Towards Control Patterns for Smart Business Networks

Vera Kartseva, Joris Hulstijn, Ziv Baida, Jaap Gordijn, Yao-Hua Tan
vkartseva@feweb.vu.nl

Vrije Universiteit, Amsterdam

Abstract

To keep a business network sustainable, controls are needed against opportunistic behavior of network participants. In this paper we present a methodology for understanding control problems and designing solutions, based on an economic value perspective. The methodology employs a library of so-called control patterns, inspired by design patterns in software engineering. A control pattern is considered as a generic solution for a common control problem. The adequacy and effectiveness of these control patterns is demonstrated by a case study about the redesign of customs procedures.

Keywords: governance and control, design patterns, business modeling

1. Introduction

Companies increasingly form smart business networks (SBNs) to jointly satisfy a complex need. Well known examples include the networked business models of Cisco Systems and Dell [28]. A *smart business network* is a group of enterprises with joint goals, linked by network technology, that collaborate and interact in such a way that the network remains sustainable and robust to defection of one of the actors, and allows each actor to increase its own value [30, p.229]. To be sustainable, a smart business network needs *control* measures, to prevent and detect opportunistic behavior of participants: a participant may leave the network, not fulfill its obligations, or commit fraud. From the early days of trade, transactions between organizations have been governed by administrative control procedures, which often rely on a transfer of *control documents* [5].

While some control procedures are of commercial nature, governmental institutions, like customs, have also introduced many control measures for trading partners. Currently, the various governmental procedures around international trade are fragmented and independent of each other. Consequently, overlaps exist between e.g., export, VAT, and excise procedures, as well as with commercial procedures. This results in a large number of different documents associated with a single container crossing the border, and in many required redundancies in business processes and information flows. Possible redesigns of customs procedures re-think why network partners work as they do, and how to use smart technological innovations to tackle control problems in the network.

We present a case study, called the *Beer Living Lab*, on the redesign of these customs procedures. In the study, governments, trading partners and technology providers collaborate in a network such that old paper-based customs procedures can be replaced by new procedures, relying on innovative technologies. All network partners benefit from the new technology-based procedures: customs administrations achieve a higher degree of control and security in international trade; trading partners achieve better control of their own supply chain, as well as increased credibility and therefore reduction in the required administrative procedures; and technology providers benefit from the research effort, to provide open-source, secure, IT-based services, using advanced technologies. The smartness of the collaboration lies in the fact that innovative third party commercial IT-based services enable customs administrations to achieve a higher degree of control, than when they would exercise control themselves.

It is crucial to guarantee that new IT-based procedures indeed provide the same safeguards as before, or better ones. To support the redesign of control procedures for smart business networks, a structured approach is needed. For individual enterprises, structured approaches to design control procedures have been developed in accounting and auditing fields (e.g. [27, 24]). But inter-organizational controls, as needed for a network of enterprises, have received only limited attention [5].

In this paper we, therefore, propose the use of *control patterns*: a way to structure existing knowledge about the design of control procedures, and make it accessible for re-use. Patterns make it possible to capture solutions for recurring problems in (business) systems that have been solved before. They provide a structured way of encoding the best practices that exist in a certain field. Such structured approaches are necessary, because domain experts often find it hard to make their knowledge explicit, and explain why a certain solution was chosen. The pattern approach has been proposed in architecture [1], and is very successful in software engineering [8]. Recently, patterns have also been applied to the business domain, for example for the design of administrative processes [21], organizational structure [7] and business process reengineering [4].

Based on a literature review, we have collected a library of generic control patterns, for the design and analysis of administrative control procedures in business networks. The adequacy and effectiveness of our library of control patterns is evaluated through a series of case studies, one of which is the Beer Living Lab. Based on the control patterns, we generate a *normative specification*, prescribing how controls should be designed according to the theory. The normative specification is then compared with real life scenarios, provided by domain experts.

The idea of using control patterns for design of business networks stems from conceptual modeling. Research on *business models* [3, 22, 29, 19, 23] utilizes conceptual modeling to provide concepts to judge and understand the viability of new (ICT-based) business initiatives. In the control patterns, we take two perspectives. Besides representing the procedural aspect of control mechanism, we also provide a representation of the *business model* that underlies the execution of the controls. We make use of modeling techniques, which stress the importance of *economic value*. With value modeling, the focus is on *what* actors do and *why*, while process modeling concentrates on *how* they do it [10]. Business models proved to be a useful support tool in a setting when decision making is done by multiple stakeholders [9]. Controls are typically not imposed by one central organization, but are negotiated among the participants of a network, where stakeholders have different interests, and different views on value propositions. This may lead to incomplete and ambiguous statements, when communicated in natural language [20]. The formal conceptualization of business ideas in a model, makes potential conflicts explicit, and may help stakeholders to resolve potential conflicts in an early stage of the development process.

The contribution of this paper is the following. *Control patterns* extend the work on formal models of controls by [5, 6] by providing a structured approach to model a wider set of control mechanisms. In addition, we aim for a methodology to design new control mechanisms, instead of checking the correctness or trustworthiness of existing control mechanisms, as in [5, 6]. Finally, we take a value perspective on design of controls, which integrates the control design in the *e^3-value* business modeling methodology.

The paper is organized as follows: Section 2 provides a theoretical background to our work: business modeling and control theory. Section 3 introduces the notion of a control pattern and Section 4 presents a library of general control patterns. Section 5 describes how we use and implement control patterns in a case study about the redesign of customs procedures.

2. Theoretical background

In the context of controls, a business network can be considered as: (1) an *ideal situation*, in which no errors, opportunistic behavior or fraud occurs, and (2) a *sub-ideal situation*, in which errors, opportunistic behavior or fraud does occur [15]. These situations must then be prevented, detected or corrected by means of control mechanisms. In accounting, ideal and sub-ideal situations are typically analyzed from a process perspective, e.g. with flow charts. From a network perspective, the ideal situation is often determined by contractual arrangements that reflect the business model of the network. Therefore, we define ideal and sub-ideal business models too.

There are other specific reasons for taking the value perspective into account, when looking at control issues. First, the value perspective is conceptually close to Transaction Cost Economics, which studies safeguards against opportunistic behavior in business relationships [32]. This facilitates a cost-benefit analysis of the control mechanisms, which in accounting control often also involves a risk assessment (e.g. [24, 27]). Second, control mechanisms are themselves services, with an additional price tag. That raises questions like: who is going to pay for a control mechanism, who is going to execute it, and how will it affect the business models of the parties involved? These questions are not particularly relevant from an internal control perspective, but in a business network controls may affect the profitability of participants, or may even lead to new business opportunities. Third, some control documents have inherent value aspects, and can for example be traded and resold (e.g. Bill of Lading, [5]).

2.1 Business Modeling

There are several methodologies that address the design issues of business models of network organizations, for example, BMO [22], value webs [28], and e^3*-value* [9, 11]. Of these methodologies, e^3*-value* is the only one that has a formal semantics, and that has a specific focus on business value transactions between enterprises. The method is ontologically and formally well founded, and is supported by graphical modeling tools (see www.e3value.com). In this paper, we therefore apply the e^3*-value* ontology for the description of ideal models [9, 11]. Sub-ideal models are expressed using e^3*-control,* a modification of the e^3*-value* ontology, which can be used to describe sub-ideal business models [15].

2.1.1 Ideal value models

An e^3*-value model* is a conceptual model (cf. the e^3*-value* ontology) of value transfers between enterprises in a smart business network. An e^3*-value model* incorporates concepts to represent the parties in a value constellation that transfer objects of economic value with other parties [9, 11]. The e^3*-value* constructs are supported by a graphical notation. Figure 1(a) shows an example of a buyer who obtains goods from a seller and offers a payment in return. According to the law, the seller is obliged to pay value-added tax (VAT). This can be conceptualized by the following e^3*-value constructs* (in bold). **Actors**, such as the buyer, seller, and the tax office are economically independent entities. Actors transfer **value objects** (payment, goods, VAT) by means of **value transfers**. For each value object, some actor should be willing to pay, which is shown by a **value interface**. A value interface models the *principle of economic reciprocity*: actors are only willing to transfer a value object, in return for some other value object. In this case that means that only if you pay, can you obtain the goods and vice versa. A value interface consists of **value ports**, which represent that value objects are offered to and requested from the actor's environment. Actors may have a **consumer need**, which, following a **path of dependencies** will result in the transfer of value objects. Transfers may be dependent on other transfers, or lead to a **boundary element**. In addition, the e^3*-value* methodology allows to assign monetary values to the value flows and to calculate profitability of actors in the business networks.

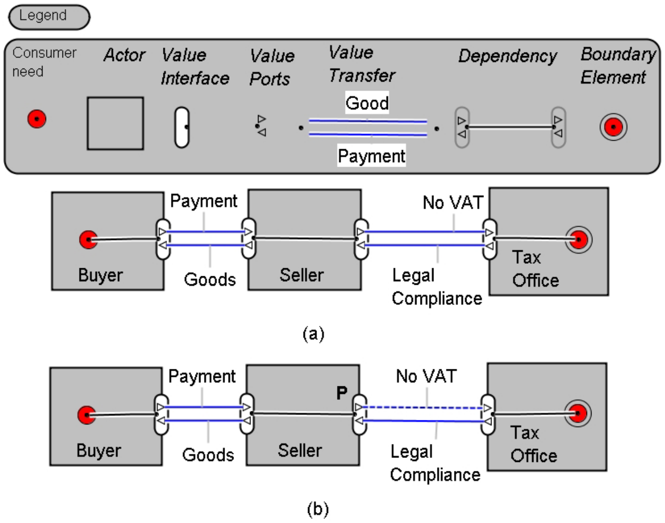


Figure 1 Example of an *e*^{3}*-value model* of a purchase with tax payment; (a) – ideal model, (b) – sub-ideal model

2.1.2 Sub-ideal value models

In *e*^{3}*-value* it is assumed that actors behave in an ideal way, meaning that all value transfers occur as prescribed. This implies, among other things, that actors respect the principle of economic reciprocity. But in practice, actors may not behave as represented in an *e*^{3}*-value model*: they can commit fraud or make unintentional errors. In *e*^{3}*-control*, these situations are modeled by **sub-ideal value transfers** [15]. These are graphically represented by a dashed arrow, and can indicate different risks: e.g. actors will not pay for the goods, not obtain the goods, or obtain the wrong goods. For example, Figure 1(b) models a situation when the seller does not pay VAT tax. **P** is a **liability token**, assigned to the actor who is responsible for the sub-ideal value transfer. In this case that is the Seller.

2.1.3 Process models

The administrative control mechanisms that we derived from the literature, described in section 2.2, are mainly applied to business processes, not to a business model . For example, delivery is verified before payment. The temporal order in which activities take place, forms a crucial part of the control mechanism. So in addition to business models we also need process models to capture control aspects. To represent the process aspects of control problems and their solutions, we need a graphical notation. For this purpose we use UML-activity diagrams [25]

2.2 Control theory

To identify the problems and controls in an organization, the internal control literature relies on a rigorous analysis of internal business processes. A general assumption is that every activity in a process is a potential source of control problems. Control problems are typically identified by an analysis of risk indicators and threats discovered in an audit process. A *control mechanism* is thus a procedural guideline on how to organize business processes in order to prevent, detect or reduce the risks posed by a control problem. This general framework is also relevant for inter-organizational controls [15].

Internal control theory is applicable to the design of inter-organizational controls only to some extend [5]. The methods for internal control solve control problems by internal organizational measures. By contrast, in inter-organizational settings, the control problems resulting from partners in a network [5], are mostly dealt with by contract arrangements Therefore, it is difficult to *directly* apply internal control guidelines for designing inter-organizational processes.

In search for a more formal theory of internal or inter-organizational controls, we studied the work of Bons et al. [5] and Chen&Lee [6]. Based on a review of internal control literature, they present a set of auditing principles (Table 1), which were developed to be implemented in a decision support system, which would detect risks, by analyzing a process and checking whether the auditing principles apply.

The auditing principles in Table 1 focus on control mechanisms, related to independent checks of performance, i.e. an audit (see also [24]). A *verification activity* (or control activity) audits the result of some *operational activity* with respect to legitimacy, quality or quantity [6]. The outcome of the operational activity is represented by a *to-be-verified document*. A *supporting document* is needed to make a decision about the correctness of the operational activity. For example, to verify whether the delivered goods were ordered, we can use the purchase order as a supporting document.

The auditing principles are clustered as follows. Principles I and II are about the order of activities. In this case, the verification has to occur after the operational activity. Principles III - VI put additional requirements on the supporting documents. For example, principle VI requires direct transfer of supporting documents: no intermediate parties should handle a document, when it is being transferred. This is crucial to avoid possible tampering, because a very high percentage of fraud cases involve the alteration of otherwise valid documents. Principles VII - X are concerned with task assignment, and ensure segregation of duties.

	Principles of precedence order of activities
I	Whenever an operational activity exists, a corresponding verification activity should also exist.
II	Whenever an operational activity and its corresponding verification activity exist, the verification activity should follow the operating activity.
	Principles of relation between information and activities
III	When a verification activity exists, it must be furnished with supporting documents.
IV	The supporting document should be the result of a previous verification activity
V	Supporting documents should be generated by a source independent of the source which generates the document to be verified.
VI	If a control activity uses a supporting document, the supporting document should be transferred directly from the verification activity that produced it.
	Principles of organizational structure (segregation of duty)
VII	A verification activity and the operational activity it intends to control should be segregated into two different positions.
VIII	A verification activity and the operating activity it intends to control should be delegated to two different agents.
IX	The position responsible for a verification activity must not be lower in the formal power hierarchy than the position of the operational activity to be controlled.
X	The agent responsible for a verification activity should be socially detached from the agent responsible for the operational activity to be controlled

Table 1 Audit principles of Chen [6], adapted for readability and coherence of terminology

In this research we focus on inter-organizational transactions in a business network . Since the principles in Table 1 are developed for internal control, not all the principles are relevant. Namely, principles VII and VIII distinguish between positions and agents. The term ’position’ refers to a set of activities that are the responsibility of a single agent in an organization. But at the network level, we only distinguish agents (or actors) [5]. Principle IX is not relevant, because at this stage of the research, we do not define hierarchical relations between organizations in a business network, as found for example in the supply chain of Cisco Systems

2.2.1 Control mechanisms and control principles

We distinguish between *control mechanisms* and *control principles*. Literature on expert systems for internal control shows that in identifying controls, experts apply deductive reasoning from *basic control principles* to more *specific control measures* (see e.g. [18]). For example, a control mechanism against stealing inventory is based on several principles: 1) periodic checks of inventory have to be executed 2) people checking the inventory must not be responsible for registering and accessing the inventory, 3) inventory has to be checked after changes in inventory have taken place. As one can see, normally, one principle is not enough to implement a control against one control problem.

Control principles define *basic* rules or norms that prescribe how a process should take place. According to Chen [6], there are 3 groups of principles (see section 2.3.1), namely (1) principles on the order of activities, (2) principles on the relation between information and activities and (3) principles of organizational structure. A control mechanism is a solution for some control problem, and can be represented as a combination of principles applied for a specific control problem.

The auditing principles in Table 1 cover only one type of control mechanism, related to auditing. Internal control theory knows other kinds of controls, which are not addressed. For example, controls related to proper authorization of transactions [24] are not covered by the auditing principles.

In this paper, the *control patterns* presented in section 4 extend [6] by providing a structured approach to model a wider set of control mechanisms. In addition, our pattern approach differs from work of [5, 6], because we aim for a methodology to design new control mechanisms, instead of checking the correctness or trustworthiness of existing control mechanisms.

3. Control patterns

3.1 Definition of a control pattern

A pattern is a description of a general and accepted solution for some recurring problem. Traditionally, a pattern has the following structure [8]: name, context, problem, solutions. As was said in section 2.2, for the design of control mechanisms we also need a problem-solution structure. Therefore, we expect that that a patterns approach would provide useful way to structure knowledge about controls.

Adapted for the domain of control, we define a control pattern as follows:

A **control pattern** is a description of a generic and re-usable control mechanism for a recurring control problem, selected on the basis of aspects of the context of application. The structure of a control pattern *(pattern template*) is the following:

1. **name**: a descriptive name of the pattern, used to select patterns from a pattern library.
2. **context**: a description of the business network to be controlled, modeled from an ideal perspective, meaning that no one behaves opportunistically. The context is represented by a value model (see section 2.1.1), and if needed, extended by a process model.
3. **problem**: a statement of one or more control problems, illustrated by scenarios that demonstrate a risk for opportunistic behavior. A control problem exists if there is some deviation of the prescribed transfers of economic value. Therefore, we model the problem by a sub-ideal value model (see section 2.1.2), using sub-ideal value transfers and liability tokens. Again, if needed, we also use a process model.
4. **solution**: a control mechanism, to detect, prevent or correct a control problem. The solution is described by process models and value models, and motivated by *control principles* (see below). A solution may have different *variations,* along with *forces*, which are conditions to select these variations. Since most of the control principles are of a procedural nature, the solution is modeled first in a process model, and then translated to a value model.
5. **related patterns**: description of the relations of the current pattern with other patterns in the library.
6. **control principles**: a set of *control principles* which express general constraints on the way a process must be performed. The principles fall into three groups, described in section 2.2.1.

3.2 Control patterns library

In this section we present a library of general control patterns. To extract the patterns, we used the so called `PattCaR method' from the patterns literature [26]. In short, the followings steps were followed:

- Step 1: Based on a literature review, we identified a preliminary set of control patterns. This literature review was based on text books in the field of internal accounting and control (among others [2, 13, 24, 27]). The list of patterns is shown in Table 2.
- Step 2: For each identified pattern we collected a number of examples both from the literature as well as from case studies. The case studies

were performed in different sectors, such as internet radio [14], renewable energy [15], international trade [17], and health care [16].

- Step 3: The examples were modeled using activity diagrams, and value models, and described in terms of the control principles (see Table 1).
- Step 4: A commonality-variability analysis [26] was performed to compare examples of each potential pattern. As a result we identified common concepts in the examples, such as activities, objects and control principles. This served as an input for encoding the patterns in pattern templates. See sections 3.3, 3.4 for two examples. The commonality-variability analysis also produced the pattern Verification (see Table 2), which was not identified as a pattern in Step 1.
- Step 5: Validation. We validate the patterns by identifying them in case studies, one of which is presented in this paper.

Pattern	Risk of primary actor	Control by primary actor
commitment confirmation	counter actor may deny to have made a commitment to primary actor	require confirmation of commitment from counter actor, before executing primary activity
commitment authorization	counter actor does not comply with norms of primary actor (e.g. sells goods of wrong quality, at inflated prices)	before making a commitment, require authorization from an actor who verifies if counter actor complies with norms
pre-execution	counter actor may not execute counter activity according to contractual agreement	require verification of counter activity before executing primary activity
execution confirmation	counter actor may claim that primary activity was not executed	require confirmation of primary activity execution from counter actor
post-verification	counter actor may execute counter activity in the wrong way	require verification of counter activity after execution of counter activity
verification	a verification activity introduced by another pattern, lacks evidence and standards	verify correctness and completeness of the results of an activity, against given standards

Table 2 Library of Control patterns

Vocabulary

Table 2 uses a specific vocabulary to describe the patterns. Identification of a common vocabulary is also a part of the PattCaR method [26]. In an inter-organizational setting [5] and also in the patterns, risks are assessed from the point of view of one actor called *primary actor*. Primary actor must decide about control mechanisms against sub-ideal behavior from another actor, called *counter actor*. From a value perspective, the primary actor transfers a primary value object (PO) to the counter actor. The counter actor transfers a counter value object (CO) to the primary actor. From a process perspective, the primary actor performs a *primary activity*, and the counter actor executes a *counter activity*, producing these transfers.

Any actor in a network can be both primary actor or counter actor. For example, both a buyer and a seller can be seen as the primary actor, when they consider the risks of a transaction with a party, whom they do not trust. As such, we consider a smart business network as a number of binary value transactions between actors. So we assume that all multi-party transactions can be decomposed as a combination of bi-party transactions.

The set of patterns in Table 2 covers the transaction cycle, which describe the four phases in which a transaction is concluded [31], see also [5, p. 30], namely (1) *preparation* or *selection* phase, (2) *negotiation* phase, (3) *performance* or *execution* phase, and (4) the *evaluation* or *acceptance* phase. These phases can be related to the patterns in the following way. The *preparation* and *negotiation* phases are dealt with by the patterns Commitment Confirmation, and Commitment Authorization. The pattern Pre-execution deals with the *execution* phase. The Execution Confirmation and Post-verification pattern cover the *acceptance* phase. Verification can occur as part of any phase.

Patterns can make use of other patterns. In this way, the execution of two different activities can be linked. In particular, the Pre-execution pattern states that the primary activity will only be executed, after a satisfactory outcome of a Verification of the counter activity. Verification is also applied as part of the Post-Verification pattern. The Verification pattern specifies additional constraints on the context and on the documents needed in any verification activity. Because these constraints are generic, we did not want to repeat them for the individual patterns. In the diagrams, this is indicated by the notation <Verify>, to illustrate that this activity is effectively a variable which needs to be supplied with more specific activities. We will now explain two patterns in detail, namely Post Verification and Verification, because these are applied in the case study.

3.3 Pattern "Post Verification"

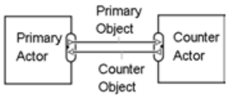
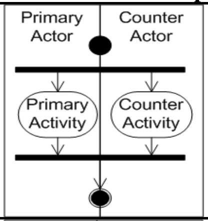
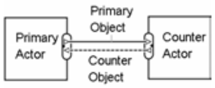
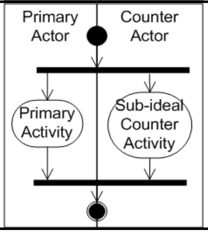
Name: Post verification
Context: The primary actor and counter actor transfer value objects, called primary object (PO) and counter object (CO). The primary actor does not trust the counter actor about execution of CO
Problem: The counter actor, if not controlled, may deliver wrong quality, wrong quantity or wrong type of counter object (CO).
Solution: The primary actor must ensure that a verification activity is executed, after the counter activity has been executed. The verification activity controls the result of the counter activity.
Force a: The primary actor is able to verify counter activity
Variation a: Add a <Verify> activity, executed by the primary actor
Force b: The primary actor is not able to verify counter activity
Variation b: The <Verify> activity is delegated to a trusted third party (TTP), who transfers an evidence document with the result of the verification activity to the primary actor.
Related patterns: This pattern uses the pattern "Verification", which puts further constraints on the <Verify> activity, and the required evidence and documents.
Control Principles: **Principles of order of activities** **1**. Whenever a counter activity exists, a verification activity <Verify> must also exist. <Verify> has to be specified with pattern Verification. **2**. The <Verify> activity must be performed after the counter activity **Principles of relations between information and activities** **3.** Variation b: the TTP has to send the primary actor an evidence document with the results of the verification. **4.** Variation b: The evidence document must be transferred directly to the primary actor **Principles of organizational structure** **5**. The <Verify> activity must be assigned to an actor who is independent and socially detached from the counter actor.

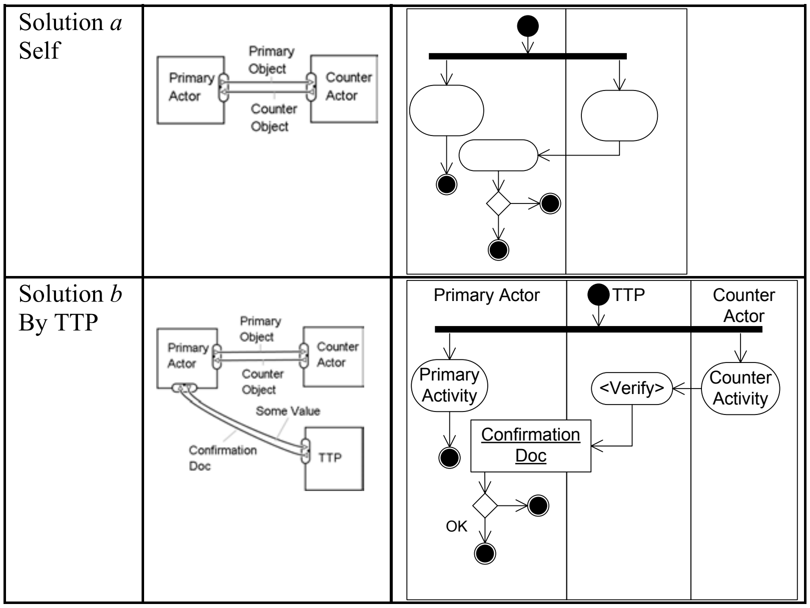
Table 3 Pattern "Post-Verification"

This pattern deals with the risk, mentioned in section 2.2, that the counter actor may not deliver, or deliver the wrong quality, quantity or type of counter object (CO). For example, a buyer runs the risk of having to accept goods or services that were not ordered, to accept damaged goods, or to pay for a wrong amount of goods or services. The suggested control mechanism is to verify *afterwards* the correctness of the goods and verify the amount on the invoice with what was agreed, and with what was actually delivered [24, 13]. We assume that any activity of a primary actor,can be delegated to a trusted third party (TTP). In this pattern, we specifically model the delegation of the <verify> activity to a TTP, since it introduces a new control document, and because it illustrates that the introduction of controls can change the value model.

The control principles listed in Table 3, are based on the audit principles of Chen [6] (see section 2.2), and also on work of Bons et.al. [5]. Principles 1 and 2 are based on Chen's Principle I and II respectively. Principle 3 is motivated by the following delegation principle from Bons et al [5]: *"If Role 1 cannot witness the performance of a counter-activity by some Role 2, then another Role 3 should testify the completion of Role 2's activity, if the party playing Role 2 is not trusted by the party playing Role 1"* . Principle 4 is based on Chen's Principle VI (direct transfer). Finally, principle 5 is based on Principle VIII and X (segregation of duties).

Figure 2 contains a graphical version of the Post Verification pattern. In Figure 2, the reciprocal value transfer of the two value objects PO and CO in the value model on the left corresponds to the execution of a primary activity and a counter activity, performed by the actor with the outgoing value port. The order in which these activities occur is not specified. This is indicated by the UML notation for parallel execution (thick horizontal bar), which is often given a so called interleaving semantics. Also notice the changed value model in solution *b*. in which the verification activity is delegated.

	Value perspective	**Process perspective**
Context	 Primary Actor Primary Object Counter Object Counter Actor	 Primary Actor Counter Actor Primary Activity Counter Activity
Problem	 Primary Actor Primary Object Counter Object Counter Actor	 Primary Actor Counter Actor Primary Activity Sub-ideal Counter Activity



Prir

Pri
Ac

Figure 2: Post Verification pattern

3.4 Pattern "Verification"

This pattern deals with additional constraints on the evidence and supporting documents needed in a verification activity. The motivation of the control principles in Table 4, is again based on work by Bons [5] and Chen [6]. The additional verification <New Verify> and resulting supporting documents of principle 2 and 3, are based on Chen's Principle III and IV. Principle 4 and 5 about the input of reconciliation, are loosely based on Chen's principle III. Like before, principle 6 is based on Chen's principle VI (direct transfer), and principle 7 is based on Chen's principles VIII and X (segregation of duties).

The Internet Radio case [14] provides an interesting example of a combination of the patterns Post verification and Verification. Three parties are involved: a listener, a rights society and a radio station. The listener provides data to the rights society about the music downloaded by him. These data are used as a supporting document for the rights society, to check how many music rights were exploited by the radio station. This in turn determines the fee to be paid by the radio station to the rights society. A transfer of evidence documents in the health care sector [16] can also be captured by these two patterns.

Name: verification
Context: The counter actor transfers a value object PO and CO. The <Verify> activity of the counter activity is executed by the primary actor, or has been delegated to a trusted third party (TTP).
Problem: The <Verify> activity is lacking the evidence and/or documents on which to base its assessment.
Force a: The primary actor has direct access to the outcome of the execution of the counter activity, and has certain standards that prescribe what the outcome of the counter activity should be.
Solution a: The <Verify> activity is instantiated with a *witnessing* activity, which involves an observation of the outcome of the counter activity and a comparison with the standard.
Force b: The primary actor is not able to witness the execution of the counter activity, or does not have standards for comparison.
Solution b: The <Verify> activity is instantiated with a reconciliation of the outcome of the counter activity, i.e. the documents to be verified, with some other supporting documents, which provide a standard or norm about what the outcome should be. Since primary actor has no access to counter activity, the supporting documents must be provided by some trustworthy actor TTP
Related patterns: This pattern is used by the patterns `Post Verification’ and `Pre-Execution’. This pattern uses the pattern Verification again, to further specify details on the <New Verify> activity that supplies the supporting documents.
Control Principles: **Solution a:** Substitute <Verify> with activity Witness. **Solution b:** **Principles of order of activities** **1.** Substitute <Verify> with activity Reconcile **2.** Add <New Verify> activity, before Reconcile and after the Counter Activity. <New Verify> has to be instantiated according to Verification. **Principles of relations between information and activities** 3. Add supporting documents as an outcome of the <New Verify> 4. Add to-be-verified document as an outcome of Counter Activity 5. Reconcile has the to-be-verified document and the supporting documents as incoming objects 6. The supporting document has to be sent *directly* to Reconcile **Principles of organizational structure** 7. <New Verify> and Reconcile have to be performed by a party socially detached from the counter actor.

Table 4 Pattern “Verification”

Figure 3 provides a graphical representation of the pattern. Remember that all activities of the primary actor, including verification, can also be delegated to a TTP (solution *b*)

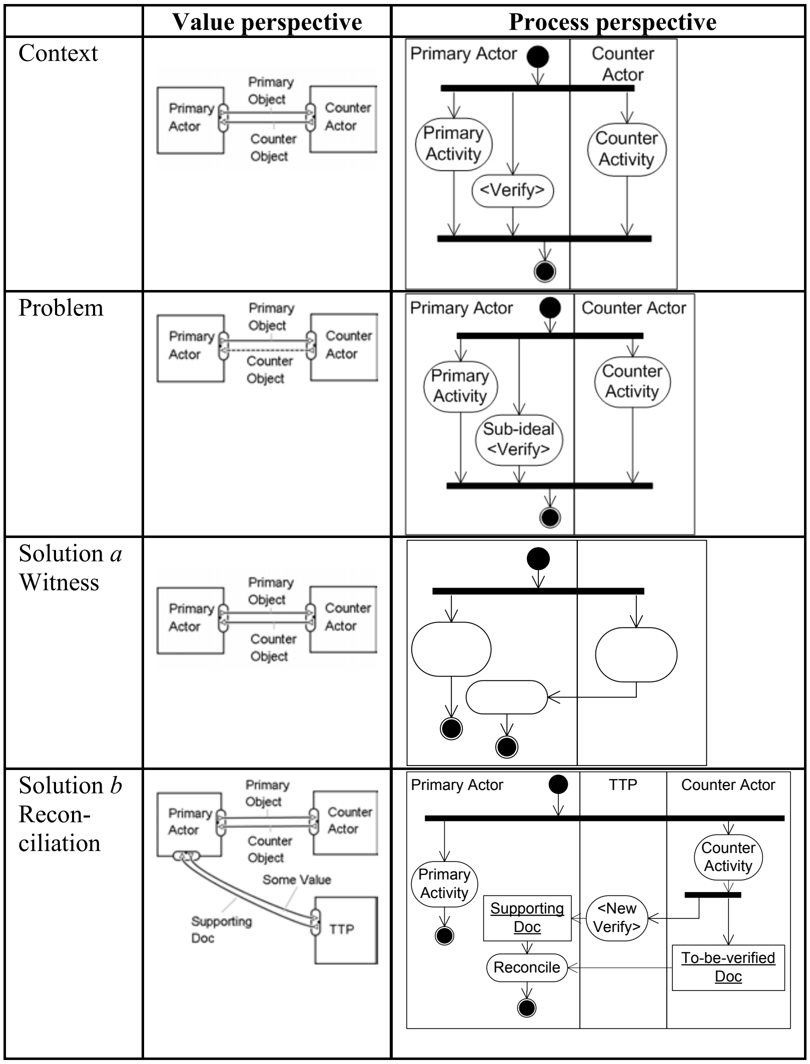


Figure 3: Verification pattern

4. Case Study: Beer Living Lab

In this section we evaluate the adequacy and effectiveness of our library of control patterns through a case study: the Beer Living Lab. The study focuses on the redesign of customs procedures for collecting excise duties.

4.1 Research Method

The use of the control patterns was studied in collaboration with domain experts. We talked to domain experts from the Dutch Customs and Tax Administration, and also to experts from a large beer producer in the Netherlands. Hence, the Beer Living Lab can be categorized as an observational case study [12].

The study was designed as follows. First, an initial analysis of the case was made, on the basis of interviews with domain experts and existing documents. In this initial phase, we identified the various actors involved, and the objects of value they transfer, and we made a partial analysis of the business processes regarding excise duties. Second, we identified control problems in the initial case description, and we applied the corresponding solution .This produced a normative specification of controls, which contains a process model of the control mechanisms that ought to be implemented. Forth, this normative specification was compared with a number of real life scenarios, provided by the domain experts.

This comparison was meant to validate our interpretation of the case and our application of the control patterns. Moreover, as part of the redesign effort of customs procedures, we wanted to find out whether the real life procedures needed to be redesigned. Note in this respect, that control patterns are *prescriptive*: they do not (only) describe facts, but rather prescribe how control mechanisms ought to be designed. So there are two possible ways in which a real life scenario can deviate from a normative specification: either (1) the specification is based on a wrong or incomplete interpretation of the relevant facts, or (2) reality does not conform to the norm, which means that we can identify some remaining control problems.

So we execute three steps, as in [15]. First, we model the case using ideal value and process models. Second, we identify the control problem, by modeling a sub-ideal value model. Third, we model a control mechanism. For this, we match the control problem with problems in the pattern library in Table 2 and select a pattern, which deals with this problem. Then we apply the solution of the pattern to our ideal model.

4.2 Case Description

When goods like beer and cigarettes, called excise goods, are sold, the seller needs to pay a special tax called excise. The general principle is that excise only has to be paid in the country in which the excise good is sold and consumed. Hence, if a beer producer in the Netherlands, say BeerCo NL, is exporting beer, possibly via the business unit BeerCo UK, to a retailer in the UK who sells the beer to English consumers, excise has to be paid by the English retailer to Customs UK[1]. In this case, the beer producer in the Netherlands can export excise-free. Clearly, this is only acceptable for the Dutch Customs and Tax Administration, if the beer producer in the Netherlands can prove that the goods were indeed shipped abroad. The procedures currently revolve around the transfer of paper documents. The core document for this excise-free export procedure is the Administrative Accompanying Document (AAD). This document is signed by a so-called excise warehouse (EW) in the UK. Customs UK subsequently signs the AAD, to confirm that the goods did indeed arrive in the UK. Finally, the AAD is returned to the Dutch beer producer as evidence that the goods have arrived in the UK and will be presented to Dutch Customs and Tax upon request. In this paper we specifically look at the control problems of the AAD procedure. We analyze the AAD procedure and indicate how the procedure can be replaced by smart technology.

4.3 Ideal value model

When BeerCo NL exports beer to the UK, no excise is due in the Netherlands. When BeerCo NL can prove delivery outside the Netherlands, it is exempted from excise duties and is considered compliant with the law (see transfer between BeerCo and Customs NL in Figure 4). BeerCo UK sells the beer to a Retailer with EW: a retailer licensed for excise handling. The retailer with EW sells the beer to UK supermarkets, for a price that covers the excise, and pays excise to Customs UK. Figure 5 shows a process model that corresponds with the business model in Figure 4.

[1] Note that in some countries excise is considered a tax issue, while in other countries it is considered a customs matter; we therefore refer to Tax and Customs organizations interchangeably.

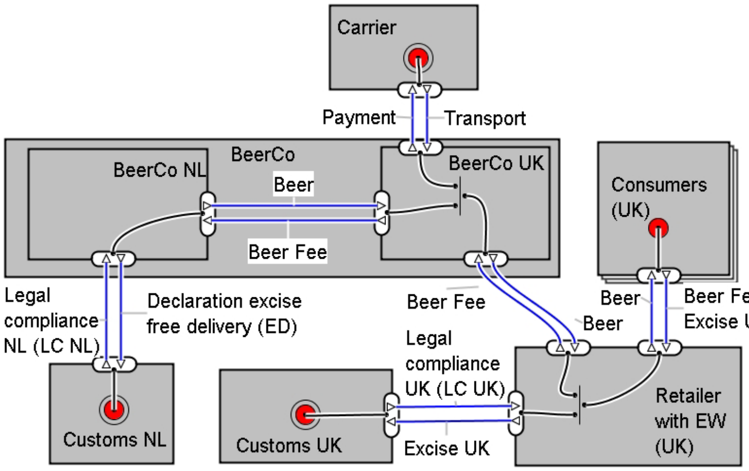


Figure 4 Ideal business model for beer export

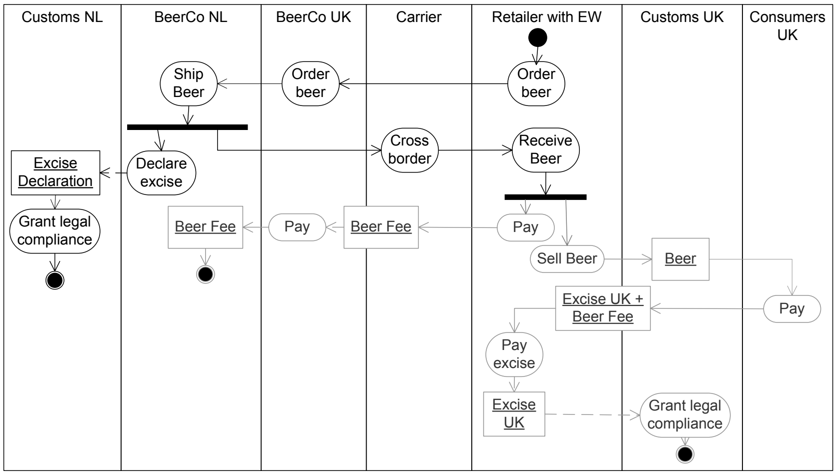


Figure 5 Corresponding (partial) process model for beer export from The Netherlands to the UK

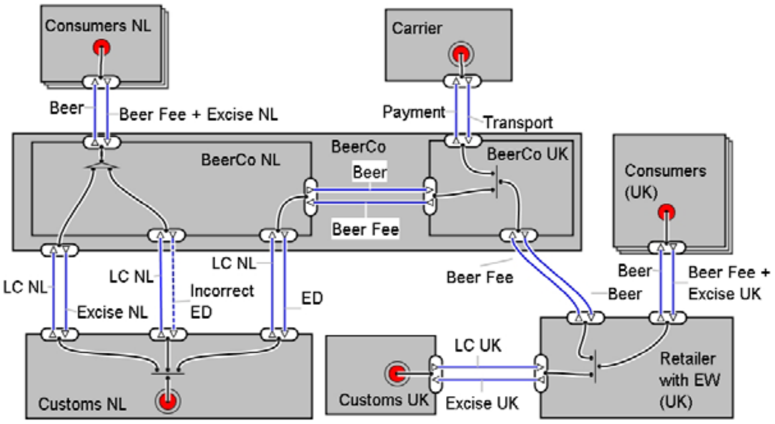


Figure 6 Sub-ideal value model for beer export from The Netherlands to the UK

4.4 *Sub-ideal value model*

The main control problem of the AAD procedure, is that BeerCo NL can declare that beer, which is in fact sold in the Netherlands, is excise free. This problem is modeled in Figure 6. BeerCo NL delivers beer to consumers in the Netherlands, for which it does have to pay excise duties to Customs NL. However, BeerCo NL behaves opportunistically: the OR-fork (triangle) at the BeerCo NL models that BeerCo NL has a choice to pay excises for the beer sold in NL (Excise NL), or to declare it as imported (incorrect ED) and still get an excise exemption (LC NL)

4.5 *Normative Specification*

Now we match the control problem to one of the patterns. Customs NL is *primary actor* and BeerCo NL is *counter actor*. The *counter activity* is Cross Border, and the *primary activity* is Grant legal compliance. Note that Cross border is taken as the counter activity, not Excise Declaration, since Cross border indicates whether the goods can be exempted from excise. The corresponding control problem is described in the patterns Pre-execution and Post-verification. Other patterns deal with different problems (see Table 2). Post-verification requires verification of every declaration of excise free beer. The initial model in Figure 5 does not have such a verification and legal compliance is granted by Customs NL for every declaration of excise. Pre-execution requires Customs NL to execute “Grant legal compliance” after the excise declaration. The latter is already the case in this process, so we apply pattern Post-verification.

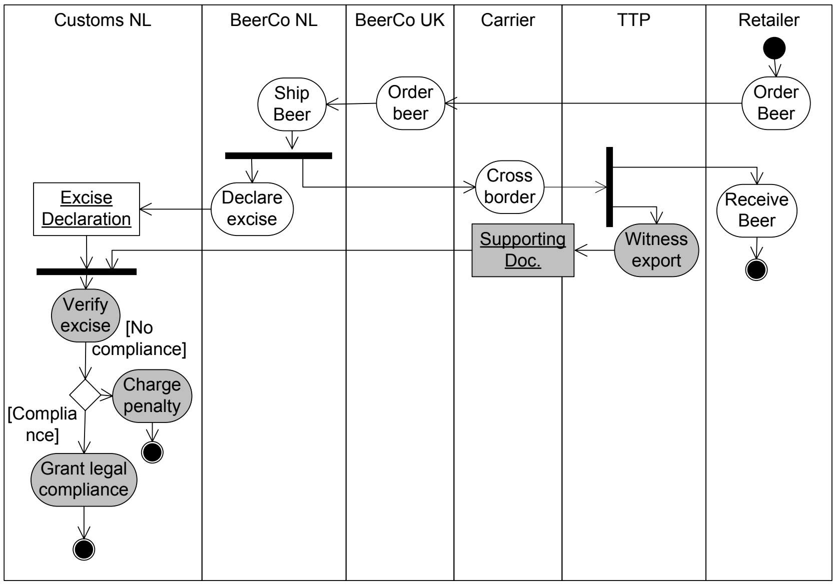


Figure 7 A normative specifications with controls

***Application of pattern Post-verification*.**

Pattern Post-verification has two variations. Variation *a* is applied when the primary actor has direct access to the results of the counter activity. In our case, Customs NL have direct access to the excise declaration, so we apply Variation *a*. According to the control principles in Table 3, the following steps are executed:

1. Add activity Verify Excise. The Verify Excise activity must verify if the beer declared by BeerCo NL to be excise free, was indeed exported. This activity is executed by Customs NL.
2. Put activity Verify Excise after activity Cross Border.
3. Verify excise is assigned to Customs NL

***Application of pattern Verification*.**

Now activity Verify Excise has to be further specified by pattern Verification. Pattern Verification has two variations. Variation *a* is applied when the primary actor can witness the performance of the counter activity, and has certain standards that prescribe what the outcome of the counter activity should be. Otherwise, variation *b* is applied. The most elegant solution would be variation *a*, in which the verification is carried out by Customs NL. But since officially there are no borders between EU member states, Customs NL cannot physically witness the export of goods. So, we choose variation *b*, where Customs NL must rely on documentary evidence confirming export from some other party. According to the control principles in Table 4, the following steps are executed:

1. Verify Excise becomes a Reconcile activity (variation *b*).
2. Add a <New Verify> activity, before Verify Excise and after Cross Border. It becomes a witnessing activity (variation *a*), called Witness Export, by another application of the Verification pattern (Table 6). This Witness Export activity is delegated to a TTP, a hypothetical party who must have direct access to the delivery.
3. Add Supporting Doc, a document that contains evidence of the outcome of Witness Export.
4. The to-be-verified document is already included in our model. This is the Excise Declaration.
5. Excise Declaration and Supporting Doc are incoming objects to the activity Verify Excise.
6. Supporting Doc is sent *directly* to Verify Excise
7. Verify Excise is performed by Customs NL, which is indeed socially detached from BeerCo NL. Witness Export is performed by a hypothetical actor TTP, who must have direct access to the delivery (see above) and must be socially detached from BeerCo NL.

Figure 7 shows the result of applying the patterns to the original model in Figure 5. This is called the normative specification. The activities in dark are the new elements added to the model by the patterns[2]. Many actors can fill the role of TTP, and supply the supporting documents. Thus, the process model in Figure 7 acts as a kind of requirements specification that can be instantiated in different ways, by specifying which actor performs the TTP's activities.

4.6 Real-world scenarios

In this section we examine current and future export practices, and compare them to the normative specification of controls. First, we analyze the current AAD procedure, modeled in Figure 8. In this process the role of the TTP is performed by a Retailer with EW. In fact, Customs NL delegates the witnessing of export to Customs UK, and Customs UK further delegates it to a Retailer with EW. Another difference between the normative specification (Figure 7) and the current practice (Figure 8) is that in reality the AAD is not transferred directly to Customs NL. The AAD is

[2] Note that for brevity Figure 7 only contains those parts of the model in Figure 7 that are related to the control mechanism. The light-grey part of Figure 5 is not included.

transferred first to a carrier, then to BeerCo NL, and finally to Customs NL. This indirect transfer violates control principle 6, which says that the supporting document AAD should be transferred directly to Customs NL, to prevent manipulations of the AAD. Only if a supporting document cannot be forged, which is not the case here, an indirect transfer is acceptable. According to the domain experts, this diversion from the normative specification indicates a real and existing control problem.

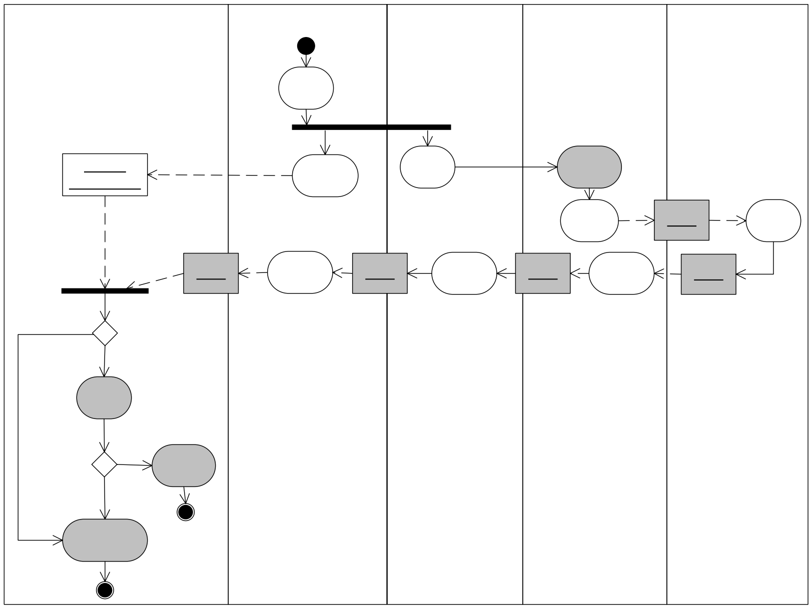


Figure 8 “As-is” model with AAD

The verification of Excise Declarations does not take place on a 100% basis, which is a control weakness. The normative specification suggests that control should take place every time the declaration is submitted.

Interesting is the role of Retailer with EW. In fact, not every retailer can perform a role of TTP, only a retailer certified by customs UK for an Excise Warehouse. This certification is also a control mechanism, covered by the pattern Commitment Authorization (see Table 2). For lack of space we do not consider that in this paper.

We have also performed an analysis of other real-life scenarios, including export to the USA, where export procedures are different from those in the EU, and export using the EMCS system. The EMCS system is currently being developed as an EU-wide information system for excise movement control. For a large part, these scenarios also proved to be instantiations of our normative specification, but just like in the AAD case, we were able to identify risks, by comparing these scenarios to the normative specification, and validating our findings with domain experts.

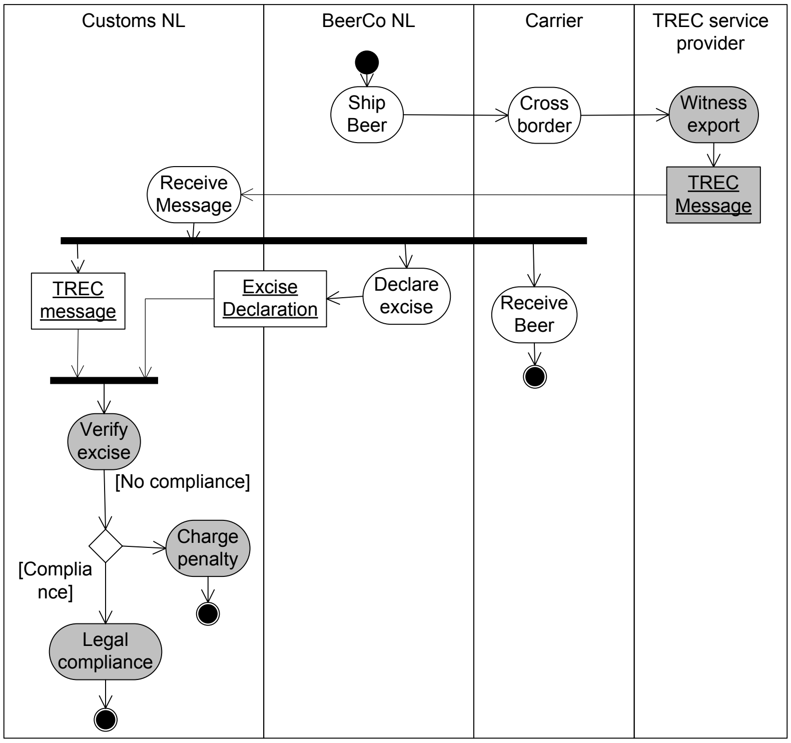


Figure 9 Export of beer using a TREC device

The control problem with the AAD that is vulnerable to tampering can be solved by means of advanced technology, in particular, by the TREC device (Tamper-Resistant Embedded Controller[3]) that is currently under development. TREC is a sensor device that can detect whether a container is opened by an authorized or non-authorized person, and send a message (e.g. to Customs NL or BeerCo NL) in case of non-authorized opening. It is intended to reduce fraud and increase security. By monitoring a container's position coordinates, a message can be automatically sent to Customs NL when the container actually leaves the Netherlands. TREC de-

[3] Further information on TREC is available at http://domino.research.ibm.com/odis/odis.nsf/pages/board.06.html

vices could therefore replace the AAD's functionality to provide evidence of export. Because the TREC technology is still under development, we propose a possible implementation of this scenario (Figure 9). Figure 9 is another instantiation of the normative specification. The TREC device performs the "Witness export" activity. In our terminology, the TREC device performs the role of TTP. This scenario is an improvement, because the TREC device can send an electronic message directly to Customs NL, at the exact moment when the container actually crosses the border. By contrast, the AAD would typically only return to BeerCo NL after three months.

A pre-requisite for this scenario is that the party operating the TREC device is independent and socially detached from BeerCo NL on the company level and on the employee level (see control principle 7 of pattern verification). This refers to the risk that an employee can send a message about crossing the border while the border was not crossed. Therefore, a TREC device must send a message without human intervention.

5.7 Lessons Learned

Regarding the redesign of customs procedures for excise duties, we can conclude that the current EU practice is vulnerable. The AAD, which should prove that goods have indeed left the country, is transferred along the supply chain, and can therefore be tampered with. This control problem can be mitigated in several ways. One of these involves a technical device, the TREC, and a new actor, the TREC service provider. The export procedure that involves TREC is currently being designed as an instantiation of the normative specification generated by the control patterns.

Regarding the use of the control patterns, we can make some initial observations about adequacy, and effectiveness. There is a general trade off between adequacy and generality. Patterns must be formulated in such a way, that they can be generalized to different contexts, but if you generalize too much, patterns may become inadequate, because they are no longer relevant. We believe that the case study shows that the control patterns in our library are not too general: they helped to reveal real control problems, and a possible solution. The case study shows that the patterns can be applied effectively in a complex domain. The patterns have proved effective, in summarizing existing principles and ideas in a concise format. In particular the value models proved to be useful when communicating with experts. They provide a means to achieve a common understanding of problems, without getting into the details.

5. Conclusions

In order to be sustainable, smart business networks need mechanisms for governance and control. However, existing approaches for the analysis and design of business models do not address the risk of fraud or network participants who fail to live up to their commitments. Such opportunistic behavior can be detected and prevented by incorporating explicit control mechanisms in the inter-organizational business processes that regulate transactions in the network. In this paper we contribute to governance and control issues in smart business networks.

We suggest *control patterns*, which structure existing domain knowledge about inter-organizational controls. The primary intent of a pattern is to provide a useful abstraction of an existing solution to a recurring problem, for the sake of reuse.

From a theoretical perspective, the control patterns are innovative in several ways. First, the library of control patterns structures existing knowledge to make it transferable to the inter-organizational domain. To elicit patterns, we use literature on formal models of control principles [5, 6]. Our contribution differs from this literature, because we aim for a methodology to design new control mechanisms, instead of checking the correctness or trustworthiness of existing control procedures. In addition, we use literature on internal controls in accounting [24, 27]. Compared to this literature, patterns (1) provide a structured way of designing controls, using conceptual modeling techniques well-known in software engineering and (2) put controls into *inter-organizational* setting. Finally, one of the objectives of our design approach is to enable evaluation of the impact of control mechanisms on the *business models* that underlie smart business networks. This is innovative compared to the field of internal control, where, conventionally, only process models are looked at to design controls, and no conceptual modeling tools are used to perform a cost-benefit analysis. For this purpose we use the *e^3-value* methodology, which enables application of scenario techniques to evaluate and compare different control mechanisms from a value perspective.

We demonstrate how to put theory into practice by a large scale case study about the redesign of customs procedures regarding excise duties: the Beer Living Lab. By applying control patterns we created a normative specification, revealing that current EU practice is vulnerable to fraud with excise payments. Furthermore, the normative specification is used in a project in which businesses and governments participate to design new customs procedures, enabled by advanced sensor technology. The new export procedure is being designed according to the control patterns. An expert panel consisting of representatives of the participating organizations confirmed that the normative specification, which is based on application of the control patterns, does manage to mitigate control problems in the business network.

One of our scenarios introduces a new actor to the business network: the TREC service provider. That means that the initial business model of the business network has to be changed, and that the financial feasibility of the new business model must be closely examined. This analysis is enabled by *e^3-value*, a business modeling methodology, our patterns approach is based on. These issues will be discussed in a separate paper.

The resulting business network comprising of trading partners, customs administrations and technology providers, is smart in two ways. First, it uses advanced technologies to replace manual and paper-based procedures, resulting in up to 100% control compared to typically 2%-5% control in the current situation. Second, to achieve this, network partners rely on 3rd party commercial services instead of implementing control by themselves. This is not new for businesses, but it is a more novel idea (and practice) for governments.

Acknowledgements The research of the first author is funded by the Post Master EDP Audit Education of the Vrije Universiteit Amsterdam. Part of the research was funded by The Freeband FRuX project, and by the EU-funded IST project ITAIDE. We are greatly indebted to the other participants of these projects for their valuable contributions to the work presented here.

References

1. Alexander, C. (1979). The Timeless Way of Building. Oxford, Oxford University Press.
2. Arens, A.A., Loebbecke, J.K. (1997) Auditing. Prentice Hall; 7th Revised edition
3. Baida, Z., Gordijn, J., Akkermans, H., Sæle, H. and Morch, A.Z. (2005). Finding e-Service Offerings by Computer-supported Customer Need Reasoning, International Journal of E-Business Research 1(3): 91-112.
4. Beedle, M. (1997) Pattern Based Reengineering. Object Magazine, January

5. Bons, R. W., Lee, R. M., and Wagenaar, R. W. (1998). Designing trustworthy inter-organizational trade procedures for open electronic commerce. International Journal of Electronic Commerce, 2(3):61–83.
6. Chen, K. (1992). Schematic Evaluation of Internal Accounting Control Systems. PhD thesis, University of Texas at Austin.
7. Coplien, J. O., Harrison, N.B. (2004) Organizational Patterns of Agile Software Development, Prentice Hall
8. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley, Boston.
9. Gordijn, J. and Akkermans, J. (2001). e3-value: Design and evaluation of e-business models. IEEE Intelligent Systems, Special Issue on e-Business, 16(4):11–17.
10. Gordijn, J., Akkermans J.M., and Van Vliet J.C. (2000), "Business Modelling is not Process Modelling". in: Conceptual Modeling for E-Business and the Web, LNCS 1921, pg 40-51 .
11. Gordijn, J. and Akkermans, J. (2003). Value-based requirements engineering: Exploring innovative e-commerce ideas. Requirements Engineering, 8(2):114–134
12. Hevner, A. R., Ram, S., and March, S. T. (2004). Design science in information systems research. Management Information Sytems Quarterly, 28(1):75–105.
13. Hollander, A.S., Denna, E., Cherrington, J.O. (1999) Accounting, Information Technology, and Business Solutions. 2nd ed McGraw-Hill
14. Kartseva, V., Gordijn, J., and Tan, Y.-H. (2005a). Designing control mechanisms for value webs: The internet radio case study. In Proceedings of the 18th Bled ElectronicCommerce Conference – eIntegration in Action (Bled'05)
15. Kartseva, V., Gordijn, J., and Tan, Y.-H. (2005b). Towards a modelling tool for designing control mechanisms in network organisations. International Journal of Electronic Commerce 10(2):57–84.
16. Kartseva, V. and Tan, Y.-H. (2005). Designing controls for a marketplace of health care services: a case study. In Proceedings of the 12th Research Symposium on Emerging Electronic Markets (RSEEM 2005). Vrije Universiteit, Amsterdam
17. Kartseva, V., Hulstijn, J., Gordijn, J., and Tan, Y.-H. (2006). Towards Value-based Design Patterns for Inter-Organizational Control. In Proceedings of the 19th Bled Conference *eIntegration in Action* (Bled'06).
18. Looi, C.-K., Tan, S. L., Teow, P. C., and Chan, H. S. (1989). A knowledge-based approach for internal control evaluation. Proceedings of the 2nd international conference on Industrial and engineering applications of artificial intelligence and expert systems, 1 (1989), 254–261.

19. 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. (1999). Tools for Inventing Organizations: Towards a Handbook of Organizational Processes. Management Science, 45 (3): 425—433.
20. Meyer B. On Formalism in Specifications. *IEEE Software*, 2, 1 (1985), 6 -26.
21. Motschnig-Pitrik, R., Randa, P., and Vinek, G. (2002). Specifying and analysing static and dynamic patterns of administrative processes. In Proceedings of the 10th European Conference on Information Systems (ECIS 2002), Gdansk, Poland.
22. Osterwalder, A. (2004), The Business Model Ontology: A Proposition in a Design Science Approach, PhD thesis, University of Lausanne, Lausanne, Switzerland.
23. Pateli, A.G., Giaglis, G.M. A Research Framework for Analysing Business Models. *European Journal of Information Systems*, 13, 4 (2004), 302-304.
24. Ronmey, M. and Steinbart, P. (2003). Accounting Information Systems. Prentice Hall, New Jersey, 9th edition.
25. Rumbaugh, J., Jacobson, I., and Booch, G. (1999). The Unified Modelling LanguageReference Manual. Addison Wesley, Reading, MA.
26. Seruca, I. and Loucopoulos, P. (2003). Towards a systematic approach to the capture of patterns within a business domain. The Journal of Systems and Software, (67):1–18.
27. Starreveld, R., de Mare, B., and Joels, E. (1994). Bestuurlijke Informatieverzorging (in Dutch), volume 1. Samsom, Alphen aan den Rijn, 4th edition.
28. Tapscott, D., Lowy, A. and Ticoll, D. (2000): Harnessing the Power of Business Webs, Harvard Business School Press, Boston
29. Tillquist, J., King, J., and Woo, C. (2002) A Representational Scheme for Analyzing Information Technology and Organization Dependency, Management Information Systems Quarterly, 26(2): 91-118.
30. Vervest, P., Preiss, K., van Heck, E., and Pau, L.-F. (2004). Introduction to smart business networks. Journal of Information Technology, 19: 228–233.
31. Weigand, H., & Moor, A. de (2003). Workflow analysis with communication norms. Data and Knowledge Engineering, 47(3): 349-369.
32. Williamson, O. E. (1979). Transaction cost economics: The governance of contractual relations. Journal of Law and Economics, 22:3–61.