

December 2004

Towards A Modelling Tool For Designing Control Mechanisms In Network Organisations

Jaap Gordijn

Department of Exact Sciences, Free University Amsterdam

Vera Kartseva

Department of Economics and Business Administration, Information Systems Group, Free University Amsterdam

Follow this and additional works at: <http://aisel.aisnet.org/bled2004>

Recommended Citation

Gordijn, Jaap and Kartseva, Vera, "Towards A Modelling Tool For Designing Control Mechanisms In Network Organisations" (2004).
BLED 2004 Proceedings. 50.

<http://aisel.aisnet.org/bled2004/50>

This material is brought to you by the BLED Proceedings at AIS Electronic Library (AISeL). It has been accepted for inclusion in BLED 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

17th Bled Electronic Commerce Conference

eGlobal

Bled, Slovenia, June 21 - 23, 2004

Towards A Modelling Tool For Designing Control Mechanisms In Network Organisations

Vera Kartseva

Department of Economics and Business Administration, Information Systems Group
Free University Amsterdam, The Netherlands
vkartseva@feweb.vu.nl

Jaap Gordijn

Department of Exact Sciences, Free University Amsterdam, The Netherlands
gordijn@cs.vu.nl

Yao-Hua Tan

Department of Economics and Business Administration, Information Systems Group
Free University Amsterdam, The Netherlands
ytan@feweb.vu.nl

Abstract

Contracts and organizational controls to monitor contract compliance are important tools to enhance trust in a fair business transaction in network organisations and electronic commerce in general. In this paper, we propose a design methodology for such contracts and supporting controls, utilizing inter-organisational value models. We argue that a framework for designing control mechanisms should include three steps: design of an inter-organizational value model, analysis of possible violations of contractual obligations underlying this value model, and design of control mechanisms to detect or prevent such violations. It is shown how the e^3 -value methodology, which was developed to design business value models, can be extended to model obligations of parties. We use concepts and ideas from deontic logic (the logic of obligations and permissions) to develop an extension of e^3 -value called e^3 -value+. The e^3 -value+ approach is a design tool for modelling violations of obligations and control mechanisms to prevent and correct these violations, which can be used in contract drafting and contingency planning for inter-organisational collaboration in network organisations.

1 Introduction

One of the limitations for the success of electronic commerce is the risk of opportunistic behaviour by legal entities (McKnight1998). For example, parties may not deliver products while they promised to do so, customers may not pay for delivered products, or others may commit frauds. Consequently, since transacting parties often do not know each other, sufficient trust enhancing facilities should be in place to facilitate trading.

Technical means, such as encryption technology and digital signatures to build trust-facilitating services are available right now. Additionally, current research topics in computer science such as 'web services' and 'peer-to-peer networking' enable the provisioning of inter-organisational business processes, required for trust enhancing procedures (Foster et. al. 2002, Barkai 2001). Finally, contracts and organizational controls to monitor contract compliance are required to improve trust. In traditional commerce, control mechanisms have already being studied extensively (Anthony and Govindarjan 2003, Vassen 2002), such as contractual obligations and formal organisational mechanisms for co-operation. For e-commerce cases however, the design of control mechanism is a rather uncharted territory.

For example, although, technically speaking, an electronic signature is much more reliable than a hand-written signature, the historically proven "paper and pen" way of signing a contract is considered by human beings to be more reliable. One can say that the reason is because a paper contract is backed with a proven infrastructure of legislative practices and underlying control mechanisms that make it possible to solve any disagreement between parties. Many research of the Transaction Cost economy (TCE) school (Williamson 1985) agree that governance mechanisms, such as control mechanisms, reduce the risk of opportunistic behaviour, and, consequently, increase the success of inter-organisational co-operation. We claim that control mechanisms are an essential part of an electronic commerce infrastructure. However, what is lacking is an understanding how to design contracts and supporting control mechanisms on top of such technical possibilities.

A contract can be seen as a control mechanism that can ensure legality and protect the interests of all parties involved in electronic commerce. Technology biased-research on electronic contracting (Ludwig and Stolze 2003) assumes that a plain textual, natural language representation of the contractual content that can be read and interpreted by business people and lawyers suffices all requirements. However, business studies (Luo 2001) indicate incomplete contracting practices that result in increased opportunism and failure of inter-organisational co-operation. The design of a contract suitable for electronic commerce as well as its supporting controls, is a multi-disciplinary task that requires solutions for incomplete contracting practises. It involves obviously legal aspects, but also computer science issues are relevant (many controls take the form of computer software), as well as inter-organisational business process design (many contracts say how, and in which order, business transactions should be carried out, and by whom). In addition, in electronic commerce a thorough understanding of the corresponding business model is often lacking (Gordijn 2002), which can make contingency planning of contracts more complex.

In (Daskalopulu and Sergot 1997, Tan and Thoen 2001) a need for contract negotiating and drafting methodology and tools is identified. The main contribution of this paper is that we propose *a methodology for designing contracts and controls, based on an inter-organizational business value model*. The result is a framework for contingency planning in contracting.

This paper is structured as follows. In section 2 two we propose an analytical framework to design control mechanisms. The framework consists of representing *ideal* and *sub-*

ideal situations, and designing control mechanisms to prevent sub-ideal situations. We address the modelling of the ideal situation in section 3, and propose some ideas for modelling sub-ideal situations and control mechanisms in section 4.

2 A Framework For Designing Control Mechanisms

The main purpose of control mechanisms in electronic commerce would be to ensure that all value exchanges between enterprises take place as planned or stated in a contractual agreement. There are many theories dealing with control mechanisms. Main elements of every control system (Anthony and Govindarjan 2003, Vassen 2002) are the entity being controlled, detection of the violation, measurement of the severeness of the violation, and behaviour alteration. Similarly, control mechanisms can be seen as measures to prevent violation of obligations. Raskin et.al. (1996) distinguish two viewpoints on the behaviour of actors with regards to obligations: (1) no violation of obligations by the actor, which we call *ideal* behaviour, and (2) violation of obligations, which we call *sub-ideal* behaviour.

A framework for designing control mechanisms should include at least the following subsequent steps:

Design of a business value model (ideal situation)

Analysis of contractual obligations and their possible violations (sub-ideal situation)

Design of control mechanisms (prevention of sub-ideal situation)

A business value model shows *what* actually is being controlled or governed by a contract. This is an essential element of every control system (Anthony and Govindarjan 2003, Vassen 2002). In electronic commerce, such a value model is often unclear because most models are innovative. Consequently, it is necessary to understand the exchanges of economic valuable objects between actors before starting to design control mechanisms for these exchanges. We deliberately assume that a business value model only deals with ideal behaviour; that there is no violation of obligations. In practice, it is sufficiently difficult to design and understand such a model (Gordijn, 2003), and the addition of obligation issues would unnecessarily complicate such a task. Therefore, we divide the design of a commercially sound value model from the analysis of violations and the design of control mechanisms.

After it is understood what value exchanges actually have to be controlled, possible violations are analysed and control mechanisms are designed. According to the description of a control system in Anthony and Govindarjan (2003), it is necessary to measure what is going on in the process being controlled, and to determine the significance of what is happening. In our terms, before introducing control mechanisms, it is needed to know *how the ideal situation can be violated*, i.e. the *sub-ideal situation* has to be modelled, and what the *severeness* of the violation is. In the next two sections we address the modelling of ideal and sub-ideal situation in electronic commerce.

3 The Ideal Situation: Value Models

A first step in designing an electronic commerce case, is the development of a business value model, focusing on how value creation (Porter 1980, Stabell and Fjeldstad 1998), distribution and consumption happens in a network of actors (see (Pateli and Giaglis

2003) for an overview). We employ the e^3 -value methodology (Gordijn 2002, Gordijn and Akkermans 2003) for designing a business value model.

The e^3 -value methodology of Gordijn (2001) allows for conceptualising a business value model by constructing a value model, representing it graphically in a rigorous and structured way, and performing an economic sensitivity analysis of the case at hand. 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 previously applied in a series of case studies including media, banking and insurance, and telecommunication companies (Gordijn and Akkermans 2003) to design value models of network organisations.

Most of the currently available design methodologies lack a value-based view stating *what* the value proposition is, rather they 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 (Uschold 1998) or the Resource Event Agent (REA) (Geerts and McCarthy 1999) conceptual framework. However, these frameworks only focus on the description of the final result, and do not support the value chain design process itself. Other business modelling methodologies (Alt and Zimmerman 2001, Osterwalder and Pigneur 2002, Petrovic et al. 2001) offer only generic conceptual frameworks, and lack the formality that is required for modelling for proper analysis. Tapscott et.al. (2000) offer a graphical diagramming approach to draw economic exchanges between enterprises, however, compared to e^3 -value, it has several drawbacks; e.g. it has no notion of economic activity, does not allow the profitability assessment of individual organisations, and lacks the proper level of formality.

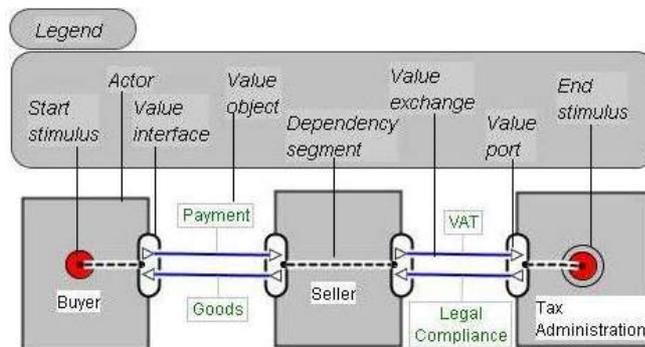


Figure 1: e^3 -Value Model Of A Purchase With Tax Payment

We shortly 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 law, the seller is obliged to pay the value-added tax (VAT). This is conceptualized 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 administration*.

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 (Buhr 1998)) 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 a model boundary.

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 to an end stimulus. The meaning of the path is that if values are exchanged via a value interface, then other value interfaces connected by the path also exchange values.

The advantage of e^3 -value is that it suggests a *minimal* number of concepts, which in practice leads to a usable modelling technique.

It is important to understand that an e^3 -value model *only* models an ideal situation. The *principle of reciprocity*, defined in e^3 -value (Gordijn 2003) as the requirement that if an actor offers something of value to someone else, this actor always gets in return something what s/he wants, assumes that *all* actors behave correctly. The violation of the principle of reciprocity (e.g. an actor obtaining something without paying for it) can be seen as a violation of an obligation, which would lead to incorrect e^3 -value models with a value interface with only one incoming or outgoing value object, i.e. delivering goods and not receiving a payment in return. In the next section we extend e^3 -value with additional concepts and rules to be able to model such sub-ideal situations as violations of the principle of reciprocity.

4 The Sub-Ideal Situation: Contractual Obligations And Controls

A large number of theories about modelling obligations and their violations have been developed in the area of deontic logic (for an overview see Meyer and Wieringa 1993). In the research of (Raskin et.al. 1996, Tan and Thoen, 1998) a method was introduced to model obligations using Petri Nets. A Petri net is a graphical formalism for modelling and

analysing discrete dynamic systems. In Raskin et al. (1996) it was suggested to model procedures and processes within and between organisations with extended Petri Nets, which model preference relations for different executions of transactions between actors.

A method based on the one described in Raskin et al. can be used to extend e^3 -value in such a way that it becomes possible to represent obligations and violations of obligations. We will call this extension e^3 -value+. Violations of obligations in e^3 -value+ are modelled by so-called sub-ideal paths. These are typically paths for which the principle of reciprocity does not hold. For example, although in a contract it was agreed that the buyer would pay the seller in return for the delivery of the goods, there can be a sub-ideal path for which the seller did deliver the good, but the buyer does not pay within the agreed period of time. So, the first extension we have to make to the standard e^3 -value tool is that we omit the principle of reciprocity and allow for this type of sub-ideal paths. The second extension is that we add weights to the segments of a path. As in Raskin et al. (1996), these weights represent a kind of fines. So, if a segment has a non-zero positive weight it reflects that this is a kind of sub-ideal behaviour that is fined with a penalty. A zero weight means that the segment does not violate the terms of the contract.

Figure 1 shows an e^3 -value model, stating the ideal situation. In Figure 2 we use an e^3 -value+ to model sub-ideal situations that can happen when one of the actors violates the contract (e.g. does not respect economic reciprocity). Modelling these sub-ideal paths are typical for contract drafting when 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. Typically, these extra contract clauses also include certain financial compensations, for example a reduction of the price if the goods are delivered too late. In organisational science research on contracting it has been observed that this reasoning about sub-ideal paths, which is called contingency planning, is one of the most important as well as labour intensive stages of contract negotiation. For example, in (McCauley 1963) it was observed that contingency planning is the most useful part of the contract, because it learns the contract partners what problems they could encounter, what reasonable financial compensations are to be paid when they do occur, and how to minimise the likelihood that these problems will occur. One of the main purposes of e^3 -value+ is to support this contingency planning process in contract negotiations.

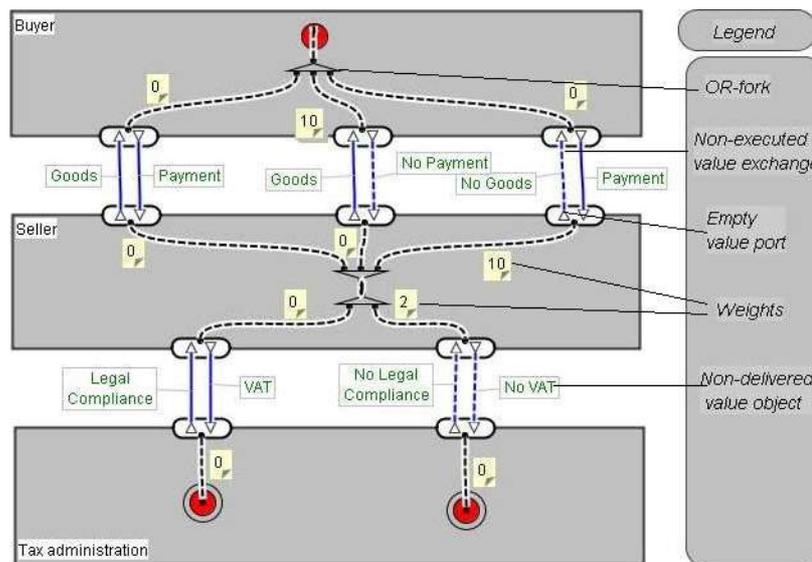


Figure 2: Modelling Violated Obligations In e^3 -value+

Compared to Figure 1, in Figure 2 we now have two additional value exchanges between the buyer and the seller, which represent two possible sub-ideal situations: (1) the seller delivers goods, and the buyer does *not* pay, and (2) the seller does *not* deliver goods, while the buyer pays. The first is the bad behaviour of the buyer, and the second is the bad behaviour of the seller. As a result, instead of one scenario path in the ideal situation in Figure 1, we now have in Figure 2 three different alternatives, modelled as one ideal and two sub-ideal paths coming out of the OR-fork. In addition, we also added weights to each of the path segments. These weights reflect the severeness of the violation. For example, the buyer can face three situations: to pay and to receive goods, to receive goods and not pay, and to pay, but not to receive goods. The worst option for the buyer is to receive goods, but not to pay, because then the buyer violates his/her obligation to pay. By assigning a higher weight to the segment, which indicates the worst case for the buyer, the incentive is modelled for the buyer not to take this sub-ideal path.

As for the exchanges between the seller and the tax administration, the ideal situation is modelled as the seller paying tax and the tax administration confirming a legal compliance. In the sub-ideal situation the principal of reciprocity can only be violated by not paying tax (we assume that a tax office cannot violate it by punishing the seller even if the tax is paid). Note that the violations of obligations in exchanges between the seller and the tax administration are independent of violations of obligations between the seller and the buyer, and the other way around. The model includes a number of possible sub-ideal paths: (a) the goods are delivered, the fee is not paid, and the tax is paid, (b) the goods are not delivered, the fee is paid, and the tax is paid, (c) the goods are delivered, the fee is not paid, and the tax is not paid, (d) the goods are not delivered, the fee is paid, and the tax is not paid, and (e) the goods are delivered, and the fee is paid, but the tax is not paid. Below we describe the extensions of e^3 -value in more detail.

Extension 1: the violation of the principle of reciprocity.

The ideal situation between the buyer and the seller is represented by value exchanges denoting transfers of goods and fees between them. *We model a sub-ideal situation as a violation of the principle of economic reciprocity.* The failure to deliver a value object is the violation of the principle of reciprocity. In e^3 -value the principle of reciprocity is “hard-wired”, and the violation of the principle of reciprocity will be reflected in the changes of the following e^3 -value concepts:

Value Interface. A value interface consists of groups of in-going and out-going value ports. It shows the value object an actor is willing to exchange *in return for* another value object via its ports. The failure to deliver value object results in an incomplete set of value exchanges coming out and in the value interface.

Value port. Value object is exchanged between actors via value ports of the value interface. If the principle of reciprocity is violated, no value object and no value exchange goes through the value port. We call this value port an **empty value port**.

Value object. The violation the principle of reciprocity results in new types of value objects. Basically, there are two types of value objects: **delivered value object** and **non-delivered value object**.

Value exchange. Value exchange is closely related to value interface, and the violation of the principal of reciprocity will result in a value exchange that it is not executed. Thus, there are also two types of value exchanges: **executed value exchanges** and **non-executed value exchanges**.

To graphically represent the violation of the principle of reciprocity we use different representations of value exchanges. Executed value exchanges are represented with solid lines, and non-executed ones with dashed lines. The value ports that are connected to non-executed value exchanges are empty. The names of non-delivered value objects are

also changed depending on the corresponding delivered value objects (like, “Payment” for the delivered value object and “No Payment” for the non-delivered one).

Extension 2: weights of value ports, path segments, and sub-paths.

We modelled five violation situations, but they are not of the same severeness. For example, one can argue that even if the buyer did not pay the seller for the delivered goods, for the seller it is still better to pay tax rather than not to pay tax, because VAT is based on the seller’s invoice and not on whether the seller actually received the payment from the buyer.

The idea of differentiating between levels of severeness of sub-ideal situations was proposed by Tan & Thoen (1998), and Raskin et.al. (1996). Following the suggested solution, we propose to model the difference between violations by assigning different weights to segments that lead to **empty** value ports. Weights represent fines: the higher the weight, the more severe the violation. Every segment leading to a non-empty value port has a zero weight, because there is no violation of the terms of the contract, and therefore no fine is assigned. Every segment leading to an empty value port has a non-zero positive weight, which reflects a kind of sub-ideal behaviour that is fined with a penalty. Note that weights are assigned only to segments leading to value ports having *outgoing* value objects: the assumption is made that the failure to deliver the value object is a responsibility of the party that offers this object, not the party that accepts it.

For example, in Figure 2 in the exchange between the buyer and the seller when the goods are delivered (value object “Goods”) and the fee is not paid (value object “No Payment”), a non-zero positive weight (in this case 10), is assigned to the segment leading to the empty outgoing value port with the non-delivered value object “No Payment”. Similarly, the segments leading to empty outgoing value ports with the associated non-delivered objects “No Goods”, and “No VAT” have non-zero positive weights 10 and 2 respectively.

Assigning weights is part of the modelling process, just as the introduction of sub-ideal paths is. The *e³-value+* methodology does not prescribe the contract partners which sub-ideal paths are relevant to consider, or which specific numbers have to be assigned to a weight. These are modelling issues that contract partners have to agree upon among each other. However, *e³-value+* does help to structure this modelling process.

To keep things simple we use absolute numbers in this paper to explain the method. Of course, these weights are rather arbitrary. In typical contract negotiations the contract partners agree among each other how to assign these weights (and typically translate these weights in financial compensations). For example, the seller might require that in case of late payment the buyer should pay a certain percentage extra on top of the sales price. A more sophisticated representation could be to use a partial ordering of all the weights rather than absolute numbers. This partial ordering will be investigated in future research.

The path can be assigned a total weight, which is a sum of weights of the segments that the path consists of. In assigning weights in Figure 2 we used the following heuristics: the more serious a violation is, the higher weight is assigned to the corresponding segment. Consequently, in the ideal scenario the total weight of a path will be zero. In sub-ideal situations the total weight of a path will be a non-zero positive number; we call these paths **sub-ideal paths**.

The total weight expresses, how good or bad it is for the actor to execute a specific path. Having weights assigned to segments it is possible to distinguish the severeness of violation from the viewpoint of every single actor in the model. Table 2 represents five sub-ideal paths of Figure 2. The last three columns include the actors. Each actor in each sub-ideal situation accumulates weights, which are assigned to the individual segments.

Table 1: Actor's View On Modelling Severeness Of Violation Weights For Different Sub-Ideal Situations

Sub-ideal path	Non-delivered value objects		
	Buyer	Seller	Tax administration
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, the fee is not paid, and the tax is not paid	No Payment (w=10)	No VAT (w = 2)	w = 0
d) 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
e) The goods are delivered, and the fee is paid, but the tax is not paid	w = 0	No VAT (w = 2)	w = 0

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 2). Consequently, we can say that for the buyer sub-ideal situations (a) and (c) 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 (d): it has the highest total weight 12 (no goods are delivered and no tax is paid). The situation (c), 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 administration every modelled situation is equal to ideal: the tax administration is modelled as a governing institution that cannot be punished for violation of obligations.

5 Conclusions

In this paper we have introduced the e^3 -value+ design methodology, which is an extension of the e^3 -value methodology for designing business models for virtual network organisations. The e^3 -value+ design methodology is being developed, in particular, for modelling inter-organisational control mechanisms. The e^3 -value+ methodology supports both the analysis of violations of obligations as well as the corresponding design of control mechanisms to minimise the likelihood of these violations. The fine systems, described in this paper, can be considered as an incentive for actors not to misbehave, and hence as a type of control mechanism. This methodology supports the contingency planning phase of contract drafting, because it helps the contract partners to understand what violations of contractual obligations they could encounter, what reasonable financial

compensations are to be paid when they do occur, and how to minimise the likelihood that these problems will occur.

References

- Alt, R. and Zimmerman H.D. (2001): Preface: Introduction to Special Section – Business Models. *Electronic Markets*, Vol. 11, No 1, pp. 3-9.
- Anthony, R. and Govindarjan, V. (2003): “Management Control Systems”, McGraw-Hill, Irwin.
- Barkai, D. (2001): “Peer-to-Peer Computing”, Intelpress.
- Buhr, R. J. A. (1998): Use Case Maps as Architectural Entities for Complex Systems, *IEEE Transactions on Software Engineering*, Vol. 24, No. 12, pp. 1131-1155.
- Daskalopulu, A. and Sergot, M. (1997): The Representation of Legal Contracts, *AI and Society*, Vol. 11, No. 1, pp. 6-17.
- Foster, I., Kesselman, C., Tuecke, S. (2001): The Anatomy of the Grid: Enabling Scalable Virtual Organisations, *International Journal of Supercomputer Applications*, Vol. 15, No. 3.
- Geerts, G. and McCarthy, W.E. (1999): An Accounting Object Infrastructure for Knowledge-Based Enterprise Models, *IEEE Intelligent Systems & Their Applications*, July-August, pp. 89-94.
- Gordijn, J. (2002): “Value-based Requirements Engineering - Exploring Innovative e-Commerce Ideas”, PhD thesis, Vrije Universiteit, Amsterdam, <http://www.cs.vu.nl/~gordijn/>.
- Gordijn, J. and Akkermans, J.M. (2003): Value-based Requirements Engineering: Exploring Innovative e-Commerce Ideas, *Requirements Engineering Journal*, Vol. 8, No. 2, pp. 114-134.
- Ludwig, H. and Stolze, M. (2003): Simple Obligation and Right Model (SORM) - for the Runtime Management of Electronic Service Contracts, *CAISE Workshop on Web Services, e-Business, and the Semantic Web (WES): Foundations, Models, Architecture, Engineering and Applications*, June 2003.
- Luo, Y. (2001): Contract, Cooperation, and Performance in International Joint Ventures, *Strategic Management Journal*, Vol. 23, pp. 903-919.
- Meyer, J-J.Ch., and Wieringa, R. (Eds.) (1993): “Deontic logic in computer science”, Wiley, Chichester.
- McKnight, D.H., Cummings, L.L. and Chervany, N.L. (1998): Initial trust formation in new organizational relationships, *Academy of Management Review*, Vol. 20, No. 3, pp. 472-490.
- Osterwalder, A. and Pigneur, Y. (2002): An e-Business Model Ontology for Modeling e-Business, “Proceedings 15th Bled Electronic Commerce Conference”, Bled, Slovenia, June 17-19, 2002.
- Pateli, A.G. and Giaglis, G.M. (2003): A Framework for Understanding and Analysing e-Business Models, “Proceedings of 16th Bled Electronic Commerce Conference”, Bled, Slovenia, June 9-11, 2003.

- Petrovic, O., Kittl, C., and Teksten, R.D. (2001): Developing Business Models for e-Business, "Proceedings of the International Conference on Electronic Commerce", Vienna, Austria, October 31-November 4, 2001.
- Raskin, J., Tan, Y. and van der Torre, L. (1996): How to Model Normative Behavior in Petri Nets, "Proceedings of the Second Modelage Workshop on Formal Models of Agents (Modelage'96)", Sesimbra, Portugal, January 15-17, 1996, pp. 223-241.
- Tan, Y. and Thoen, W. (1998): A Logical Model of Transfer Obligations in Trade Contracts, Accounting, Management and Information Technologies, Vol. 8, pp. 23-38.
- Tan, Y. and Thoen, W. (2001): A Survey of Electronic Contracting Related Developments, "Proceedings of 14th Bled Electronic Commerce Conference", Bled, Slovenia, June 25-26, 2001.
- Tapscott, D., Ticoll, D. and Lowy, A. (2000): "Digital Capital – Harnessing the Power of Business Webs", Nicholas Brealy Publishing, London.
- Uschold, M., King, M., Moralee, S. and Zorgios, Y. (1998): The Enterprise Ontology, The Knowledge Engineering Review, Vol. 13, No. 1, pp. 31-89.
- Vassen, E., (2002): "Accounting Information Systems, a Managerial Approach", Wiley, Chichester.
- Williamson, O.E. (1985): "The Economic Institutions of Capitalism", The Free Press, New York.