Towards a Modelling Tool for Designing Control Mechanisms for Network Organisations

Vera Kartseva, Jaap Gordijn and Yao-Hua Tan

ABSTRACT: In network organisations and electronic commerce control mechanisms to guarantee inter-organisational contract compliance are important tools to enhance trust in a fair business transaction. We propose the *e^3-value/control* design methodology for designing controls for network organisations. The *e^3-value/control* design methodology is an extension of the *e^3-value* design methodology for business models, which is focussed on modelling economic value exchanges between the organizations of a network, rather than operational processes between them. We argue that a methodology for designing control mechanisms should include three steps: (1) the design of an inter-organisational value exchange model of a network, (2) the analysis of inter-organisational control problems within the network, i.e. the analysis of possible violations of contractual obligations related to the value model, and (3) the design of control mechanisms to detect or prevent such control problems. We illustrate the usability of the *e^3-value/control* design methodology in a case from the electricity industry.

KEYWORDS AND PHRASES: Virtual network organisations, value webs, inter-organisational controls, design methodology.

One of the limitations for the success of electronic commerce is the risk of opportunistic behaviour by legal entities [16]. For example, parties may not deliver products while they promised to do so, customers may not pay for delivered products, or other types of fraud may be committed. Consequently, since transacting parties often do not know each other in electronic commerce, sufficient trust enhancing facilities should be in place to facilitate trading. In this paper we look at e-business network organisations from a control theory perspective, which implies that network participants can behave *opportunistically*, and cause *control problems*, and, thus, require *control mechanisms* to create trust.

Control mechanisms to prevent and detect opportunistic behaviour of members of the network are an important component for sustainability of e-business networks. Extensive research has been carried out on inter-organisational control mechanisms in the fields of economics, strategy, organisational research and management accounting (see [5] for an overview). However, these studies are mainly exploratory in nature, and do not address the *design* of control mechanisms. With respect to problems in hierarchical organisations, the design of controls is addressed in internal control literature [23, 28]. However, little attention is paid in this research to the development of design tools to design control mechanisms for e-business networks.

An important aspect of business network organisations (compared to hierarchical organisations) is that in business networks controls are typically not imposed on the network by one central organisation, but are negotiated among all the partners of the network. During this negotiation process, stakeholders (e.g. business analysts, system developers, CIOs, CEOs etc.) typically use natural language to represent and communicate their statements. However, the stakeholders often have different views on value propositions and different interests, which, when communicated in natural language, may lead to incomplete and ambiguous statements [17]. Conceptual modelling, which is a well-known methodology from the field of requirements engineering, plays an important role in recent research on e-business modelling [1, 10, 18, 20, 21, 26, 29]. The advantage of business modelling is that the conceptualisation of business ideas makes conflicts explicit, which can occur during the negotiation process. This helps stakeholders to resolve these conflicts in an early stage of the development process. Although business modelling addresses the design issues of business models of network organisations, it does not explicitly address the design of inter-organisational controls.

The design of control mechanisms for e-business is a rather uncharted territory. Current research topics in computer science such as 'web services' and 'peer-to-peer networking' enable the provisioning of inter-organisational business processes, which are required for control procedures [2, 7]. This is, however, not the same as the design of control procedures. The design of a contract, as well as its supporting controls, is a multi-disciplinary task. It involves obviously economic and legal aspects, but also computer science issues are relevant (many controls are implemented in software code), as well as inter-organisational business process design (many contracts stipulate how, and in which sequence, business transactions should be carried out, and by whom). In organisational science research on contracting it has been observed that contingency planning of a contract is one of the most important as well as labour intensive stages of contract negotiation. For example, in [15] it was observed that contingency planning is the most useful part of a contract, because it shows the contract partners what problems they could encounter, what reasonable financial compensations have to be paid when they do occur, and how to minimise the likelihood that these problems will occur.

Based on these shortcomings described above, we propose a *graphical conceptual modelling technique* for modelling control mechanisms in business networks, to support contract negotiating and drafting as explained in [4] and [25]. Related research has been done by Elsas [6], Lee and Bons [13]. Elsas developed in [6] a theory of auditing based on Petri Net modeling, dealing with the trustworthiness of an organisation's annual account. However, his focus was on preservation results of transformations in these Petri Net models. In other words, he studied when an effective control is still effective when the underlying adminstrative organsation or processes is changed. He did not specifically focus, as we do, on the development of tools that support the control design process. Lee and Bons developed in [13] a tool, based on so-called Documentary Petri Nets, to model and test the trustworthiness of the document flow in trade procedures with. This tool focuses on the document process aspects of inter-organisational control mechanisms rather than the value exchanges as we do in this paper.

In our work we take a different perspective on the design of controls than the abovementioned authors. As a starting point of our methodology we take the value modelling tool *e^3-value* [10], which focuses primarily on modelling *economic value exchanges* in business networks rather than *business processes*. Various authors in business modelling research, e.g. [1, 10, 26], have emphasized that the design of business networks should not start with business process design, but with modelling the exchange of value between the organisations in a network. One reason to focus on value exchanges rather than on business processes is that the focus on business processes can easily lead to an unmanageable complexity in the modelling activity. For example, in the *Business Process Handbook* project [14], an ontological business process modelling approach was developed for inventing organisations. They proposed an ontology with about 3400 different activities with 20 levels of specialisation and 10 levels of decomposition. Such a complex ontology is too complicated for exploring new business models for network organisations, where the modelling object is not just a single organisation, but rather a network for a large group of organisations. We do not claim that process aspects of controls should be ignored, but we want to emphasize that there are reasons to consider modelling controls with value models. An important reason to develop the value perspective in addition to the business process view is that the primary objective of an inter-organisational control mechanism is the value exchange rather than a specific business process. Furthermore, in previous research [11, 12] we observed that many elements of controls in business networks have a value component. Many control mechanisms are commercial services themselves, hence they could be viewed as value exchanges. Consider, for example, the Bill of Lading that we analysed in [11, 12] or the ROC example that we will later explain in this paper. Both controls constitute new value exchanges; e.g. Bills of lading and ROCs are tradeable objects.

The *e^3-value* methodology [10] was developed to model business networks with the objective to estimate the value that a network delivers. The *e^3-value* methodology assumes non-opportunistic behaviour of network participants, which is a simplification that enables stakeholders to concentrate primarily on the business opportunity and not on the implementation initially. Although *e^3-value* models value exchanges, it is not primarily a tool for economic modelling in the sense of being a support tool for financial analysis or econometric simulation. It does support a basic financial analysis functionality in the so-called profitability sheets (see [9] and [10] for further explanation of this), but this is not the main goal of the methodology. Since the *e^3-value* ontology does not have concepts to model anything related to the control aspect of value exchanges, we present in this paper *e^3-value/control*, which is an extension to *e^3-value* with features for modelling controls.

The remaining of the paper is structured as follows. First, we introduce a framework for designing control mechanisms, describing the steps that should be executed to design control mechanisms for business networks. Then we give a brief introduction on value model design with *e^3-value* methodology. Subsequently, we introduce the *e^3-value/control* methodology, and we show how this methodology supports the analysis and design of controls related to violations of economic value exchange obligations. Furthermore, we show how to model and to analyse a real-life example from the electricity industry with *e^3-value/control.*

A Framework for Designing Control Mechanisms in Network Organisations

Extensive research has been done on developing a theory of control mechanisms design for hierarchical organisations (or internal control) (see [23, 28]). The comprehensive framework for the design of control mechanisms used in this literature basically includes the following steps: (1) business process analysis, (2) control problem analysis and (3) design of control mechanisms. For example, the business processes in different wholesale organisations are similar; the loss of revenues in a wholesale organisation can indicate billing errors (control problem), and the control mechanisms are pre-billing (preventative controls) or reconciling purchases with sales (detective controls).

For the implementation of a control mechanism it is crucial to take the actual business processes into account that are used to realise a specific value exchange, but to some extent one should abstract away from these process details to understand the primary purpose of a control mechanism. For example, when a seller is not confident that a buyer will pay for the goods that he ordered, then the seller might either ask for a prepayment, or a letter of credit procedure, which is basically a prepayment arrangement provided by a bank. Clearly, these two control mechanisms are quite different from a process point of view, but the underlying concern of the seller is the same; namely how to guarantee the value exchange of the payment for the goods. Hence, we propose to start the design of control mechanisms with modelling economic value exchanges rather than business processes.

However, the value exchange perspective as modelled in *e^3-value*, presupposes an ideal world, in which parties in the value web do not commit a fraud or behave opportunistically with respect to each other. Clearly, this is not realistic. When designing controls, risk factors have to be taken into account [10], and one has to identify what can go wrong in the value web. Therefore, we distinguish two states of a network of organisations: (1) no errors, opportunistic behaviour or fraud occurs, according to the terminology in [22], *ideal behaviour* of actors, further referred to as an *ideal situation*, and (2) errors, opportunistic behaviour or fraud does occur, referred to as a *sub-ideal situation*. Hence, the framework for designing control mechanisms should include at least the following subsequent three steps:

(1) Design of the ideal situation within value exchange perspective,

(2) Control problem analysis: or the analysis of possible sub-ideal situations

(3) Design of inter-organisational control mechanisms (IOCs), including *detecting*, *preventing* and *correcting* control problems.

To address these three steps of the framework, we need to be able to model both ideal and sub-ideal situations. A value model expressed in *e^3-value* is a model of an inter-organisational contract, where parties have obligations to exchange value objects. An *e^3-value* model assumes that parties in a business network act according to that contract, therefore *e^3-value* models are referred further as *ideal value models*. The control problems occur when parties in the business network behave sub-ideally and violate obligations to exchange values. Since *e^3-value* does only model ideal behaviour, we need to extend the *e^3-value* with concepts to represent sub-ideal behaviour and we call this extension *e^3-value/control*. An *e^3-value/control* model enables representation of control problems, resulting in a *sub-ideal value model*. The three-step framework is incorporated in the proposed methodology in the following way: first, an ideal value model is designed with *e^3-value*, secondly, the control problem analysis is supported by designing a sub-ideal value model with *e^3-value/control*, and, thirdly, control mechanisms are designed to reduce the likelihood that these sub-ideal situations will occur.

Ideal value models

The *e^3-value* methodology [10] supports the conceptualisation of a business network by constructing a value model, representing it graphically in a rigorous and structured way, and performing an economic sensitivity analysis for all organisations involved. In particular, the *e^3-value* methodology provides modelling concepts for showing which parties exchange things of *economic* value with whom, *and* expect *what* in return. The methodology has been validated in a series of case studies including media, news, banking and insurance, electricity power, and telecommunication companies to design value models of network organisation [10].

A distinguishing feature of the *e*^{3}*-value* methodology is that it takes a value-based view stating *what* the value proposition is, while most of the currently available design methodologies focus on business processes stating *how* a value proposition is implemented. There are a few value chain design methodologies, which provide concepts for describing value constellations; for example, the AIAI Enterprise conceptual framework [27] or the Resource Event Agent (REA) [8] conceptual framework. However, these frameworks only focus on the description of the final result, and do not support the value chain design process. Other business modelling methodologies (see [21] for an overview) offer only generic conceptual frameworks, and lack the formality that is required for proper analysis. Tapscott *et al.* in [26] offer a graphical diagramming approach to represent economic exchanges between enterprises. However, compared to *e*^{3}*-value*, this approach has several drawbacks; e.g. it has no notion of economic reciprocity, or economic activity, nor does it support the profitability assessment of individual organisations. The *e*^{3}*-value* methodology avoids these shortcomings by focusing on the value viewpoint and introducing a *minimal* number of concepts, which leads to a concise and efficient modelling technique.

[Insert about here Figure 1. *e*^{3}*-value* model of a Purchase with Tax payment.]

We briefly describe the concepts of the *e*^{3}*-value* methodology using a simple example. In Figure 1 a buyer obtains goods from a seller and offers money in return. According to the law, the seller is obliged to pay the value-added tax (VAT). This is conceptualised by the following e^3-*value* constructs:

Actor. An actor is perceived by its environment as an independent economic (and often legal) entity. An actor makes a profit or increases its utility. In a sound, sustainable, business model *each* actor should be capable of making profit. The example shows a number of actors: a *buyer*, a *seller*, and a *tax office*.

Value Object. Actors exchange value objects, which are services, products, money, or even consumer experiences. The important point here is that a value object is *of value* for one or more actors. *Good* and *payment* are examples of value objects, but *legal compliance* to pay tax is also a value object.

Value Port. An actor uses a value port to show to its environment that it wants to provide or request value objects. The concept of port enables to abstract away from the internal business processes, and to focus only on how external actors and other components of the business model can be 'plugged in'.

Value Interface. Actors have one or more value interfaces, grouping reciprocal, opposite-directed value ports. A value interface shows the value object an actor is willing to exchange, *in return for* another value object via its ports. The exchange of value objects is atomic at the level of the value interface.

Value Exchange. A value exchange is used to connect two value ports with each other. It represents one or more *potential* trades of value objects between value ports.

With the concepts introduced so far, we can explain who wants to exchange values with whom, but we cannot yet explain what happens in response to a particular end-consumer need. For this purpose we include in the value model a representation of *dependency paths* (based on [3]) between value interfaces. A dependency path connects the value interfaces in an actor and represents triggering relations between these interfaces. A dependency path consists of dependency nodes and segments.

Dependency node. A dependency node is a stimulus (represented by a bullet), a value interface, an AND-fork or AND-join (short line), an OR-fork or OR-join (triangle), or an end node (bull's eye). A stimulus represents a consumer need, an end node represents final state in a model.

Dependency segment. A dependency segment connects dependency nodes and value interfaces. It is represented by a link.

Dependency path. A dependency path is a set of dependency nodes and segments that leads from a start stimulus (also called a **consumer need**) to an end stimulus. A path indicates that if values are exchanged via a value interface, then other value interfaces, connected by the path, also exchange values.

Sub-ideal Value Models: Modelling Control Problems and Control Mechanisms

To be able to perform the second step of the framework and to model sub-ideal situations, it is not enough to use only the *e^3-value* methodology. For example, one of the sub-ideal situations in the case, modelled in Figure 1, is when the seller does not deliver a good, while the buyer does pay. To model this situation, we need to model an exchange with a single value object "Payment", but this is not allowed in *e^3-value*. The *e^3-value* methodology has a requirement called the *Principle of Reciprocity*, which says that actors only exchange value objects in return for one or more other value object, or they do not exchange at all. However, in the exchange of one single value object between two actors the Principle of Reciprocity does not hold. The exchange of the single value object "Payment" is in *e^3-value* terms, an invalid construct, and the attempt to model it will result in an error message by the *e^3-value* model checker. However, in our example of the sub-ideal situation single value exchanges can occur as a result of a violation by one of the actors. To design similar sub-ideal situations we extended *e^3-value* to the *e^3-value/control* formalism to be able to model *what* a violation is and when the Principle of Reciprocity does not hold. Furthermore, *e^3-value/control* has constructs to represent explicitly which party in the network is responsible for a violation and how serious this violation is. In particular, in *e^3-value/control* penalties can be assigned to an actor who violates the obligation to exchange. The seriousness of a violation is expressed in economic terms. As in [22], the penalties represent a kind of fines, which an actor gets if he does not follow the behaviour prescribed by an e^3-*value* model. The weight of a penalty represents the *costs* of the violation, and, therefore, penalties have economic consequences for actors. Thus, the introduction of penalties into the value model creates incentives for actors to change their behaviour. A regulator (e.g. government) has to set the weight of penalties to ensure that the incentives work.

The Classification of the Elements in e^3-value/control

We distinguish two types of violations resulting in an incorrect value exchange. The first type – the **exchange violation** - is when value exchanges are executed incorrectly (e.g. late delivery of goods, or late payment), or not executed at all (e.g. no delivery, or no payment). The second type – the **object violation** - is an exchange of incorrect value objects (e.g. delivering a CD instead of the ordered DVD).

Exchange Violation: Incorrect Value Ports

The value exchange is an exchange of a value object between two ports. Every port has at least one value exchange. If the exchange does not occur, the value object remains at a port. Thus, we can distinguish between **correct** and **incorrect value ports**. Incorrect value ports can have different properties to represent different sub-ideal situations: non-executed exchange, late delivery, damage of the value object etc. This list can easily be extended to represent other situations.

[Insert about here Figure 2. Some sub-ideal scenarios in the primary value web modelled with *e*^{3}*-value/control*.]

Figure 1 shows an ideal *e*^{3}*-value* model. In Figure 2 we use *e*^{3}*-value/control* to model sub-ideal situations that can happen when one of the actors violates the exchange obligations stated in the ideal model (i.e. does not respect the principle of reciprocity). To represent the possible exchange violations between the buyer and the seller and between the seller and tax office, the following value exchanges have been added in Figure 2, compared to Figure 1:

- Value Exchange 2: The goods are delivered, but the payment is not done;
- Value Exchange 3: The goods are not delivered, but the payment is done;
- Value Exchange 5: Neither the goods are delivered, nor the payment is done;
- Value Exchange 7: The tax is not paid, so in return the legal compliance is not granted.

Object Violation: Incorrect Value Objects

The second type of the sub-ideal exchange, called **object violation,** refers to an exchange of incorrect value objects (for example, delivering a CD instead of the ordered DVD). If the value object exchanged between two ports is not equal to the value object that would be exchanged in the ideal situation, we call it an **incorrect value object**. Incorrect objects can be used to model such situations as, for example, a delivery of a good or a service, which is of a lesser quality than the one referred to in the ideal value model. For example, in Figure the incorrect object "Other goods" is delivered 2 in value exchange 4, instead of the intended object "Goods".

Compared to Figure 1, there are four additional value exchanges in Figure 2 between the buyer and the seller, and one additional value exchange between the tax office and the seller. The scenario path starts at the Buyer and follows with the OR-fork that splits into five sub-paths leading to value interfaces with value exchanges 1-5. Thus, we actually model that one of these five exchanges can happen, including the value exchange 1 that represents the ideal behaviour. The second OR-fork, that is located in the seller's box, splits into two paths leading to the exchanges between the seller and the tax office, modelling that the seller has two choices: either to pay taxes, or not to pay taxes. With the path modelled this way, we assume that decisions about exchanges between the seller and the tax administration are independent of decisions about exchanges between the seller and the buyer, and vice versa.

Sub-ideal Value Exchanges, Ideal and Sub-Ideal Dependency Paths

Value objects can also be **correct** or **incorrect**. As shown in [Insert about here Table 1**, value exchanges** can be ideal or sub-ideal, depending on the configuration of ports and objects connected to this value exchange. Ideal value exchanges are connected to an ideal port and an ideal value object. In sub-ideal value exchanges, there is either an incorrect value port, or an incorrect value object, or both. Thus, the following three configurations of a port and an object make the value exchange sub-ideal: (1) an ideal value port and an incorrect value object (e.g. delivery of a CD instead of the ordered DVD), (2) an incorrect value port and an ideal value object (e.g. late delivery), and (3) an incorrect value port and an incorrect value object (e.g. late delivery of an incomplete order).

[Insert about here Table 1. Definition of ideal and sub-ideal value exchanges]

Graphically, sub-ideal value exchanges, in case of both an object and exchange violation, are marked with a dashed line, as in Figure 2. For example, when no good or no payment is delivered, or no tax is paid. Additionally, the value objects of sub-ideal exchanges are given a different name than the corresponding value objects in the ideal value model, indicating that the right object was not exchanged (like, “No Goods”, “No Payment”, “Other Goods”).

Dependency segments, connected to interfaces, can be ideal or sub-ideal depending on the type of value exchanges connected to the value interface. An **ideal scenario segment** is connected to a value interface that has only ideal value exchanges. A **sub-ideal segment** triggers at least one sub-ideal value exchange.

Consequently, we define an **ideal path** as a dependency path that contains only ideal segments and has the same structure of segments and AND- and OR-forks as in the ideal value model; an ideal path represents an ideal situation. A **sub-ideal path** is a path that contains at least one sub-ideal segment. Thus, a sub-ideal path represents a sub-ideal situation. In Figure 3 the sub-ideal paths go through value exchanges, marked with dashed line (see value exchanges 2,3, 4, 5, 7).

Penalty Weights

In Figure 2 we modelled five exchanges where the principle of reciprocity does not hold. To support the design of control mechanisms, the following aspects of control problems should be addressed in an *e^3-value/control* model: who is the violating actor and how serious is the violation.

To address both issues, we introduce **penalty weights**. Following the solution suggested in [22] and [24], the violating actor will be assigned a penalty weight. The weight of the penalty can be considered as the *economic costs of violation*: the more severe the damage caused by the violation, the higher the costs of the violation, and therefore, the fine, i.e. penalty, for the actor should be higher. If there is no violation, the actor has a penalty of zero. Graphically, penalties are assigned to the segments connected to sub-ideal exchanges (see Figure 3).

[Insert about here Figure 3. Modelling penalties in *e^3-value/control*]

Subsequently, we address two issues related to penalties. First, we consider how to determine a violating actor, and secondly, how to determine the weight of the penalty.

Determining the violating actor. In some cases it is easy to determine the violating actor: if the goods are not delivered, then this is a violation by the seller. However, in the case of damaged goods, the responsibility for this violation can be more complex (e.g. the seller sent damaged goods, or the goods were damaged by the carrier etc.), and it depends on the specific terms of the contract how this responsibility is allocated. We assume that the execution of the sub-ideal value exchange is a responsibility of the actor that offers this object, not the party that accepts it. Hence, in Figure 3, where for simplicity reasons we included only value exchanges 2, 3 and 7 of the model in Figure 2, the non-zero penalties are assigned to the segments of violating actors. In the sub-ideal exchange 2, modelling non-payment, the buyer is the violating actor. In the sub-ideal exchange 3, modelling non-delivery of goods, the seller is responsible. This simple rule does not suffice in the sub-ideal exchange 5 in which there are two non-executed value exchanges, making it impossible to determine who is the violating actor. We need more information to settle this issue. In this model we assume that the tax office cannot violate an obligation (i.e. if the VAT is paid the legal compliance is always granted). This leaves us with the conclusion that in the sub-ideal exchange 5 the non-payment of VAT is the responsibility of the seller.

The weight of the penalty. In Figure 3, the segment at the buyer, modelling that the buyer has not paid, is assigned a penalty weight of 10; the segment at the seller for not delivering goods is assigned a penalty weight of 10, and the segment for not paying VAT is assigned a penalty weight of 2. To keep things simple, we use absolute numbers for weights in this paper to explain the method.[1] Obviously, the weights in figure 3 are rather arbitrary.

The weight of penalties are modelling issues that contract partners have to agree among each other. It is typical for contract drafting that the contract partners negotiate about the possible problems and contingencies that can occur during the execution of the contract and mutually agree on additional clauses to cover risks of non-performance of the other party. These extra contract clauses also include certain financial compensations, for example a reduction of the price if the goods are delivered too late. These are typical things that can be modelled with penalty weights, and they vary in different contracts. For these reasons, it is not possible to develop a precise method, which, without any domain knowledge, prescribes the contract partners, which specific numbers have to be assigned to a weight.

---

[1] In future research we will explore how these absolute weights can be replaced by a partial ordering on the penalty weights.

Although there is no precise method, we can give some heuristic guidelines how to determine penalty weights. Suppose, the seller ordered a box of DVDs, and the buyer delivered CDs; what should be the penalty for the seller? One argument could be that the seller should be punished less, than for not delivering them at all, since the seller did not violate the obligation to deliver, thus the value exchange occurred, only the value object was different. According to another argument, the seller should be punished more, because not only did the seller deliver an incorrect good, this good has to be stored for some time at buyer's side, and has to be transported back, which involves higher costs than with the non-delivered goods. The chosen argument will determine the weight of the penalty.

Weight of a Port

Because we assume that the failure to execute a value exchange or delivering an incorrect value object is a responsibility of the party that offers this object, not the party that accepts it, weights can only be assigned to *outgoing value ports* (value ports, having *outgoing* value objects) of the violating actor. Since the weight is related to the costs of the violation for executing sub-ideal value exchange, the following rule is used to assign weights:

A port connected to an ideal value exchange has a zero weight. A value port connected to a sub-ideal value exchange has a non-zero positive weight.

Weight of a Segment

A segment connected to a value interface can accumulate all penalties of the ports in this value interface, thus representing the total costs of violation if the segment is followed. Thus, the weight of a segment, connected to a value interface with at least one sub-ideal exchange, is always larger than the weight of the segment, connected to a value interface with no sub-ideal exchanges. The following rule is used to determine the weight of a segment:

The weight of a segment equals the sum of the weights of all value ports in the value interface to which this segment is connected.

For example, in Figure 4, the buyer and the seller exchange goods, and the buyer has to make two payments: one for goods, and another for the delivery. The model on the left shows that the buyer receives two fines: the weight 10 is assigned to the outgoing value port for not paying goods, and the weight 2 is assigned to the outgoing value port for not paying for the delivery. Thus, the buyer accumulated 12 points of fine as a segment weight. The model on the right shows that the buyer did not pay for the delivery, thus receiving 2 points of fine as in the previous case, but the value port with the "Goods payment" object has a zero weight, because the buyer pays for the goods; in total the segment weight of the buyer is 2. Graphically, weights are represented only at segments, not at value ports. This is done for the better readability of the models.

[Insert about hereFigure 4. Weight of the segment as a sum of the weights of value ports with outgoing value objects.]

Weight of a Dependency Path

Because the dependency path consists of scenario segments, and penalty weights can be assigned to segments, the dependency path can also have a weight.

The weight of the dependency path is the sum of the weights of all segments of the path.

The weight of an ideal dependency path is always 0, because the ideal path always goes through the value interfaces with ideal value exchanges and, thus, through value ports with zero weights. The weight of a sub-ideal dependency path is always a non-zero positive number: At least one segment of a sub-ideal path is connected to a value interface with a sub-ideal value exchange, which implies that the weight of the segment has a non-zero positive value.

Weight of an Actor

Each actor exchanges value objects via value ports. Actors can execute multiple exchanges with violations. Because penalty weights are numbers, they can be accumulated into a total penalty for the actor. Table 2 represents five sub-ideal paths for Figure 3. The last three columns indicate the actors. Each actor in each sub-ideal situation accumulates penalty weights, which are assigned to the individual segments.

[Insert about here Table 2. Penalties of actors: an actor’s view on modelling penalty weights for different sub-ideal paths]

For example, the “No Payment” value object is assigned to the buyer, because the value interface with outgoing value object “No Payment” belongs to the buyer, notifying that the buyer did not pay (see Figure 3). Consequently, we can say that for the buyer sub-ideal situations (A) and (D) are the worst because they have the highest weights 10, while other paths are equal to the path of the ideal situation. Similarly, for the seller, the worst sub-ideal situation is (E): it has the highest total weight 12 (no goods are delivered and no tax is paid). The situation (D), when the buyer did not pay for the goods, and the seller did not pay taxes, is worse for the buyer than for the seller: the buyer accumulates the weight 10, while the seller has only 2. For the tax office, every modelled situation is equal to ideal: the tax office is, in our example, supposed not to violate obligations.

To summarize, very often in network organisations, especially if a regulator is involved, penalty weights can be used to design incentive systems to influence the behaviour of actors. Thus, penalties not only identify a violating party, but also include an element of the control mechanism by modelling costs of violation and suggest a required punishment.

Modelling Remedies: Repair Value Exchanges

So far we only discussed penalty weights, related to punishments; however there also can be another type of penalty weights, which are related to remedy measures. The remedy measures can be added to a penalty system to create an opportunity for an actor to repair his violating behaviour. For example, a government can offer an opportunity for businesses that evaded tax payment, to pay this tax afterwards with some extra fee, thus, avoid the penalty, which in this case is, if being caught, to go to court and face severe charges such as imprisonments and bankruptcy. Thus, the government creates an incentive for businesses, which already violated the tax law, to pay taxes, and create an opportunity to reduce the penalty.

In deontic logic such a remedy, which is evoked when another obligation is violated, is called a *Contrary-to-Duty* obligation (CTD) (see [22]). A CTD tells you what you should do, if you have violated an obligation to exchange, or you have exchanged an incorrect object. For example, a CTD obligation of the buyer may be to return the received goods in case s/he did not pay. In this section we introduce **repair value exchanges** to model how actors can use CTD obligations to mitigate their non-compliance with the principle of reciprocity after they made a violation. Repair value exchanges model remedy measures for sub-ideal value exchanges.

In Figure 5 we model the obligations that the seller and the buyer have to fulfil to repair their sub-ideal behaviour. As in Figure 3, in Figure 5 the start stimulus at the buyer leads to the OR-fork, which branches into three sub-paths. One sub-path indicates the ideal situation: the seller delivers goods and the buyer pays in return. Hence, this segment is assigned a zero penalty. As in Figure 3, two other sub-paths lead to sub-ideal situations; in the first case, the buyer does not pay for the goods, and, in the second case, the seller does not deliver the goods. Non-executed value exchanges are modelled with dashed lines, and the fines of the buyer and the seller are modelled with weights assigned to the scenario segments at the violating actor.

[Insert about here Figure 5. Modelling Repair Transitions]

If the buyer did not pay for the goods, one possible solution that can help him to repair his behaviour is to pay for the goods at a later stage. In e^3-*value* terms, there is a value exchange of the value object “Payment” between the buyer and the seller. The corresponding value object “Goods” of the object “Payment” has already been exchanged in the earlier value exchange. Therefore, we model the late Payment with a pair of value interfaces that exchange only one value object (see Figure 5).

In case of two non-executed value exchanges in one value interface as in the exchange of VAT, the repair value exchange is to pay VAT and an additional fee (for example, administration costs caused by the delay in VAT payment). In this case the repair value exchange has two value objects, and is also valid in e^3-*value* terms.

The logic behind the repair value exchange is that it reduces the costs of the violation, and, therefore, reduces the punishment for the violating actor: By executing some remedy behaviour, an actor reduces the costs of violation, and, therefore, pays a smaller penalty. Therefore, the weight of the path that goes through the repair value exchange is reduced by the remedy:

Weight of Actor = Penalty Weight – Remedy Weight,

where *Weight of Actor* is the weigh the actor accumulated when executing the sub-ideal path, *Penalty Weight* – the weight of the segment connected to sub-ideal value exchange, *Remedy Weight* – the weight of the segment connected to the corresponding repair value exchange. In Figure 5, the weights assigned to the repair value exchanges are indicated with the "minus" sign.

The repair value exchange reduces the penalty paid by an individual actor, and can potentially reduce it to zero. However, on the other hand, one of the main ideas of penalty weights is to make an actor to prefer ideal paths and not to violate at all. Therefore, we propose that although by executing the repair exchange the actor corrects its behaviour, the total penalty weight for this path should always be larger than the penalty for ideal path (which is zero). Hence, the following rule holds:

If the path passes through at least one repair value exchanges (the remedy path), then the total weight of this path should be more than the total weight of the ideal path.

In Figure 5 the scenario segment connected to the buyer's repair value exchange of "Payment" has the weight –9, the scenario segment connected to the seller's repair value exchange of "VAT" has the weight –1. This makes the total weight of the sub-ideal path that contains these exchanges 10 + 0 + 0 - 9 = 1 (if the seller paid VAT), or 10 + 0 + 0 + -9 + 2 - 1 = 2 (if the seller did not pay VAT). Similarly, the scenario segment connected to the seller's repair value exchange of the value object "Goods" has the weight –4 which makes the whole dependency path weight equal to 1 (if the seller paid VAT) or 2 (if the seller did not pay VAT). All the weights of sub-ideal paths are larger than 0, indicating that these are sub-ideal scenario paths.

Modelling the Renewable Energy Case with *e^3-value/control*

An interesting case of a complicated business network is the renewable energy case. In order to comply with international environmental agreements, such as the Kyoto protocol, governments implement different regulations, that would ensure sufficient generation and supply of green energy. The technologies to generate renewable electricity, such as wind turbines, photovoltaic panels, hydro generators, etc., require high initial investments, and, therefore, the price of the renewable electricity (or green electricity) is higher than the price of electricity produced by conventional generators (further referred to as grey electricity). Consequently, government regulation is needed to guarantee that electricity companies will use these, commercially less attractive, technologies.

The regulation implemented in Great Britain, starting from April 2002, is based on the *Renewable Obligation* (RO) regulation [19], which *prescribes* that a certain percentage of energy delivered by energy suppliers to consumers should be green energy. The RO regulation is complex and a special government organisation, *Office of Gas and Electricity Markets* (Ofgem), has been established to carry out the administration of the RO regulation. Ofgem accredits electricity producers that are capable of generating electricity from eligible renewable sources. Suppliers are required to produce evidence to the Ofgem of their compliance with the RO rules. An important evidence token is the so-called *Renewable Obligation Certificate* (ROC). The supplier receives a ROC, when it buys electricity from an accredited renewable energy producer. A ROC can also be traded, and a supplier can resell its ROCs to other suppliers. This has created a market for ROCs. Alternatively, a supplier can be discharged from its Renewable Obligations, in whole or in part, by paying the so-called *buy-out* price (a penalty) to Ofgem, if he is short on ROCs. Additionally, the fund of buy-out fees, collected by Ofgem, is distributed by Ofgem among the suppliers, proportional to the number of ROCs the supplier has. Hence, a supplier receives a kind of additional bonus for each ROC it possesses. In the next sections, we analyse this RO regulation using the *e^3-value/control* modelling framework.

Ideal Value Model

Figure 6 presents an ideal value model for the renewable energy case. In order to satisfy a customer's need for energy, the supplier provides electricity and obtains a retail fee in return for that. The supplier has a choice (as denoted by the OR-fork, represented by the triangle in the segment of the supplier). He may decide to obtain grey electricity from a non-renewable generator and pay in return a fee; or he can decide to buy electricity from a renewable generator and pay a fee plus an extra fee, because green electricity, produced by the renewable generator is more expensive than the grey electricity produced by non-renewable generator. We assume in this model that for the consumer there is no price difference whether s/he buys grey or green electricity. Other cases that exist are, for instance, where the supplier charges a customer an additional fee for green energy.

[Insert about here Figure 6. Ideal value model.]

The supplier has to comply with the renewable obligation (RO) and buy at least 10% of its supply from renewable generators. The government sets this limit of 10%. The government organisation to which suppliers have to report about RO-compliance is Ofgem. Ofgem operates on behalf of the government and requests ROCs from the supplier and in return acknowledges that the supplier complies with the RO regulation. Ideally, every supplier has to show to Ofgem 10% of green supply, and then Ofgem will grant the RO compliance.

Sub-Ideal Value Model

A typical case of sub-ideal behaviour is that the supplier buys a lower percentage of green energy than the percentage prescribed by the government. As the second step of the framework requires, we develop in Figure 7 a sub-ideal model to identify these control problems. This is modelled by the second OR-fork that appears in the most-left sub-path, and which leads to two sub-paths. The left sub-path leads to the exchanges with the renewable generator, and the right sub-path leads to the exchanges with the non-renewable generator. The right sub-path, has a non-zero penalty (Penalty > 0), indicating that this is a sub-ideal path, while the left sub-path has a zero penalty weight (Penalty = 0), thus it is an ideal sub-path.

The sub-ideal path is the situation where the supplier buys insufficient green electricity. It represents that the supplier does not comply with the norms set by the government: the supplier buys less than the required amount from a renewable generator (for example, 2%), and the remaining amount (8%) from a non-renewable one. Therefore, we introduce additional sub-ideal value exchanges between Ofgem and the supplier, modelling that the RO compliance is not granted to a supplier who fails to deliver an evidence of 8% green electricity supply.

[Insert about here Figure 7. Sub-ideal value model.]

The sub-ideal value exchange is between Ofgem and the supplier, and it has two sub-ideal value exchanges: *"No RO compliance"* by Ofgem, and *"No* *evidence of 8% green electricity supply"*. As in the Tax Office example (see Figure 3)

According to our framework for modelling IOCs; the first step is to analyse the control problems in terms of sub-ideal scenarios, and after that the second step is to design appropriate control mechanisms. The first question is how the government actually can detect a sub-ideal situation; this refers to *detective controls*. A second question is how the government can prevent a supplier to take a sub-ideal path; this refers to *preventive controls*. A third question is how remedies may be applicable in case a supplier does violate the RO rules.

Detective Controls

To ensure that Ofgem can detect the execution of a sub-ideal path, the government introduced *Renewable Obligation Certificates* (ROCs). Each renewable generator applies to Ofgem for accreditation, and if it can prove that it generates green energy, the renewable generator receives the right from Ofgem to issue ROCs. The supplier receives from an accredited renewable generator one ROC for each MWh electricity it buys from this generator. Periodically, the supplier has to declare to Ofgem the total amount of electricity it sold and the number of ROCs it possesses. So, ROCs are something of value, because the supplier can prove with these ROCs that he fulfilled his obligation to supply sufficient renewable energy. The resulting value model is presented in Figure 8. In this model the supplier receives a ROC for each MWh renewable electricity it purchased, which is used to prove that the electricity is indeed from a renewable source. Additionally, we model that all renewable generators receive ROCs, thus, we model only those generators that were accredited to produce green energy.

[Insert about here Figure 8. Sub-ideal value model with ROCs.]

The AND fork shows that the supplier exchanges values with the renewable generator and Ofgem. Clearly, in case of exchanges with a non-renewable generator, no ROCs are exchanged. Ofgem can prove when suppliers have executed a sub-ideal path, because then they have a shortage of ROCs relative to the amount of electricity they have sold.

The sub-ideal path is the one that goes through the sub-ideal exchange with value objects "No ROCs" and "No RO compliance". In this case the supplier does not buy sufficient ROCs and is charged with a *Non-compliance penalty. Non-compliance penalty* represents costs of violation for the supplier. This is a fee that the supplier has to pay to the the government when it did not comply with RO obligation and did not buy sufficient green electricity.

[Insert about here Figure 9. Value model representing trading of ROCs.]

There are a number of reasons why suppliers do not always follow the ideal path. For instance, apart from fraudulent behaviour, it can happen that there is simply insufficient renewable electricity capacity available at a certain point in time. Consequently, the government introduced a mechanism that allows for trading of ROCs (see Figure 9). In case of insufficient renewable electricity power, the supplier may decide to buy grey electricity, *and* to buy ROCs from another supplier. In this way, the supplier can still prove that he fulfilled his obligations. Obviously, the supplier selling ROCs will only do so if he possesses already sufficient ROCs. We consider this scenario still an ideal path (thus marked with the zero weight), because seen from a green electricity generation perspective, overall sufficient renewable electricity is generated. Note that there are three paths that can lead to the exchange between the supplier and the non-renewable generator; two are ideal paths and one is a sub-ideal path.

Preventative Controls and Remedies

Despite a lack of renewable electricity power, situations can occur where the supplier really performs the sub-ideal path. In Figure 10 the supplier may still decide to buy non-renewable electricity, despite his obligation to buy renewable electricity. He then follows the right-most part of the second OR fork denoting his choice to execute the sub-ideal path. In this part, an AND fork occurs, representing that the supplier obtains grey electricity *and* that he pays a so-called buy-out fee to Ofgem for obtaining non-renewable electricity. The buy-out fee can be considered as a kind of remedy by the supplier for not complying with the obligation to present sufficient ROCs to Ofgem. Ofgem can calculate this buy-out fee, because we have shown that there is a detective control that is used by Ofgem to check by counting the ROCs whether suppliers buy sufficient renewable electricity.

[Insert about here Figure 10. Remedies for sub-ideal behaviour of Supplier.]

In this case, the possibility to pay the buy-out fee offers a repair exchange for the supplier. By paying the buy-out fee the supplier is not immediately penalized with maximum charges, but it is given the opportunity to repair its sub-ideal behaviour by executing the repair value exchange. As shown in Figure 10, the remedy equals *Buy-Out Fee – Extra,* where *Extra* is the difference in price between one MWh of renewable electricity and non-renewable electricity, and *Buy-Out Fee* is the buy-out price of one ROC. This reflects the logic that a supplier, who did not buy sufficient green energy will loose the economic advantage, because he has to pay the difference in price of the buy-out fee and the extra costs he would have made if he had bought the green energy. Ofgem regulates the penalty of the supplier on behalf of the government. The penalty assigned to the sub-ideal path is *Non-compliance Penalty – (Buy-Out Fee - Extra).* For a successful regulation of this case, the following equations should hold:

$$\textit{Non-compliance Penalty} - (\textit{Buy-Out Fee} - \textit{Extra}) > 0,$$

$$\textit{Buy-Out Fee} - \textit{Extra} > 0.$$

By satisfying these equations the government creates a preventative control to ensure that suppliers prefer to perform the ideal path and buy green energy. If the buy-out fee is more than *Extra*, then the penalty weight assigned to the sub-ideal path is more than zero, and the model shows that the supplier has an incentive to take the ideal path rather than the sub-ideal one. If the buy-out fee equals *Extra*, the weight of the sub-ideal path becomes zero, i.e. equal to the weight of the ideal path; and then there is no difference for the supplier whether to buy ROCs or to pay the buy-out. If the buy-out fee is less than *Extra*, then the weight of the sub-ideal path is less than zero. In this case, the supplier has no longer an incentive to buy green energy, and will prefer to pay the buy-out fee instead. The *Extra* component of the equation is determined by free market mechanisms, and cannot be regulated by Ofgem. Hence, Ofgem has to set the levels of the non-compliance penalty and the buy-out fee such that it keeps the weight of the sub-ideal path well above zero.

An Additional Preventive Control: the Pot of Buy-Out Fees

The buy-out fees collected by Ofgem are used as an additional preventative control (see Figure 11). All collected buy-out fees, are periodically returned by Ofgem to the suppliers proportional to the number of ROCs they presented to Ofgem in that period. This is modelled with the value exchange "Pot Fee". So, ROCs can be seen as shares in a so-called Pot of collected buy-out fees. For example, if a supplier S presented *x* percent of the total number of ROCs presented in that period, then it will also receive back *x* percent from the Pot of collected buy-out fees in the period.

The "pot" fee is an additional incentive for suppliers to behave ideally; they receive money for each ROC presented to Ofgem, and, therefore, it can be considered as an additional preventive control. The control aspect of the "pot" fee is reflected in the change in the remedy formula. Now the remedy of the supplier is larger. For example, if the difference in price *Extra* is 4 cent, the *Pot Fee* is 3 cent, and the *Buy-Out Fee* is 10 cent, then in case the "pot" fee is not paid, the remedy for taking sub-ideal path for supplier is *Buy-Out Fee- Extra = 6 cent*, and in the case the pot fee is paid, the remedy of the supplier is *Buy-Out Fee – Extra + Pot Fee = 9 cent,* which means that supplier's costs would be reduced even more, if he would choose the ideal path. Additionally, in this case Ofgem should set the *Non-compliance Penalty* at such a level that *Non-compliance Penalty* > *Buy-Out Fee – Extra + Pot Fee*.

[Insert about here Figure 11. The Buy-Out Fee Pot as an additional preventive control.]

To summarize, in this case we identified a control problem with the supplier as violating actor, and we focussed on one specific sub-ideal value exchange. To model this control problem it appeared useful to add the sub-ideal exchange and extend the dependency path with a sub-ideal path. To model the control mechanisms, we added actors to the model, namely another supplier market segment. Additionally, the control mechanisms, based on penalties, suggest some guidelines for regulative decisions of Ofgem, such as the level of the buy-out fee. The final model contains new value exchanges compared to the initial ideal model (see Figure 6), indicating how the modelling of the control mechanism requires an extended network structure. The penalty weight in this case depends on the value of other value objects in the model, namely on the price of the green and grey electricity.

Conclusions

In this paper we introduced the design tool e^3*-value/control* for designing controls for network organisations. The e^3*-value/control* tool is part of a methodology to analyse and design inter-organisational control mechanisms. This methodology is based on the idea that the design of control mechanisms for network organizations should start with the modelling of economic value exchanges rather than business processes. Clearly, business processes are essential for implementing and realizing economic value exchanges, but we argued that for the proper analysis of the control issues one should abstract from the processes and focus on the underlying value exchanges instead. Hence, we agued that a methodology for designing control mechanisms should consist of the following sequence of steps:

1. Design of the value exchanges in a business model;
2. Analyse control problems of this business model, and modelling of these problems in sub-ideal scenarios
3. Design of Inter-organisational control mechanisms (IOCs) to solve these control problems.

Since we take the modelling of value exchanges as starting point in our methodology, we based our tool development on the business modelling tool e^3-

value, which focuses completely on the value modelling aspect of business models rather than modelling the process details how to realize these value exchanges. However, the e^3*-value* methodology has as primary focus the design of a business *value* model, and it does not provide support for the analysis of control problems and the design of explicit control mechanism. This is most noticeable in the *Principle of Reciprocity*, which assumes that value exchanges always occur in the proper way. This is clearly an idealization, since the main reasons to introduce control mechanisms is precisely that value exchanges often are not guaranteed to occur. Hence, we called this the ideal model, and we extended the e^3*-value* tool with concepts to model sub-ideal scenarios, which violate the Principle of Reciprocity, hence we called these extra scenarios in the value model the sub-ideal scenarios or paths. We also argued that these sub-ideal paths could be viewed as possible violations of the contractual obligations underlying this value model. We used concepts and ideas from deontic logic (the logic of obligations and its violations) to develop an extension of e^3*-value* called e^3*-value/control*. The two most important extensions of e^3*-value/control* are the modelling of sub-ideal exchanges and paths, and the introduction of penalty weights. The penalty weights indicate the costs of the violation of the obligation to exchange value. These extensions are implemented in the e^3*-value/control* design tool for modelling violations of obligations to exchange and control mechanisms to prevent and detect these violations. We used the ROC case to show how the tool can be applied in a complicated real-life case. This case study showed that the e^3*-value/control* tool is useful for designing control mechanisms for inter-organisational collaboration in network organisations.

Acknowledgements. The research of the first author is funded by the Post Master EDP Audit Education of the Vrije University Amsterdam.

REFERENCES

1. Alt, R. and Zimmerman H.D. Preface: Introduction to Special Section – Business Models. *Electronic Markets*, 11, 1 (2001), 3-9.
2. Barkai, D. *Peer-to-Peer Computing.* Santa Clara, CA: Intel Press, 2001.
3. Buhr R.J.A. Use case maps as architectural entities for complex systems. *IEEE Transactions on Software Engineering*, 24, 12 (1998), 1131–1155.
4. Daskalopulu, A. and Sergot, M. The Representation of Legal Contracts. *AI and Society*, 11, 1 (1997), 6-17.
5. Dekker, H.C. Control of inter-organisational relationships: evidence on appropriation concerns and coordination requirements. *Accounting, Organisation and Society*, 29, 1 (2004), 27-49.
6. Elsas, P.I. *Computational Auditing,* PhD Thesis. Amsterdam: Vrije Universiteit Amsterdam, 1996.

7. Foster, I.; Kesselman, C.; Tuecke, S. The Anatomy of the Grid: Enabling Scalable Virtual Organisations. *International Journal of Supercomputer Applications*, 15, 3 (2001).

8. Geerts, G. and McCarthy, W.E. An Accounting Object Infrastructure for Knowledge-Based Enterprise Models. *IEEE Intelligent Systems and Their Applications*, 14, 4 (July-August 1999), 89—94.

9. Gordijn, J. *Value-based Requirements Engineering - Exploring Innovative e-Commerce Ideas*, PhD thesis. Amsterdam: Vrije Universiteit Amsterdam, 2002. (Accessed on "http://www.cs.vu.nl/~gordijn").

10. Gordijn J. and J.M. Akkermans. Value-Based Requirements Engineering: Exploring Innovative E-Commerce Idea. *Requirements Engineering Journal*, 8, 2 (2003), 114-134.

11. Gordijn, J. and Tan, Y.-H. A Design Methodology for Modeling Trustworthy Value Webs. *International Journal of Electronic Commerce*, 9, 3 (2005).

12. Kartseva, V., Gordijn, J. and Tan, Y.-H., Value-based Business Modelling for network organizations: Lessons Learned From the Electricity Sector. *Proceedings of the 12th European Conference of Information Systems*. Turku (June 2004), on CD.

13. Lee, R.M. and Bons, R.W.H. Soft-Coded trade procedures for Open-EDI. *International Journal of Electronic Commerce*. 1, 1 (1996), 27-49.

14. Malone, T. W., Crowston, K., Pentland, B., Dellarocas, C., Wyner, G., Quimby, J., Osborn, C.S., Bernstein, A., Herman, G., Klein, M., O'Donnel, E. Tools for Inventing Organizations: Towards a Handbook of Organizational Processes. *Management Science*, 45, 3 (1999), 425—433.

15. McCauley, S. Non-contractual relations in business: A preliminary study. *American Sociological Review*, 28 (1963), 55-67.

16. McKnight, D.H., Cummings, L.L. and Chervany, N.L. Initial trust formation in new organizational relationships. *Academy of Management Review*, 20, 3 (1998), 472-490.

17. Meyer B. On Formalism in Specifications. *IEEE Software*, 2, 1 (1985), 6—26.

18. Morris, M.; Schindehutte, M.; and Allen, J. The entrepreneur's business model: Toward a unified perspective. *Journal of Business Research*, 58 ,6, (2005), 726-735.

19. Office of Gas and Electricity Markets. *The Renewables Obligation, Ofgem's first annual report*. (February 2004), (Accessed on "www.ofgas.gov.uk").

20. Osterwalder, A. and Pigneur, Y. An e-Business Model Ontology for Modeling e-Business. *Proceedings of 15th Bled Electronic Commerce Conference*. Kranj: Moderna Organizacija, 2002, on CD.

21. Pateli, A.G., Giaglis, G.M. A Research Framework for Analysing Business Models. *European Journal of Information Systems*, 13, 4 (2004), 302-304.

22. Raskin, J., Tan, Y. and van der Torre, L. How to model normative behavior in Petri nets, *Proceedings of the 2nd Modelage Workshop on Formal Models of Agents*, Sesimbra, 1996, 223-241.

23. Ronmey, M.B. and Steinbart, P.J. *Accounting Information Systems (9th edition)*. New Jersey: Prentice Hall, 2003.

24. Tan, Y. and Thoen, W. A Logical Model of Transfer Obligations in Trade Contracts. *Accounting, Management and Information Technologies*, 8 (1998), 23-38.

25. Tan, Y. and Thoen, W. A Survey of Electronic Contracting Related Developments, *Proceedings of 14th Bled Electronic Commerce Conference*, Kranj: Moderna Organizacija, 2001.

26. Tapscott, D., Ticoll, D. and Lowy, A. *Digital Capital – Harnessing the Power of Business Webs.* London: Nicholas Brealy Publishing, 2000.

27. Uschold, M., King, M., Moralee, S. and Zorgios, Y. The Enterprise Ontology. *The Knowledge Engineering Review*, 13, 1 (1998), 31—89.

28. Vaassen, E.H.J. *Accounting Information Systems, a Managerial Approach*. Chichester: Wiley, 2001.

29. Weill, P. and Vitale, M.R. *Place to Space : Migrating to eBusiness Models*. Boston: Harvard Business School Press, 2001.

30.

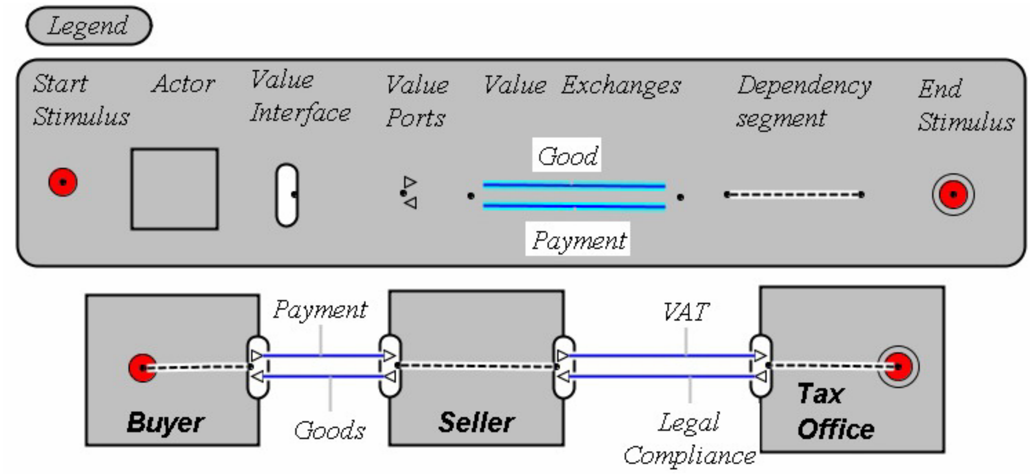

Figure 1. e^3-*value* model of a Purchase with Tax payment.

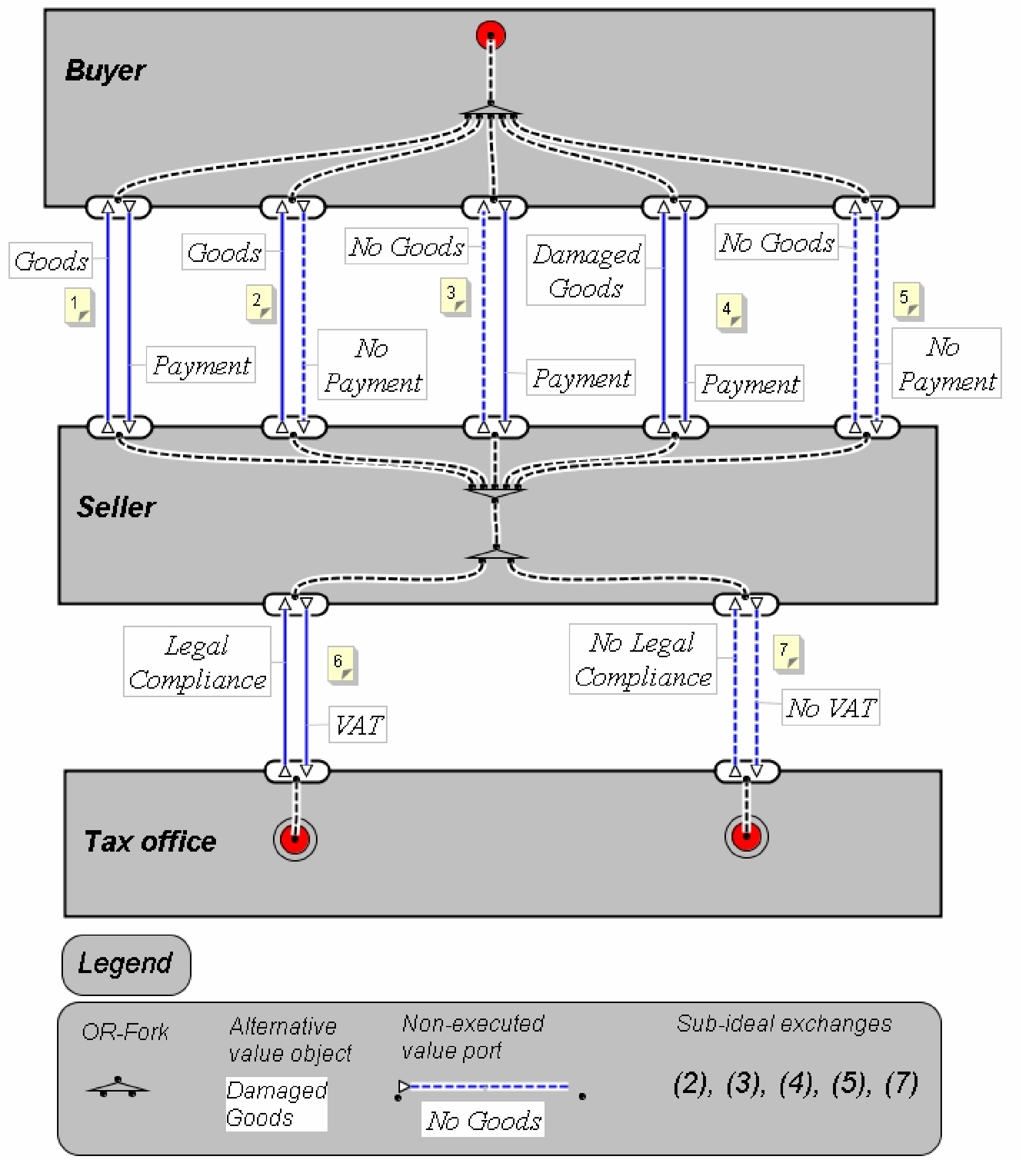

Figure 2. Some sub-ideal scenarios in the primary value web modelled with e^3-*value/control*.

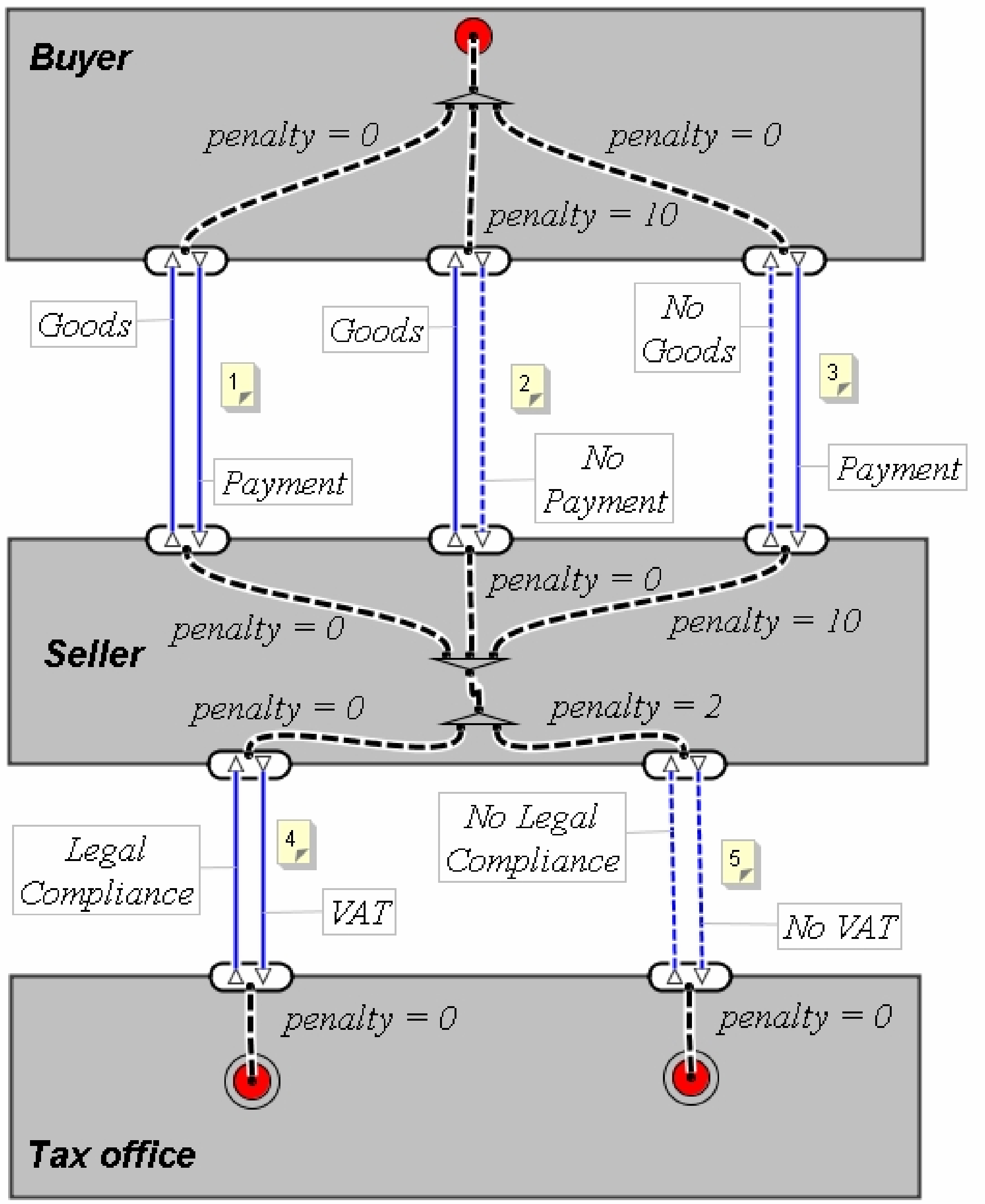

Figure 3. Modelling penalties in e^3*-value/control*.

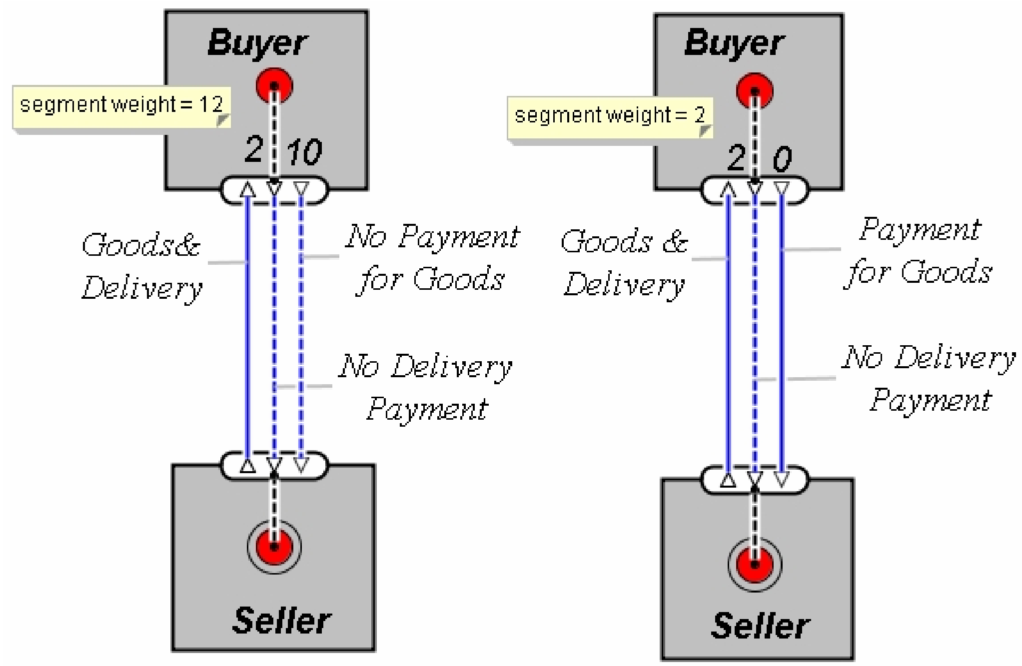

Figure 4. Weight of the segment as a sum of the weights of value ports with outgoing value objects.

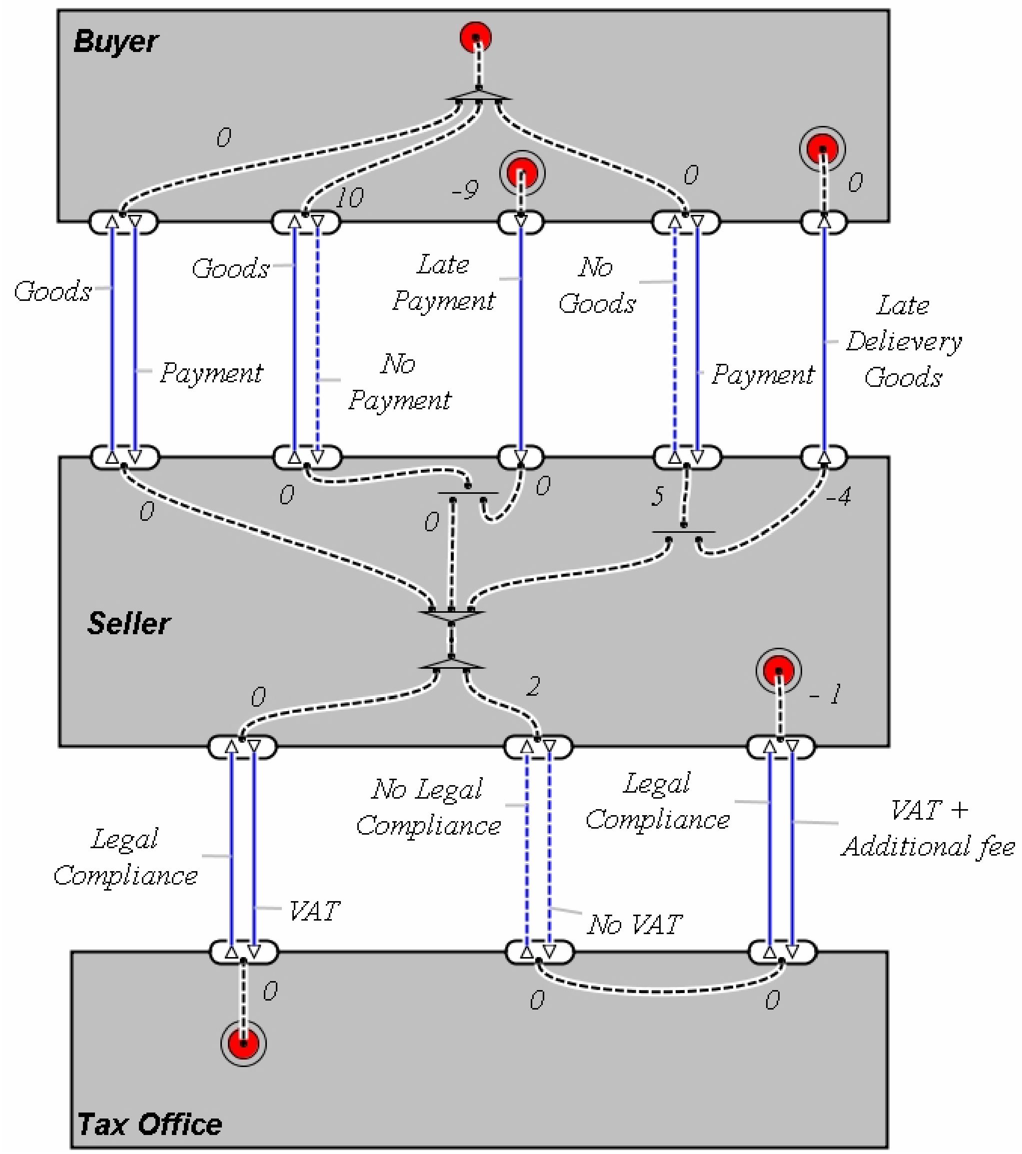

Figure 5. Modelling Repair Transitions.

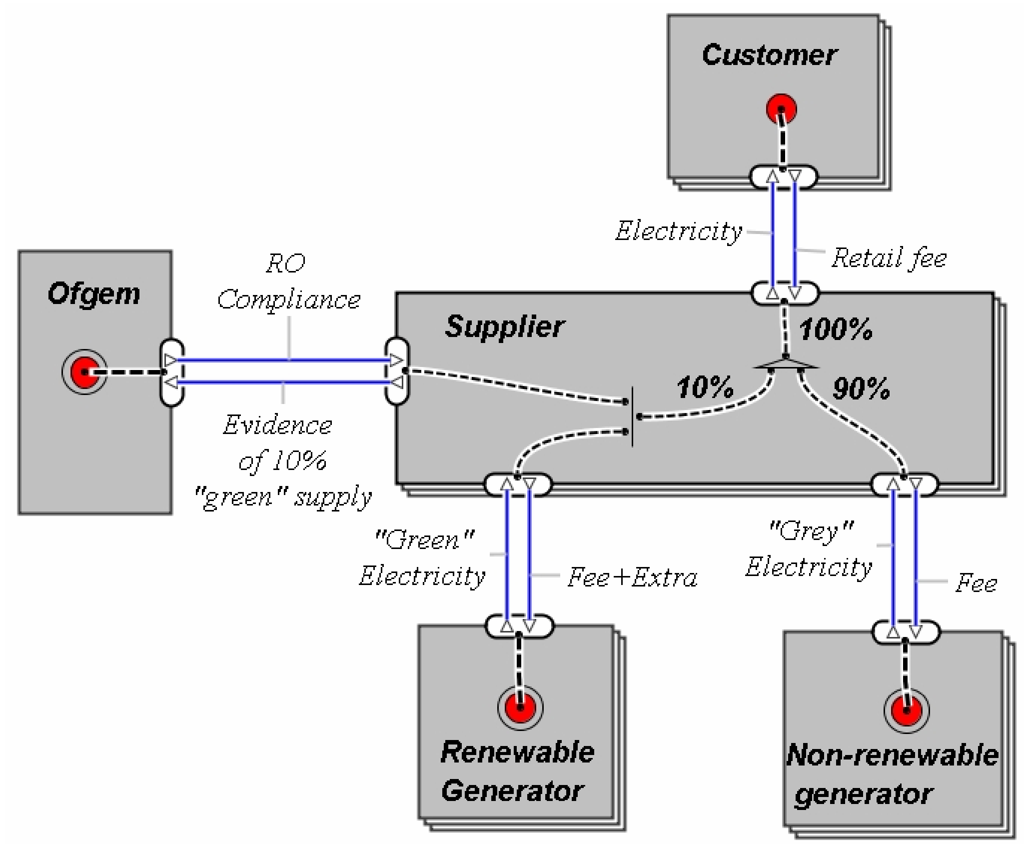

Figure 6. Ideal value model.

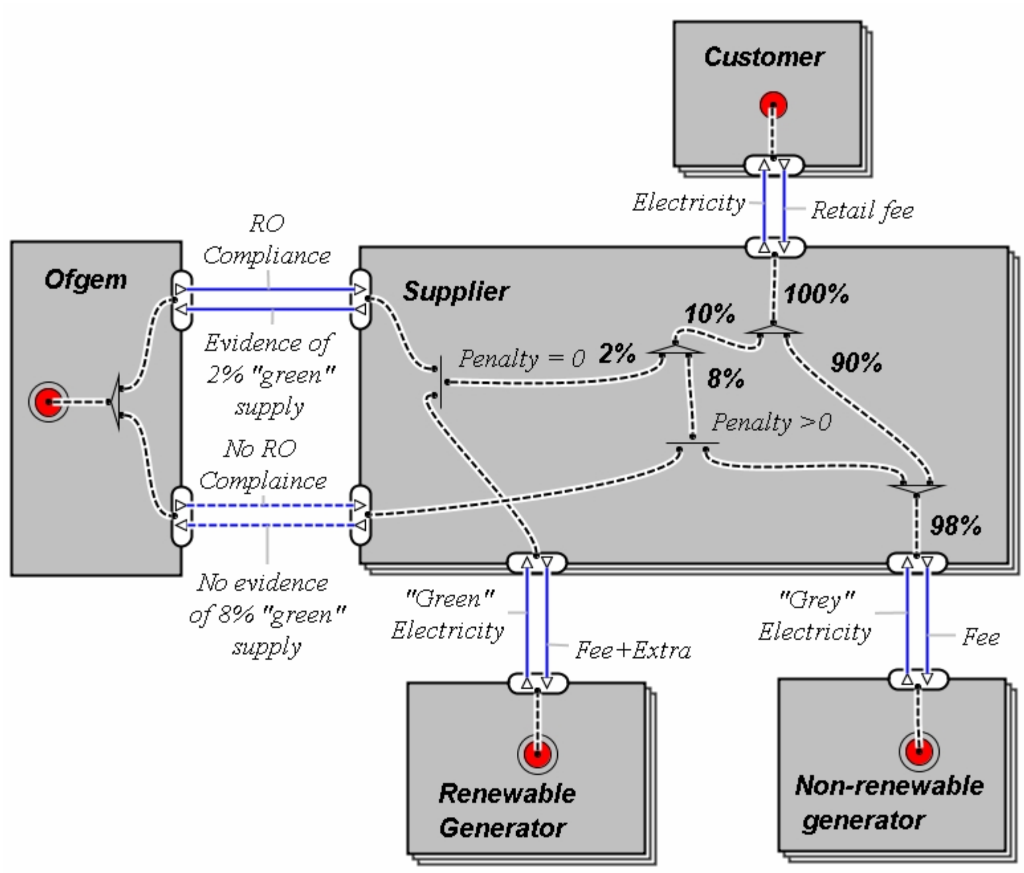

Figure 7. Sub-ideal value model.

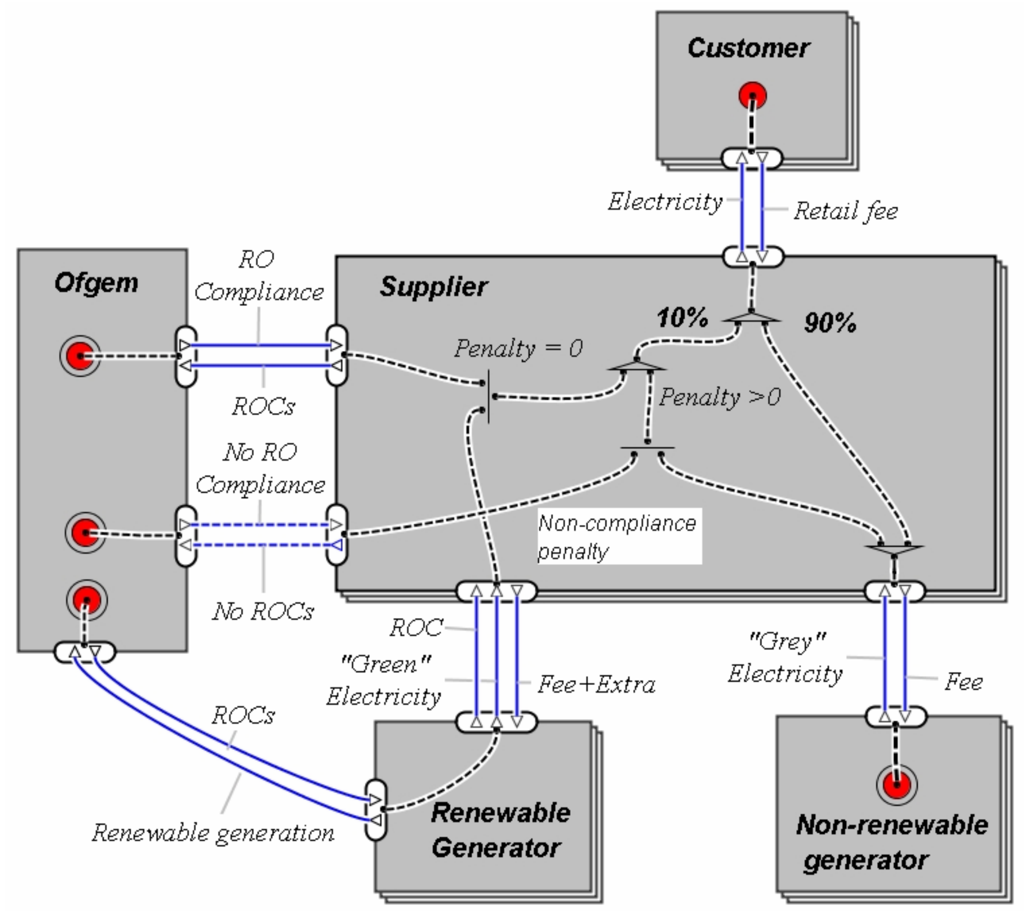

Figure 8. Sub-ideal value model with ROCs.

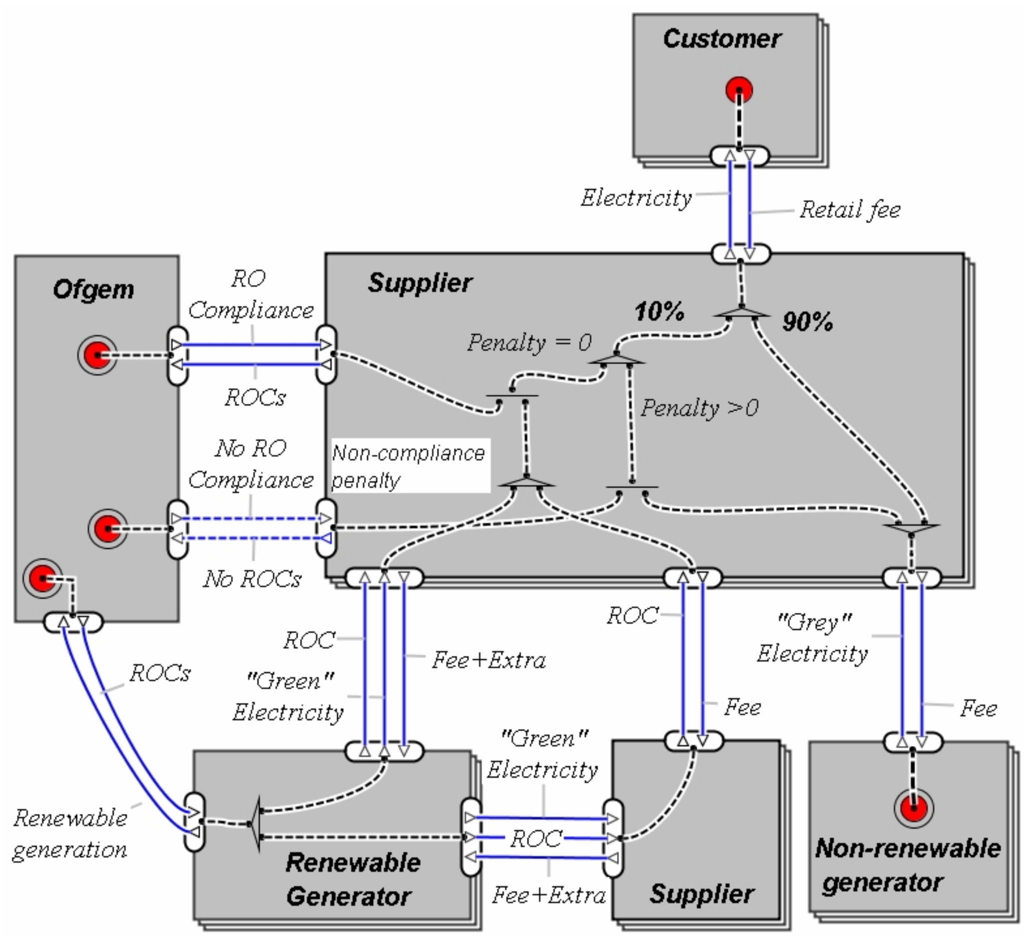

Figure 9. Value model representing trading of ROCs.

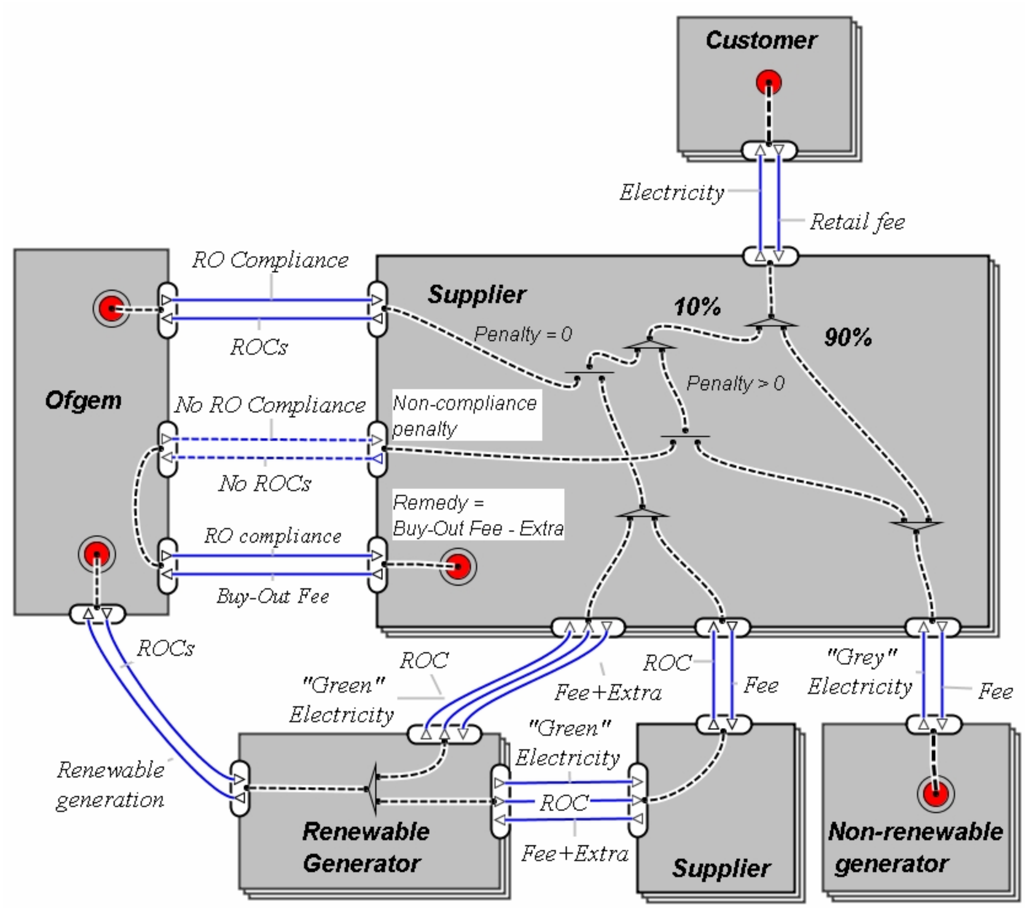

Figure 10. Remedies for sub-ideal behaviour of Supplier.

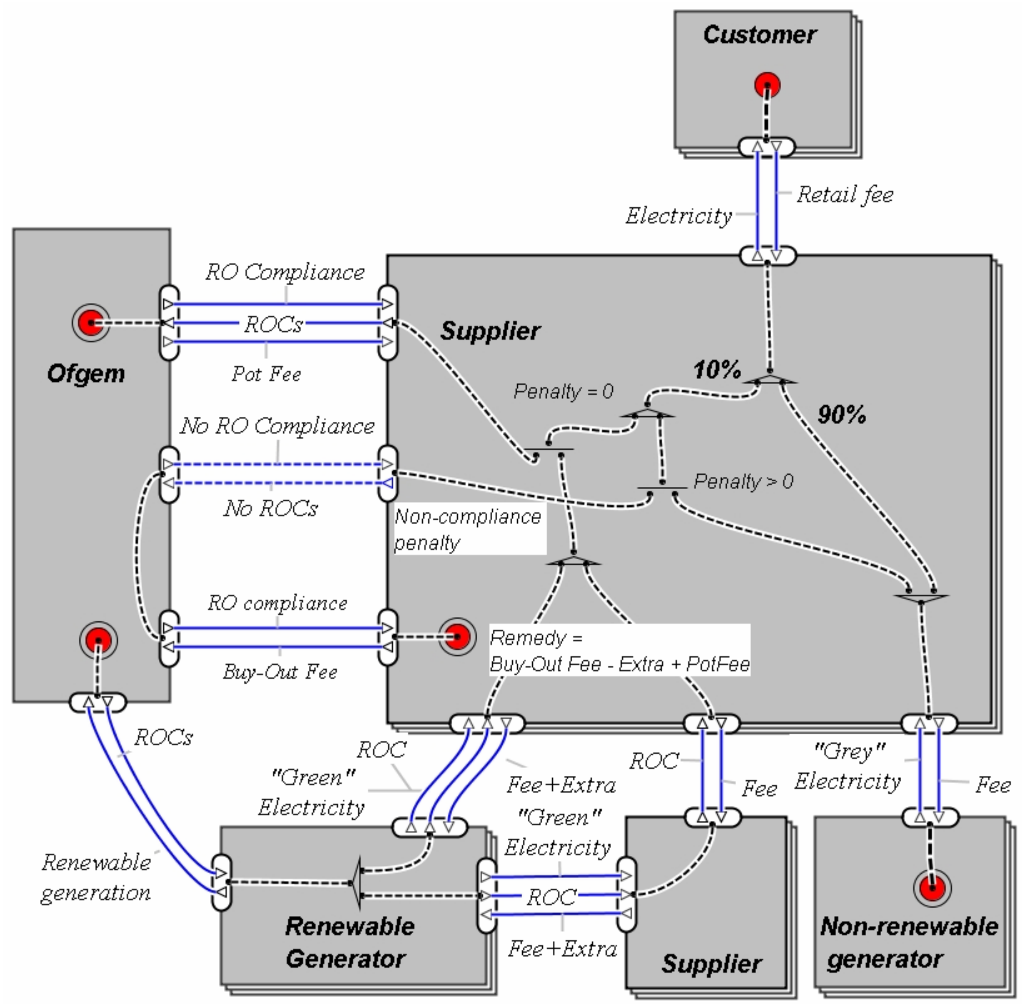

Figure 11. The Buy-Out Fee Pot as an additional preventive control.

		Value object	
		Correct	**Incorrect**
Value port	**Correct**	Ideal value exchange	Sub-ideal value exchange
	Incorrect	Sub-ideal value exchange	Sub-ideal value exchange

Table 1. Definition of ideal and sub-ideal value exchanges

	Sub-ideal path	Empty value objects		
		Buyer	Seller	Tax Office
A	The goods are delivered, the fee is not paid, and the tax is paid	No Payment (w=10)	w = 0	w = 0
B	The goods are not delivered, the fee is paid, and the tax is paid	w = 0	No Goods (w=10)	w = 0
C	The goods are delivered, and the fee is paid, but the tax is not paid	w = 0	No VAT (w = 2)	w = 0
D	The goods are delivered, the fee is not paid, and the tax is not paid	No Payment (w=10)	No VAT (w = 2)	w = 0
E	The goods are not delivered, the fee is paid, and the tax is not paid	w = 0	No Goods (w=10) No VAT (w =2)	w = 0

Table 2. Penalties of actors: an actor's view on modelling penalty weights for different sub-ideal paths