange of book versus money could not happen as depicted in Fig. 1 since both parties are in different locations. Indeed, the book needs to be transported to customer's location and the money has to be somehow collected by *Amazon*. In order to change the *Book*'s location property, *Amazon* relies on a *shipping company* who provides it with transportation rights. Similarly, the change of *Money*'s location involves a *financial institution* handling the credit card transaction, which, in turn, involves transaction fees charged by the *financial institution*.

The REA framework allows the description of the business process at different levels of abstraction. It is theoretically possible to develop a model that encompasses all economic phenomena. However, this needs to be balanced against the need to keep our models manageable and scalable. In [11], Geerts et al. suggest to decompose business processes down to the level needed "to plan, control and evaluate" the business process

In our example, for the sake of simplicity, we suppose that the transportation service has been previously acquired from the shipper (see E2, Fig. 2). This service is consumed to ship the book, resulting in the E3 REA conversional exchange, shown on Fig. 2. As for the banking transaction fees, we made the (modeling) decision to not account for them as a full fledged resources, along with the corresponding events and duality relationship. Thus, we chose to represent them as an attribute of the *Receive Payment* event (E4). By continuing further down we notice that the book cannot be shipped as is and must be packaged

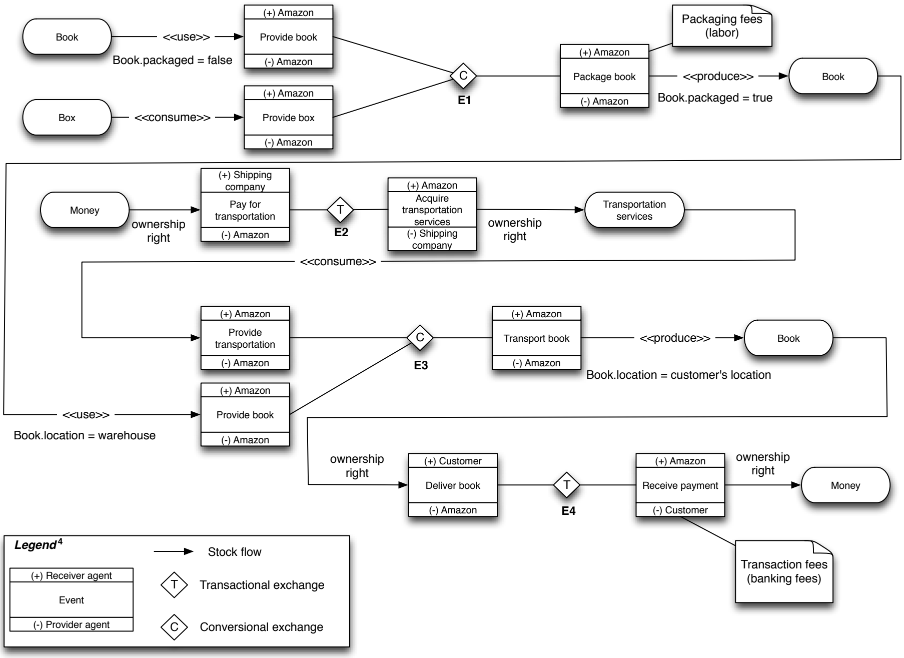

Fig. 2. REA value chain of *Amazon*'s process

prior to shipping. This transformation requires the consumption of a box as well as the labor of an employee (E1). As we did with banking fees, we decide to account for employees' labor as an attribute of the packaging activity.

Once we have determined the REA exchanges at the desired level of abstraction, we construct the global value chain by connecting the exchanges. Fig. 2 depicts the global value chain for our *Amazon* example. Section 3.2 explains the compensation decision factors needed to compensate for the various exchanges.

3.2 Compensation Factors

Consider the exchange E1 from Fig. 2, which creates a "book in a box" from a book and a packing service. If the process aborts after E1 has been performed, we need to figure out how to reverse this exchange. This, in turn, requires us to understand, among other things, what activities have taken place, and how they affect the manipulated resources: (1) a box has been consumed and is no longer usable for future exchanges, (2) the book is now enclosed in a box, and (3) labor has been consumed– in this case, represented as an attribute of E1. Intuitively, we can return the book to its previous state, by spending a bit more labor. However, the box is "lost forever". This illustrates two compensation decision parameters, 1) the *type of process*, and 2) the way it affects its input resources (the book versus the box). Luckily, these parameters have a handful of values each: 1) an exchange can be either *conversional* or *transactional*, and 2) the effect of a conversional exchange on a resource can be either *reversible* or not.

We have identified seven compensation decision factors, which fall into two groups, 1) what we called *class factors,* that depend on things such as *types* of (REA) events and resources, and that determine *whether* an exchange or resource needs to be compensated, and *how*, and 2) *instance* factors, which determine, for a particular exchange *instance*, *whom* and *how much* to compensate. We present the two groups in turn.

Class Factors

What is the Type of the Exchange? In order to compensate for a given activity, one should be able to asses the effects of the activity on the economic resources. The REA framework distinguishes two types of exchanges: *transactional exchanges* and *conversional exchanges* [19].

Transactional exchanges involve an exchange of a set of resources rights. A typical example would be a sale exchange where the company relinquishes its *ownership right* on the product it is selling and gaining an *ownership right* on customer's money. Other types of rights may include usage rights, copyrights, etc. Thus, resources involved in transactional exchanges are perceived as a collection of rights. Compensating a transactional exchange requires us to reverse the exchange by returning those rights to their original provider. For example, compensating the exchange E4 from Fig. 2 would involve returning the ownership of the money to the customer and the ownership rights of the book to *Amazon.*

However, business activity do not revolve solely on transactional aspects. Although most business collaboration processes comprise a transactional activity at some point, they are usually combined with transformational activities that either *use/alter* or *consume* some resources in order to gain new or enhanced resources having an added value praised by the customer. These are called conversional exchanges and resources involved are defined by a set of properties (as opposed to a set of rights) being altered by the economic events.

Consumed resources cease to exist and cannot be restored to the provider agent. Thus, the provider agent will be compensated by receiving a resource of an equivalent value. We propose to compensate a consumed resource by relinquishing an abstract economic resource called *a claim* that could be settled in the future by an equivalent value - usually money. Claims have been introduced by McCarthy as "(...) not tangible resources for and against the enterprise. Claims derive from imbalances in duality relationships where an enterprise has either (1) gained control of a resource and is now accountable for a future decrement (...) or (2) relinquished control of a resource and is now entitled to a future increment (. . .)"[20]. In our example, the box has been consumed and is therefore compensated by a claim.

On the other hand, a used resource is one whose properties- or a subset thereof- are altered by the conversional exchange. Compensation consists then of attempting to restore those properties to their original values. For example, an REA event that constructs a machine by assembling its parts can be compensated by disassembling those parts. However, if some parts are altered during the assembly (e.g. welded), they may not recover their original value upon disassembly. Here again, the original provider of the part would need to be issued a claim for the lost value.

What is the Type of the Resource? I walk into a food store, pick-up a can of soup that has an expiration/consume-before date. If I change my mind and decide to return the can, the store will likely take it back if it is before the consume-before date. If I return it *after* that date, the can would have lost all of its value. The soup can belongs to the category of *perishable resources* whose value goes to zero at a given date/instant. Other perishable resources include hotel room bookings, flight seats, or rock concert tickets. If, on the other hand, the purchaser decides to return the resource *before* the consume-before date, the seller may credit them for *part of the purchase*, depending on how difficult it is for the seller to turn around and resell it while it still has value. This simple example illustrates one *dimension/subcategorization* of resource types that influences, a) whether or not a resource can be compensated, and b) how and how much. We identified foor such (non-orthogonal) dimensions, explained below: *reversibility*, *perishability*, *depreciation*, *and discreteness.*

- Reversibility: Shipping the book from *Amazon*'s warehouse to customer's location does not normally affect the book's physical condition. This means that the book will not sustain any value loss on its way to the customer. Consequently, we consider the book as being a reversible resource that we compensate by restoring its original property. However, the reversibility factor must be linked to the resource's property being altered by the event. Thus if, say, the same book has been autographed before being shipped (i.e. a conversion process altering its *autographed* property), then it has been irreversibly altered by the event. A non-reversible resource involves a loss in value, in full or in part, sustained by the owner, that a compensation process must take into account, for example by redeeming the lost value.
- Perishability and depreciation/appreciation: As explained above, resource is *perishable* if it completely loses its value at a given date. By contrast, *depreciation* corresponds to a *gradual* loss of value, notwithstanding the loss of value due to wear and tear. For example, notwithstanding generous product return policies, over time, computers and cars depreciate, even when they are not used, because of the advances in technology (and fashion). Conversely, art and collector items tend to increase in value, with time. Both perishability and depreciation/appreciation involve compensating for the resource's lost value as a function of time (i.e. time of process interruption).
- Discreteness: I need a four-foot long wooden beam, with a two-by-four inch base. The local hardware store sells only beams that are eight foot long. If I

cut and use a four-foot segment, I cannot return the other half. Similarly, if I need five four-foot segments, I will need to buy three eight-foot beams. With discrete resources, exchanges and compensation are measured in discrete units, even if the actual quantity used is continuous. By contrast, if I consume 5MW of power, I will pay for only– and exactly– that. Non-discrete resources tend to be substance-like [7], in the sense that if we divide it into two (or more) parts, the parts are of the same nature as the original resource.

Going back to our Amazon, the book is a) reversible on both its packaged *and* location *properties, b) non perishable and c) discrete. Therefore, returning the book would not involve any loss in value to consider for compensation.*

Is it a Gradual Event? Assume that I want to paint my living room a light sunny blue and I hire a painter to work by the hour. The painter needs to mix the paint first, to obtain a gallon-plus of light blue paint, and then paint the living room with it. The 'mix the paint' activity/REA event *consumes* the 'input' cans at the beginning, i.e. as soon as I pour *one drop* of color paint into the white paint container. Indeed, from that point on, neither can of paint can be reused nor repurposed. By contrast, the actual painting of the living room consumes the labor *gradually*: if the painter stops at any time during the activity, I will only pay for time used.

This example illustrates the difference between an activity (REA event) that consumes its input resources *gradually* (laying the paint on the walls) from one that consumes its resources *atomically* at some point during the activity, typically at the beginning (mixing the paint), or at the end.

Going back to our amazon example, for practical and business reasons, all of the activities (REA events) consume their resources atomically. For example, once we start packing the book (exchange E1 in Fig.2), the box is altered, and we consider that the labor required for the packing has been consumed. Similarly, once the book leaves the warehouse, we consider that the shipment service has been consumed.

Are there Event Costs? Many business activities, and REA exchanges in particular, involve labor. Theoretically, a value chain model should show labor as an economic resource, and represent its consumption by separate economic events. However, such a granularity of representation may result in large and hard to understand models. As discussed earlier in section 3.1, Geerts et al. argued that analysts should stop modeling at the level enabling to plan, control and evaluate the business process. It is not clear that accounting for labor consumption, *in all circumstances*, with separate economic events helps to plan, control and evaluate the business process. For example, in a car manufacturing process (a *conversional exchange*), labor accounts for a significant part of the cost, and it should be represented as a *resource*, and *its consumption* on the assembly line should be represented by an *economic event*. By contrast, approving supplier invoices for payment should not. This is *not* to say that these costs should not be counted. We suggest to use a 'cost' attribute attached to events, that aggregates all of the costs that we deemed too low-level to merit a full REA treatment. If a completed activity is to be compensated, in addition to the resources that it did consume (reversibly or not, atomically or not, etc.), we need also to compensate for the activity cost.

In our example, the labor used for packing (exchange E1 in Fig. 2) is included in the 'cost' property of the exchange E1. Similarly, the banking transaction charges (exchange E4 in Fig. 2) are included in the 'cost' property of exchange E4.

Are there Compensation-Specific Business Policies and Rules? Going back to the process presented in Fig. 2, suppose that the customer decides to cancel his/her order after the package was sent out for shipping. The previous factors determine whether the book is returnable, and how much resources have been irreversibly used or consumed, both in the going forward process, and in the compensation process. There remain a number of issues / choices, which are typically driven by organization/company-specific policies or rules. Most notably:

1. should the resource(s) that is/are at the heart of the process (book and cash) be recoverable, in what form will they be returned to their original owners? in our amazon example, the customer may either have his credit card credited for the refund, or receive a credit voucher with the equivalent value towards future purchases
2. how much should the customer be credited, anyway? as we saw from our amazon example, the cancellation of a book order carries a number of non-reversible costs, including the box used for packing, the shipping to and back from the customer, the banking transaction costs, and the labor involved in packing and unpacking. It is a matter of business policy to choose which costs to incur to the customer, and under what circumstances (see 'who is the accountable agent' below).

We see business policies as business domain and corporate specific refinements of the compensation factors discussed above. A business policy does *not* change the *nature* of an event or the *type* of the resource. However, it may influence the choice of which compensation activity to choose, among many, or how much to compensate for.

In the amazon example, customers who cancel orders that have been shipped are liable for, a) the initial shipping charges, b) the return shipping charges. However, they are not liable for the labor costs or the banking transaction costs. Further, the money they are entitled to is credited back to their credit cards.

Instance Factors. The *class factors* introduced above enabled to determine the compensation activities required for a given exchange with given dual events and involving given resources. They are general factors applicable to any execution of the business process. Conversely, instance factors apply to a specific execution of the process and enable us to assess to which extent one should compensate for a given resource. We identify two instance factors: the time of interruption and the accountable agent.

What is the Time of Interruption? Abortion of a business process may happen anywhere between the instant it starts and the instant it ends. Thus knowing the time at which a process aborts is a critical information in order to establish (1) which of the activities completed and thus need to be compensated and (2) how much of the resources involved during the last running event(s) need to be accounted for.

Who is the Accountable Agent? If one cancels his trip a week prior to the departure date he expects to pay for cancellation fees that may go as far as the price of purchase for the trip. Indeed, aborting a business process may involve losses sustained by either the company in charge of the business process and/or an actor involved in the process. Some of these losses may be absorbed by the organization fully or partially as dictated by business policies while some won't, thus rendering one of the agents responsible for the losses. In order to identify the accountable agent of a given abortion, we conjecture him being:

– The provider agent of the economic event causing the process abortion in a case of process failure (e.g. The travel agency if the plane seats have been overbooked); or
– The agent who triggered the canceling event (e.g. the customer who cancelled his order).

3.3 Modeling Compensation Processes

The compensation factors presented in the previous subsection along with a given REA business process value chain will enable us to infer corresponding compensation activities for each REA exchange. Our methodology relies on a catalog of <REA exchange, REA compensation exchange> patterns based on the factors identified above that we will build. For example, we might have different pairs for the different types of conversions, based on the resource types, the gradualness of the consumption, etc. Note that the pure transactional exchanges are reversible, modulo the transaction costs (and if different, the reverse transaction costs). For conversion processes, the factors discussed above should help us design the REA compensation exchanges at a high level.

Given a value chain model, we consider every exchange and classify it according to the catalog outlined above by asking questions to the analyst. Such question may include: *Is your resource <A> perishable (discrete, etc.)? Does your event <E> consume its resources at the beginning, gradually or at the end?*. Based on the answers, we will be able to select the appropriate pattern/pair <REA exchange, REA compensation exchange>, and instantiate the REA compensation exchange part using her domain terminology. An illustration of such compensation exchange instantiation applied to exchange E1 from our *Amazon* example is shown in Fig. 3.

Finally, once the relevant compensation exchanges have been identified, we propose to construct a compensation value chain by composing compensation exchanges in reverse order following the Sagas[10] approach and in respect to the resource flow of the original value chain.

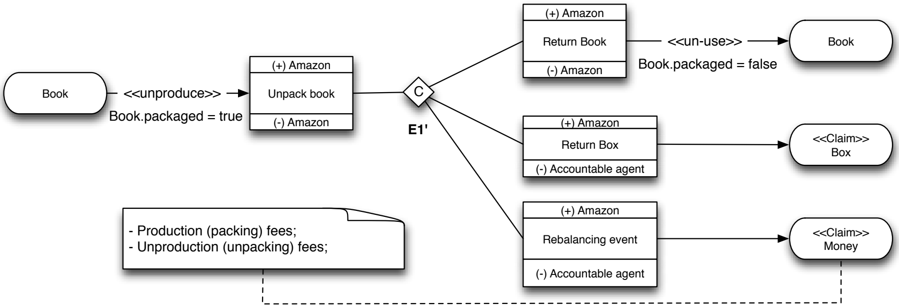

Fig. 3. Compensation exchange of E1 from Fig. 2

4 Discussion

4.1 Related Work

Our work builds on transaction management in the fields of distributed databases and long running workflows. Most of current implementations of error recovery in business process enactment engines rely on the Sagas[10] approach first introduced by Garcia-Molina. A saga is a chained transactions technique aiming at ensuring global transaction atomicity by slicing a process into a set of ACID transactions. In case of an error and the process needs to abort, successful ACID transactions are compensated in their reverse execution order by invoking their compensation handlers. WS-BPEL[22], a standard language for service composition execution, uses a Saga-like approach to handle errors by implementing *fault, compensation and termination* (FCT) handlers. In order to achieve model checking and ensure reachability in error handlers, authors (e.g. [9], [4]) expressed FCT mechanisms in formal semantics based on pi-calculus and Petrinets.

In recent work, authors proposed alternate approaches to handle process execution failures, mainly to achieve process self-healibylity. Techniques inspired by aspect programming(e.g. [14,26]) permitted to separate process design from failure handling by treating error paths as crosscutting concerns. Advances in semantic web services allowed implementing transactional support through negotiating agents([3]). Some approaches emphasize the human involvement in recovering from business process exceptions. In order to achieve organizational resilience, Antunes proposes a framework integrating both machine and human involvement in error recovery[1]. His approach relies on a *control switch* concept supporting ad-hoc human interventions by moving control out of the BPM enactment engine whenever a certain type of exception occurs. Although we find these directions promising, we argue that the problem of compensation remains to handle backward recovery. All the approaches above mentioned focus essentially on technical aspects from an operational perspective such as language constructs, message exchanges and coordination. To the best of our knowledge, no work has been done in supporting compensation design at the analysis level.

On the business process design, different authors stressed the advantages of business modeling prior to process modeling in order to assess the rationale of the business process and to express business objectives at a high level, from which implementation should be derived. Many business ontologies have been proposed that fall into two groups, value-oriented (e.g. [20,15,24]) and goal-oriented approaches (e.g. [27,2,17]). Although authors have tackled the problem of process modeling based on abstract business objectives, none has applied these aspects to error recovery and compensation. Thus our work aims at filling this gap and offers to consider business modeling as a valuable option to the error recovery.

4.2 Validation Approach

Generally speaking, there are two aspects of our approach we need to validate. Firstly, as mentioned in section 3.3, we will construct an exhaustive catalog of *<REA exchange, REA compensation pattern>* pairs based on our compensation factors. The catalog will be validated using existing business ontologies and on business process modeling experts who will be asked to assess the soundness of our high level compensation patterns.

The second aspect of our validation will concern the resulting compensation processes obtained using our methodology for which we will take a two steps approach. First, we will compare our results to a catalog of classified business processes. We chose the reference model of the SAP business blueprints[6] because we believe it is at the right level of abstraction enabling us to validate on a wide range of business domain activities. The reference model describes best-business practices from many different industries by modeling, both the 'happy' and alternate paths, of generic business processes. We will extract those happy paths and model their value chain using the REA ontology. Then we will walk through our methodology in order to generate compensation processes and compare them to the business error handling paths of SAP's model. We expect to generate more compensation processes than can be found in SAP blueprint because our method will cover the cases exhaustively. As a second step, we will validate on real world cases by consulting a panel of subject matter experts within specific organizations. Our approach will be to work with their business process and see whether (1) we ask the right questions (perishability, discreteness, etc.), and (2) the resulting compensation processes are sound from their business perspective.

Note that we have developed a prototype implementing our approach and applied our factors and our framework to a number of real world examples. This enabled us to refine our initial set of compensation factors and to establish the ground rules of our approach.

5 Conclusion

In this paper, we proposed a business-oriented approach in order to assist the business process designer in establishing the compensation activities. We argue that despite apparent numerous ways of compensating for a given business process, the compensation is mainly a business problem; therefore, the solution should be tackled from a business standpoint.

Extensive research focused on the compensation from a technical programming perspective and, to the best of our knowledge, none has addressed the problem from a business standpoint. Our work aims at filling that gap. The main contribution of our work lies in establishing the decision factors involved in designing compensation activities. Relying on these factors, we were able to determine compensation activities and elicit their requirements is a systematic fashion, hence providing the business analysts with a much-welcomed guidance.

Although this work is still at an early stage, this paper focuses on establishing our ground ideas and our major directions. We are currently working on constructing a catalog of high level business activities paired with corresponding compensations in order to apply the compensation factors programmatically.

References

1. Antunes, P.: BPM and Exception Handling: Focus on Organizational Resilience. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 41(3), 383–392 (2011)
2. Behnam, S.A., Amyot, D., Mussbacher, G.: Towards a Pattern-Based Framework for Goal-Driven Business Process Modeling. In: 2010 Eighth ACIS International Conference on Software Engineering Research, Management and Applications (SERA), pp. 137–145 (2010)
3. Bocchi, L., Ciancarini, P., Rossi, D.: Transactional aspects in semantic based discovery of services. In: Jacquet, J.-M., Picco, G.P. (eds.) COORDINATION 2005. LNCS, vol. 3454, pp. 283–297. Springer, Heidelberg (2005)
4. Butler, M., Ferreira, C.: A Process Compensation Language. In: Grieskamp, W., Santen, T., Stoddart, B. (eds.) IFM 2000. LNCS, vol. 1945, pp. 61–76. Springer, Heidelberg (2000)
5. Butler, M., Ferreira, C.: An Operational Semantics for StAC, a Language for Modelling Long-running Business Transactions. In: De Nicola, R., Ferrari, G.-L., Meredith, G. (eds.) COORDINATION 2004. LNCS, vol. 2949, pp. 87–104. Springer, Heidelberg (2004)
6. Curran, T., Keller, G., Ladd, A.: SAP R/3 Business Blueprint. Understanding the business process reference model. Prentice Hall (1998)
7. CYCorp. CYC Knowledge Base
8. Dayal, U., Hsu, M., Ladin, R.: Business Process Coordination: State of the Art, Trends, and Open Issues. In: 27th Very Large Databases Conference, VLDB 2001, Roma, pp. 3–13 (2001)
9. Eisentraut, C., Spieler, D.: Fault, Compensation and Termination in WS-BPEL 2.0 - A Comparative Analysis. In: Bruni, R., Wolf, K. (eds.) WS-FM 2008. LNCS, vol. 5387, pp. 107–126. Springer, Heidelberg (2009)
10. Garcia-Molina, H., Salem, K.: Sagas. In: SIGMOD 1987: Proceedings of the 1987 ACM SIGMOD International Conference on Management of Data, pp. 249–259. ACM Request Permissions, New York (1987)
11. Geerts, G.L., Mccarthy, W.E.: Modeling Business Enterprises as Value-Added Process Hierarchies with Resource-Event-Agent Object Templates. In: Business Object Design and Implementation, pp. 94–113. Springer (1997)

12. Geerts, G.L., Mccarthy, W.E.: The Ontological Foundation of REA Enterprise Information Systems. In: American Accounting Association Conference, Philadelphia (2000)
13. Geerts, G.L., Mccarthy, W.E.: An ontological analysis of the economic primitives of the extended-REA enterprise information architecture. International Journal of Accounting Information Systems 3(1) (2002)
14. Ghidini, C., Francescomarino, C.D., Rospocher, M., Tonella, P., Serafini, L.: Semantics-Based Aspect-Oriented Management of Exceptional Flows in Business Processes. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 42(1), 25–37 (2012)
15. Gordijn, J., Akkermans, H., van Vliet, H.: Business Modelling is not Process Modelling. In: Mayr, H.C., Liddle, S.W., Thalheim, B. (eds.) ER Workshops 2000. LNCS, vol. 1921, pp. 40–51. Springer, Heidelberg (2000)
16. Gordijn, J., Wieringa, R.: A value-oriented approach to e-business process design. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, pp. 390–403. Springer, Heidelberg (2003)
17. Gordijn, J., Yu, E., van der Raadt, B.: E-service design using i* and e/sup 3/ value modeling. IEEE Software 23(3), 26–33 (2006)
18. Greenfield, P., Fekete, A., Jang, J., Kuo, D.: Compensation is not enough (fault-handling and compensation mechanism). In: Proceedings of the Seventh IEEE International Enterprise Distributed Object Computing Conference 2003, pp. 232–239. IEEE Comput. Soc. (2006)
19. Hruby, P.: Model-Driven Design Using Business Patterns. Springer (2006)
20. Mccarthy, W.E.: The REA Accounting Model - A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review 57(3), 554–578 (1982)
21. Mili, H., Godin, R., Tremblay, G., Dorfeuille, W.: Towards a Methodology for Designing Compensation Processes in Long-Running Business Transactions. In: Montreal Conference on eTechnologies, MCETECH 2006, Montreal, pp. 137–148 (2006)
22. OASIS. Web Services Business Process Execution Language (BPEL). OASIS (2007)
23. OMG. Business Process Model and Notation (BPMN). OMG (January 2011)
24. Osterwalder, A.: The Business Model Ontology. PhD thesis, Université de Lausanne École des Hautes Études Commerciales (2004)
25. Porter, M.E.: Competitive advantage: Creating and Sustaining Superior Performance, New York, USA (1985)
26. Sonntag, M., Karastoyanova, D.: Compensation of Adapted Service Orchestration Logic in BPEL ' n ' Aspects. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 413–428. Springer, Heidelberg (2011)
27. Yu, E.S.K.: Models for supporting the redesign of organizational work. In: COCS 1995: Proceedings of Conference on Organizational Computing Systems. ACM Request Permissions (August 1995)
28. Yu, S.C.: The Structure of Accounting Theory. The University Press of Florida (1976)