# Designing Value-Based Inter-organizational Controls Using Patterns

Vera Kartseva[2], Jaap Gordijn[1], and Yao-Hua Tan[2]

[1] VUA, Faculty of Sciences, De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
gordijn@cs.vu.nl
[2] VUA, Faculty of Economics, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands
{kartseva,ytan}@feweb.vu.nl

**Abstract.** Networked organizations, consisting of enterprises who exchange things of economic value with each other, often have participants who commit a fraud or perform other actions, not agreed in a contract. To explore such opportunistic behavior, *and* to design solutions to mitigate it, we propose the $e^3$*control* approach. This approach takes the valuable objects, which are exchanged between enterprises, as a point of departure, and proposes a control patterns library to find solutions for various types of opportunistic behavior in network organizations. The practical use of the patterns is illustrated by a case study in the field of renewable electricity supply in UK.

**Keywords:** Inter-organizational control, value network, control pattern.

## 1 Introduction

Organizations increasingly organize themselves as networks: Collections of enterprises that jointly satisfy a complex consumer need, each utilizing their own specific expertise, products, and services [1]. These networks are enabled by innovative technologies such as web-services, allowing for timely coordination of enterprises. Due to this innovation, new networks emerge, for instance in the field of energy supply, Internet service provisioning, or digital content [2,3,4,5].

Techniques, such as goal- and value modeling [6,3] play an important role in the early requirements engineering phase for information systems supporting and enabling these networks. For instance, in [3] we report how to explore an IT-enabled network of enterprises from a *business value* perspective using the $e^3$*value* technique, and in [4] we explain how such exploration can be done in combination with multi-actor *i** goal analysis. In brief, $e^3$*value* analyzes *what* objects of *economic value* are exchanged between enterprises, and *what* actors request in return for these objects (usually other objects of value). So, the $e^3$*value* approach is an *early* requirements engineering technique with the aim to understand business *value* requirements in a model-based way. Understanding of the network's business value requirements provides a starting point for analyzing requirements of information systems.

The $e^3$*value* approach deliberately supposes that enterprises behave *honest*, as otherwise resulting models would soon become rather complex, and disturb executive decision making. In $e^3$*value* , 'honest behavior' refers to actors who - if they obtain an

K. Lyytinen et al. (Eds.): Design Requirements Workshop, LNBIP 14, pp. 276–301, 2009.

object of value from their environment - *always* provide another object of value to their environment in return, as a economic reciprocal exchange. To our experience, it is initially already sufficiently difficult to design a network under such perfect-world conditions. The next step however is to assume *opportunistic* behavior of enterprises. To this end, we have proposed $e^3$*control* [7]. As *i** and $e^3$*value* , the $e^3$*control* technique is a requirements engineering technique to understand the context of multi-actor information systems. The $e^3$*control* models have close similarities with $e^3$*value* models with one important difference: An $e^3$*value* model supposes that each enterprise in a network behaves honestly, or *ideally*, whereas an $e^3$*control* model allows opportunistic, or *sub ideal*, behavior. An $e^3$*control* model however still focuses on the *value objects* exchanged.

To discourage opportunistic behavior, *control mechanisms* can be applied. Such controls are often *value-based*, e.g. penalties and incentives, or reconciliation of valuables. Also, controls may require specific business processes, or rely on information technology (e.g. security protocols). As processes are significantly controlled and executed by IT, understanding of these controls are important for the IS requirements.

To design controls in networked enterprises, $e^3$*control* can be used as a general framework, but the design process still requires a vast amount of knowledge on organizational controls *themselves*. To make this knowledge available within $e^3$*control* , we propose a library of *control patterns*, which describes organizational controls for networked organizations. These patterns are the main contribution of this book chapter. The $e^3$*control* approach and the supporting patterns are unique because they are grounded in an *economic value* perspective, while connecting properly to the processes putting the controls into operation. It is the transfer of valuable objects in a network that has to be controlled in first place. This contrasts to existing process-only approaches for controls (see e.g. [8]), or even EDP-auditing (e.g. [9]).

The controls have been collected from agency theory (e.g. [10]), internal control theory (e.g. [8]) and management control theory (e.g. [11]). Examples of such organizational controls are *detective* controls such as monitoring and verification (e.g. quality control, reconciliation of accounting records with material reality), but also *preventative* controls, such as economic incentives and penalties. In addition to literature, the patterns are based on four real-life case studies we performed in the drinks industry [12], international trade [13], the entertainment industry [14], and electricity supply industry [15]. In this chapter, we elaborate on the latter case study.

This chapter first introduces the notion of value-based controls for networked value constellations (Sec. 2). Then, we present three of our control patterns in detail in Sec. 3 as well a summary of the rest of the patterns. In Sec. 4, we show the three patterns can be practically applied in a case study. Finally, Sec. 5 presents our conclusions.

## 2 Value-Based Design of Controls for Networked Constellations

We illustrate the design of controls for networked constellations by a small example (see also Fig. 1). For this example, we suppose that someone buys a product or a service from a seller, and the seller has to pay Value Added Taxes (VAT) to the Tax office. In $e^3$*control* , we follow three subsequent steps to analyze networks for sub ideality.

**Step 1: Elicit and Model the *Ideal* Network Using an *e*$^3$*value* Model.** We use the *e*$^3$*value* technique to first understand the network, assuming that all actors would be behave ideally, or, in other words, would be honest. The *e*$^3$*value* technique allows to represent which enterprises in a network exchange which objects of economic value with which other enterprises. Fig. 1 exemplifies a buyer obtaining goods from a seller and offering a payment in return. Due to the law, the seller must pay a value-added tax (VAT). This can be conceptualized with the following *e*$^3$*value* constructs.

*Actors*, such as the buyer, seller, and the tax office are economically independent entities. Actors transfer *value objects* (payment, goods, VAT) by means of *value transfers*. For value objects, some actor should be willing to pay, which is shown by a *value interface*. A value interface models the *principle of economic reciprocity*: only if you pay, you can obtain the goods and vice versa. A value interface consists of *value ports*, which represent that value objects are offered to and requested from the actor's environment. Actors may have a *consumer need*, which, following a path of dependencies will result in the exchange of value objects. Transfers may be *dependent* on other transfers, or lead to a *boundary element*. In the latter case, no transfers are considered anymore.

The important point here is that an *e*$^3$*value* model *by definition* supposes that all actors behave ideally. This is reflected by the explicit notion of 'economic reciprocity': All agreed transfers are required to happen, or should not happen at all. Performing this step results in understanding of the valuable objects that should be transferred, and thus which objects should be subject to control.

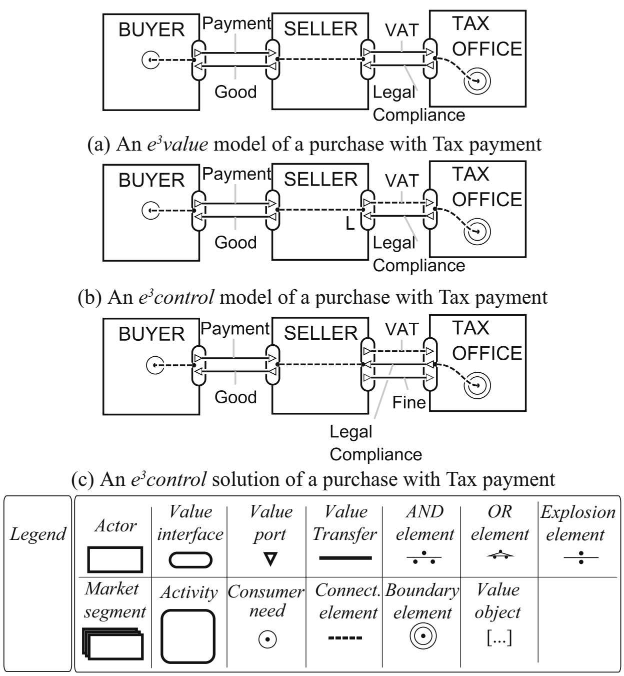


**Fig. 1.** Example of an *e*$^3$*value* model of a purchase with tax payment

**Step 2: Analyze *Sub ideality* in a Network Using an $e^3$*control* Model.** In reality, actors ideal often behave sub-ideally: they commit a fraud or make unintentional errors. In $e^3$*control* , these situations are modeled by *sub-ideal value transfers* [7]. These are graphically represented by dashed arrows, and can indicate different risks: e.g. actors not paying for goods, not obtaining the goods, or obtaining wrong goods. For example, Fig. 1 (b) models a situation that the seller does not pay VAT. ‘L’ is a *liability token* [7], assigned to the responsible actor for the sub-ideal value transfer, here the seller.

**Step 3: Reduce *Sub ideality* by Adding Controls.** We now add control mechanisms that *reduce* the control problem. Hardly any control mechanism can *remove* a control problem completely. A combination of mechanisms, so-called ’control mix’, is usually required [8]. For example, Fig. 1 (c) introduces fining. In case the seller does not pay taxes, he is charged with a high fine. The fine is modeled as a value object, transferred from the seller to the tax office. As can be seen by the dashed transfer, the model is still sub-ideal, but at least the Tax Office receives adequate compensation if the seller behaves sub-ideally, and if such behavior is detected.

## 3 Control Patterns

The design process in Sec. 2 is general and requires quite some design knowledge on well accepted control problems and solutions. To increase the usability of $e^3$*control* , it is therefore important to bring in this *accepted* knowledge; therefore we propose a series of *inter-organizational* control patterns (cf. [16]). These patterns and their use is the main contribution of this chapter.

### 3.1 Elicitation and Representation of Control Patterns

**Elicitation Method.** Pattern development usually consists of the *identification*, *collection* and *codification* of existing knowledge [17]. The PattCaR method developed by [18], suggests more specific guidelines for patterns elicitation: (1) analysis of the *domain* and *context* of the patterns, (2) definition of a *vocabulary*, (3) a thorough domain analysis and extraction of *patterns candidates*, (4) a collection of several examples of each *pattern candidate*, (5) *encoding* of patterns by modeling the examples and performing a commonality-variability analysis, and (6) a description of *relations* between patterns.

**Domain of Controls for Networks.** There are several theories that attempt to describe the domain of controls, including accounting control theory (e.g. [19]), management control theory (e.g. [11]), and agency theory (e.g. [10]). There is also specific work on inter-organizational controls, such as [20,21].

In this chapter, we consider controls in terms of the principal-agent framework. This framework makes a distinction between a *primary* actor (or principal) and a *counter* actor (or agent). The counter actor behaves *sub-ideally* and the primary actor wants to reduce the loss caused by such behavior. The agency theory describes several control problems and mechanisms to mitigate sub ideal behavior, namely *Screening*, *Signaling*, *Monitoring*, and *Incentives*. The Screening and Signaling mechanisms are used to counter the *hidden information* problem. This occurs if the primary actor does not have enough knowledge about the counter actor, increasing the risk that the counter actor will perform his activities in a sub ideal way. The screening and signaling mechanisms recommend checking the counter actor's abilities and characteristics before signing a contract with him. The Monitoring control is used to counter the *hidden action* problem, which means that the counter actor performs his activities in a sub ideal way. The Monitoring mechanism recommends verifying the counter actor's performance before rewarding him. If monitoring is difficult or costly, the counter actor can be stimulated to behave ideally by Incentives which can be positive (reward) or negative (punishment).

Two additional inter-organizational controls are described in [20]. The Commitment Evidence control applies to a situation in which the counter actor inappropriately denies his commitment to the primary actor. The Execution Evidence control addresses a counter actor inappropriately claiming that the primary actor executed his activities sub ideally. Both commitment evidence and execution evidence controls require the creation of evidence that can be used in (legal) disputes against the counter actor. These two controls stem up from the audit trail principle of the internal control theory.

Usually, a distinction is made between *ex-ante* controls, i.e. controls executed before the contract between two actors is settled, and *ex-post* controls, i.e. controls executed after the contract is settled. The Screening, Signaling and Settlement of incentives are ex-ante controls, while Monitoring, Commitment evidence, Execution evidence and Execution of incentives (actual rewarding or punishment) are ex-post controls. A further distinction can be made between *contractual* controls and *procedural* controls. Contractual controls employ value-based mechanisms to stimulate the counter actor to behave ideally. Procedural controls employ process-level mechanisms to repressively prevent or detect the counter actor's sub ideal behavior. With the exception of incentives, all the groups of controls considered here are procedural.

**Pattern Representation and Vocabulary.** Cf. [22], a pattern has the following structure: *name*, *context*, *problem*, *solutions*. We consider a *control pattern* as a description of generic and re-usable control mechanism for a recurring control problem. We describe the context, problem, and solution slots by taking a business value ($e^3$*value* or $e^3$*control* ) and business process (UML activity diagrams [23]) perspective, thereby following the principles of multi viewpoint requirements engineering [24]. Examples of patterns can be found in the appendix of this chapter.

We use the following vocabulary[1] to describe a control pattern. There are two actors, a *primary actor* and a *counter actor*. From a value perspective, the primary and counter actors exchange value objects: the primary actor transfers a *primary value object* (PO) to the counter actor, and the counter actor transfers a *counter value object* (CO) in return. From a process perspective, the exchange of PO corresponds to execution of a *primary activity*, and the exchange of CO corresponds to execution of the *counter activity*. These activities can also be collections of multiple operating activities.

Sub ideal behavior and, consequently, sub ideal transfers are defined from the point of view of the primary actor, who is the *principal*. Sub ideal behavior is executed by the counter actor, who is the *agent*. The primary actor expects the counter actor to

[1] The terminology is inspired by [25].

behave sub ideally with respect to the execution of the counter activity. The result of this opportunistic behavior is a sub ideal transfer of the CO. Obviously, actors can all play the role of principal or agent, depending on the perspective taken.

Furthermore, based on [25,26,8], we have also developed a vocabulary of *control activities* and *control principles*, which form the building blocks for control mechanisms. The activities include e.g. *verify*, *witness*, *testify*, and *authorize*. The control principles are normative rules of relations between activities, objects and actors [26]. As an example of such a rule, segregation of duties requires the party who executes a verification activity to be independent and socially detached from the party who executes the activity being verified. Also, the ordering of activities is motivated by control principles.

**Extraction of Pattern Candidates.** Due to lack of space, we can not present the pattern extraction process itself (see [27] for more details), rather we focus on a few considerations with respect to this process, and the results.

A first consideration is that our candidate patterns should represent a unique combination of problems and solutions. To do so we require that (1) two different control problems should fall in two different patterns, (2) two different control mechanisms for one control problem should be represented by two different patterns.

A second consideration is that, in line with the approach of $e^3$*value* [3], we want the patterns library to be lightweight. This means that the fewer patterns we have to describe all the considered controls, the better it is. To achieve this, we abstract from domain-specific details, which are present in the internal control theory. Firstly, we do not consider any specific roles of actors, such as a supplier or a customer. We describe a transaction in terms of the principal-agent framework. So, we distinguish between a *primary* and a *counter actor*. The counter actor behaves *sub ideally* and the primary actor wants to reduce the loss caused by this sub ideal behavior. The actors can delegate their activities to *trusted parties*. Secondly, we do not differentiate controls if they only involve different types of documents, e.g. a purchase order or a contract.

Our domain analysis resulted in Screening, Signaling, Monitoring, Commitment Evidence, Execution Evidence and Incentives controls. We call these *sub domains* of the control domain, and we consider these to be a good starting point for elicitation of the patterns. We compare the sub domains with each other, and re-group them to select the unique problem-solution pairs. These pairs will form the control patterns. Effectively, this process is about commonality-variability analysis of the domains.

*Screening and signaling.* The screening and signaling sub domains mitigate the same control problem of hidden characteristics, however they are different control mechanisms in terms of solution. In screening, the primary actor verifies the *activities* of the counter actor. In signaling, the primary actor verifies indirect *signals* and not the activities. Such signals have a historical correlation with the expected performance of the actor in the future. The difference between screening and signaling is that screening is based on information collected by direct observation of the counter actor's activity, while signaling is based on information collected from a third party. However, if we ignore delegation, the difference between the two mechanisms disappears. So, screening and signaling describe the same control problem and the same control mechanism. This results in one pattern, called *Partner Screening*, in which the primary actor screens his partner.

*Screening and monitoring.* Screening and monitoring mitigate the same control problem: The sub ideal execution of contractual agreements by a counter actor. On the other hand, the screening control also considers the condition of hidden characteristics, and, therefore, employs an ex-ante control. As explained before, screening verifies activities performed by a counter actor in the past with the assumption that the past resembles the future. The monitoring mechanism does not carry the assumption of hidden characteristics. As a result, it is an ex-post control performed in the context of an existing contract and it suggests verification of activities under the contract. Because of this difference, screening and monitoring controls result in different patterns. The pattern for monitoring control is called *Execution Monitoring*, meaning that the execution of the counter actor's activities is monitored by the counter actor.

*Positive incentives and negative incentives.* Incentives may be positive or negative. Positive incentives stimulate the counter actor to behave ideally by rewarding him while the negative incentives do the same by punishing the counter actor. These two mechanisms require different changes in $e^3$*value* models. Namely, positive incentives can be created by adding an incoming value object to the counter actor in the case of ideal behavior, while negative incentives can be created by adding an outgoing value object to the counter actor in the case of sub ideal behavior. We therefore put positive and negative incentives into two different patterns. Positive incentives are described in the *Incentive* pattern and negative incentives are described in the *Penalty* pattern.

*Monitoring and incentives.* Both the monitoring and the incentive mechanisms mitigate the same control problem - that of a sub deal execution of contractual agreements by the counter actor. On the other hand, the two controls are different, as the former is a procedural control, while the latter is a contractual control. Incentives also require monitoring mechanisms to prove when the reward or punishment has to be issued or not. Such proof can be modeled with the pattern *Execution Monitoring*, while the actions related to the punishment are a part of the Penalty pattern. In fact, the incentive mechanism is a *variation* of the monitoring mechanisms.

*Execution evidence and commitment evidence.* The execution and commitment evidence controls involve the same activity: The counter actor should provide the primary actor with an evidence document, which can later be used in a legal dispute. As a commitment evidence control, the evidence document contains a testimony of the counter actor's commitment to a future transaction with the primary actor. This evidence document is normally represented by a contract. For the execution evidence control, the evidence document contains the counter actor's testimony that the primary actor executed his obligations as stated in the contractual agreement. An example of such an evidence document is a receipt given as proof of payment.

So, the processes behind these mechanisms are technically the same, only the role of the evidence document is different. This is because the two controls address different control problems. In addition, the commitment evidence control is executed ex-ante, while the execution evidence control is executed ex-post. For these reasons, we describe these two controls in different patterns.

The pattern for the commitment evidence control is described in the *Proper Contracting* pattern. As the name implies, the control provides guidelines on a correct contracting process. The pattern for the execution evidence control is called *Execution Confirmation*. The name reflects the essence of the mechanism, which is to provide evidence about the execution of primary activities.

*Delegation.* The agency theory and the work of Bons consider controls in a relationship between two actors: A primary and a counter actor. In addition, Bons also considers some network aspects. He considers networks as being derived from a two-actor network as a result of the *delegation* of activities to other actors. We do not include delegation issues in our patterns (e.g. as variants), but rather factor it out. First of all, a very large number of different networks (and so patterns) can be formed by using delegation, as e.g. third parties can further delegate activities. If we also consider that actors not only execute primary and counter activities, but also other activities associated with controls (e.g. reconciling, witnessing, verifying), even more possibilities for delegation arise. Furthermore, inclusion of delegation in our patterns would describe similar control problems and mechanisms and only differ in the way activities are delegated, which, strictly speaking, is not a control issue.

Therefore, we describe each pattern only for a transaction between a primary and a counter actor. In order to describe delegation situations, we introduce *delegation patterns* (see [27] and http://www.e3value.com/e3family/e3control/patterns), which provide guidelines on how the two-actor model of a control pattern should be properly changed into the multi-actor model. They ensure that when an activity is delegated, the controls prescribed by the control pattern, are still in place.

**Examples of Patterns.** In addition to the literature review, elicitation and validation of usability of the patterns was done through a series of case studies.

- *Beer Living Lab.* The case study is about an excise collection procedure inside and outside the EU. This case study [12] contains the patterns Execution Monitoring, Partner Screening Certification, and includes multiple situations when activities by a principal or an agent are delegated to other trusted parties.
- *Dutch health care services.* The case study is about processes in Dutch health care system. It contains the patterns Execution Monitoring and Certification and provides a test of patterns for a non-profit sector [15].
- *International trade.* The case study is about a bill of lading procedure in international trade. It contains the patterns Execution Monitoring, Proper Contracting and Execution Confirmation. It demonstrates the application of patterns in a complex situation when control mechanisms in a network are conflicting [13].
- *Internet Radio.* The case study about an Internet service of free radio broadcasts. It contains, by applying the Execution Monitoring pattern, a mechanism for controlling how many listeners the radio station has, using data collected from distributed listeners [14].

**Control Patterns.** Based on the mentioned literature and case studies, we have identified the following control patterns: Partner Screening, Proper Contracting, Execution Monitoring, Execution Confirmation, Incentive and Penalty. The patterns are summarized in Table 1; for three of these patterns, the $e^3$*value* , $e^3$*control* , and UML activity models are shown in the Appendix of this chapter.

**Table 1.** Library of Control Patterns

| *Name* | *Control Problem* | *Solution* |
|---|---|---|
| Partner Screening | Counter Actor executes his commitment sub ideally. | Primary Actor verifies credentials of Counter Actor before making any commitments |
| Execution Monitoring | Counter Actor executes his commitment sub ideally. | Primary Actor verifies Counter Actor's execution of the commitment, before executing own commitments |
| Incentive | Counter Actor executes his commitment sub ideally. | Primary Actor provides a reward for the ideal execution |
| Penalty | Counter Actor executes his commitment sub ideally. | Primary Actor provides a punishment for the sub ideal execution |
| Proper Contracting | Counter Actor denies to have made a commitment to Primary Actor | Counter Actor provides an evidence document, which confirms his commitment |
| Execution Confirmation | Counter Actor denies that Primary Actor executes commitments ideally, and refuses to execute his commitments in return, or requires a compensation for executing his commitments | Counter Actor provides an evidence document, which confirms that Primary Actor executes his commitment ideally |

# 4 Case Study: Renewable Energy in the UK

## 4.1 Introduction

One of the industries with interesting and complicated control problems is the renewable electricity industry. To comply with international environmental agreements, such as the Kyoto protocol, governments must ensure that a sufficient amount of electricity is produced with technologies that do not use fossil fuels. Examples of $CO_2$-friendly technologies are wind turbines, photovoltaic panels and hydro generators. Such technologies are called *renewable* or *green* technologies. At present, these technologies require high initial investments, meaning that the price of green electricity is higher than the price of electricity produced in the conventional way using fuel-based technologies [5]. Many government regulated schemes have been implemented to make renewable technologies commercially more attractive, e.g. tax cuts and subsidies on initial investments, premiums for generated electricity, etc. In this chapter we examine more closely one such scheme, which was implemented in the United Kingdom (UK).

In the UK, the Renewable Obligation (RO) regulation law was introduced to stimulate the generation of renewable electricity. The first Renewable Obligation regulation in the UK came into force in April 2002. The law places an obligation on electricity *suppliers*, licensed to supply electricity in the UK, to source a certain proportion of electricity from renewable sources [28]. When the regulation was introduced, this portion constituted 10% of the total supply of a UK supplier. In 2006/07 a UK supplier is obliged to generate 6.7% of its supply from renewable sources.

Suppliers prove that they meet their obligations by presenting Renewable Obligation Certificates (ROCs), each representing one Mega Watt/hour (MWh) of produced renewable electricity output. ROCs can be acquired by suppliers from *producers* of green electricity. The producers get ROCs from a government agency, the Office of Gas and Electricity Markets (*Ofgem*) for each MWh of renewable electricity output they produce. In addition, Ofgem maintains a register of all ROCs it has issued.

The suppliers must therefore provide ROCs as evidence of how much MWh of green electricity they have supplied. If a supplier does not have sufficient ROCs to cover his obligation, he must make a deposit into a *buy-out fund*. The buy-out fee is a fixed price per MWh shortfall and is adjusted in line with the Retail Prices Index each year. Premiums from the buy-out fund are paid back to suppliers in proportion to how many ROCs they have presented.

In this case study we apply the $e^3$*control* methodology and patterns to understand and find controls. We reverse engineer the ultimate ROC-scheme, by means of the patterns, to show why this scheme is needed from the control perspective. For example, we illustrate how the patterns explain the necessity of introducing ROCs.

We first assume that ROCs do not exist yet. We explain the control problems that may occur in the network. We explicitly take the government's point of view and only describe the problems as perceived by the government, which is represented by Ofgem. Then, step by step, we design the ROC scheme by applying the patterns. As a result, we will demonstrate that the ROC scheme can be explained by means of $e^3$*control* patterns.

The case study material is based on participation in the EU BusMod project [5], as well as the Ofgem web site (www.ofgem.co.uk), including [28,29].

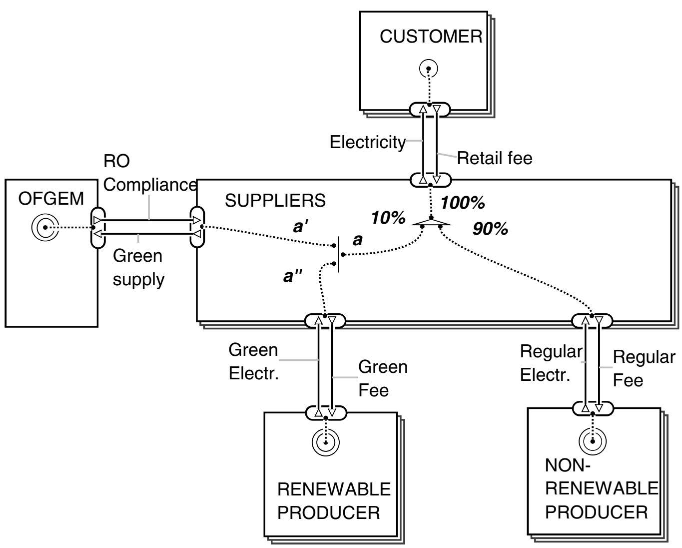


**Fig. 2.** An ideal value model of the ROC case study

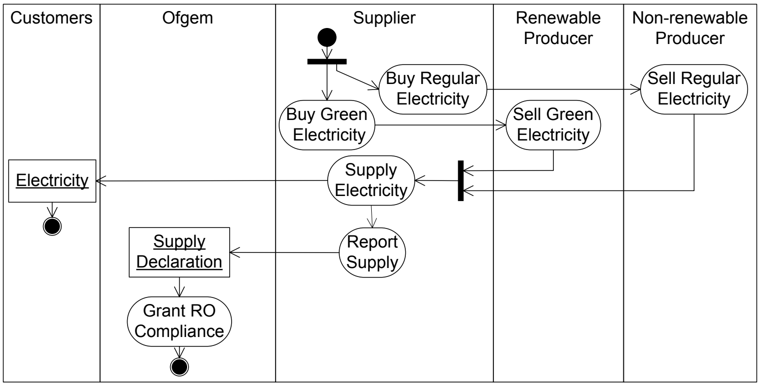


**Fig. 3.** The ideal process model of the ROC case study

### 4.2 The First $e^3$*control* Cycle: Non-tradable ROCs

#### Step 1: Ideal Situation

*Value view.* Fig. 2 presents an $e^3$*value* model for the ROC case study. The dependency path in Fig. 2 starts at the *customer*, the final electricity consumers in the UK. The customer buys *Electricity* from the *supplier* and pays the supplier a *Retail Fee* in return[2]. As denoted by the OR-fork at the supplier, the supplier can buy electricity from two sources: from *non-renewable producers* or from *renewable producers*. In the first case, the supplier buys *Regular Electricity* and pays *Regular Fee* in return. In the second case, the supplier buys *Green Electricity* and pays a *Green Fee*. Because green electricity is produced by more expensive renewable technology, the renewable producer asks a higher price for electricity than the non-renewable producer.

According to the RO regulation, a supplier has to obtain 10% of electricity from renewable sources[3]. In $e^3$*value* terms, this means that the electricity delivered by buying *Green Electricity* in Fig. 2 has to account for at least 10% of the *Electricity* supplied to the customers. We also assume that the suppliers behave ideally and always buy 10% of their supply from renewable producers. Therefore, if a supplier buys *Green Electricity*, then he also reports the supply of green electricity to *Ofgem* and receives a statement of compliance with the renewable obligation. This is modeled by the objects *Green Supply* and *RO Compliance* accordingly (see path *a'*).

*Process view.* In Fig. 3 we represent an ideal process model that corresponds to the ideal $e^3$*value* model. The process starts at the supplier who, as in the value model, has the choice of buying electricity from a renewable or a non-renewable supplier. In the first case, the supplier executes *Buy Regular Electricity* , followed by *Sell Regular Electricity* of the non-renewable supplier. In the second case, the suppler executes

[2] In this model, the customer buys both green and conventional electricity for the same price.

[3] When the regulation was introduced in 2002, the limit was around 10%. Currently in 2006/07 it is 6.7% and 2.6% in Northern Ireland.

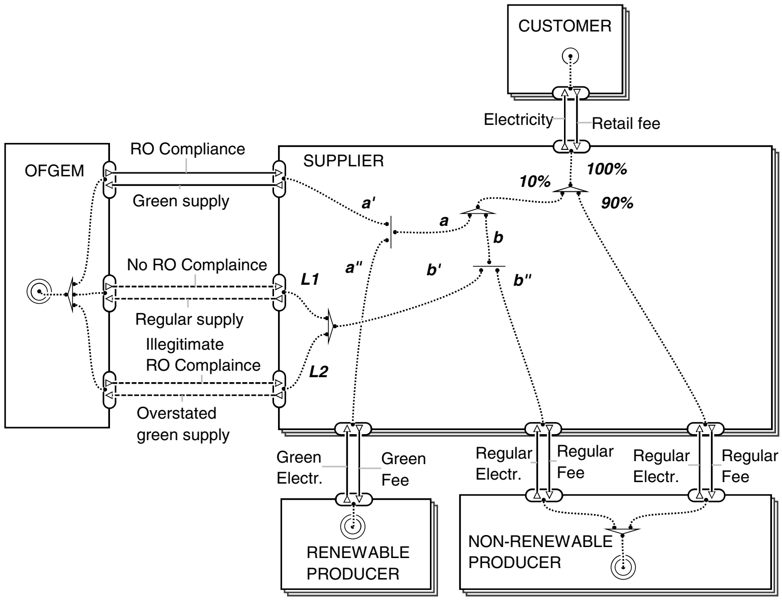


**Fig. 4.** Sub ideal value model for the problem of not supplying green electricity

*Buy Green Electricity* , followed by *Sell Green Electricity* executed by the renewable supplier. After that, in both cases *Supply Electricity* is executed by the supplier, which results in a transfer of an object *Electricity* from the supplier to the customer. Further, the supplier reports information about his supply (in MWh) to Ofgem by transferring a statement *Supply Declaration* to Ofgem. In the *Supply Declaration*, the supplier reports how much green electricity was supplied and what part of this electricity was green.

Since this model represents an ideal situation, the supplier is always assumed to behave ideally. In other words, the supplier always buys at least 10% of green electricity. Therefore, at the end of the process the *RO compliance* is always granted.

**Step 2: Sub Ideal Situation.** There are two types of sub ideal behavior. Firstly, not every supplier complies with the renewable obligation as a supplier can buy a lower percentage of green electricity than the 10% prescribed by the regulation. In this case, the *RO compliance* is not (completely) granted. Secondly, some suppliers can *overstate* the percentage of green supply in order to obtain the RO compliance illegally.

*Value view.* The sub ideal value model in Fig. 4 models both the ideal and sub ideal behavior of a supplier. The second OR-fork leads to the ideal path *a* and sub ideal path *b*. The ideal path *a* shows the same as in the ideal value model. The sub ideal path *b* corresponds to the two types of sub ideal behavior. In both cases, the supplier buys *Regular Electricity*, instead of *Green Electricity*. This corresponds to the exchanges in the sub path *b"*. Further, the *OR-fork* at the sub path *b'* indicates two possibilities of sub ideal behavior. The sub path, marked with a liability token *L1*, corresponds to a

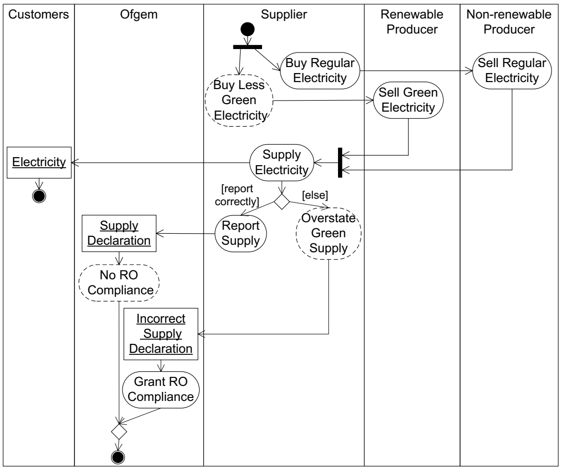


**Fig. 5.** Sub ideal process model for the problem of not supplying green electricity

situation in which the supplier reports his low supply of green electricity and does not get the RO compliance. At the sub path, marked with a liability token *L2*, the supplier either overstates his low supply of green electricity or understates the total supply. As a result, he gets the RO compliance illegitimately. The value objects that correspond to this situation are marked as sub ideal with dashed value transfers.

*Process view.* In Fig. 5 we represent only the sub ideal behavior of the supplier. The supplier buys insufficient green electricity. We model it with a sub ideal activity *Buy Less Green Electricity* instead of the *Buy Green Electricity*, as in the ideal process model. The supplier has the choice to report the true supply of green electricity or to overstate it. In the first case, the supplier transfers a *Supply Declaration* in which he informs Ofgem about insufficient green supply and, as a result, he does not get the RO compliance. In the second case, the supplier overstates the percentage of green supply and transfers an *Incorrect Supply Declaration*. As a result, the supplier gets the RO compliance illegitimately.

**Step 3: Reduce Sub Ideality by Applying Control Patterns.** In order to solve the control problems, Ofgem should implement one or more control mechanisms. We exemplify how the Penalty and Incentive patterns can be used to motivate suppliers to supply the right amount of renewable electricity. We have a process for selecting appropriate patterns for found controls problems (see [27]), which we do not explain due to lack of space. We illustrate below how the Penalty and Incentive patterns contribute to solving the found control problems.

*Process view.* Both the Penalty and Incentive patterns require the primary actor Ofgem to check the outcome of the sub ideal counter activity *Buy Green Electricity* and, then to reward or punish the supplier. To model this, we first add *Verify Compliance* to Fig. 6. This is an instance of the *Verify* activity in the patterns (see Figs. 13 and 14).

Furthermore, according to the pattern, *Verify Compliance* requires two inputs: *Supporting Statement* and *To-be-verified Statement* (see the Appendix). The *To-be-verified Statement* gives information about a percentage of electricity supplied from renewable sources. This corresponds to the object *Supply Declaration*, presented in the model.

According to the pattern, the to-be-verified statement *Supply Declaration* must be produced by an activity that *witnesses* the activity *Buy Green Electricity*. In addition, the pattern requires this witnessing activity to be executed either by the primary actor Ofgem or by a party independent and socially detached from the counter actor Supplier. If this is not the case, the *violation of segregation of duties* occurs, since the supplier who is responsible for buying green electricity also reports about it.

In the ideal model, the to-be-verified statement *Supply Declaration* is produced by the Supplier. This violates the requirements of the pattern. Therefore, we explicitly model the witnessing activity *Witness Green Supply*, instead of only the reporting activity. Secondly, we assign this activity to an actor, who is independent from the Supplier and is able to produce trustworthy information about the supply. Ideally such an actor is the primary actor Ofgem. However, as it is just an administrative body, Ofgem does not have the resources to control each supplier. So, Ofgem delegates control of the suppliers to some trusted party.

The role of such a trusted party can be played by the renewable producer who supplies electricity to suppliers. This party is therefore physically able to keep track of how much green electricity is bought by each supplier. In Fig. 6, the activity *Witness Green Supply* is assigned to the renewable producer.

The third change we make is to rename the to-be-verified statement. This statement, previously called *Supply Declaration*, is now called *Renewable Obligation Certificate (ROC)*. One ROC is issued for each Mega Watt/hour (MWh) of eligible renewable output. According to the pattern, the ROC is fed into the activity *Verify Compliance*, which compares whether the ROCs of one particular supplier represent 10% of his total supply.

The ROC only represents the amount of green electricity. However, the important criterion for granting RO compliance is the *share* of the green electricity within the supplier's total electricity supply. Therefore, we add another to-be-verified statement *Total Supply*, which represents this information. As with *ROC*, the *Total Supply* must be generated by an actor who is independent and not acting in the interests of the supplier. For instance, the data about the total supply could be retrieved from the final customer or from the supplier's annual accounts, assuming they are trustworthy. In the model we show that the *Total Supply* is generated by Ofgem. Data concerning the total supply of each supplier is easily accessible to a governmental organization like Ofgem.

The *Verify Compliance* activity requires a supporting statement, namely, information, which is needed to decide whether the *RO Compliance* should be granted. Such a document is a *RO legislation*, stating e.g. the required percentage of green electricity (which we assume is 10%), which producers are qualified to hold the status of 'renewable', to which customers should the reported green electricity be supplied, etc.

In addition, the activities are assigned in a proper order, as required by the control principles. The activity *Witness Green Supply* is executed after the *Buy Green Electricity* activity and before *Verify Compliance* activity. In addition, the *Verify Compliance* activity is executed before the primary activity *Grant RO Compliance*.

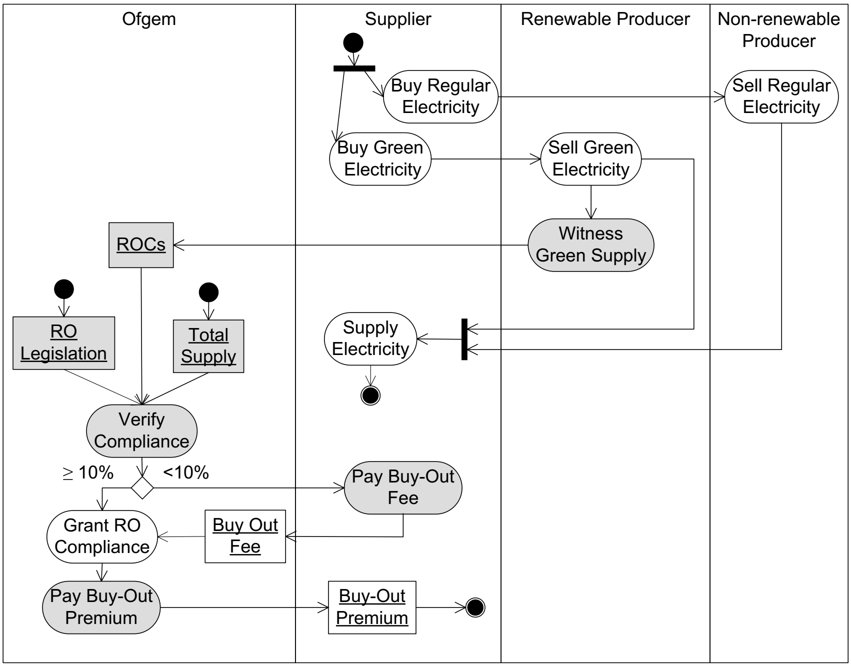


**Fig. 6.** Solution process model with penalties and incentives

According to the penalty pattern, the object *Penalty* should be added. The penalty should be transferred to Ofgem by a supplier who supplies less than 10% of green supply. According to the RO regulation, the supplier who does not have sufficient green supply to cover his obligations must make a deposit into the *buy-out fund* [28]. Such a payment corresponds to the object *Penalty* of the Penalty pattern. In Fig. 6 the *Buy-out Fee* is paid by the suppliers according to the RO regulation. According to the pattern, if the outcome of the *Verify Compliance* states that the green supply is less than 10%, the RO compliance is granted only after the buy-out fee is paid by the supplier. The *Pay BuyOut Fee* activity corresponds to the *Pay Penalty* activity of the Penalty pattern.

According to the solution given by the Incentive pattern, the *Incentive* object should be added. This object should be transferred to the supplier who supplies at least 10% of green supply. The buy-out fund is paid back to suppliers in proportion to how much green electricity they have purchased [28]. This payment, henceforth called *Buy-out Premium*, represents the incentive. We add it to Fig. 6.

In addition, according to the pattern, we model that if the outcome of the *Verify Compliance* states that the green supply is more than 10%, then RO compliance is granted and the *Buy-out Premium* is paid. The *Pay BuyOut Premium* activity corresponds to the *Pay Incentive* activity of the Incentive pattern.

Note that in the application of this pattern we have also used the Simple Delegation pattern (see http://www.e3value.com/e3family/e3control/patterns). This is needed

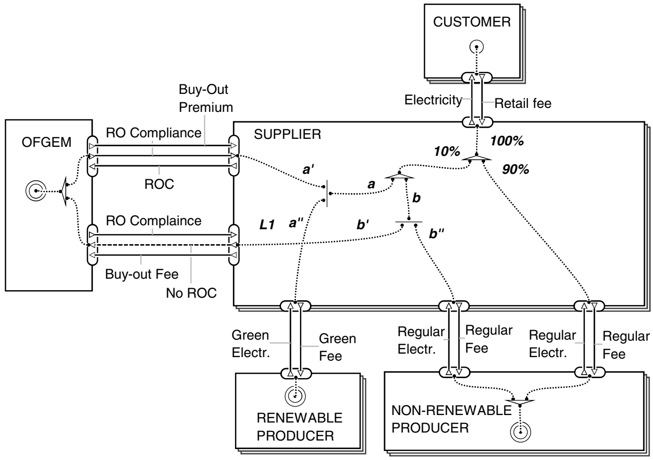


**Fig. 7.** Solution value model with penalties and incentives

to model that the *Witness Green Supply* activity is delegated by Ofgem to the trusted actor Renewable Producer.

*Value view.* We now make appropriate changes in the ideal value model. The changes have been caused by the introduction of penalties and incentives as well as by the delegation of the witnessing activity.

We add a new value transfer, indicating a penalty, as an outgoing value object of the counter actor Supplier in the sub ideal path *L1*. We add it to the transfer of *No RO Compliance* and *Regular Supply*, and change *No RO Compliance* to *RO Compliance*.

Also, since we have renamed the *Supply Declaration* to *ROC* in the process model, the ROC also appears as value object in the value model. We model *ROC* instead of the value object *Green Supply*, and *No ROC* instead of the value object *No Green Supply*.

The resulting value model is presented in Fig. 7 and this corresponds to reality. The penalty is represented by the value transfer *Buy-out Fee*. So, at the sub ideal path *b*, where the supplier does not supply enough green electricity, he is obliged to pay a buy-out fee in order to cover the RO.

The Incentive pattern requires that an incoming value object *Incentive* should be added to the Supplier in the ideal value transfer. The incentive is the buy-out premium paid by Ofgem to compliant suppliers. The incentive value object *BuyOut Premium* is added to the transfer of *Green Supply* and *RO Compliance*.

### 4.3 The Second $e^3$*control* Cycle: Tradable ROCs

The process model in Fig. 6 represents only a part of the actual ROC scheme. Due to the nature of the electricity business, suppliers can buy and sell electricity several times to other suppliers before it reaches the final customer. According to RO regulation, ROCs can be claimed by the supplier who delivers the associated green electricity to final customers. If a supplier sells green electricity to another supplier, the ROCs should also

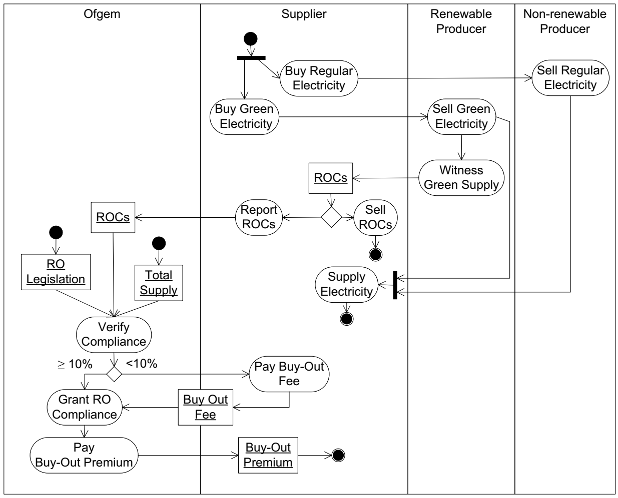


**Fig. 8.** An ideal process model with tradable ROCs

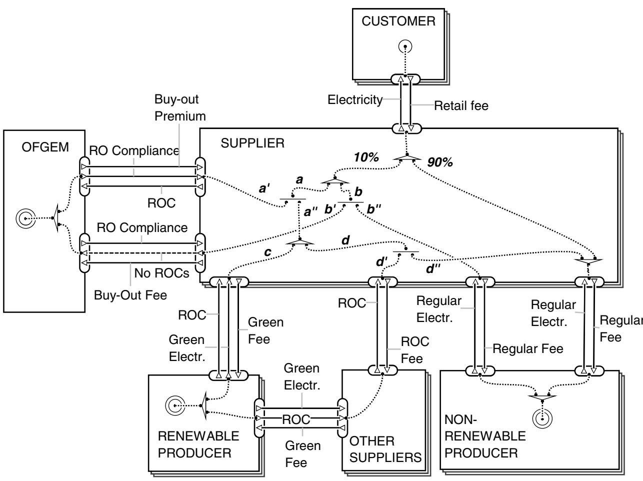


**Fig. 9.** An ideal value model with tradable ROCs

be transferred to this other supplier. In addition, ROC's can be traded themselves on the market. The ROC is in fact a security similar to stocks and bonds. As will be explained later, the ROC market was created to stimulate green electricity production. Therefore, additional controls are required.

**Step 1: Ideal Situation.** The ideal process model of the scenario with tradable ROCs is shown in Fig. 8. Unlike in the solution process model in Fig. 6, the ROCs are transferred to the supplier before being transferred to Ofgem to comply with the required percentage of supplied renewable energy. The supplier has the choice of selling the obtained ROCs or of reporting them to Ofgem. This choice is denoted by the UML decision element at the supplier. If the supplier reports the ROCs to Ofgem, the verification process of charging the buy-out fee or the paying buy-out premium remains the same as before. However, if the supplier sells the ROCs, they are not presented to Ofgem.

The corresponding value model is shown in Fig. 9. The following changes have been made compared to the situation without tradable ROCs in Fig. 7. Firstly, suppliers obtain ROCs from the renewable producers. Secondly, unlike in the solution value model in Fig. 7, the ideal dependency path now offers a choice between (1) obtaining ROCs for free while buying the (more expensive) green electricity in path *c* or (2) buying ROCs separately and purchasing the (cheaper) regular electricity in the path *d*. Thus, because ROCs can be traded, they are modeled as value objects, and not as process objects only.

**Step 2: Sub Ideal Situation.** The new ideal models in Figs.8 and 9 do not comply with the prescriptions of the Penalty and Incentive patterns in Figs. 7 and 6. Specifically, to perform the *Verify Compliance* activity, Ofgem has to rely on information received from the supplier. This implies that the supplier performs the *Witness Green Supply* activity. As already explained, this contradicts with the pattern, since the supplier should not report his own activities. For example, the supplier can forge ROCs and overstate the number of supplied green electricity.

In Fig. 10 we show the sub ideal process model of the scenario with tradable ROCs. The supplier overstates the number of ROCs he has, which is modeled by the activity *Overstate Green Supply*. This corresponds to the transfer of *No ROCs* by the supplier in the sub ideal value model in Fig. 11. Because the overstatement remains undetected, the supplier receives the *RO Compliance* and even gets the *BuyOut Premium* in return. This path is marked with the liability token L3.

Note that the supplier has an illegitimate interest in overstating the number of ROCs, which does not depend on whether he can cover the RO obligation or not. If the supplier has enough ROCs to cover the obligation, he may overstate ROCs to receive the buy-out premium (see path *a*). If the supplier has not enough ROCs to cover the obligation, he is motivated to overstate ROCs to avoid the buy-out fee penalties (see path *b*).

**Step 3: Reduce Sub Ideality by Applying Control Patterns.** We now apply the Execution Monitoring pattern to reduce sub ideality.

*Process view.* Following the solution given by the Execution Monitoring pattern, we add a new verification activity *Verify ROC* to Ofgem. It verifies the *Present ROCs* activity of the Supplier. To do so, it checks if a *ROC*, submitted by a supplier, corresponds to a ROC reported by the renewable producer.

In addition, unlike in the ideal model with ROCs in Fig. 8, the renewable producer not only issues a ROC to the supplier, but also reports the number of issued ROCs to Ofgem. This is modeled with an object *ROC Register*. The ROC register is an electronic, web-based system, supported by Ofgem, which allows generators and suppliers to view the ROCs they hold and to transfer ROCs to other parties. In this way, Ofgem can verify the authenticity of each ROC.

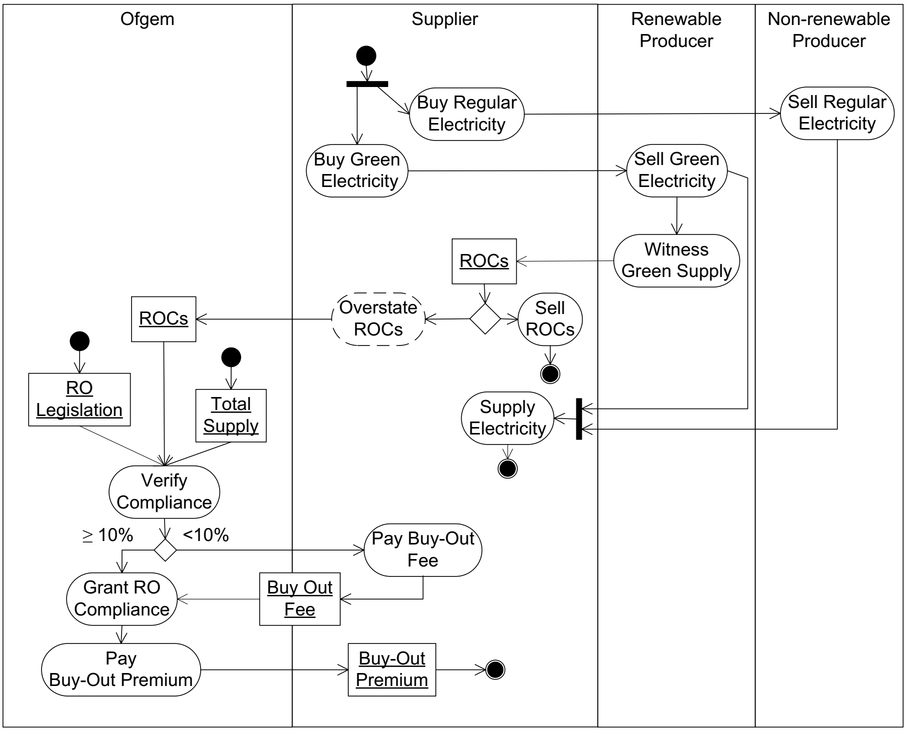


**Fig. 10.** A sub ideal process model with tradable ROCs

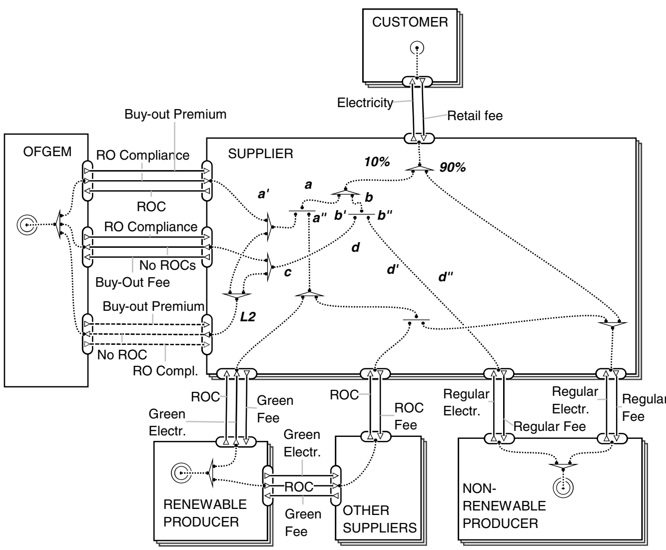


**Fig. 11.** A sub ideal value model with tradable ROCs

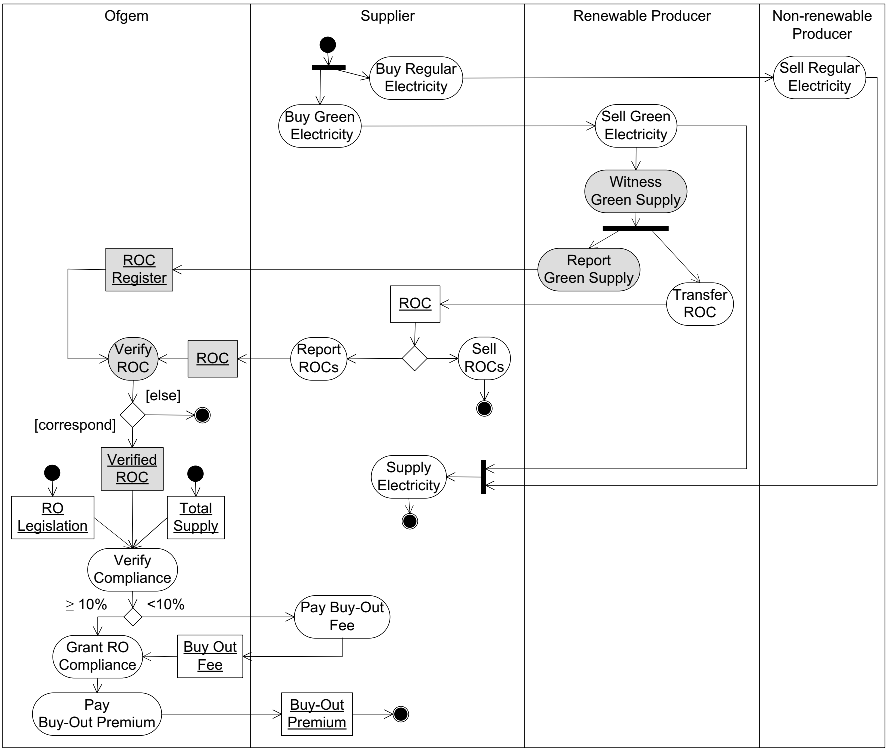


**Fig. 12.** Controls in a process model with tradable ROCs

Note that the activity *Report ROCs* is added not because of a pattern's prescription. It is added because the UML language restrains modeling the exchange of the object (*ROC Registry* in this case) directly from the AND-join (the black thick bar). We therefore add an activity in between.

In this solution, an *ROC* plays the role of the to-be-verified statement, while the *ROC Register* plays the role of the supporting document. Thus, the renewable producer plays the role of the provider of a supporting document.

After the verification of an ROC, the *Verified ROC* object is used. As in the previous $e^3$*control* cycle, the *Verified ROC* plays the role of the to-be-verified statement for the verification activity *Verify Compliance*.

The value model does not change and is the same as the ideal value model in Fig. 9.

## 5 Conclusion and Discussion

In this chapter, we have presented a series of patterns for the value-based design of inter-organizational controls, and demonstrated three patterns in a case study. The proposed patterns take an explicit *economic value* perspective on the control problem: It is important to understand first the valuable objects to be safeguarded, before designing controls ensuring proper transfer of objects.

We have contributed a structure for stating control patterns in terms of ideal value&activity models, sub ideal value&activity models representing the fraud, and possible solutions expressed using similar model types. The patterns themselves stem from two sources: (1) accepted theory on the principal-agent relations, accounting, and auditing, and (2) four industrial case studies we have performed to design inter-organizational controls.

For $e^3$*control* , many further research issues can be identified, and we present two of them. First, there is the issue of the cost of controls. So far, we have studied controls that have inherent economic value aspects (e.g. incentives and penalties), and therefore have an impact on the $e^3$*value* model of a networked organization. However, implementing controls usually comes with a price. This 'cost of control' should also be considered when analyzing control issues in value networks. Second, controls may interact. If we first analyze and select controls for actor A, and thereafter actor B, the resulting set of the controls for the *network* can be different compared to first considering actor B, and thereafter actor A. Actually, this can be seen as an example of feature interaction, and a more structured approach is needed to deal with this interaction.

More in general, the $e^3$*control* methodology is a member of the $e^3$*value* modeling suite, focusing on understanding the exchange of valuable objects in networks of enterprises. Although value modeling has proven to be useful in numerous case studies, there exist also a number of research challenges to be addressed. First, there is the issue of *valuation*. Value objects reflecting money are relatively easy; the value of such objects coincides with the amount of money exchanged. However, in value modeling it is sometimes needed to assign economic value to non-money objects also. For instance, a final customer of a service should value the service outcome obtained. Essentially, we are then interested in the economic utility function of the customer, which is difficult to obtain. But even objects directly reflecting money pose interesting problems. Usually, a 'money object' refers to a price to be paid for a product or service outcome. However, the pricing scheme can only be partly based on market considerations (e.g. the amount of money the customer is willing to pay). Another factor is the cost needed to produce the product or service outcome. As in our field, the services are usually *ICT* services, there should be a clear, and understandable, relationship between the valuable ICT service offered, and the costs of the ICT service. In addition to that, the value network takes a commercial perspective on a network of actors. But, in the case of ICT, other perspectives are required, such as perspectives on the supporting ICT, and the business processes that should be carried out, to develop the value network at hand. One of the key questions is then how to properly align these perspectives during the design of the actor network, and how to ensure that these perspectives remain aligned.

*Acknowledgments.* This work is partially funded by the Post Master EDP Audit Education of the Vrije Universiteit Amsterdam and by the EC funded IP-project FENIX.

## References


1. Weill, P., Vitale, M.R.: Place to Space: Migrating to eBusiness ModelsEconomy and Society. Harvard Business School Press, Boston (2001)
2. Gordijn, J., Akkermans, J.M.: *e3-value*: Design and evaluation of e-business models. IEEE Intelligent Systems, Special Issue on e-Business 16(4), 11–17 (2001)
3. Gordijn, J., Akkermans, J.M.: Value-based requirements engineering: Exploring innovative e-commerce ideas. Requirements Engineering 8(2), 114–134 (2003)
4. Gordijn, J., Yu, E., van der Raadt, B.: E-service design using i* and e3value modeling. IEEE Software 23(3), 26–33 (2006)
5. Gordijn, J., Akkermans, J.M.: Business models for distributed energy resources in a liberalized market environment. The Electric Power Systems Research Journal 77(9), 1178–1188 (2007), doi:10.1016/j.epsr.2006.08.008
6. Castro, J., Kolp, M., Mylopoulos, J.: Towards requirements-driven information systems engineering: the tropos project. Information Systems 27, 365–389 (2002)
7. Kartseva, V., Gordijn, J., Tan, Y.-H.: Developing a modelling tool for designing control mechanisms in network organisations. International Journal of Electronic Commerce (2005)
8. Ronmey, M.B., Steinbart, P.J.: Accounting Information Systems, 10th edn. Prentice Hall, New Jersey (2006)
9. ISACF: Control objectives for information and related technology (cobit). Technical report, Information Systems Audit and Control Foundation (ISACF) (2005)
10. Eisenhardt, K.M.: Agency theory: An assessment and review. Academy of Management Review 14(1), 57 (1989)
11. Anthony, R.N., Govindarajan, V.: Management Control Systems, 11th edn. McGraw Hill Higher Education, New York (2003)
12. Kartseva, V., Hulstijn, J., Baida, Z., Gordijn, J., Tan, Y.-H.: Towards control patterns for smart business networks. In: Vervest, P., van Heck, E. (eds.) Proceedings of the Smart Business Networks Initiative Discovery Session. Springer, Heidelberg (2006)
13. Kartseva, V., Hulstijn, J., Tan, Y.-H., Gordijn, J.: Towards value-based design patterns for inter-organizational control. In: Waldena, P., Gricar, J. (eds.) Proceedings of the 19th Bled Electronic Commerce Conference on eValues, Bled, Slovenia. University of Maribor (June 2006)
14. Kartseva, V., Gordijn, J., Tan, Y.-H.: Designing control mechanisms for value webs: The internet radio case study. In: Vogel, D.R., Walden, P., Gricar, J., Lenart, G. (eds.) Proceedings of the 18th BLED conference (e-Integration in Action), pages cdrom, Maribor, SL (2005)
15. Kartseva, V., Hulstijn, J., Gordijn, J., Tan, Y.-H.: Modelling value-based inter-organizational controls in healthcare regulations. In: Proceedings of the 6th IFIP conference on e-Commerce, e-Business, and e-Government, Turku, Finland (2006)
16. Alexander, C.: The Oregon Experiment. Oxford University Press, New York (1975)
17. Fowler, M.: Analysis Patterns: Reusable Object Models. Addison-Wesley, Reading (1997)
18. Seruca, I., Loucopoulos, P.: Towards a systematic approach to the capture of patterns within a business domain. The Journal of Systems and Software (67), 1–18 (2003)
19. Hollander, A.S., Denna, E., Cherrington, J.O.: Accounting, Information Technology, and Business Solutions, 2nd edn. McGraw-Hill, Irwin (1999)
20. Bons, R.W.H.: Designing Trustworthy Trade Procedures for open Electronic Commerce. PhD thesis, University of Rotterdam (1997)
21. Bons, R.W.H., Dignum, F., Lee, R.M., Tan, Y.-H.: A formal analysis of auditing principles for electronic trade procedures. International Journal of Electronic Commerce 5(1), 57–82 (2000)

22. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley, Boston (1995)
23. Fowler, M., Scott, K.: UML Distilled, 2nd edn. Addison Wesley, Reading (2000)
24. Nuseibeh, B., Kramer, J., Finkelstein, A.: A framework for expressing relationships between multiple views in requirements specification. IEEE Transactions on Software Engineering 20(10), 760–773 (1994)
25. Bons, R.W.H., Lee, R.M., Wagenaar, R.W.: Designing trustworthy interorganizational trade procedures for open electronic commerce. International Journal of Electronic Commerce 2(3), 61–83 (1998)
26. Chen, K.T., Lee, R.M.: Schematic evaluation of internal accounting control systems. EURIDIS Research Monograph, RM-1992-08-1, Erasmus University Rotterdam (August 1992)
27. Kartseva, V.: Designing Controls for Network Organization: A Value-Based Approach. PhD thesis, Vrije Universiteit Amsterdam (2008), `http://docs.e3value.com/bibtex/pdf/Kartseva2008.pdf`
28. OFGEM. The renewables obligation: Annual report 2003-2004. Technical report, Office of Gas and Electricity Markets, United Kingdom (2004)
29. OFGEM. Renewables Obligation - Atlantic Electric and Gas Limited and Maverick Energy Limited. Technical Report R/65, Office of Gas and Electricity Markets, United Kingdom (2004)

## Appendix: The $e^3$*control* patterns

Below we present three control patterns that we use in the case study of this chapter. Other patterns can be found at http://www.e3value.com/e3family/e3control/patterns.

### 5.1 Penalty Pattern

The penalty pattern punishes the counter actor for sub ideal behavior by introducing a value object that decreases the accumulated value of the counter actor in the *ideal path*.

### 5.2 Incentive Pattern

The incentive pattern rewards the counter actor for ideal behavior by introducing a value object that increases the accumulated value of the counter actor in the *ideal path*.

### 5.3 Execution Monitoring Pattern

The *Execution Monitoring* pattern describes the control problem in which a counter actor does not execute his commitments or executes them in a sub ideal way (e.g. not as agreed in the contract). The control mechanism requires the primary actor to monitor counter activity. This solution is very similar to the Partner Screening pattern, the only difference being that verification concerns the counter activities under the contract and not an actor's past counter activities.

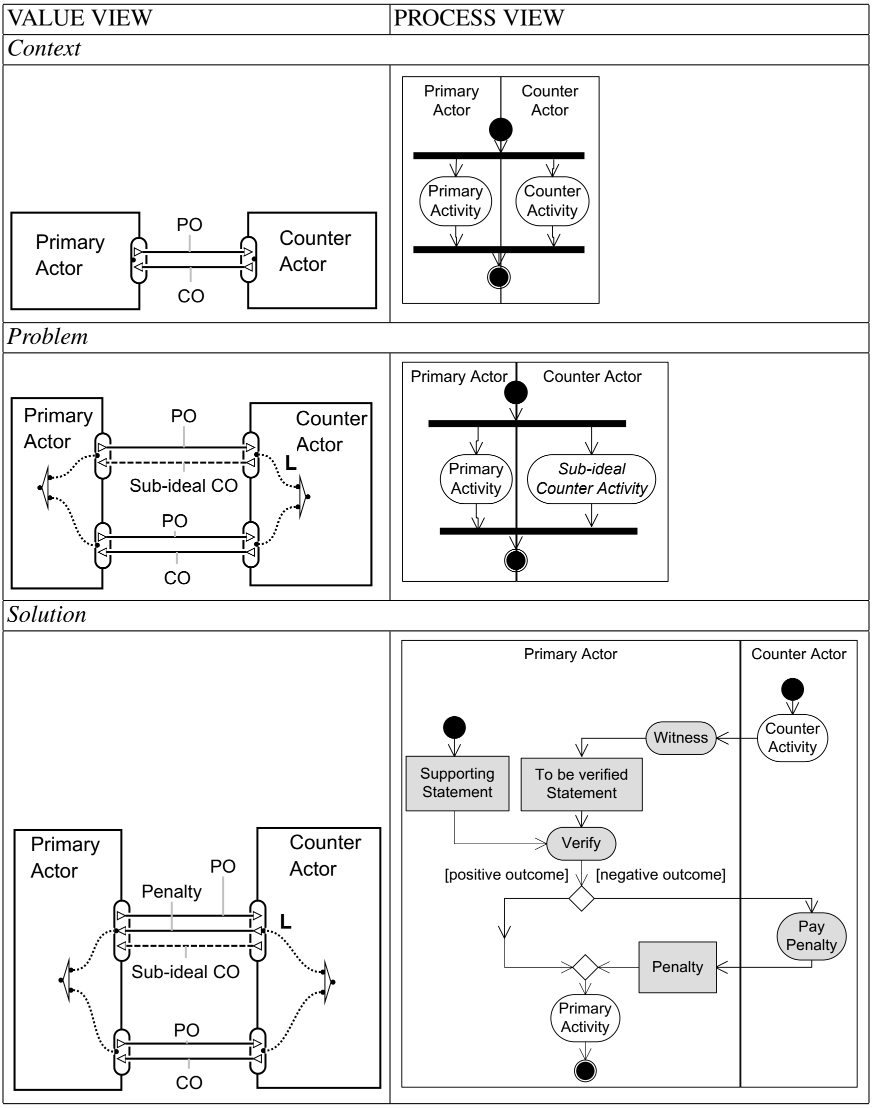


**Fig. 13.** Penalty pattern

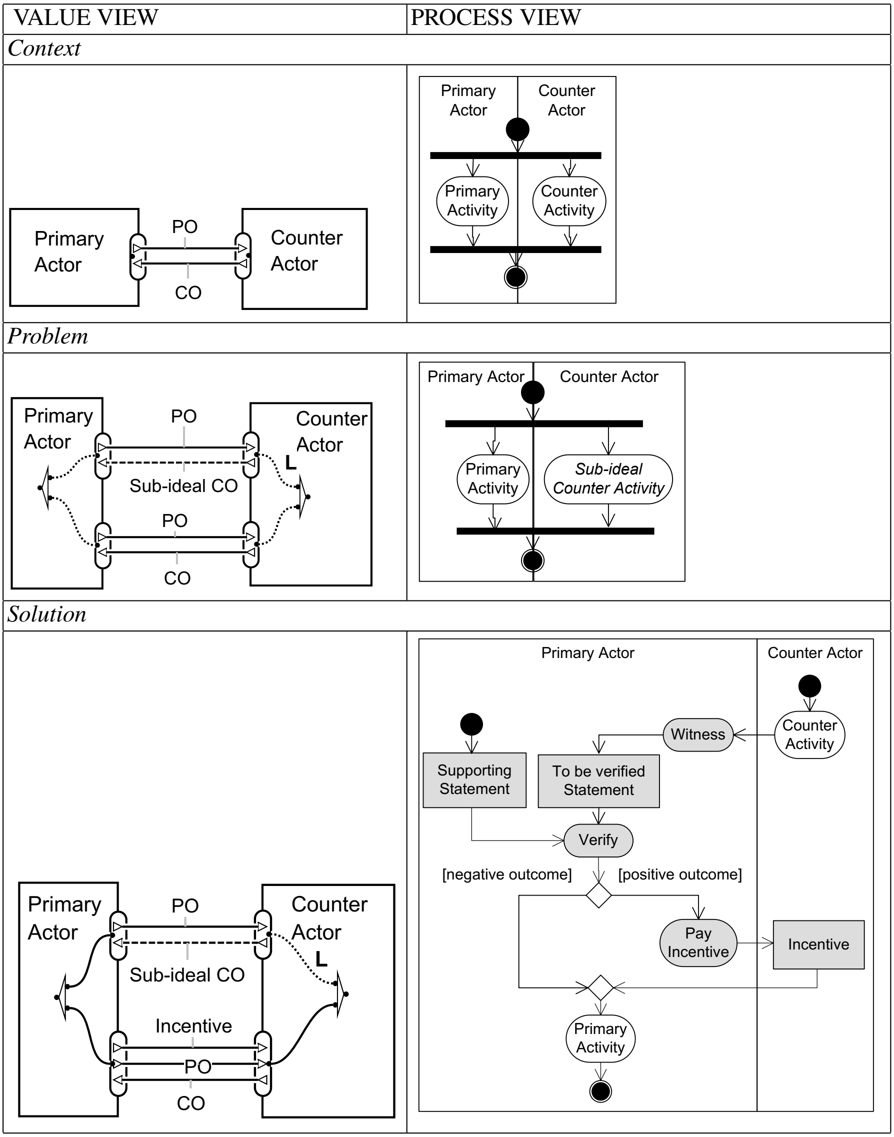


**Fig. 14.** Incentive pattern

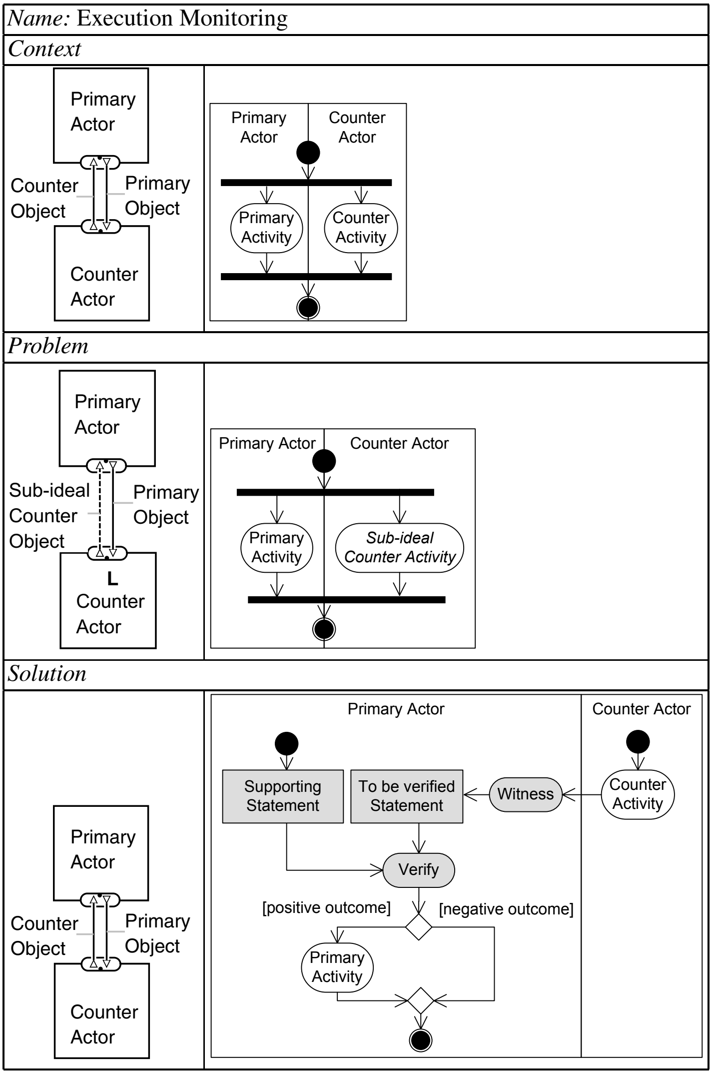


**Fig. 15.** Pattern 'Execution Monitoring'