Analysing preventative and detective control mechanisms in international trade using value modelling

Vera Kartseva
Department of Economics and Business Administration
Free University Amsterdam
De Boelelaan 1105, 1081 HV
Amsterdam, The Netherlands
+31 20 444 6059
vkartseva@feweb.vu.nl

Jaap Gordijn
Department of Informatics
Free University Amsterdam
De Boelelaan 1105 1081 HV
Amsterdam, The Netherlands
+31 20 444 7769
gordijn@feweb.vu.nl

Yao-Hua Tan
Department of Economics and Business Administration
Free University Amsterdam
De Boelelaan 1105, 1081 HV
Amsterdam, The Netherlands
+31 20 444 6057
ytan@feweb.vu.nl

ABSTRACT

Exploration and development of e-business models takes a series of viewpoints. One important perspective is the value web perspective, which can be modelled using the *e^{3}value* methodology. However this perspective supposes a perfectly honest world, an assumption which turns out to be not true in practice. As a consequence, it is important to explore fraudulent behaviour of actors in value web, preferably by using a lightweight and graphical approach similar to the *e^{3}value* methodology. Using a real-life e-business scenario (Bill of Lading) we show that *e^{3}value*, with some extensions, is capable of modelling a control perspective that captures mechanisms for preventive and detective controls. Additionally, we show that such controls themselves can be seen as a kind of commercial services.

Keywords

e-business modelling, viewpoints, value webs, control mechanisms, *e^{3}value*

1. INTRODUCTION

Various definitions of the notion of 'business model' (see [9] for an overview) agree that a business model shows a way of doing business from multiple perspectives. One of the viewpoints on ‘business model’ is found in the *e^{3}value* methodology. This perspective, called a value web, represents the creation, distribution, and consumption of economic value in a network of multiple enterprises and end-consumers and [5] supposes that each enterprise behaves correctly. The concern here is to come up with a model that seems to be economically sustainable for each actor involved.

Another perspective to be addressed while developing a business model is the fraudulent behaviour of actors in network of organisations. Such a perspective would enable the design of *control mechanisms* to prevent possible damage from the fraudulent behaviour of some actors. It is an addition to a value web that, in contrast, assumes honest behaviour of actors. Many authors ([1],[3],[6],[7],[8],[12],[13],[14]) have been working on modelling control mechanisms and related to that, trust issues. From a business modelling and development perspective, it is important to do such modelling sufficiently lightweight, and preferably graphically. During business development, there is only limited time for modelling; a graphical representation of control and trust issues is preferable because only then these issues can be easily communicated to various stakeholders. In this paper we explore the usability of the already lightweight and graphically oriented *e^{3}value* methodology to model control and trust issues. We present an example from the international trade, the Letter of Credit procedure, and we suggest how the theory of control of a network of organisations can be build based on the *e^{3}value* methodology.

The remainder of this paper is structured as follows. Section 2 focuses on modelling value webs with *e^{3}value*. Section 3 describes the control perspective on networks of organisations, and proposes how to analyse sub-ideal scenarios, as well as preventive and detective control mechanisms in networks of organisations using *e^{3}value*. Finally, in section 4 we draw conclusions and suggest a further research agenda.

2. MODELLING NETWORK ORGANISATIONS WITH *E^3-VALUE*

The *e^{3}value* methodology [4] has been developed to model a value web consisting of actors who create, exchange, and consume things of *economic value*. It has been used to model value webs in various industries, e.g. the music, finance, internet service provisioning, news and energy industry [5]. Moreover, tool-support is available (see http://www.cs.vu.nl/~gordijn/research.htm).

In Figure 1 a value web is represented, modelling that a supplier offers some object of value to a customer and obtains a fee in return. It is one of the easiest *e^{3}value* models we can make. This simplicity has two purposes; first we need to introduce the *e^{3}value* concepts clearly, second it allows to focus on control issue rather than on modelling the value web itself. In addition to that, most real-life *e^{3}value* models fit on a few pages. This is actually a main contribution of the methodology: to communicate concisely a value web to the stakeholders involved.

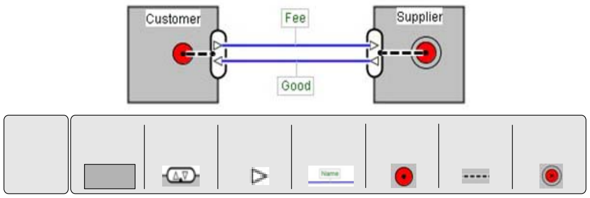

Figure 1: A supplier and a customer exchanging objects of value.

Figure 1 consists of the following e^3*value* base constructs:

- **Actor.** An actor is perceived by its environment as an independent economic (and often also legal) entity. By performing *value activities* (see below) an actor makes profit or increases its utility. In a sound, viable, business value model *every* actor should be capable of making profit. Actors are represented as rectangles. Customer and supplier are actors.
- **Value Object.** Actors exchange value objects. A value object can be a service, right, good or even a consumer experience. The important point is that a value object represents a *value* for one or more actors. Value objects are shown as text next to arrows. Value objects in Fig. 1 are Good and Fee.
- **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 a port is important, because it enables to abstract away from the internal business processes, and to focus on how external actors and other components of the e-business value model can be 'plugged in'. Ports are shown as small black circles.
- **Value Interface.** Actors have one or more value interfaces. A value interface consists of individual value ports offering or requesting value objects. It shows the value object(s) an actor is willing to exchange *in return for* other value object(s). Such willingness is expressed by a decision function on the value interfaces, which shows under what conditions an actor wants to exchange a value object for another value object. The exchange of value objects is *atomic* at the level of the value interface. Either *all* exchanges occur as specified by the value interface or *none* at all. Note that a value interface does not indicate the time ordering of objects to be exchanged on its ports. It only indicates which value object is available, in return for some another value object. A value interface is shown by a rounded box, connected to an actor. In Fig. 1, value interfaces denote that actors offer/request a good and request/offer fee in return.
- **Value Exchange.** A value exchange is used to connect two value ports with each other. A value exchange represents one or more *potential* trades of value objects between value ports. As such, it is a prototype for actual trades between actors. According to the Enterprise Ontology 0 a value exchange would be called a potential sale. It shows which actors are willing to exchange value objects with each other. A value exchange is shown by an arrow.

Scenarios are used to relate an actor's value interfaces. Whereas value exchanges show *inter*-actor dependencies, scenarios represent *intra*-actor dependencies. They show via which value interface(s) an actor must exchange value objects, given the exchange of objects via another interface of that same actor. The main purpose is to facilitate the counting of value exchanges in an entire value web, as a result of a consumer need. This facilitates profitability analysis on a per actor basis (see for more detail [5]). Note that our scenarios do not represent *time-ordering*. Scenarios are only used to present *dependencies* between value exchanges of objects via value interfaces. For representation of scenarios, a simple form of Buhr's Use Case Maps [2] is used. The two constructs used are *dependency elements* and *connection elements*. Connection elements interconnect dependency elements like value interfaces, resulting in scenario paths.

- **Dependency element.** A scenario is expressed by dependency elements, interconnected by connection elements. Essentially, a scenario gives dependencies between value interfaces (a kind of connection element) so that we can reason for an entire value model what happens with other value interfaces if we exchange values via one particular value interface. Dependency elements are denoted by normal lines.
- **Connection element.** A connection element connects various dependency elements. Connection elements can be start or stop stimuli, AND/OR forks or joins and value interfaces. Connections elements are denoted differently depending on their specific kind.
- **Stimulus element.** Scenarios start with one or more *start stimuli*. A start stimulus represents an event, possibly caused by an actor. In most cases, such a stimulus represents a consumer need. If an actor causes an event, the start stimulus is represented in the actor box. A scenario also has one or more *end stimuli*. They have no successors. A start stimulus is represented by a filled circle, an end-stimulus is represented by a line perpendicular to the line denoting a dependency element.
- **AND and OR connection elements.** An *AND fork* connects a dependency element to one or more dependency elements, while the *AND join* connects one or more dependency elements to one other dependency element. It splits a scenario into several sub-scenarios or merges sub-scenarios into one scenario (see for a path the discussion below). An *OR fork* models a continuation of the scenario into one direction, to be chosen from a number of alternatives. The *OR join* merges two or more sub-scenarios into one scenario. An *AND* fork/join is shown as a line, perpendicular to the lines visualising dependency elements. An *OR* fork/join is represented by a number of lines joining into one (a join), or by a line splitting into more lines (a fork).

With these constructs a conceptual model of a value web can be constructed. Value webs typically consist of multiple enterprises, represented by multiple stakeholders. The e^3*value* modelling constructs help to create a shared understanding of the value web. In addition to this ontology-based graphical design tool e^3*value* also supports the calculation of the economic benefit of a value model via the so-called profitability assessment (see for full details [5]). It consists of two main steps: (1) profitability sheet generation, and (2) evolutionary scenario-based assessment.

It is important to understand that, given the semantics of the e^3*value* concepts, this value model states that a supplier is only willing to provide a good *if and only if* it obtains a fee (of course, the reverse requirement holds for the customer). In other words, the supplier is only willing to exchange objects via *all* ports of its value interface, or *none* at all. Hence, in this value model economic reciprocity is assumed to hold. *How* this is ensured is not an issue when designing a value web in the first place; during value web design we focus only on the value proposition itself and not on operational or trust issues.

3. CONTROLS IN NETWORKS OF ORGANISATIONS

Internal accountancy and auditing theory provides an excellent starting point for the analysis of inter-organisational controls. In [1] this theory is used as a basis to develop an audit theory of trade procedures. Within intra-organisational context, an organisation's financial resources (e.g. cash and merchandise inventory) must be protected from activities such as loss, waste, or theft by the organisation's employees. In the internal auditing theory, there are two basic systems considered: the administrative organisation system and the internal control system. Administrative organisation is the organisation's information system that most often contains the accounting information, and the procedural embedding of this information system in the organisation. The goal of the control mechanisms within the organisation is to secure trustworthiness of registered information within administrative organisation, and to control potential error in both administration and business operation [11]. A part of the control system in an organisation is *control procedures*, which are policies and procedures that help ensure management directives are carried out. For example, properly developed control procedures are important to help to ensure that necessary actions are taken to address risks related to achievement of a company's objectives. Intra-organisational control procedures are often classified into three major types: preventive controls, detective controls, and corrective controls [10]. Starreveld describes the control system as consisting of control measure as well as check and control activities [11]. (Internal) control measures are of preventive nature, which means they have to prevent errors (preventive controls). The (internal) check activities are of detective nature, which means they enable organisation to notice the occurrence of error (detective controls). Control activities do not only detect errors but they also correct them (corrective controls).

Internal auditing theory suggests utilising document process models of internal accounting information systems as a source of norms, and based on those, to analyse or develop the control system within the organisation [10],[11]. In the network perspective, using document process models of internal accounting information systems is rather complicated task, requiring the integration of multiple information systems. There is no other framework developed that provides controls and norms to analyse trustworthiness of transactions in networks of organisations. Inter-organisational audit theory of trade procedures developed by [1], focuses on activities and controls within the scope of a single transaction between two organisations, specified as trade procedure. For example, he viewed preventive controls to be documents or messages arranged in such a way, that any manipulation with these documents by one party would be noticed sooner or later by the other party, which will be an incentive not to misbehave. The detective controls would be also in a scope of single transaction.

We suggest that e^3*value* provides a good model to design control mechanisms for networks. In this paper we show examples of how e^3*value* models can be a basic formalism to design inter-organisational controls without going into details of internal organisational systems.

3.1 Extending e^3value to model sub-ideal behaviour

Value webs, expressed using e^3*value*, suppose a perfectly honest world, in which actors do not commit a fraud. When designing controls, consideration must be given to a risk factor [10], and to the identification of what can go wrong in the value web. Thus, from the control perspective, we distinguish two states the network of organisations can be in: (1) no presence of errors or sub-ideal behaviour of actors, which further are refereed as an *ideal situation*, and (2) errors or cases of sub-ideal behaviour of actors are present, which are referred as a *sub-ideal situation*.

In Figure 1 the *ideal situation* between the buyer and the seller is represented by value exchanges denoting transfers of goods and fees between them. In case of two value objects exchanged via the value interfaces of two actors (see Figure 1), we consider three sub-ideal situations: (1) the seller delivers goods, and the customer does *not* pay, (2) the seller does *not* deliver goods, while the customer pays, and (3) the actors do not exchange objects at all.

In a pure e^3*-value* setting, the principle of reciprocity (see section 2) states that the supplier is only willing to exchange objects via *all* ports of its value interface, or *none* at all. The situations (1) and (2) result in a failure to deliver a value object, i.e. in a value exchange, when one of the objects is not exchanged, which is the violation of the principle of reciprocity.

The situation (3), when there are no objects exchanged between the actors, is not the violation of the principle of reciprocity, since the latter has no restrictions on not exchanging value objects. The problem starts when we model a dependency path connected to the value interface of the consumer. The dependency element connects start stimulus, notifying a consumer need, and the value interface. In the case when no value objects are exchanged via a value interface, the consumer need is not satisfied (the customer does not receive goods!). Since the scenario path model a dependency between value interfaces, the dependency path is not prolonged at the reciprocal actor.

In the e^3*-value* methodology, the principle of reciprocity as well as the notion of the consumer need is "hard-wired". The attempt to model invalid, in e^3*value* terms, constructs will result in an error by an e^3*value* ontology validator, which is part of the tool-support mentioned in the previous section.

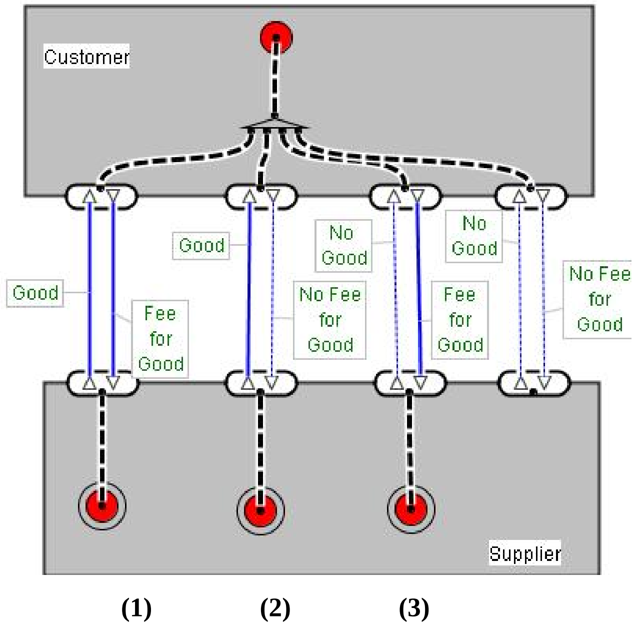

Figure 2: Some sub-ideal scenarios in the primary value web (Figure 1)

To model sub-ideal situations described above, in figure 2 we adjusted some e^3-*value* concepts:

Value Interface. A value interface consists of groups of in-going and out-going value ports. It models a value object an actor is willing to exchange *in return for* another value object. In a sub-ideal situation it is possible to have an incomplete set of value exchanges within the value interface. We call such value interfaces *incomplete*. Incomplete value interfaces can have at least one value object not exchanged. In Figure 2, in order to graphically represent the incomplete value interfaces we use different types of value exchanges. The value exchanges, which are executed as stated in the ideal scenario (Figure 1), are represented with solid lines, and non-executed value exchanges are drawn with *dashed lines*. The names of the value objects that are not delivered properly are also different from the names of the corresponding value objects in the ideal situation (like, "Fee" for the value object in the ideal situation, and "No Fee" for the in the sub-deal situation).

Dependency path. The dependency path that goes through the incomplete value exchanges, thus describing the execution of the sub-ideal situation, we call *sub-ideal paths*. The OR-fork is used to model the set of ideal and sub-ideal paths as alternatives. Thus, the customer in Figure 2 has four alternatives to happen. In the case of no exchange, the path is not prolonged at the supplier, the reciprocal actor, modelling that the consumer need is not satisfied.

3.2 Preventive control as value model: Letter of credit

Certain control mechanisms should be designed to prevent sub-ideal situations identified in the previous section. In daily trading practice, the Letter of Credit is a control mechanism that prevents the sub-ideal behaviour of the customer, and ensures that the seller gets paid before the goods are shipped. The Letter of Credit procedure is specifically tailored to secure the interests of the seller.

Banks introduced Letter of Credit procedure in order to solve the following problem in international trade. Suppose we have a seller in Hong Kong and a buyer in the Netherlands. The agents are geographically far apart, and the goods have to be transported by a carrier from the seller to the buyer (we assume by sea). On the one hand the seller does not want to ship the goods onto the carrier's vessel (and thereby lose control over them) without first receiving payment from the buyer. On the other hand the buyer does not want to pay the seller (and thereby lose control over the money) before the goods have been shipped. In other words, the actors prefer a simultaneous exchange of the shipment of the goods in return for the money. To solve this deadlock situation banks introduced the letter of credit, which is an agreement that the bank of the buyer, the so-called issuing bank, will arrange the payment for the seller as soon as the seller can prove to the bank that the goods are shipped. The seller proves this shipment by presenting the Bill of Lading to the bank, the so-called corresponding bank. The seller receives the Bill of Lading from the carrier, when the seller shipped the goods. The seller's bank transfers the Bill of Lading to the customer's bank and the customer's bank gives the Bill of Lading to the customer as soon as the customer pays. The customer can receive the shipped goods from the carrier in return for the Bill of Lading, or can sell the Bill of Lading to some other company, which can then receive the shipped goods from the carrier in return for the Bill of Lading. The Bill of Lading is an example of a multimodal transport document that has an evidentiary effect. This evidentiary effect is even stipulated in a special convention of the United Nations. The *United Nations Convention on International Multimodal*

Transport of Goods (CIMTG) describes this function as follows [15]:

> **Article 10** - *Evidentiary effect of the multimodal transport document*
>
> *Except for particulars in respect of which and to the extent to which a reservation permitted under article 9 has been entered:*
>
> *The multimodal transport document shall be prima facie evidence of the taking in charge by the multimodal transport operator of the goods as described therein; and*
>
> *Proof to the contrary by the multimodal transport operator shall not be admissible if the multimodal transport document is issued in negotiable form and has been transferred to a third party, including a consignee, who has acted in good faith in reliance on the description of the goods therein.*

The Letter of Credit procedure can be considered from multiple perspectives. Seen from a control mechanism perspective, the Letter of Credit procedure is a preventive control mechanism, which contributes to increasing confidence in reliable and fair exchanges of goods between actors, who do not know each other in advance. From a value web perspective, the Letter of Credit procedure can be seen as a commercial service *itself* facilitating the sale and delivery of another good or service. Actually, the letter of credit is a commercial service, because the buyer has to pay a fee to the bank that issues the letter of credit. If a value web is considered as a set of actors performing economic exchanges with each other, we can view the Letter of Credit procedure as an economically valuable service in a *secondary* value web, facilitating a *primary* value web consisting of actors exchanging goods or services.

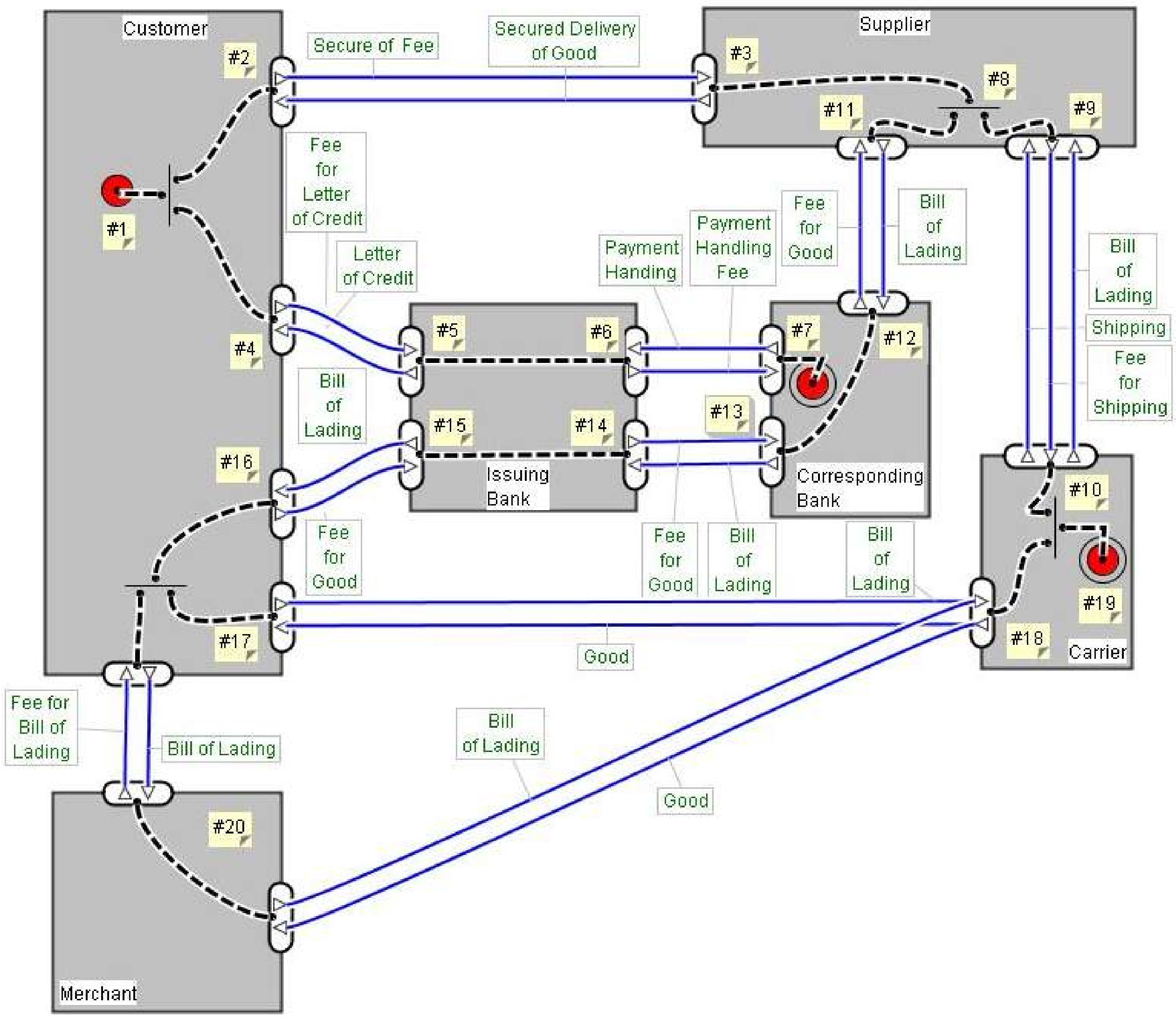

Figure 3: Secondary value web for Letter of Credit

The Letter of Credit procedure considered from a commercial service perspective is represented in Figure 3 and illustrates that the *customer* must guarantee that the *supplier* gets paid for the good. This is depicted by the AND-fork (a kind of connection element, see #1), indicating that if the consumer wants a good, he must exchange values via interfaces #2 and #3 (a good for a fee) *and* via interfaces #4 and #5. The latter is the receipt of a Letter of Credit, a service that ensures that if the supplier ships a good, then he gets paid. The customer obtains a Letter of Credit from an *issuing bank* and the customer pays a fee for this. Typically, the issuing bank of the Letter of Credit is in the same country as the customer, but often has no branch in the supplier's country. In such a case the issuing bank needs to involve a *corresponding bank*, which is physically close to the supplier. This corresponding bank pays the supplier when the supplier presents the bill of lading to the bank as evidence that he has shipped the good. In return for this service, which is an intrinsic part of the Letter of Credit procedure, the corresponding bank charges the issuing bank a fee (see interfaces #6 and #7). Note that the Letter of Credit, of which the supplier is notified by the issuing bank, is a guarantee for the supplier that he will be paid. This is reflected by the value web by the exchange *secure of fee*, rather than just *fee*. Consequently, the fee itself is not directly exchanged between consumer and supplier, only the *guarantee* that the issuing bank will arrange the payment of the fee for the good.

As a result of securing the fee, the supplier exchanges objects of value via two of its interfaces, represented by AND-fork #8. Via interfaces #9 and #10, the supplier ships the ordered good via a *carrier*. The carrier charges the supplier a shipping fee, and the supplier obtains a Bill of Lading. It is important to understand that the Bill of Lading is of economic value. It is a so-called negotiable document, which can be traded and therefore be seen as similar to paper money. Additionally, from a trust perspective, it is important to understand that the carrier is seen as a trusted third party; all actors involved assume that the carrier only gives a Bill of Lading if he obtains the good to be shipped. As soon as this Bill of Lading is presented by the seller to the corresponding bank, then the bank pays the fee for the good to the supplier. Hence, this is a kind of secured pre-payment arrangement for the seller. So, via interfaces #11 and #12, the supplier offers the Bill of Lading, obtained from the carrier to a bank, and in return obtains a fee for the good. Note that Fig. 4 only shows dependencies between exchanges of value, not their temporal ordering of actual events.

The Bill of Lading is transferred by the corresponding bank to the issuing bank (interfaces #13 and #14). As a consequence, the issuing bank exchanges with the customer the Bill of Lading for the fee for the good (interfaces (#15 and #16). The customer has an option to sell the Bill of Lading to Merchant or to keep it (#20). The carrier transports the good to an actor (customer or merchant) that possesses the Bill of Lading, and releases the good to the customer or merchant, in return for the Bill of Lading (interfaces #17 and #18). The AND join, annotated with #19, models that the Bill of Lading as issued by the carrier, should also be obtained by the carrier once the good is delivered. After that, the Bill of Lading is useless.

The analysis of fraudulent behaviour of the customer or seller resulted in a number of the sub-ideal paths (see Figure 2). The Letter of Credit procedure is a preventive control mechanism to prevent the sub-ideal situation 1, when the buyer does not pay the seller. However, from the perspective of the buyer the letter of credit control does not secure the buyer against the sub-ideal behavior of the seller or the carrier. There still can be some sub-ideal situations within secondary value web of the letter of credit, which result in the execution of sub-ideal path 2 or 3 in the primary value web. Other preventive controls have to be used by the buyer; for example, the buyer normally buys the insurance to secure the goods.

3.3 Detective controls in value models: forged Bill of Lading scenario

Viewing the secondary value web not as a control mechanism, but as a value web, in this section we demonstrate how *detective controls* are build in the e^3*value* constructs can be used to detect sub-ideal situations that have taken place. The fraudulent situation, considered in this section, is an attempt to obtain the goods by the third party by showing a *forged* Bill of Lading to the carrier. In Figure 4 we introduce a new actor, named Illegal Claimant. The illegal claimant gives the carrier the forged Bill of Lading, and the carrier, who does not know that the Bill of Lading is forged, gives the illegal claimant the goods. Since it is permitted by law to sell the Bill of Lading (to a merchant, as modelled in Figure 3, #20), the carrier will ship the goods to the party who shows the Bill of Lading. Thus, from the carrier's point of view, the value exchange with the illegal claimant would be similar to the value exchanges with the merchant in Figure 3.

Although from the carrier perspective, the value exchange with the illegal claimant is similar to the valid value exchange with a merchant (see Figure 4), and thus, cannot be detected by analysing the processes between the carrier and the other party. From the network perspective, the exchange with the illegal claimant generates the *invalid* value model. It violates the following requirement in the e^3*value* formalism: *the number occurrences going into an AND join (annotated #19) should be equal*. This is explained below.

Suppose, the number of start stimuli at the customer is 10: meaning that the customers bought 10 letters of credits from the issuing bank, and the supplier shipped goods 10 times, which generated 10 Bills of Lading. Then all value interfaces except #18 and #20 should also 'fire' 10 times. These interfaces are all connected via direct paths or AND-forks, and these do not result in changes in the number of exchanges. If we suppose that the number of start stimuli of 'illegal claimant' is 2, value interface #20 fires two times, whereas value interfaces #18 fires 10+2=12 times.

The AND join annotated #19 merges two paths. The path connected to interface #18 fires 12 times, whereas the path connected to interface #10 only fires only 10 times. The semantics of the AND-join is that both incoming paths should have equal numbers. Consequently, the validation tool reports that this AND join is wrong. This makes the value model in Figure 4 invalid.

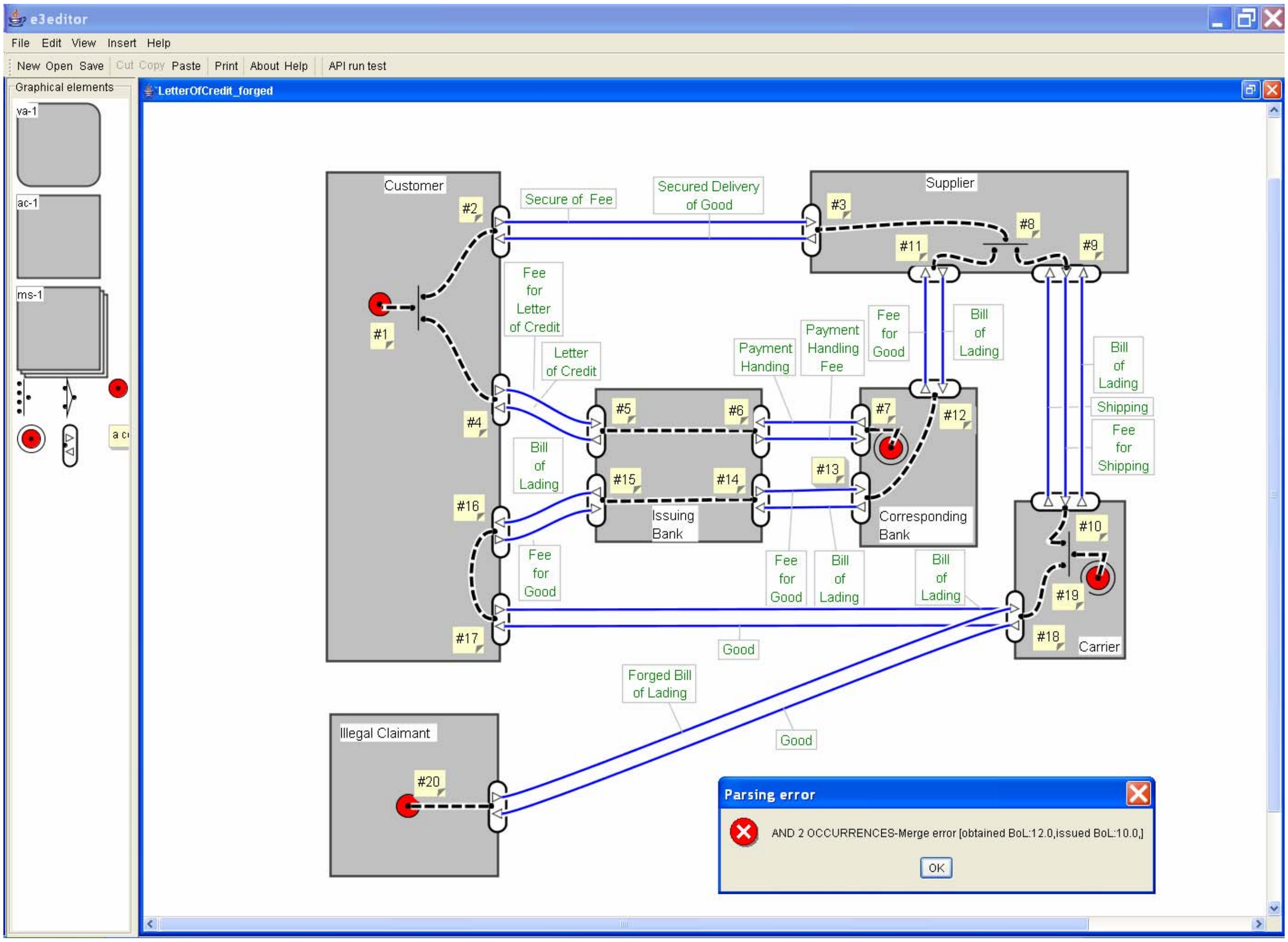

Figure 4: Forged Bill of Lading

This example clearly shows that an e^3*value* model is capable of modelling a *detective* control. The carrier's AND fork models that each Bill of Lading issued, should match with a Bill of Lading obtained from a customer, and vice versa. So, a forge Bill of Lading will sooner or later be detected by the Carrier, because the number of issued Bills of Lading do not match with the obtained Bills. The requirement that the number of start stimulus should equal the number of end stimulus plays a role of the *norm* for the detective control. This type of detective control is only possible in the network setting when information is tracked throughout the network of companies. It is sufficient enough to have data needed to build e^3*value* model to detect the forged Bill of Lading.

4. CONCLUSIONS AND FUTURE RESEARCH

In this paper we suggested to analyse the control mechanisms in networks of organisations using the e^3*value* methodology. By example, we demonstrated how some aspects of control can be analysed. First, we showed how to perform contingency planning of e^3*value* models distinguishing ideal and sub-ideal situations. Then we modelled a preventive control mechanism the Letter of Credit as a value mode. Finally, we demonstrated how detective control mechanisms in networks of organisations could be analysed based on e^3*value* models. As can seen from the examples, e^3*value* models are formal enough to develop a theory of controls for the specific case at hand. Future research will concentrate on other controls to explore the boundaries of the power inexpressiveness of e^3*value*.

5. ACKNOWLEDGMENTS

Our thanks to Post Master EDP-auditing Education of Free University Amsterdam for funding this research

6. REFERENCES

[1] Bons, R.W.H.; Lee, R.M.; and Wagenaar, R.W. Designing trustworthy inter-organizational trade procedures for open electronic commerce, Global Business in Practice, In Gricar, J., Pucihar, T. (eds.), *Proceedings of the Tenth International Bled Electronic Commerce Conference*, Kranj: Moderna Organizacija, 1997.

[2] Buhr, R. J. A. Use case maps as architectural entities for complex systems. *IEEE Transactions on Software Engineering*, 24, 12 (1998), pp. 1131–1155.

[3] Das, T.K. and Teng, B.S. Between Trust and Control: developing confidence in partner cooperation in alliances. *Academy of Management Review*, 23, 3 (1998), pp. 491-512.

[4] Gordijn, J. and Akkermans, J.M. Designing and Evaluating e-Business Models. *IEEE Intelligent Systems - Intelligent e-Business*, 16, 4, (2001), pp. 11-17.

[5] Gordijn, J., and Akkermans, J.M. Value based requirements engineering: Exploring innovative e-commerce idea. *Requirements Engineering Journal*. Springer Verlag, 8, 2, (2003), pp. 114-134.

[6] Holland, C.P. and Lockett, A.G., Business trust and the formation of virtual organizations. In Sprague, R.H. (ed.), *Proceedings of the 31st Annual Hawaii International Conference on System Sciences.* Los Alamitos: IEEE Computer Society, 1998, pp. CDROM.

[7] Lee, R.M. and Bons, R.W.H. Soft-Coded trade procedures for open-edi. *International Journal of Electronic Commerce*. 1, 1, (1996), pp. 27-49.

[8] Mayer, R.C.; Davis, J.H.; and Schoorman, F.D. An integrative model of organizational trust. *Academy of Management Review*. 20, 3, (1995), pp. 709-734.

[9] Pateli, A.G. and Giaglis, G.M. A Framework for Understanding and Analysing eBusiness Models. In Wigand, R.T., Tan, Y.H., Gricar, J., Pucihar, T. and Lunar, T. (eds.), *Proceedings of 16th Bled Electronic Commerce Conference: eTransformation*, Kranj: Moderna Organizacija, 2003, pp. 329-348.

[10] Romney, M.B., and Steinbart, P.J., Accounting Information Systems Prentice Hall, 2003

[11] Starreveld, R.W., De Mare, H.B., Joels, E.J., Bestuurlijke Informatieverzorging, Samsom Bedrijfsinformatie, Part I&II, Alphen a/d Rijn, The Netehrlands, (in Dutch), 1995

[12] Tan, Y.H. and Thoen, W. Toward a Generic Model of Trust for Electronic Commerce. *International Journal of Electronic Commerce*. 5, 2, (2000), pp. 61-74.

[13] Tan, Y.H. and Thoen, W. Formal aspects of a generic model of trust for electronic commerce. *Journal of Decision Support Systems*. 33, (2002), pp. 233-246.

[14] Tan, Y.H.; Thoen, W.; and Gordijn, J. Modeling Controls for Value Exchanges in Virtual Organizations. to appear in the *Proceedings of the 2nd Conference on Trust Management*, Berlin: Springer Verlag, 2003.

[15] United Nations Convention on International Multimodal Transport of Goods. http://www.jus.uio.no/lm/un.multimodal.transport.1980/index.html, Geneva: United Nations, accessed on February 26, 2004.

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