Methodology and Tool for Business Process Compensation Design

Anis Boubaker, Hafedh Mili, Yasmine Charif and Abderrahmane Leshob
LATECE Laboratory, University of Quebec at Montreal (UQÀM)

***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. 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 the REA ontology [19], including things such as the type of activity, the type of resource, and organizational policies. We propose a three-step 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 along with an implementation algorithm and propose a business ontology for compensation design.**

I. Introduction

Consider an order and delivery process used by an e-retailer such as *Amazon*. The process starts when a customer gets online and orders books. Then Amazon checks for the availability of the ordered books. 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 the debit authorization fails, the process aborts and the order is canceled. Otherwise, the ordered books 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.

This process description establishes the "*happy path*". However, many things could go wrong. Amazon's inventory could report a wrong stock, or the payment be rejected by the customer's financial institution, or the wrong book be shipped, or the order be delivered to a wrong address or, more simply, the order be cancelled by the customer at any moment during the process.

A typical e-business transaction takes hours or days to complete, involves a number of partners, and comprises 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], [6]). 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 [20] 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[22].

Some studies report that nearly 80% of the time spent on implementing a business process is spent on handling possible exceptions/failures [17]. An overriding issue is 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 their 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. 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 or 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* factors 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. We start with a literature review in section 2 followed by a brief description of the REA framework in section 3. Section 4 provides an overview of our approach using our *Amazon* running example. In section 5 we will discuss our algorithm and implementation. We will then propose a compensation ontology architecture leveraging the usability of our approach in section 6 before discussing outstanding issues and concluding in section 7.

II. Related Work

Our work builds on transaction management in the fields of distributed databases and long running workflows. Most current implementations of error recovery in business process enactment engines rely on the Sagas approach first introduced by Garcia-Molina[10]. A saga is a chained transactions technique aiming at ensuring global transaction atomicity by slicing a process into a set of ACID transactions. In the case of an error that requires aborting, successful ACID transactions are compensated in their reverse execution order by invoking their compensation handlers. The WS-BPEL[21] 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], [6]) 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-healability". Techniques inspired by aspect programming permitted to separate process design from failure handling by treating error paths as crosscutting concerns [14]. 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 [2] proposes a framework integrating both machine and human involvement in error recovery. 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.

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 that have been proposed fall in one of two groups: goal-oriented and value-oriented approaches. Goal driven ontologies try to model business activities by capturing and formalizing business goals (ex.: [26]). Value-oriented ontologies, on the other hand, focus on exhibiting the creation of value within business activities. Such ontologies include REA, presented in section 2, e^3-value[15] and BMO[23]. Andersson et al. [1] have compared these three ontologies and proposed a reference ontology. Many authors ([16], [25]) proposed to apply business value modeling to support business process definition. Zdravkovic et al. [28] propose a model-driven approach to construct a high level business model (Computer Independent Model) using the REA and Open EDI Business Transaction (OeBTO) ontologies.

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. 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 conceptual modeling as a valuable option to the error recovery.

III. The REA Framework

McCarthy[19] 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; abstracting away from the dynamic aspects of the process.

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 use valuable

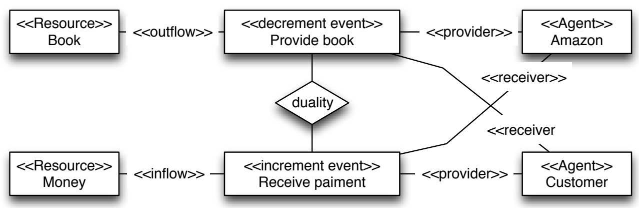

Figure 1. A high level REA model of *Amazon*'s process.

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" [19].

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 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 [27] definition of an economic event 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.

The REA framework distinguishes two types of exchanges: *transactional exchanges* and *conversional exchanges* [18]. Transactional exchanges involve the 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 gains an *ownership right* on the 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.

However, business activities 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.

Increment and decrement events of a given economic phenomenon are also linked through a *duality* relationship exhibiting the *exchange* between involved agents. 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 allows us to represent. First, REA enables us to exhibit the business transactional nature of a 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 the reasons *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[24].

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

IV. Approach Overview

We argued in [20] that current implementations of compensation in service-oriented architectures have inherent problems with 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 exception 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 scope of compensation responses that organizations can deploy to a given failure, there lay a handful of principles that are codifiable [20]. This, in turn, relies on our ability to analyze the activities within a typical business process in pure economic and business terms, abstracting away the idiosyncrasies of the corresponding record-keeping by the IT system.

We laid the ground rules of our approach in [5], [4] and we proposed a three steps methodology to analyze and infer business process compensation activities 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.

In the following subsections we describe each of these steps.

A. Value chain design

In order to assess business process compensation activities we need to define and encode the business rationale embodied by the business process. As we have explained in the previous section, the REA ontology provides us with a relevant abstraction enabling us to answer the three following questions:

1) *what entities of importance to the company are involved in the process?* These entities are embodied by the economic resource concept;
2) *what treatment these entities are subjected to during the business process execution?* i.e. are we exchanging them or converting them?
3) *who are the actors involved in such treatments?* In other words, who is entitled to receive or relinquish ?

We discussed in [4] a top-down methodology for designing an REA model of a business process. Quickly stated, we consider the process from a high level of abstraction and start by modeling the main exchange being the main objective of the business process. The exchange illustrated in Fig.1 represents the main exchange of our *Amazon* example. The exchange could be a transaction (ex.: sales or hiring process) as well as a conversion (ex.: production process); this mostly depends on the business domain to which the process belongs.

From that point we dig deeper to determine the intermediate resources that are acquired, used or consumed in order for the main exchange to happen. Going back to our *Amazon* example, it appears clear that the sale could not happen as illustrated on Fig.1 since both agents are not in the same physical location. Thus, we must take further actions in order to enable the sale by transporting the book to the customer's address while ensuring that he pays for his order and his money payment is somehow credited back to Amazon's bank account. Transporting the book and perceiving the customer's money are both REA conversional exchanges in their own account; the former changes the *physical location* property of the book and the latter the *bank account* where the money is stored.

Going further down, we can imagine that the book needs to be placed in a labelled box. Similarly, Amazon needs to acquire a transportation service from a shipping company to ship the packed book to the customer. The analyst should continue decomposing the process to the level sufficient in order "to plan, control and evaluate"[11] the business process. Once we have determined the REA exchanges at the desired level of abstraction, we construct the global value chain by connecting the exchanges.

B. Compensation factors

Let's consider the "Order preparation" activity from our Amazon example that creates a "book in a box" from a book and a packing service. If the process aborts after the preparation 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 packings;
2) the book is now enclosed in a box; and
3) labor has been spent.

Intuitively, we imagine it possible to return the book to its previous state, by consuming 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 each have a handful of values: 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 such things as *types* of (REA) events and resources, and that determine *whether* an exchange or resource needs to be compensated and if so, *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.

1) Class factors.:

a) 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* [18].

Transactional exchanges involve the 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 the 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.

Conversional exchanges are 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. The resources involved in a conversion are characterized 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 or quantity. We propose to compensate a consumed resource by relinquishing an abstract economic resource called a *claim* that could be settled in the future by the same resource in equal quantity or by a resource of 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"[19]. In our Amazon example, the box has been consumed and is therefore compensated by a claim.

On the other hand, a used resource is one whose properties are altered by the conversion. Compensation then consists in to restoring these 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 state upon disassembly. Here again, the original provider of the part would need to be issued a claim for the lost value.

b) What is the type of the resource?: I walk into a food store, pick up a can of soup that has an expiry date. If I change my mind and decide to return the can, the store will likely take it back if this return takes place prior to the expiry date. If I return it *after* that date, the can will have lost all of its value. The can of soup belongs to the category of *perishable resources* whose value goes becomes null 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 expiry 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 three such (non-orthogonal) dimensions, explained below: *reversibility*, *perishability*, *depreciation* and *discreteness*.

- Reversibility: Shipping the book from *Amazon*'s warehouse to the 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 location of the book as being a reversible that we compensate by restoring its original property. Note that the reversibility is not intrinsic to a resource and depends on the properties being altered. 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 / depreciation / appreciation: As explained above, resource is *perishable* if it completely loses its value at a given instant. By contrast, *depreciation* corresponds to a *gradual* loss of value, notwithstanding the loss of value due to wear and tear. For example, the value of computers and cars depreciate over time, regardless of their usage but simply because of advances in technology and style. Conversely, some 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, and more specifically the time elapsed between the transaction affecting the resource and the process interruption.
- Discreteness: I need a four-foot long wooden beam. 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.

Considering again resource reversibility, we explained that this may be inferred from the set of properties that have been altered by the REA event. In this regard, we have refined the concept of properties and identified two dimensions against which resource properties may be classified: the *reversibility dimension* and the *value dimension*.

On the reversibility dimension we distinguish three types of reversible properties: *reversible*, *non-reversible* and *semi-reversible* properties. Reversible properties are those that may be set back to their original state (ex.: book's location) while non-reversible properties could not be recovered (ex.: the autographed book's property). Some properties are however reversible but the resource still suffers from unrecoverable damages (ex.: wear-and-tear). As an example we may consider screwing two pieces together; unscrewing is obviously possible but usually leaves stains on the screwed piece.

The value dimension, refers to the importance of a property to a given resource. We distinguish *primary* and *secondary* properties. Primary properties are those that directly contribute to the perceived utility of the resource in such a way that if the property is altered, the resource would have no value for the company (ex.: the *elasticity* of a spring, the *availability* of a plane seat, etc.). To the contrary, secondary resources do not contribute to the utility of the resource and the resource still preserves some value even if the property has been irreversibly altered.

c) Is it a gradual event?: Assume that I want to paint my living room a light 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 or 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 the hours of labor.

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.

d) Are there event costs?: Many business activities involve labor. Theoretically, an REA 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 complex models. As discussed earlier in section 4.1, 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*, as well as *its consumption* on the assembly line should be represented by an *economic event*. By contrast, labor involved in approving supplier invoices for payment by someone from accounts payable should not. This is *not* to say that these costs should not be accounted for. 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. Thus, if a completed activity is to be compensated, in addition to the explicit resources that it did consume (reversibly or not, atomically or not, etc.), we also need to compensate for the activity cost.

In our example, the labor used for packing the order or the banking transaction charges to collect the money from the customer may be accounted for as event costs.

e) Are there compensation-specific business policies and rules?: Going back to the Amazon example, 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 and specific policies or rules. Most notably:

1) should the resources that are at the heart of the process (book and cash in this example) be recoverable? And if so, 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;
2) how much should the customer be credited? 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, etc. 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').

We see business policies as general domain / organization 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 and b) the return shipping charges. However, they are not liable for the labor costs or the banking transaction costs. Further, the money is credited back to their credit cards.

2) Instance factors.: The *class factors* introduced above enabled us to determine the compensation activities required for a given exchange. 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.

a) What is the time of interruption?: Abortion of a business process may happen anywhere between the instant it begins 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 are 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.

b) 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 high as the price of the trip. Indeed, aborting a business process may involve losses sustained by the company in charge of the business process or an actor involved in the process. Some of these losses may be absorbed by the organization partially or fully 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 to be:

- 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).

C. 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 1) a compensation ontology organized in a layered architecture pattern that defines the core compensation concepts and their relations (see section 6) and 2) a set of compensation rules based on the factors identified above. 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 ask the analyst to classify its main entities (i.e. the resources, the events and the type of exchange) according to our ontology using concepts from her domain terminology. Based on the classification, we will be able to infer 1) what properties a given resource embodies and which are altered by the corresponding event, and 2) how to compensate for the exchange relying on our compensation rules. This will give us the set of REA compensation exchanges that we may in-turn compose into a compensation REA value-chain following the Saga approach.

Algorithm 1 Compensation modeling algorithm

```
Compensate(ValueChain v){
 ValueChain saga = v.reverseChain();
 List claimsToSettle;
 for(REAProcess p : saga){
  if(p instanceof Conversion){
   foreach Event {
    compensate each property alteration;
    add generated claims to claimsToSettle;
   }
  }
  apply compensation rules;
 }
 settleClaims(claimsToSettle);
}
```

However, the obtained compensation value-chain considers the whole process (i.e. as if the process ran entirely and we wanted to reverse it once it ended). As mentioned earlier, our 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-chain[1]. To this end, we need to establish a correspondence between the dynamic view of the business process (ex. expressed in BPMN) and the REA-value chain. We will rely on a tagging approach by asking the analyst to identify the activity - or the set thereof - corresponding to each REA event. Given this set of <Activity, Event> couples, we are able to determine which REA events occurred for each failure point and instantiate the compensation value-chain accordingly [2].

V. Compensation algorithm

In Algorithm 1 we present the basics of the implementation of our approach to mechanize the generation of business process compensation models. We assume, from now on, that the business analyst has modeled the REA value chain of the business process. The REA value model will be used as an input to our algorithm.

The first step, inspired by the Saga approach [10] (see Section 1), consists of reversing the value chain graph where, for each REA process, we generate an inverse REA exchange with inflow (resp. outflow) resources becoming outflow (resp. inflow) resource. Property alterations as well as right transfers are also reversed at this step; properties recover their initial values (for the time being) and rights are affected to their initial owner agent. The resulting *saga* returned will be a syntactically valid REA value chain.

[1]The instantiated value-chain is a subgraph of the full value-chain.

[2]As mentioned earlier, the REA value-chain abstracts away from the dynamic aspects of the business process. Thus, the order in which events are executed cannot be determined only by analyzing the value chain.

	Reversible	Semi-reversible	Non-reversible
Primary	Recover the property	- Recover the property; - Issue a claim on *the lost residual value*	- Issue a claim on *the resource*; - Delete the branch from the REA graph rooted on the altered; resource; - Ignore other properties;
Secondary			Issue a claim on the lost residual value

Table I
Compensation based on property alterations

Then we iterate through every exchange[3] of the saga and consider each *conversional exchange* in order to determine how to compensate for each altered resource. To this end, we analyze each property alteration and infer how to compensate for the resource relying on a set of rules summarized in Table I. As mentioned in the previous section, we classify each property against two dimensions: its reversibility and its primary value. Thus, we compensate by reversing each reversible and semi-reversible properties. Semi-reversible and non-reversible properties will involve a value loss sustained by the resource's provider agent. Hence, we compensate for the loss by issuing a claim in favor of the provider agent. We must however quantify the loss involved depending on whether 1) the whole resource will lose its value (non-reversible/primary property) or 2) if the alteration will only lessen the resource's value while still conserving its main purpose (semi-reversible and non-reversible/secondary properties). Thus, in case of a non-reversible and primary property the whole resource is lost. It is therefore useless to continue compensating for the same resource in subsequent exchanges. To this end, we delete the subgraph rooted on the lost resource.

Regarding transactional exchanges, we mentioned in section 4.2.1 that compensation simply involves restoring the exchanged rights to their original owners. Therefore, compensating for a transaction comes down to reversing the exchange; which is already handled by the *saga* mentioned above.

The last step in the for-loop in Algorithm 1 concerns the application of compensation rules which are derived from the compensation class factors introduced in Section 4.2.1. As an example of such rules, for a perishable resource, we would issue a claim on the lost value that is relative to the timeframe between the time of the exchange and the time of the process interruption.

Each claim issued in the previous steps is added to a list of claims to settle. As we explained previously, we consider that claims should be answered for by the accountable agent (i.e. the agent responsible for the process interruption). Claims are settled by introducing a transactional exchange having the claims as inflow resources in exchange for a settling

[3]Note that REA exchanges are partially ordered by the inflow/outflow relationship of their corresponding resources. A exchange E1 is smaller than an exchange E2 if E1 has an outflow resource being an inflow resource of E1. Iteration through exchanges is made with respect to this partial order.

outflow resource (usually *money*). It is to be noted, however, that claims settling will mainly depend on business policies (whether or not to hold a particular agent responsible for a given claim).

A. Implementation

In order to validate our approach, we implemented our process metamodel using the Eclipse Modeling Framework™ (EMF). EMF is a Java modeling framework that implements a core subset of the Meta Object Facility (MOF) API in a package called *ecore*. EMF enables us to serialize our models using the built-in XMI serializer.

Our meta-model is an extension of the REA business ontology (See Fig. 2). Classes presented in shaded grey boxes and their associations are the main REA concepts as introduced by Geerts and McCarthy [11]. The root class of our meta-model is the ValueModel class which will be the container class for the business process definition (i.e. the "happy path") and the compensation process. Therefore, a *BusinessProcessDefinition* represents the root object of an REA model and is composed of a set of ordered *Exchange*s. An Exchange is an abstract concept that could be specialized either by a *Transaction* or a *Conversion*. Exchanges are linked by a containment relationship to two sets of REA *Events* (decrement events and increment events). Each event is in turn linked to a *Resource* or a *Claim*. To this end, we generalized *Resources* and *Claims* under the *Valuable* abstract class, thus associating *Event*s with a Valuables. The partial order of the REA Exchanges are maintained by a *predecessor/successor* bi-directional attribute linking each Exchange to the set predecessor/successor with regards to resource dependencies.

Central to our approach are the concepts of *Rights* and *Properties* of resources eventhough the REA ontology does not give them such treatment and considers them as simple resource attributes. *Rights* apply to both *Resources* and *Claims* and are therefore linked to their common ancestor: the *Valuable* class. By contrast, *Properties* apply only to economic *Resources*. REA Events are linked to the set of altered *Properti*es and exchanged *Rights* through the qualified associations[4] *RightTransfer* and *PropertyAlteration*, respectively.

We have implemented a set of tools based on the metamodel presented in Fig.2. First, we developed a graphical modeling tool called *REAModeler* on top of the Eclipse Graffiti framework[5]. REAModeler enables an analyst to design the value chain of his business process as a graph of REA exchanges. We have also implemented a tabular REA model editor on top of the code generated by the *EMF.edit* framework, as well as the set of rules and axions dictated by the REA ontology into EMF.edit generated code in order to ensure syntactic and semantic correctness.

We implemented our algorithm in a tool called *REACompensator*. In short, REACompensator takes an REA model (serialized in the XMI format) and, following Algorithm 1, produces an REA model of compensation process both in XMI and as visual representation.

[4]These appear as regular classes in the meta-model on Fig.2 because qualified associations are not supported by EMF.

[5]http://www.eclipse.org/graphiti/

VI. Towards an ontology of economic resources and Events

In the previous sections, we presented our approach to model REA compensation models going from a value chain of the so-called "happy path" of the business process. We proposed a set of factors enabling us to infer how to compensate for a given business activity/REA exchange. This approach relies on the ability of the designer to identify the type of resources, the properties and rights embodied by each resource, and the type of events involved in the process. Going back to our Amazon example, the book should be identified as a resource having a *location* property, a *packed* property and an *ownership* right. The packing event - where the book gets enclosed in a box - must specify the altered property and be classified as a non-gradual event requiring event costs (to account for the labor involved). Each of the aforementioned properties must be classified according to their reversibility (*reversible*, *semi-reversible*, or *non-reversible*) and their primary value (*primary* or *secondary*). In our example, the *packed* and the *location* properties are reversible and secondary; i.e. we can unpack a book and return it to the warehouse and both properties do not constitute the primary value of the book. On the other hand, once we pack the order the *Box* resource looses its *physical integrity* property, which is a primary and non-reversible property.

Classifying each concept in this manner seems cumbersome and time consuming. As a matter of fact we found it to be a tedious process that does not serve our purpose in helping the business analyst design compensation processes. To make things worse, the terminology and cognition process involved in order to model the full-fledged value model defining the set of properties/rights and alterations does not seem appropriate from an analyst standpoint. Hence, we should offer the analyst a framework that only involves describing the process from a business perspective while our approach automates - as much as possible - the identification of resources (with appropriate rights and properties). In other words, the analyst would only specify to which type the resource "Book" belongs (say, *Printed Literature*) and to which type the event "Ship the order" belongs to (say, *Ground Shipping*). Given this classification, we should be able to determine that the book has a *location* property that is altered by the "Ship the order" event.

Luckily, we believe that we can go further down in our generalization and identify subconcepts of similar REA economic concepts. Consider, for example, a book, a car and a computer. One of the characteristics these objects have in common is that they are located in a given physical space at a given time and may be transported from one place to another. In fact, these resources may be grouped as the concept of *corporal resources* sharing a *location* property among others (ex.: *physical integrity*).

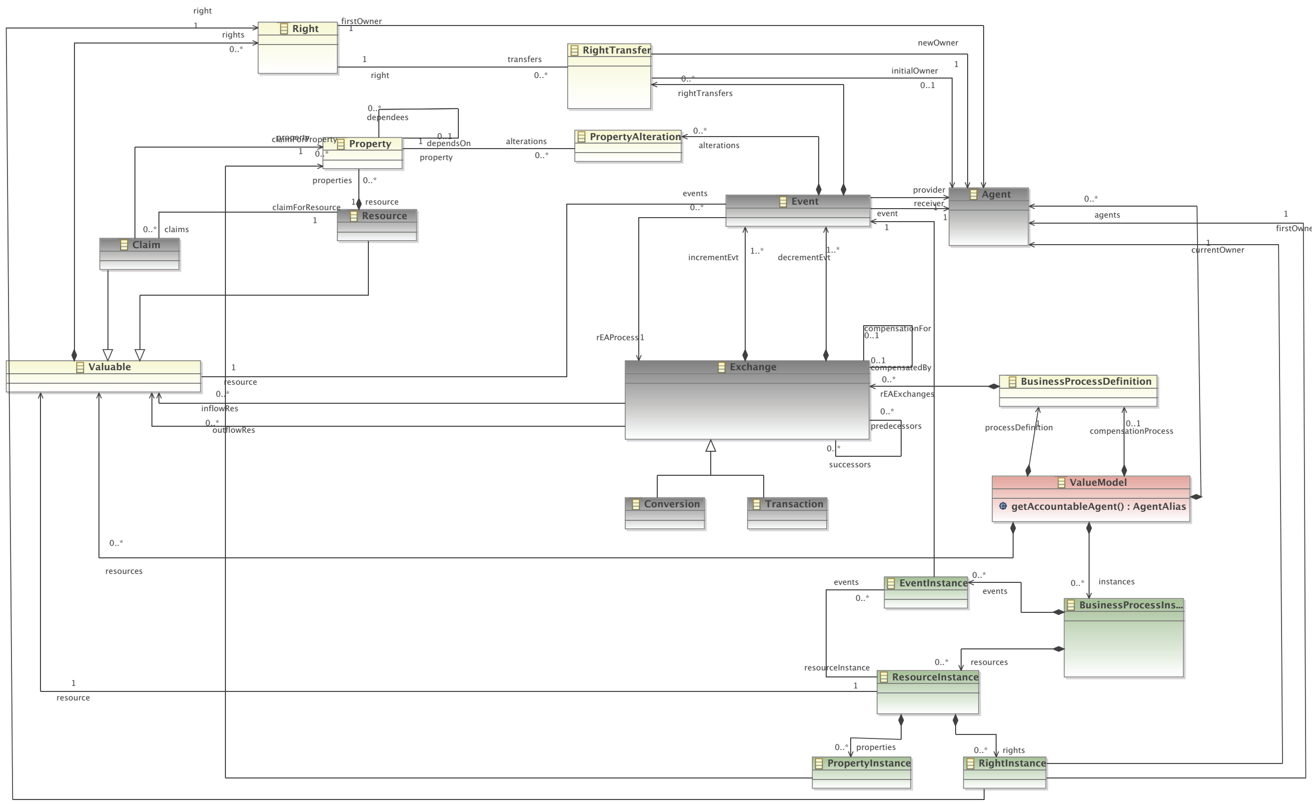

Figure 2. Meta-Model of our Implementation

This classification should not be an orthogonal (ie. taxonomy like) classification. Take the book as an example; the same book could be sold as a printed version or as an electronic version for on-screen reading. While the printed version shares the properties of tangible resources, the electronic version does not. However, both have an *intellectual right* belonging to the book's author and a *divulgation property* (ie. even if a book is returned to Amazon unaltered, the customer may have read it before returning it and should therefore pay for it). Many other resources that we could regroup under an *intellectual resources* concept share these properties as well as being part of the *tangible resources* group - from which we may cite a movie, a computer software or an electronic device.

We could apply the same reasoning to REA events or to resources properties. A shipping event shares many specifics whether it is done by air, ground or sea; they all alter a resource's location property. However, if the company relies on a shipping company, it loses the *control* over the resource once it is handled by the shipping company; this is not the case if the company does self-shipping for domestic deliveries. Therefore, we can envision a class of *shipping events* along with a class of *subcontracting services*; shipping by a shipping company belongs to both while self-shipping only belongs to the former.

Going this road, one might ask where to stop going down into the abstraction hierarchy? We believe that it will always be room for more specialization, and that we can't possibly record all existing variations. Furthermore, there are inherent variations between business domains that one must take into account. Considering again the printed book example. Its *physical integrity* is of primary importance to Amazon while it is a secondary property from the perspective of a paper recycling company. Therefore such a classification should be high level enough to be valid for every industry and application domain, while it should be adaptable to particular business domain and enterprise needs.

Therefore, we propose to consider a four-layered ontology as illustrated in Fig. 3 where each layer is an abstraction level. The *core ontology* constitutes the basic ontology of our approach and includes the REA concepts as well as our compensation factors. The *high level* ontology will include generic industry neutral variations of the concepts we are planning to engineer relying on a comprehensive economic resources catalogue (ex.: The United Nations Central Product Classification[6]) and a formal concept analysis. The *business domain* ontology constitutes the industry specific shared among the same business domain while the *corporate* ontology would embody enterprise specific variations. We propose to implement this ontology architecture as an OWL-DL ontology that will enable us to achieve:

[6]See http://unstats.un.org

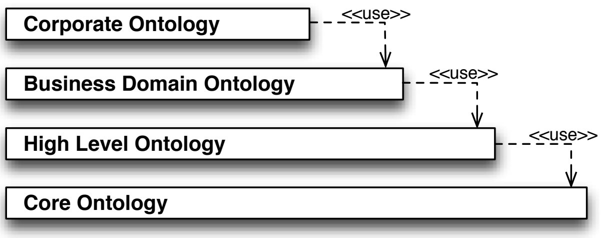

Figure 3. Architectural View of the Proposed Compensation Ontology

- a greater separation of concerns implying better reusability;
- scalability and variability to industry specifics;
- a greater usability with terminology closer to the analyst; and
- reasoning capabilities letting us 1) insure semantic soundness of the concepts (ex.: a given property cannot be reversible and non-reversible), 2) infer property alterations from the classification of a given resource and a given event, and 3) discover fortuitous classifications.

VII. Conclusion and Future Work

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 in a systematic fashion, hence providing the business analyst with a much-welcomed guidance.

We also proposed a methodology along with an implementation algorithm in order operationalize our approach, as well as a compensation ontology architecture that enables us to improve the usability of our approach. Our implemented prototype allowed us to ensure the syntactic validity of the resulting compensations value chains (i.e. valid REA models) and to perform a preliminary validation of their semantic soundness. However more thorough validation is still to be performed. More precisely, two aspects of our approach are to be validated: 1) the completeness and expressivity of our ontology, and 2) the semantic soundness of the results by relying on a group of business analyst experts.

Although this work is still at an early stage, this paper establishes our ground ideas and our major directions. The engineering of our compensation ontology is an ongoing project that follows an iterative process.

References

[1] B. Andersson, M. Bergholtz, A. Edirisuriya, T. Ilayperuma, E. Dubois, S. Abels, A. Hahn, B. Wangler, and H. Weigand. Towards a Reference Ontology for Business Models. In *25th International Conference on Conceptual Modeling*, pages 482–496. Springer, 2006.

[2] P. Antunes. 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.

[3] L. Bocchi, P. Ciancarini, and D. Rossi. Transactional aspects in semantic based discovery of services. In *Coordination Models and Languages*, pages 283–297. Springer, 2005.

[4] A. Boubaker, H. Mili, Y. Charif, and A. Leshob. Towards a Framework for Modeling Business Compensation Processes. In *14th International Conference on Business Process Modeling, Development, and Support (BPMDS'13)*, pages 139–153, 2013.

[5] A. Boubaker, H. Mili, A. Leshob, and Y. Charif. A Value-Oriented Approach to Business Process Compensation Design. In *2nd IEEE Int. Conference on Information Technology and e-Services*, 2012.

[6] Michael Butler and Carla Ferreira. An Operational Semantics for StAC. In *6th International Conference*, Pisa, 2004. Springer-Verlag.

[7] CYCorp. CYC Knowledge Base, http://cyc.com/cyc/.

[8] U. Dayal, M. Hsu, and R. Ladin. Business Process Coordination : State of the Art , Trends , and Open Issues. In *27th Very Large Databases Conference, VLDB 2001*, pages 3–13, Roma, 2001.

[9] C. Eisentraut and D. Spieler. Fault, Compensation and Termination in WS-BPEL 2.0. In *WS-FM*, pages 107–126. Springer, 2009.

[10] H. Garcia-Molina and K. Salem. Sagas. In *Proc. of the international conference on Management of data*, pages 249–259. ACM, 1987.

[11] G. Geerts and W. McCarthy. Modeling Business Enterprises as Value-Added Process Hierarchies with Resource-Event-Agent Object Templates. Springer-Verlag, 1997.

[12] G. Geerts and W. McCarthy. The Ontological Foundation of REA Enterprise Information Systems. In *American Accounting Association Conference*, Philadelphia, 2000.

[13] G. Geerts and W. McCarthy. An ontological analysis of the economic primitives of the extended-REA enterprise information architecture. *International Journal of Accounting Information Systems*, 3(1), 2002.

[14] C Ghidini, C.D Francescomarino, M Rospocher, P Tonella, and L Serafini. Semantics-Based Aspect-Oriented Management of Exceptional Flows in Business Processes. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(1):25–37, 2012.

[15] J. Gordijn, H. Akkermans, and H. van Vliet. Business Modelling is not Process Modelling. In *1th International Workshop on Conceptual Modeling Approaches for e-Business*, pages 40–51, 2000.

[16] J. Gordijn and R. Wieringa. A value-oriented approach to e-business process design. In *Adv. Information Systems Eng.* Springer, 2003.

[17] 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.

[18] P. Hruby. *Model-Driven Design Using Business Patterns*. Springer, 2006.

[19] 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.

[20] H. Mili, R. Godin, G. Tremblay, and W. Dorfeuille. Towards a Methodology for Designing Compensation Processes in Long-Running Business Transactions. In *MCETECH*, pages 137–148, Montreal, 2006.

[21] OASIS. WS Business Process Execution Language (BPEL), 2007.

[22] OMG. Business Process Model and Notation (BPMN). OMG, 2011.

[23] A. Osterwalder. *The Business Model Ontology*. PhD thesis, Université de Lausanne École des Hautes Études Commerciales, 2004.

[24] M. E. Porter. Competitive advantage: Creating and Sustaining Superior Performance. New York, New York, USA, 1985.

[25] J. Vom Brocke, J. Mendling, and J.C. Recker. Towards a value-oriented approach to business process modelling. In *Information Systems*, pages 12–16, 2008.

[26] E. S K Yu. Models for supporting the redesign of organizational work. In *Proceedings of conference on Organizational computing systems*, 1995.

[27] S. C. Yu. *The Structure of Accounting Theory*. The University Press of Florida, 1976.

[28] J. Zdravkovic and T. Ilayperuma. A Model-Driven Approach for Designing E-Services Using Business Ontological Frameworks. *Enterprise Distributed Object Computing Conference*, pages 121–130, 2010.