Control Patterns in a Health Care Network

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

[1] Faculty of Economics and Business Administration, Vrije Universiteit, Amsterdam
De Boelelaan 1105, 1081 HV Amsterdam
{ vkartseva, jhulstijn, ytan}@feweb.vu.nl
[2] Faculty of Sciences, Vrije Universiteit, Amsterdam
de Boelelaan 1081a 1081 HV Amsterdam
gordijn@few@vu.nl

Abstract. In this paper we present *control patterns* for the analysis and design of administrative control mechanisms in a network organization. A control pattern is a description of a generic and reusable control mechanism that solves a specific control problem, to be selected on the basis of the context. To represent the context and solution, we analyze a network organization as a set of actors who transfer objects of economic value. The usefulness and adequacy of the control patterns is demonstrated by a case study of the governance and control mechanisms of the Dutch public health insurance network for exceptional medical expenses (AWBZ).

Keywords. governance and control, network organizations, value modeling

1 Introduction

Multi-agent systems are computer systems which are composed of a number of autonomous agents, i.e., pieces of software, interacting to achieve the global systems objectives. Applications can for example be found in transport logistics [9], manufacturing scheduling [38], and social simulation [35]. Increasingly, developers of multi-agent systems have turned to concepts from the social sciences like organization structures and norms, as a guideline for design, e.g. [48][29]. Because of the desired global system objective, the autonomy of the individual agents must generally be restricted: the decisions of agents should be bound by rules, regulations and norms. Especially in an open environment, in which heterogeneous agents may freely enter, leave and interact, a multi-agent system becomes a *normative* multi-agent system [13]. Normative multi-agent systems are

> “sets of agents (human or artificial) whose interactions can fruitfully be regarded as norm-governed; the norms prescribe how the agents ideally should and should not behave. [...] Importantly, the norms allow for the possibility that actual behavior may at times deviate from the ideal, i.e., that violations of obligations, or of agents rights, may occur” [23].

Generally, there are two good reasons to employ multi-agent systems: (1) as a paradigm for software engineering, or (2) as a conceptual tool for understanding human societies [47, p.7]. In the first case, multi-agent techniques are applied, because the

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/915

solution to some automation problem is inherently distributed. This is for example the case in manufacturing scheduling [38]. A centralized solution, as in operations research, would be possible, but would be less robust, and sometimes less effective or efficient. In the second case, multi-agent techniques are applied, because the *application problem* is inherently distributed, and a centralized solution is not feasible or legally possible. Consider for example a work-flow system for criminal investigations, in which parties with different legal rights (prosecution, defense, forensic expert), must or may not have access to the documents at certain stages of the proceedings.

In this paper, we provide a case study of the second type of application: the health care sector. We model a complex network organization in which autonomous entities (government agencies, care providers, patients) collaborate to achieve a global objective. Although we do not strictly use the terminology and techniques of the multi-agent community – we use theories from management and accounting – we do believe that this work is relevant to multi-agent systems community. First, because it provides a set of *control patterns*, practical guidelines for the analysis and design of the implementation of norms and regulations in a network organization. Such control patterns may also be of use in the design of normative multi-agent systems. Second, because it deploys the notion of *economic value* from business modeling to motivate the regulations. The use of economic value allows specification of the business model of a network organization, at a relatively high level of abstraction. But to understand these contributions, we first have to explain the case study.

1.1 A Social Chart for Dementia Care

In the Netherlands, the health care sector is making a transition from a supply-driven structure, in which health care providers (e.g. hospitals) decide what health care services are delivered, towards a more demand-driven structure, in which the patient can select health care services (e.g. treatment, physiotherapy, domestic care) from different providers. Such a 'market for health care' requires that patients know enough about the available services and providers to make an informed choice. Currently, patients generally perceive the health care on offer as fragmented, and not fitted to their needs [33]. Few information is available about the services offered by different care providers, and about their quality of those services.

This research is carried out as part of the FRUX project [14]. Among other things, the project develops a *dynamic interactive social chart for dementia care* (DEMDISC): an interactive web-site that will provide an overview of the health care services available in a region, and provide personalized advice about possible combinations of health care services, so called *service bundles*. The Social Chart is designed for the relatives and informal care givers of patients with Alzheimer's disease (dementia). One of the aims is to develop a generic method for generating bundles of services, tailored to the specific needs of a user [2].

1.2 Control Patterns and Value Modeling

Our role in the project is to model the governance and control aspects of a complex information service like the Social Chart. Before starting a software requirements engineering process, it is crucial to get an understanding of the business model that underlies the health care network. The health care sector is highly regulated. Moreover, the regulations that govern the health care system are subject to change. Because changes are the outcome of a political process with many stake holders, the way they develop can remain unclear. Moreover, when health care regulations change, the business opportunities for a system like the Social Chart may well change too. For these reasons it is important to develop a generic method for analyzing and developing governance and control aspects, and for discovering corresponding business opportunities. Important elements of such a method are

(i) a representation language and graphical modeling tool for the analysis and redesign of control procedures, based on accounting principles,
(ii) a set of general guidelines to assist in analyzing and redesigning control procedures,
(iii) a library of generic control mechanisms, that have been validated and can serve as 'best practices'.

Regarding (i), the e^3-*control* methodology provides a representation language and graphical modeling tool [21]. The e^3-*control* methodology is based on the e^3-*value* ontology, which analyzes network organizations in terms of the transfer of *economic value* between participants [21]. The resulting model is called a *value network*. In this paper we extend this work, and present a set of *control patterns*. A control pattern is a description of a generic and re-usable control mechanism that solves a control problem, to be selected on the basis of the context of application. A control problem is an identifiable risk for opportunistic action by one of the other network participants. So control patterns combine guidelines (ii), with best practices (iii). Control patterns are inspired by the design patterns approach, which was proposed in architecture [1] and is now very successful in software engineering. More recently, it has also been applied in the business domain, to administrative processes [36], organizational structures [11], and business process re-engineering [4].

The usefulness and adequacy of control patterns is validated on a case study of the value network for the Dutch public health insurance system AWBZ (Exceptional Medical Expenses Act). The governance and control aspects of this system are interesting, because it is funded by taxes, and lacks direct feedback on the quality of services. The system is undergoing changes, one of which is the introduction of a personal budget. This facilitates the development of a kind of market for care providers. Although an analysis approach based on economic value works well in commercial settings, one could question its suitability for the public sector. We are therefore especially interested in the applicability of control patterns in a highly regulated value network, which involves many public-private partnerships. See also [26].

The remainder of the paper is structured as follows. In Section 2 we present a definition, and our library of control patterns, explaining the underlying control theory. In Section 3 we apply the control patterns to the case study, reverse engineering the way in which the network may have developed.

2 Control Patterns

A sustainable network organization needs mechanisms to govern and control the interaction among network participants. In most cases, interaction is encoded in contractual arrangements, and implemented through procedures and regulations. But regulations can be violated. In the context of control theory, a network organization is therefore considered to be either in an *ideal situation*, in which no opportunistic behavior in the form of violations, errors, or fraud will occur, or in a *sub-ideal situation*, in which violations, errors, or fraud do occur. The terminology ideal – sub-ideal is taken from deontic logic [34]. The ideal situation is in some sense normative or obliged, or (alternatively) desired by the dominant actor in the network. Sub-ideal situations can be prevented, detected or corrected by administrative measures called *control mechanisms*. An example of a control mechanism is the three-way verification by the accounting department of a payment for a purchase, against the invoice and the inventory.

In the accounting literature, control mechanisms are typically analyzed from an operational or procedural perspective, with process models and flow charts [43][39]. In a network organization, however, the ideal situation is often determined by the business model of the network. Therefore, we need a form of business modeling, to analyze the reasons for implementing a control mechanism.

2.1 Business Modeling

There are several methodologies that address the design of business models for network organizations, like the Business Modelling Ontology [37], value webs [45], and e^3*-value* [21]. Of these, e^3*-value* is the only one with a formal semantics, and a specific focus on value transfers between enterprizes. The method is ontologically well-founded, and is supported by graphical modeling tools. We therefore apply the e^3*-value* ontology [21] for the description of so called ideal models, which express organizations that behave in compliance with the procedures and regulations. Sub-ideal models are expressed using e^3*-control*, a modification of the e^3*-value* ontology, used to describe opportunistic behavior of actors [21].

Ideal Value Models An e^3*-value* model provides a conceptual model of the value transfers in a business network, encoded in the e^3*-value* ontology [21]. The e^3*-value* constructs have 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. So only if you pay, can you obtain the goods and vice versa. A value interface consists of **value ports**, to represent that value objects are offered to and requested from the actor's

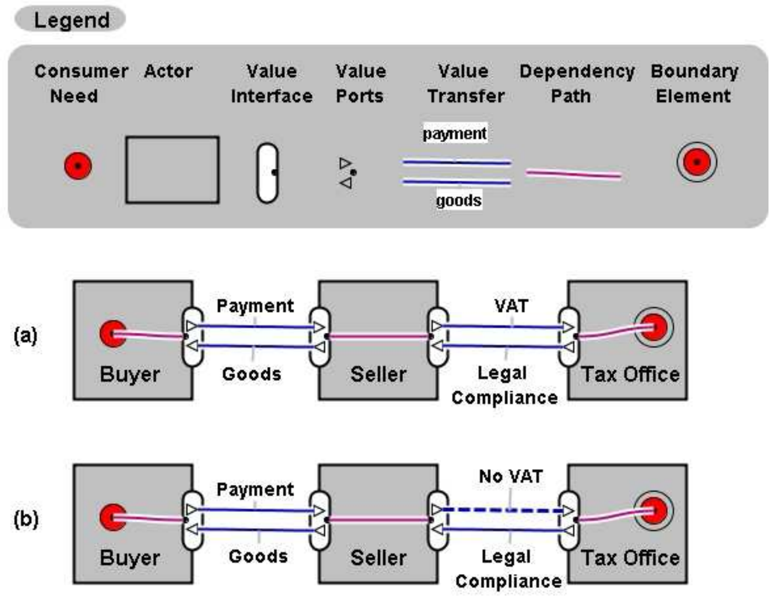


Fig. 1. Example of an e^3-*value* model of a purchase

environment. Actors may have a **consumer need**, which, following a path of **dependencies** will result in the transfer of value objects. Transfers are either dependent on other transfers, or lead to a **boundary element**. The e^3-*value* methodology also allows the designer to assign monetary values to value transfers in a spreadsheet and calculate profitability of actors in a network.

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 reality, actors may commit fraud or make unintentional errors, e.g. actors will not pay, or not obtain the right goods. In e^3-*control* such sub-ideal value transfers are graphically represented by dashed arrows [21]. For example, Figure 1(b) shows a sub-ideal situation in which the seller does not pay VAT.

Process Models Value models consider the transfer of value objects, like money, goods or services. This is essentially the transfer of ownership rights [25]. But such transfer requires operational activities to be performed, by multiple actors, which can only be shown using a process model. Moreover, in value models the temporal order in which objects are transferred is abstracted over: it only represents that objects are transferred, but not in which order. The order in which activities take place, forms a crucial part of many control mechanisms. So in addition to value models we need process models to capture control aspects. We represent process aspects of control problems and their solutions by UML-activity diagrams [40].

Information Models Evidence documents, like receipts or tickets, form an important part of the administrative control mechanisms studied in this paper. Nowadays, documents are often certified files in a distributed information system, of which the paper document is only a trace. For an analysis of the structure and content of documents, UML class diagrams are a suitable representation format. Other issues are related to the management of information. Which party should store the documents? How is privacy, accuracy and reliability of the data preserved? For such issues, see Ronmey and Steinbart [39, ch8]. In this paper, we focus only on the procedural role of documents in a control mechanism.

2.2 Theoretical Framework

The control patterns proposed later in this section, are based on a combination of agency theory, control theory and ideas about trust.

Agency Theory A well known theory in sociology and management accounting is *agency theory*, also called *principal-agent* theory. See Eisenhardt [16] for a survey. Agency theory studies the relationship between two parties: the *principal*, who delegates some activity, and the *agent*, to whom the activity is delegated. The theory argues that if (1) the principal and the agent are utility maximizers with bounded rationality and (2) there is information asymmetry in favor of the agent, the agent may behave opportunistically. Agency theory distinguishes two types of opportunistic behavior.

The first type is caused by *hidden information*: the principal can not be sure that the agent accurately presents his ability to do the work. For example, a producer (agent) generally has better information about the product he is producing, than someone who wants to buy the product (principal). The generally accepted control mechanism against hidden information is *screening* [28]: the principal collects information about the reliability of the agent, before agreeing on a transaction.

The second type is caused by *hidden action*: the principal can not be sure whether the agent did his work according to the contract or not. For example, the producer may use low quality components to produce a product. As a result, the quality of the product is lower than agreed in the contract. The generally accepted control mechanisms against hidden action are *monitoring* the agent, the and *creation of incentives* to motivate the agent not to behave opportunistically [16].

Control Theory Control problems are typically identified by an analysis of risk indicators and threats discovered in an audit process. A control mechanism prescribes how to organize business processes in order to prevent, detect or reduce the risks posed by a control problem. Internal control theory is concerned with administrative and organizational measures inside an organization [43]. But in inter-organizational settings, risks related to the behavior of partners in a network, are mostly dealt with by contractual arrangements. So it is difficult to apply internal control guidelines directly. We also used the more formal work of Chen [10] and Bons et al. [8][7] on *inter-organizational trade procedures*. Chen concentrates on *detective controls*, as they are applied in purchasing and procurement. Bons et al also deal with forms of *preventative control*.

The distinction between detective and preventative control can be made clear by the well-known example of the Paris metro [18]. In many countries, the norm that one should not board a train without a ticket, is implemented by random checks. This is a form of detective control; it assumes that there are enough incentives for a passenger to avoid violation, given that the chance of being caught and the (social) sanction are large enough. By contrast, the Paris metro has an elaborate system of automatic gates, which makes it physically impossible to board a train without a ticket. This is a form of preventative control. In practice, the two types of control are often combined.

Trust A network organization can be interpreted as a number of binary value transactions between actors. When parties in a network do not have an existing business relationship, lack of trust is likely. Trust has been defined as

> "The willingness of a party to be vulnerable to the actions of another party based on the expectation that the other party will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party"[32, p.712].

Without prior trust, the party who invests in a transaction, called the *trustor*, is uncertain whether the other party, the *trustee*, will perform its part of the deal or will defect and behave opportunistically. The uncertainty of the trustor about possible opportunistic behavior also comes out in Gambetta's definition:

> "Trust is the subjective probability by which an individual a expects that another individual b performs a given action on which its welfare depends" [19].

However, trust does not have to depend on the trustor and trustee alone [44]. Institutional control measures can be used to guarantee performance according to contract. Legal provisions against fraud are an example of a detective kind of control mechanism. An example of a preventative control mechanism for international trade, is an Escrow service [22]. In such a scenario, the money of the buyer is deposited with a trusted third party, called Escrow agent. Only when the Escrow agent confirms to the seller that the money has been deposited, the goods are shipped. This solves the risk of the seller, that the buyer will or cannot pay. Only when the buyer has confirmed to the Escrow agent, that the goods have been delivered as agreed, the money is released and transferred to the seller, with a handsome fee subtracted. This solves the risk of the buyer, that the seller will not deliver as agreed. So in general, the purpose of inter-organizational control mechanisms is to reduce the uncertainty of the trustor, and provide enough guarantees for both parties to engage in a transaction. If we model the reasoning of the trustor, we get a kind of game theoretic reasoning, comparing a scenario in which a control mechanism applies, with a scenario in which there are no institutional control measures [5]. This kind of reasoning, about the risks of engaging in a transaction, can be applied from the perspective of both the buyer and the seller.

It is not a coincidence that the effectiveness of an Escrow service has been shown by game theoretic means [22]. The same kind of reasoning – using recursive models of the expected behavior of agents according to other agents – can also be modeled by the 'qualitative' game theory developed by Boella and Van der Torre [6]. Although we do not explicitly use this kind of recursive modeling, it is implicit in the way control patterns are to be applied.

2.3 Conceptual Framework

The three components of the underlying theory, agency theory, control theory and trust, can be combined. Based on this mixed background, we identify a vocabulary of terms (in **bold**), to be used in the definition of control patterns.

Consider a transaction scenario, as shown in Figure 2. There are two agents, called **primary actor** and **counter actor**. From a value perspective, we say that the primary actor transfers a value object, called **primary object**, to the counter actor, and the counter actor returns a value object called **counter object**. From a process perspective, such a transaction is modeled by two operational activities: the primary actor performs a **primary activity**, and the counter actor executes a **counter activity**, each resulting in the corresponding value transfers[3]. Figure 2 shows a value model of such a transaction on the left, and the corresponding process model on the right. The order in which the primary activity and counter activity occur is not specified. This is indicated by the UML notation for parallel execution (thick horizontal bar).

As in agency theory, in this scenario the primary actor has delegated some activity to the counter actor. So we could say that **primary actor = principal**, and **counter actor = agent**. The control risks of the transaction are generally assessed from the point of view of the primary actor, who – by definition – does not trust the counter actor, and must therefore design control mechanisms against possible sub-ideal behavior of the counter actor. So we could also say that **primary actor = trustor**, and **counter actor = trustee**.

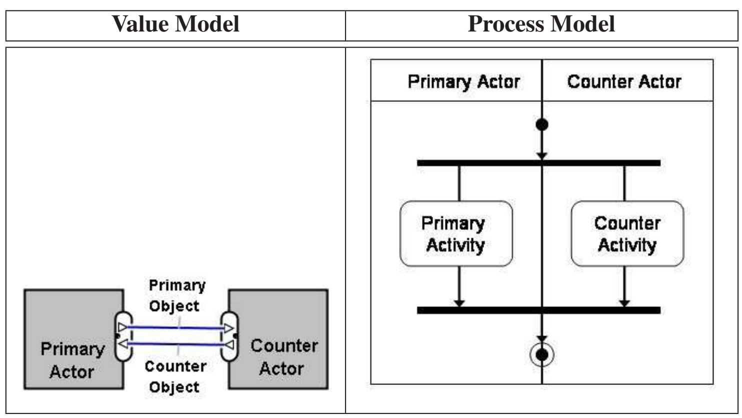


Fig. 2. Transaction Scenario, in which primary actor = principal = trustor, and counter actor = agent = trustee.

Regarding detective control, we need to make some more distinctions. Detective control [10] is generally concerned with *verification*: comparing the results of an op-

[3] The primary-counter terminology is based on Bons [7], and is inspired by contract law, which uses the phrases *primary obligation*, and *counter party*.

erational activity with some claim about its legitimacy, quality or quantity. The claim is represented by a *document to-be-verified*. One or more so called *supporting documents* represent evidence about the operational activity. The result of a verification is usually a decision to perform some action or not, or else an evidence document stating the decision. A template of a verification activity is shown in Figure 3. To simplify the diagrams, in the remainder of the paper, we will only show the positive outcome of a verification activity, since the negative outcome always leads to the end of the process. One way of obtaining evidence about an activity, is by a **witness** activity. When parties sign a contract, evidence of commitments is generated by **confirm**. For the transfer of a value object, we use activities **request** and **provide**. When the value object is a fee, we use the activity **pay**.

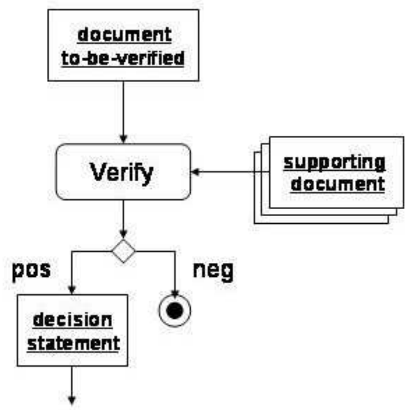


Fig. 3. Verification Activity

2.4 Pattern Definition

Control patterns are inspired by the use of design patterns in architecture [1] and software engineering [20]. The idea is to capture the 'best practices' about the design of buildings, software, and later also organizational structures [4][36], for different applications. Traditionally, a design pattern has four essential elements: pattern name, problem, solution, and consequences [20, p.3]. We have transferred the idea of using design patterns to the control domain. We separated the description of the context in which a pattern is to be applied, from the problem which motivates the selection of a pattern [11]. In our interpretation, the context describes the value network with the actors, their relationships, like trust, and the activities to be controlled. The problem specifies a risk to be detected or prevented by the solution of the pattern. The solution describes the value network and the corresponding process model, after implementing the control mechanism encoded in the pattern.

Definition 1 (control pattern). A *control pattern* is a description of a generic and reusable control mechanism for a recurring control problem, selected on the basis of the context. A control pattern consists of the following elements:

name: a descriptive name of the pattern, used to select patterns from a pattern library.
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, and if needed for understanding the context, also by a process model.
problem: a statement of risks for opportunistic behavior. A control problem exists if there is some deviation of the prescribed transfers of economic value. So we model the problem by a sub-ideal value model, using sub-ideal value transfers. Again, if needed, we also use a process model.
solution: description of a control mechanism, to detect, prevent or correct the control problem.

The solution is described by both process models and value models. Table 1 contains an overview of our library of reusable control patterns. They were elicited using the PattCaR method [42]. Based on literature and case studies, we identified potential patterns. These were modeled with e^3*-control* and activity diagrams, and compared using a commonality-variability analysis. The resulting patterns are validated in case studies, one of which is presented here.

In addition we present some *organizational design patterns* listed in Table 2. Compare for example [31][11]. We use delegation of an activity to some external actor. For this pattern, we assume that the actor to whom the activity is delegated, is trusted by the primary actor, so no additional control measures are required. Otherwise, the usual control patterns apply. Decomposition of activities is typically used for efficiency. It is also needed to separate an activity into two parts for control reasons, as for example when a payment is made in two installments, one before and one after delivery (down payment).

2.5 Library of Control Patterns

We will now present a library of reusable control patterns. We illustrate the patterns by a simplified transaction scenario, in which a buyer (primary actor) has ordered some goods, and does not trust the seller (counter actor) to deliver. So in this case the primary activity is payment; the counter activity is delivery of the goods.

The **Execution Monitoring** pattern requires the primary actor to verify execution of the counter activity, *before* executing the primary activity [10][7]. In previous versions of this research [27], the pattern was therefore called Pre-execution. It roughly corresponds to the monitoring practice of agency theory [16]. The associated control problem for the primary actor is that without the verification, the primary actor is not certain whether the counter actor will execute the counter activity as agreed. In our example, the buyer is not certain about delivery. So the buyer will only pay for the goods, after having inspected that the right goods were delivered. When both actors apply this pattern, we get into conflict, and an intermediate solution like a down-payment must be applied [27].

Name	**Control Problem**	**Solution**
Execution Monitoring	primary actor is not certain whether counter actor will execute the counter activity as agreed	verify counter activity, before executing primary activity
Execution Confirmation	counter actor may deny that primary activity was executed as agreed, and refuse to execute counter activity, or require compensation	require confirmation from counter actor that primary activity was executed as agreed
Commitment Confirmation	counter actor may deny to have made a commitment to primary actor	require confirmation of commitment from counter actor
Partner Screening	counter actor may not be a reliable partner to make commitments with before making any commitments	verify credentials of counter actor
Certification	counter actor may not be reliable to perform counter activities	require verification of past behavior of counter actor

Table 1. Library of Control Patterns

Name	Objective	Solution
Delegation	activity can not be effectively or efficiently done	delegate activity to a specialized trusted external actor
Decomposition	activity can not be effectively or efficiently done	decompose activity in parts, which can be effectively or efficiently carried out

Table 2. Organizational Design patterns

The **Execution Confirmation** pattern requires the counter actor to confirm that the primary activity was performed as agreed [7]. Think of a receipt. The associated control problem is that, in case of a dispute, the primary actor will not have independent evidence to prove that the primary activity was executed as agreed. In our example, the buyer will require a quittance from the seller, as evidence of payment.

The **Commitment Confirmation** requires the counter-actor to confirm, i.e. provide documentary evidence, of the transaction commitment [7][46]. Normally, this is done by signing a contract. The associated control problem is that otherwise the counter actor may refuse to recognize that he made a commitment to the primary actor, and may therefore not execute the counter activity. In our example, the buyer will require a price quote or offer, which commits the seller to deliver at a certain price.

The **Partner Screening** pattern require the primary actor to collect evidence about the past conduct of the counter actor, and verify whether the past record and reputation of the counter actor conform to standards of trustworthiness, before making any commitments [28]. The underlying control problem is the risk of making a commitment to an unreliable partner. Often the research and verification are delegated to a specialized party. In our example, the buyer can have an agency like the Chamber of Commerce check the credentials of the seller, before making any commitments.

The **Certification** pattern establishes the authorization by the primary actor that the counter actor meets the requirements to be allowed to perform a counter activity of that type. This kind of regulatory control is can be implemented by different kinds of evidence documents, like certificates, licenses or accreditations. Certification is needed when the primary actor is at least partly responsible for the counter activity. The primary actor can be a regulatory body, or an actor who has delegated the execution of the counter activity to another actor. For example, companies offering Escrow services must be licensed and accredited by the financial regulatory bodies of the country in which they are based, such as the central bank.

In the Appendix, the patterns are summarized using the graphical notation discussed above.

2.6 How to apply the Control Patterns?

Patterns can be applied using the three steps of the e^3-*control* methodology [24]:

1. Define ideal value models
2. Identify control problems; model them in sub-ideal value models and sub-ideal process models
3. Select a pattern to fit the control problem; (re-)design control mechanisms, using the pattern.

To select a pattern from the patterns library, a control problem in a case description must be matched with the control problem in one of the patterns. First sub-ideal value and process models of the case are developed. Then, based on the value model, one can identify the primary actor, counter actor and the sub-ideal value transfer. The primary actor is the one who may expect sub-ideal behavior from the counter actor. After a pattern that matches the problem description is found, the process model is adjusted according to the pattern.

3 Validation In Health care

In this section, we show how the control patterns can be applied to a health care network. In particular, we focus on the fact that the case exemplifies a highly regulated value network that involves many private-public partnerships. The case study is presented in two parts. The first part tries to explain, by means of the patterns, how the current governance and control mechanisms may have developed. In the second part, we study the control problems that arise from the introduction of a personal budget, and show how the Social Chart could provide a solution. We discuss three possible future scenarios for exploiting an information service like the Social Chart. The data for the case study was collected by semi-structured interviews with five experts from different health care organizations. The resulting e^3-*control* models were verified by the experts. In addition, we used publicly available policy documents, like [17], and government regulations.

3.1 Reverse Engineering the AWBZ

In the Netherlands, the AWBZ[4] deals with long-term and chronic diseases, such as protracted illness, invalidity, learning disability, mental disorders and geriatric diseases. Because this kind of care is too expensive to insure in a regular way, the system is arranged as public health care. A patient only pays a small part of the costs; the largest part of the costs is reimbursed to the care provider by a government fund, collected from taxes.

Suppose we have a hypothetical health care system (Figure 4). There are two parties: the patient and the care provider, for example a general practitioner. The patient receives care in return for fees. The corresponding process model is relatively simple: the patient pays only after receiving the care, according to the Execution Monitoring pattern, applied from the perspective of the patient (Table 3). So there is direct quality feedback. The task of paying the fees may be delegated to a trusted party, for example the family of the patient.

Name	Execution Monitoring
Context	Patient (primary actor) receives care (counter object) from care provider (counter actor) in return for fees (primary object).
Problem	Patient is not certain if the care provider will provide care, as agreed.
Solution	Patient must verify that care is provided, before paying the fees.

Table 3. Applying Execution Monitoring

Note that because of differences in expertise and status, the patient, or the family of the patient, are often not in a position to verify the quality of the care. Patients tend to trust care providers. Generally, care providers do not only provide care to get paid, but also to help patients. So, applying accounting models and professional distrust seems

[4] Algemene Wet Bijzondere Ziektekosten (Exceptional Medical Expenses Act)

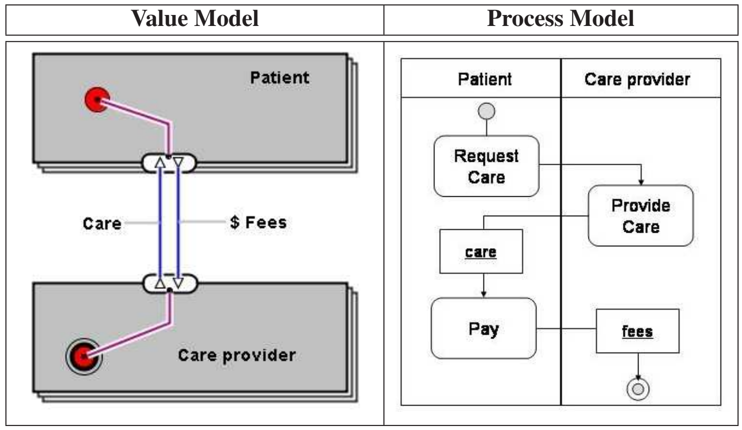


Fig. 4. Hypothetical Care Transaction

inappropriate for many care situations. Nevertheless, it may help to understand the governance and control of the health care system. This caveat applies throughout the rest of the case.

In this hypothetical situation, there is a problem. Not all citizens fall ill, but those who do, are faced with very high costs. For long-term and chronic diseases, these costs can not even be carried by individual health insurance policies. Some kind of solidarity is needed between healthy citizens and chronic patients, managed by the government [15]. Such an exceptional health care system is shown in Figure 5. The solidarity is shown by the fact that value transfers may be summed over a 'stack' of actors, for a specific time period.

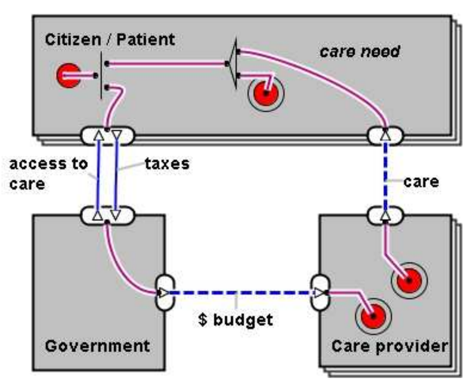


Fig. 5. Solidarity

There are essentially two value transfers. The first one, on the left, is concerned with access to health care. Citizens pay taxes, in return for the government funding care providers. When citizens do fall ill and become patients, depicted on the right, they can use the infrastructure. These possibilities are linked by a choice-fork (triangle). The model is sustainable, as long as the income from the taxes of citizens is sufficient to cover the average costs of providing care services ($\Sigma\ taxes \geq \Sigma\ budget$).

This is not a valid e^3*-value* model. First, the value transfer 'care', is a single value transfer. The reciprocal value transfer is made indirectly, through taxes. For this reason, there is no direct feedback about the quality of the care provided, to the actor who decides about funding. Such indirect dependencies are typical for the public sector. Second, there is no reciprocal evidence, that the care requested by a patient, is actually needed. In a regular insurance system, this would correspond to the verification that a claim is eligible according to the insurance policy and conditions. Third, for the value transfer 'budget' no reciprocal evidence is required to ensure accountability of the care provider. Moreover, in this model, funding of the care provider is not linked to the number of care services offered. Typically, a yearly budget is set.

We first focus on the control problems of the care provider.

Control Problem 1a. A care provider receives a budget, without being held accountable for the money. Control theory dictates, that every activity should be verified, with evidence of its execution [10][43]. Here, that means that the two dependency paths should be linked. On the basis of the control problem, that the care provider may receive a budget (primary object), that stands in no relation to the care actually provided (counter object), we therefore select the Execution Monitoring pattern (Table 4).

Name	Execution Monitoring
Context	Government (primary actor) pays budget (primary object), so that the care provider (counter actor) may provide care to patients (counter object).
Problem	Government is not certain if the budget is in proportion to the care provided to patients.
Solution	Government must verify what care is provided, before paying the budget. The control activities 'witness' and 'verify' are added, as well as 'supporting document', namely the evidence.

Table 4. Applying Execution Monitoring to Problem 1a

Control Problem 1b. All the costs of the care provider are reimbursed by the government. Therefore, there is no incentive for care providers to try and work more efficiently. Such an 'open-ended' system is one of the general reasons behind the increase in health care spending. Currently, there is much interest in budgeting schemes to reduce this problem, for example by output budgeting. We interpret budget agreements between government and care provider as a kind of mutual commitment, and select the Commitment Confirmation pattern (Table 5). Before committing to a certain budget, the government needs a commitment from the care provider that they can and will provide care for such a budget. In some cases the budget is calculated by standardized Care Intensity Packages (ZZP).

Name	Commitment Confirmation
Context	Government (primary actor) pays budget (primary object), so that the care provider (counter actor) may provide care to patients (counter object).
Problem	Care provider may claim to have no commitment to provide care for a given budget, and hence refuse to continue to provide care.
Solution	Government only commits to a specific budget, if care provider makes a commitment to provide the agreed care for that budget. Efficiency gains can be reinvested by the care provider.

Table 5. Applying Commitment Confirmation to Problem 1b

The result of applying these two patterns is shown in Figure 6 (value network) and Figure 7 (process model). The task of controlling the budgets of care providers, and verifying the evidence, is delegated to an independent local agency, called Administration Office. For its administration task, the Administration Office receives a yearly budget from the government. This budget is fixed; it does not depend on the amount of care delivered by the care providers. By contrast, the budget of the care provider (1) depends on the number of care services actually delivered (n).

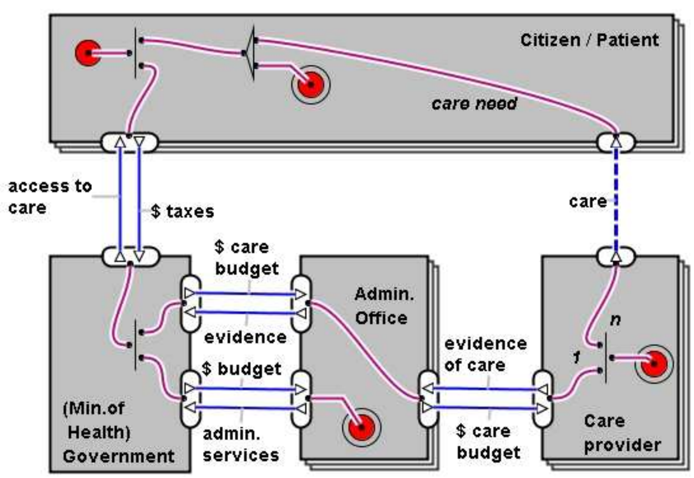


Fig. 6. Output Budgeting and evidence (value model)

Control Problem 2a. Now we concentrate on the patient. Although every citizen is entitled to health care when needed, access should be restricted to patients whose care is

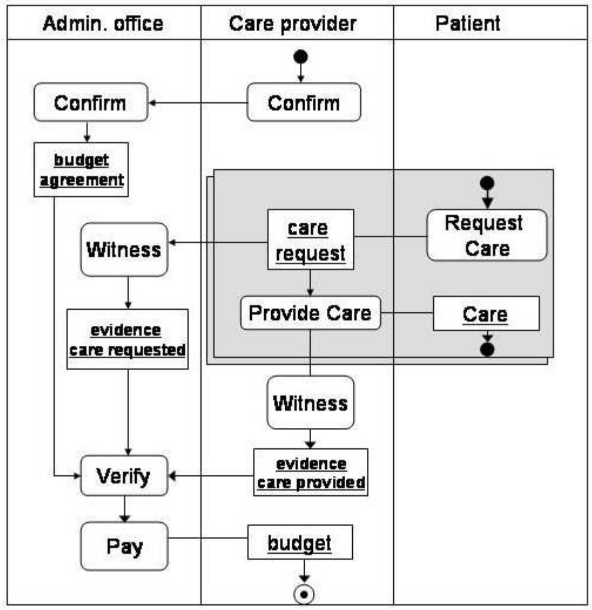


Fig. 7. Output Budgeting and evidence (process model)

(medically) necessary. Again we frame this problem as an instance of execution monitoring. The government or some government agency, must verify on the basis of evidence, like medical tests or a diagnosis from a physician, to which kinds of exceptional care the patient is entitled to.

Control Problem 2b. Initially, before March 2003, needs assessment was combined with the task of allocating actual care. Both were carried out by Regional Indication Centers (RIO). However, in some cases the needs assessments turned out to be inadequate: patients from different regions were given different products for the same diagnosis. These problems can be addressed by two applications of execution monitoring, combined with the general accounting principle of *standardization* [43]. In the AWBZ, care is categorized in standardized *functions*: domestic care, personal care, nursing, supportive assistance, activating assistance, treatment and institutional care. For each function, there are different *classes*, which specify the intensity of the treatment. Which care needs correspond to which functions and classes, is laid down in the Care Entitlement Regulation[5] .

In addition, a patient needs to pay a personal contribution. The solution, which more or less represents the current situation, is shown in Figure 8. Now there are two verification steps (Table 6): In the first step, Evidence of Care Needs (Supporting Document) is compared with the Care Entitlement Regulation (Document-to-be-verified). The result is modeled as a Right for Functions. This assessment is performed by a national

[5] Besluit Zorgaanspraken AWBZ

agency, called Central Indication Center (CIZ)[6]. In the second step, the local Administration Offices translates the Right for Functions into an actual Right for Care. Because the administration office has an overview of the available care providers in a region, this actor is in a position to advice patients on where to get the best care they need.

Name	Execution Monitoring
Context	Government (primary actor) entitles all citizens (counter actor) to exceptional health care (primary object).
Problem	2a. Government is not certain if the entitlement for care corresponds to actual care needs of a patient, as detailed in the Care Entitlement Regulation 2b. Needs assessment must be uniform, but the available care supply depends on the region.
Solution	Step 1. based on evidence of care need, CIZ makes an assessment of the care needs of a patient, and issues an evidence document: Right for Functions. Step 2. The Administration Office translates the Right for Functions into Right for Care, allocating care services and care providers.

Table 6. Applying Execution Monitoring to Problem 2a and 2b

Control problem 3. Until now, no actor in the network controls the quality of care products. As a result, provisioning of care of low quality could remain undetected. In the previous step, administrative evidence is used for budget control of care providers, but no evidence of the quality of care is used.

However, a basic form of quality control does exist. The government is in a position to select care providers. Before being allowed to enter the network in the first place, the ability of care providers to provide the care functions for which they are known, must be assessed. This accreditation is the result of applying the pattern Certification. Accreditation is delegated to the Health care Insurance Board (CVZ)[7]. So in Figure 8, the Administration Office can only assign patients to care providers who have an accreditation from the CVZ. The CVZ cannot provide real quality control, but it based on administrative evidence, it can at least ensure that the care provider has adequate facilities.

3.2 Personal Budget and the Social Chart

The exceptional health are system described above has a number of problems. The right for health are, through needs assessment, is disconnected from the care that is actually available. For each care provider, the budget from the government has a limit. Care providers do not have an incentive to provide services above their budget. Moreover, this supply-driven system results in a fragmented and unbalanced care supply in some regions. Therefore the government is moving towards a demand-driven system.

[6] Centrum Indicatiestelling Zorg (CIZ)
[7] College van zorgverzekeringen (CVZ).

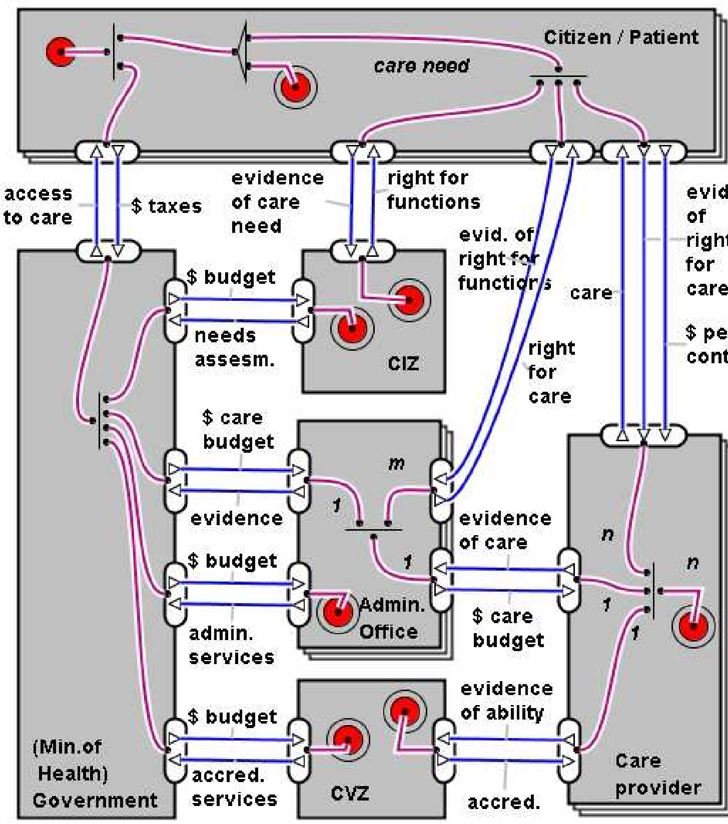


Fig. 8. Current Exceptional Health care Network

Name	Certification
Context	Government (primary actor) provides budget (primary object) through Administration office, to care providers (counter actor), who in return provide care (counter activity) for patients.
Problem	Government is not certain if the quality of the care delivered corresponds to certain basic quality standards and requirements.
Solution	The basic abilities of the care provider are verified, The result is an accreditation.

Table 7. Applying Certification to Problem 3

To try and solve these problems, the government introduced the possibility for patients to buy their own care services with a *Personal Budget*[8]. The personal budget is allocated by the Administration Office on the basis of the Right for Functions. The Personal Budget can cover care from any of the following six functions: domestic care, personal care, nursing, supportive assistance, activating assistance, and short term institutional care. The budget does not cover medical treatment, permanent institutional care or medication. Furthermore, the rules for care providers are liberalized. A care provider may now be any institution or private person. This liberalization has led to an enormous growth in the number of care providers, which creates more choice for patients, and in some cases allows them to regain control over their lives. We call these new care providers alternative care providers. This situation is shown in Figure 9.

From a control perspective, we can observe the quality control problem again. Because of the large number of alternative care providers, there is no way that the CVZ can accredit all of them. Therefore, alternative providers are generally not required to have an accreditation from the CVZ.

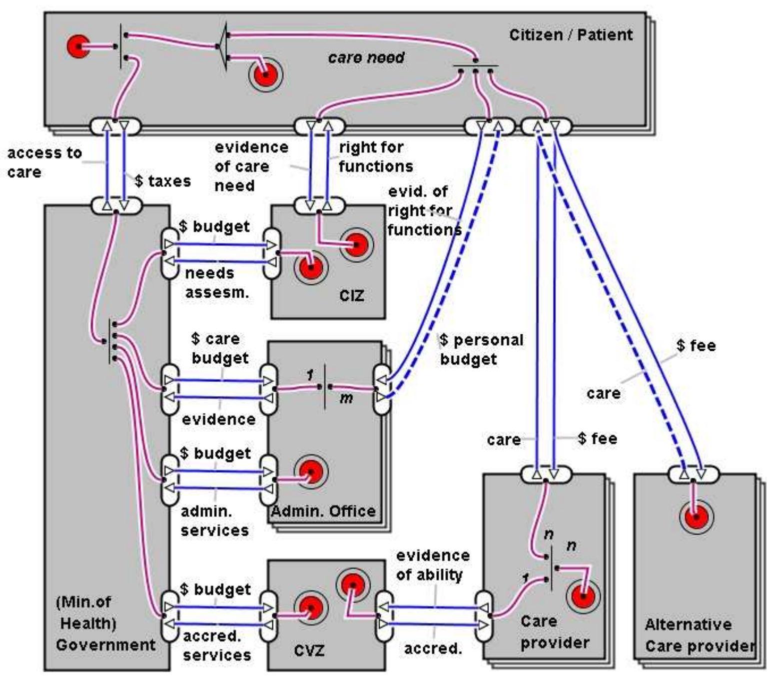


Fig. 9. Personal Budget

8 Persoonsgebonden Budget (PGB)

Control problem 4. Patients and relatives are not adequately informed about the available care, and care providers in a region. Only information about accredited care providers is available from the Administration Office. Patients therefore tend to select traditional care providers, rather than alternative care providers. This may stifle the development of the market for alternative care providers. This observation corresponds to the general idea that *information asymmetry*, a situation in which the customer has less information about a product than the provider, has a negative effect on the emergence of new markets [3]. In Figure 9, this control problem is represented by a sub-ideal exchange (dashed line), labeled 'care'.

Control problem 5. There is a risk that low quality care is delivered by alternative care providers (dashed line for 'care'). We can also see the problem from the government's point of view. If patients select care providers that provide inadequate quality, public money is wasted. This is indicated by a dashed line for 'Personal Budget'.

In an e-commerce setting, the first problem would typically be solved by an information broker, who matches supply and demand. The second problem would be solved by an agency verifying reliability. Since these activities require special expertise, it makes sense to delegate them to a separate agent. To solve these problems, we must select a pattern from the pattern library. We dismissed the possibility of a regulatory body assessing quality, because of the large number of alternative care providers. So the Certification pattern can not be used. Control problem 4 (lack of information) seems to be related to the Partner Screening pattern. However, as it stands, the Partner Screening pattern does not deal with the general information needed to collect a set of feasible providers (see definition in Section 2.5). It only deals with the second half: reliability assessment. Now we could adapt the patterns to accommodate this problem, but in this validation test, we have chosen to keep the patterns as they are, and conclude that they do not completely cover all aspects of the case study.

Although not generated by a pattern, we do have a possible solution. The information problem can be solved by providing an information service, such as the Social Chart introduced at the beginning of this paper. In Figure 10 this solution is represented.

Regarding problem 5, quality control is a general concern in Dutch health care. Since 2004, an independent Health Inspectorate (not in the model) must supervise the quality of institutional care providers. But this organization cannot feasibly control the large number of alternative care providers. We therefore propose that an initiative like the Social Chart should enable a kind of informal quality control. It could provide, for example, a web-forum with testimonials, an online community peer review, a reputation mechanism, or collaborative filtering techniques [41]. In this manner, knowledge about the quality of care providers can be shared throughout the community of patients and relatives. Community-based quality control only works when users contribute to the community. That is why in the scenario shown in Figure 10, the Social Chart receives Quality Assessment from (some) patients.

3.3 Exploitation of the Social Chart

Figure 10 presents only one of many possible exploitation scenarios. The Social Chart could be set up for example by the patients' association, by commercial parties like an

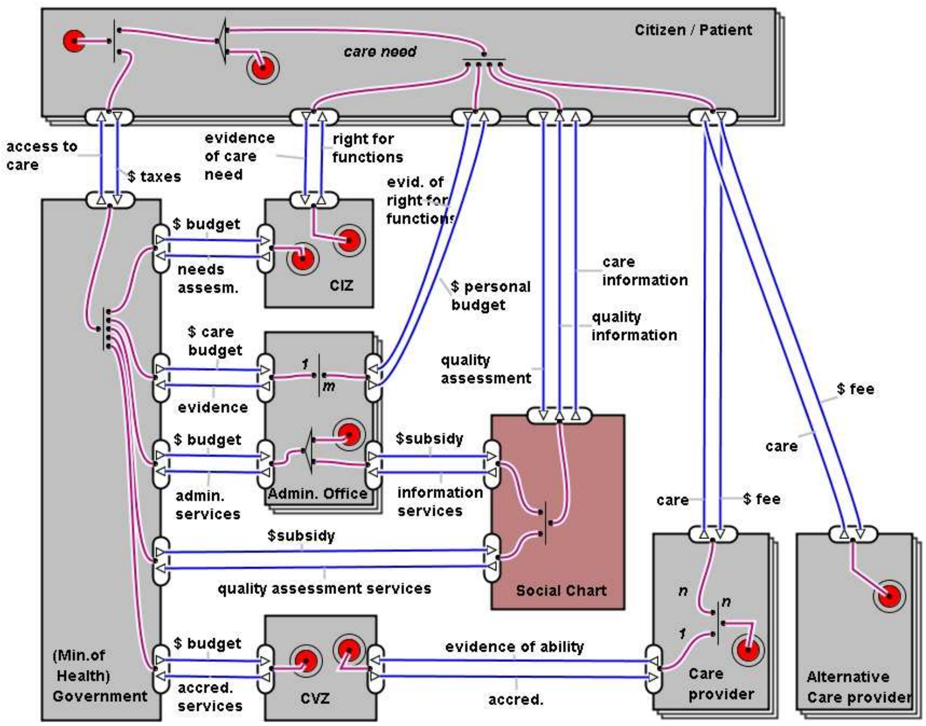


Fig. 10. Possible Scenario for Social Chart

insurance company or information broker, or by government. There are some indicators that virtual communities of patients are able to provide a form of community-based quality control [12]. There are already many successful virtual communities for patients with a chronic disease. For example, Leimeister et al [30] discuss the characteristics of a forum for cancer patients. Regarding dementia care, we find a relatively successful virtual community, hosted by patients association 'Alzheimer Nederland'. In the framework of the National Dementia Programme, Alzheimer Nederland collaborates with local government to improve dementia care [33]. So a combination of the patients' organization and local government seems a viable option for setting up and exploiting a kind of Social Chart.

Getting reliable online feedback is difficult [41]. Personal testimonials tend to be biased. Luckily, also quantitative approaches exist. For example, a Dutch information broker, independer.nl, is using a panel of general practitioners to get statistically valid feedback on the quality of hospitals.

3.4 Discussion

The objective of the case study is to validate the use of the control patterns. In particular, we are interested in modeling a highly regulated setting, with public-private partnerships.

Regarding the use of Control patterns Control patterns have a 'constructive' element, but reconstruction can also be used for analytic purposes. In this case study, we applied the control patterns to a hypothetical health care system based on solidarity. We have re-engineered the administrative control mechanisms in the AWBZ. We have shown that crucial aspects of the AWBZ controls can be motivated by the application of control patterns. In particular, the evidence documents (needs assessment, evidence of ability to provide care) are generated by application of the Execution Monitoring pattern. Quality control is established by applying the Partner Screening pattern, or by applying the Certification pattern when executed by a regulatory body. Budget control arrangements can be seen as a kind of transaction, with mutual commitments laid down in a contract. In addition, the delegation of tasks to separate agencies makes a large difference. So although delegation is not a control pattern, it does play a role as an organizational pattern. However, providing information about care providers and the care supply, the first function of the Social Chart, could not be established as the result of a control pattern.

In addition, the analysis demonstrates that there is a quality 'control gap' for alternative care providers. We have indicated a particular solution to fill this control gap: an interactive Social Chart, which could provide information about care supply and care providers in region, and possibly provide community-based quality control. The use of quality feedback to make a decision can be interpreted as application of the Partner Screening pattern.

We have only highlighted some aspects, ignoring others. In particular, we have had little to say about the information aspects of the administrative procedures. For this case study, the control patterns are too much focused on processes and administrative control, and too little on information and evidence.

Regarding highly regulated environments The use of a value-based modeling technique in a highly regulated setting raises some issues for discussion. For more on these issues we refer to [26].

Indirect reciprocity In this case study, the economic reciprocal relation is often indirect. This is modeled in e^3*-value* by a broken dependency path (thin line). People pay only for access to a service (triangle). In any insurance, the insurance premiums (here from taxes) should cover the claims from patients.

Regulatory rights Public-private partnerships are heavily regulated. Regulation can take the form of a system of legal rights, to restrict access to a service. Examples in the case study are Right for Functions and Right for Care. Legal rights can be seen as a value object in the e^3*-value* ontology.

Evidence documents This case shows the need for various evidence documents. Although collection and interpretation of evidence is usually modeled as part of the regular business process, here evidence documents are seen as value objects.

Community-based reciprocity A community-based quality control, like a recommender system, only works when members contribute Sharing and exchanging information, like experiences about care providers, can be based on solidarity in a community.

Control services A control service like needs assessment can be seen as a separate service, which can be delegated. This is highlighted in the model. All parties, including government agencies, need to be funded

Regarding Multi-agent Systems Multi-agent systems are used when either the solution to an automation problem, or the problem itself is inherently distributed. In this paper we discuss a case study of the second kind. Although the AWBZ health care system used to be centralized, with a large influence for the government (Ministry of Health), the case study illustrates the development towards a more distributed structure. Semi-independent government agencies like the CIZ or the Administration Office, have taken over government tasks regarding access control the the AWBZ. In particular the introduction of the Personal Budget and the acceptance of alternative care providers show that the health care network is becoming more and more ‘open’. For alternative care providers accreditation before being allowed to enter the network is no longer compulsory. On the other hand, quality control and information provision have become more important, now that access is no longer restricted to traditional care providers.

4 Conclusions

The health care sector is subject to a constant revision. In general, it is much harder to set up and maintain an information service, when the context is subject to change. When regulations change, the business opportunities may well change too. Therefore a generic method for analyzing and developing governance and control mechanisms for network organizations is needed. Control patterns provide such a method.

A *control pattern* is a description of a generic and re-usable control mechanism for a recurring control problem, to be selected on the basis of the context of application. Like design patterns, control patterns capture 'best practices' in a domain. Based on accounting literature and various case studies, we have developed a representation language for expressing control patterns, and a library of generic control patterns.

In this paper we have validated the control patterns, on a case study in health care. We have reconstructed the development of the governance and control mechanisms of the AWBZ system for provision of exceptional care. The case study shows that crucial aspects of the administrative controls can be motivated by the control patterns. In particular, evidence documents, like the needs assessment are generated by the Execution Monitoring pattern. Quality control can be established by partner screening or certification. Budget control arrangements can be seen as the application of the Commitment Confirmation pattern, just like in business contexts. So the control patterns have proved to be useful and adequate in analyzing this case study.

However, there are also some limitations. The patterns focus on process aspects and administrative controls. Much less attention is paid to information and evidence collection. This is unfortunate for this case study, because information provision is one of the main functions of the Social Chart. Management information issues have not been studied in this paper.

Acknowledgments We would like to thank the members of the FrUX project for their valuable input, in particular Rose-Marie Dros and Franka Meiland of the Vrije Universiteit Medical Center (VUmc).

References

1. Alexander, C. *The Timeless Way of Building*. Oxford, Oxford University Press, 1979.
2. Baida, Z. *Software-aided Service Bundling: Intelligent Methods and Tools for Graphical Service Modeling*. PhD thesis, Vrije Universiteit, Amsterdam, 2006.
3. Bakos, Y. The emerging role of electronic marketplaces on the internet. *Communications of the ACM*, 41(8):35–42, 1998.
4. Beedle, M. Pattern based reengineering. *Object Magazine*, January, 1997.
5. Boella, G., Hulstijn, J., Tan, Y.-H., and van der Torre, L. Transaction trust in normative multiagent systems. In Falcone, R. and Barber, S., editors, *Proceedings of the AAMAS workshop on Trust in Agent Societies (Trust'05), Utrecht.*, 2005.
6. Boella, G. and van der Torre, L. A game theoretic approach to contracts in multiagent systems. *IEEE Transactions on Systems, Man and CyBernetics - Part C*, 36(1):68–79, 2006. Special issue on Game-theoretic Analysis and Stochastic Simulation of Negotiation Agents.
7. Bons, R. W., Dignum, F., Lee, R. M., and Tan, Y.-H. A formal analysis of auditing principles for electronic trade procedures. *International Journal of Electronic Commerce*, 5(1):57–82, 2000.
8. Bons, R. W. H. *Designing Trustworthy Trade Procedures for open Electronic Commerce*. PhD thesis, University of Rotterdam, 1997.
9. Brckert, H.-J., Fischer, K., and Vierke, G. Holonic transport scheduling with TeleTruck. *Applied Artificial Intelligence*, 14(7):697725, 2000.
10. Chen, K. *Schematic Evaluation of Internal Accounting Control Systems*. PhD thesis, University of Texas at Austin, 1992. revised version available as Chen, K. and Lee, R.M. (1992), EURIDIS Research Monograph RM-1992-08-1.
11. Coplien, J. O. and Harrison, N. *Organizational Patterns of Agile Software Development*. Prentice Hall, 2004.

12. Dannecker, A. and Lechner, U. Success factors of communities of patients. In Ljungberg, J. and Andersson, M., editors, *Proceedings of the 14th European Conference on Information Systems (ECIS 2006)*, page 12 pages on CDROM. Göteborg University, Sweden, 2006.
13. Dastani, M., Hulstijn, J., Dignum, and F. Meyer, J.-J. C. Issues in multiagent system development. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, pages 922 – 929. IEEE Computer Society, Washington, DC, 2004.
14. Dröes, R., Meiland, F., Doruff, C., Varodi, I., Akkermans, H., Baida, Z., Faber, E., Haaker, T., Moelaert, F., Kartseva, V., and Tan, Y.-H. A dynamic interactive social chart in dementia care. In Bos, L., Laxminarayan, S., and Marsh, A., editors, *Medical and Care Compunetics 2*, volume 114 of *Studies in Health Technology and Informatics*. IOS Press, 2005.
15. Drummond, M. *Methods for the economic evaluation of health care programmes*. Oxford University Press, 2005.
16. Eisenhardt, K. M. Agency theory: An assessment and review. *Academy of Management Review*, 14(1):57–74, 1989.
17. Exter, A., Hermans, H., and andR. Busse, M. D. Healthcare systems in transition: Netherlands. Technical report, WHO Regional Office for Europe, Copenhagen,, 2004.
18. Firozabadi, B. S. and van der Torre, L. Towards an analysis of control systems. In Prade, H., editor, *Proceedings of the Thirteenth European Conference on Artificial Intelligence (ECAI'98)*, pages 317–318, 1998.
19. Gambetta, D. *Trust*, chapter Can we trust trust?, pages 213–237. Basil Blackwell, New York, 1988.
20. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, Boston, 1995.
21. Gordijn, J. and Akkermans, J. Value-based requirements engineering: Exploring innovative e-commerce ideas. *Requirements Engineering*, 8(2):114–134, 2003.
22. Hu, X., Lin, Z., Whinston, A., and Zhang, H. Hope or hype: On the viability of escrow services as trusted third parties in online auction environments. *Information Systems Research*, 15(3):236–249, 2004.
23. Jones, A. and Carmo, J. Deontic logic and contrary-to-duties. In Gabbay, D., editor, *Handbook of Philosophical Logic*, pages 203–279. Kluwer, 2002.
24. Kartseva, V., Gordijn, J., and Tan, Y.-H. Towards a modelling tool for designing control mechanisms in network organisations. *International Journal of Electronic Commerce*, 10(2):57–84, 2005.
25. Kartseva, V., Gordijn, J., and Tan, Y.-H. Inter-organisational controls as value objects in network organisations. In *Proceedings of the 18th Conference on Advanced Information systems Engineering (CAiSE 2006), Luxembourg*, 2006.
26. Kartseva, V., Hulstijn, J., Gordijn, J., and Tan, Y.-H. Modelling value-based inter-organizational controls in healthcare regulations. In *Proceedings of the 6th IFIP conference on e-Commerce, e-Business, and e-Government, Turku, Finland, (I3E'06)*, 2006.
27. Kartseva, V., Hulstijn, J., Gordijn, J., and Tan, Y.-H. Towards value-based design patterns for inter-organizational control. In *Proceedings of the 19th Bled Conference: eValues (Bled'06)*, page CDROM, 2006.
28. Keil, P. Principal agent theory and its application to analyze outsourcing of software development. In *Proceedings of the International Workshop on Economics-Driven Software Engineering Research (EDSER'2005)*, pages 1 – 5. IEEE Computer Society, 2005.
29. Kolp, M., Giorgini, P., and Mylopoulos, J. Multi-agent architectures as organizational structures. *Journal of Autonomous Agents and Multi-Agent Systems*, 13(1):3–25, 2006.
30. Leimeister, J. M., Daum, M., and Krcmar., H. Towards mobile communities for cancer patients: the case of krebsgemeinschaft.de. *International Journal on Web Based Communities*, 1(1):58–70, 2004.

31. Malone, T. and Crowston, K. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1), 1994.
32. Mayer, R., Davis, J., and Schoorman, F. An integrative model of organizational trust. *Academy of Management Review*, 20(3):709–734, 1995.
33. Meerveld, J., Schumacher, J., Krijger, E., Bal, R., and Nies, H. Werkboek landelijk dementieprogramma. Technical report, Nederlands Instituut voor Zorg en Welzijn (NIZW), 2004.
34. Meyer, J.-J. and Wieringa, R. *Deontic Logic in Computer Science: Normative System Specification*. John Wiley & Sons, 1993.
35. Moss, S., Gaylard, H., wallis, S., and Edmonds, B. SDML: A multi-agent language for organizational modelling. *Computational and mathematical Organization Theory*, 4(1):43–70, 1998.
36. Motschnig-Pitrik, R., Randa, P., and Vinek, G. Specifying and analysing static and dynamic patterns of administrative processes. In *Proceedings of the 10th European Conference on Information Systems (ECIS 2002), Gdansk, Poland*, 2002.
37. Osterwalder, A. *The Business Model Ontology: A Proposition in a Design Science Approach*. PhD thesis, University of Lausanne, Lausanne, Switzerland, 2004.
38. Parunak, H. V. D. A practitioners' review of industrial agent applications. *Autonomous Agents and Multi-Agent Systems*, 3(4):389–407, 2000.
39. Ronmey, M. and Steinbart, P. *Accounting Information Systems*. Prentice Hall, New Jersey, 10th edition, 2006.
40. Rumbaugh, J., Jacobson, I., and Booch, G. *The Unified Modelling Language Reference Manual*. Addison Wesley Longman, Reading, MA,, 1999.
41. Schubert, P. and Ginsburg, M. Virtual communities of transaction: Personalization in electronic commerce. *Electronic Markets*, 10(1):45–55, 2000.
42. Seruca, I. and Loucopoulos, P. Towards a systematic approach to the capture of patterns within a business domain. *Journal of Systems and Software*, 67:1–18, 2003.
43. Starreveld, R., de Mare, B., and Joels, E. *Bestuurlijke Informatieverzorging (in Dutch)*, volume 1. Samsom, Alphen aan den Rijn, 4th edition, 1994.
44. Tan, Y.-H. and Thoen, W. Formal aspects of a generic model of trust for electronic commerce. *Decision Support Systems*, 33(3):233 – 246, 2002.
45. Tapscott, D., Lowy, A., and Ticoll, D. *Harnessing the Power of Business Webs*. Harvard Business School Press, Boston, MA, 2000.
46. Weigand, H. and de Moor, A. Workflow analysis with communication norms. *Data and Knowledge Engineering*, 47(3):349–369, 2003.
47. Wooldridge, M. *An Introduction to Multiagent Systems*. John Wiley and Sons, Chichester, 2001.
48. Zambonelli, F., Jennings, N. R., and Wooldridge, M. Organisational rules as an abstraction for the analysis and design of multi-agent systems. *International Journal of Software Engineering and Knowledge Engineering*, 11(3):303–328, 2001.

Appendix: Control Pattern Library

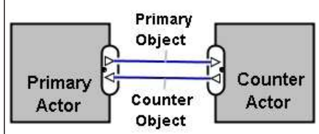
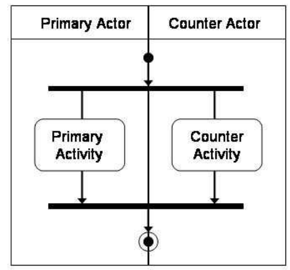
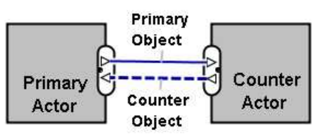
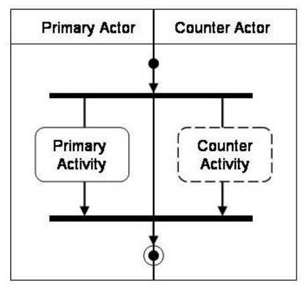
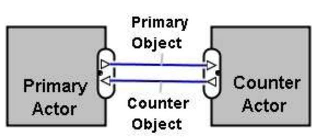
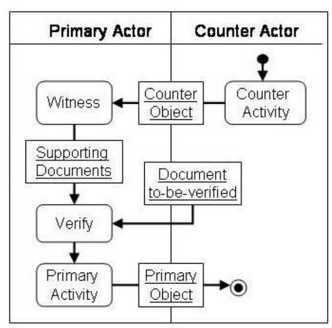
<table>
<tr><th colspan="3">Execution Monitoring</th></tr>
<tr><td>Context</td><td>
</td><td>
</td></tr>
<tr><td>Problem</td><td>
</td><td>
</td></tr>
<tr><td>Solution</td><td>
</td><td>
</td></tr>
</table>

Table 8. Execution Monitoring

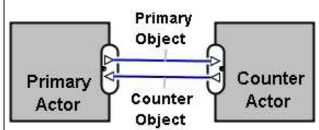
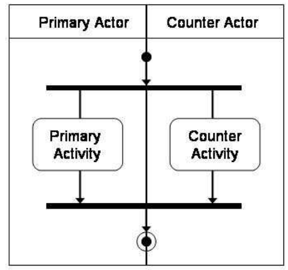
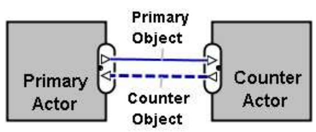
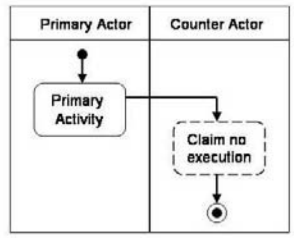
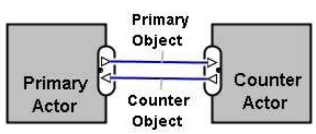
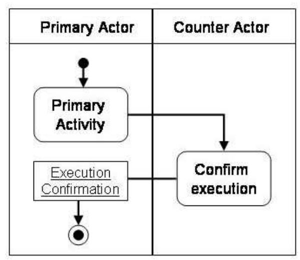
<table>
<tr><th colspan="3">Execution Confirmation</th></tr>
<tr><td>Context</td><td>
</td><td>
</td></tr>
<tr><td>Problem</td><td>
</td><td>
</td></tr>
<tr><td>Solution</td><td>
</td><td>
</td></tr>
</table>

Table 9. Execution Confirmation

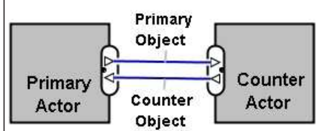
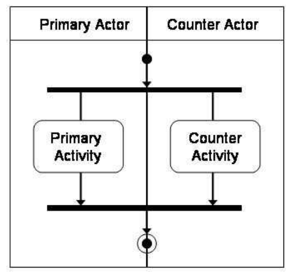
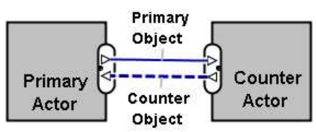
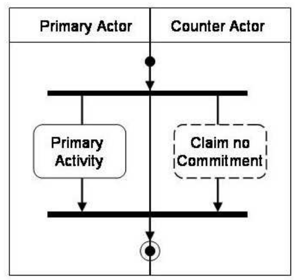
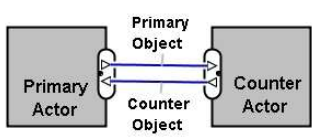
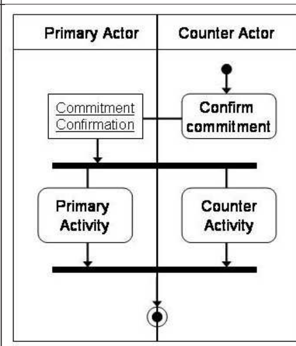
<table>
<tr><th colspan="3">Commitment Confirmation</th></tr>
<tr><td>Context</td><td>
</td><td>
</td></tr>
<tr><td>Problem</td><td>
</td><td>
</td></tr>
<tr><td>Solution</td><td>
</td><td>
</td></tr>
</table>

Table 10. Commitment Confirmation

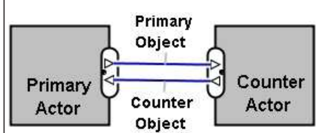
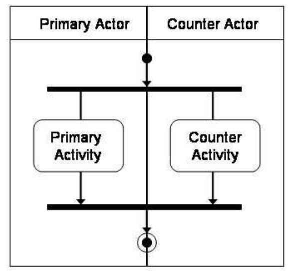
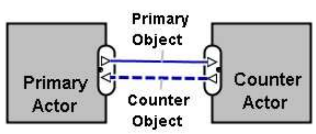
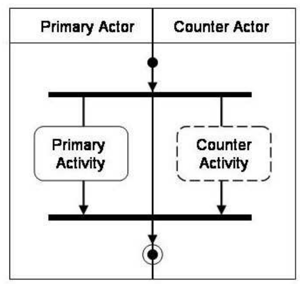
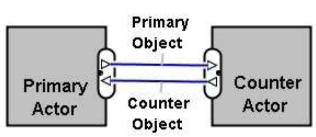
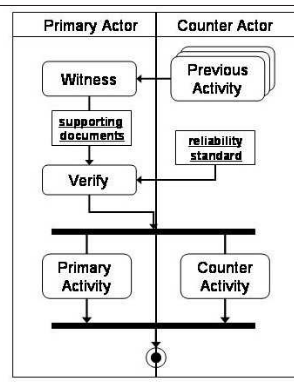
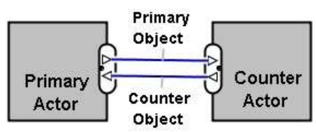
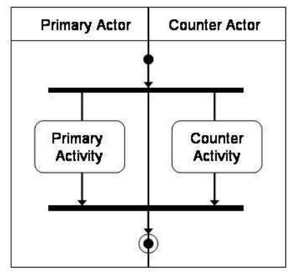
Partner Screening		
Context		
Problem		
Solution		

Table 11. Partner Screening

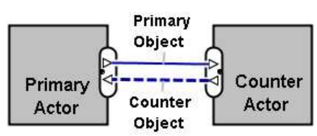
Certification

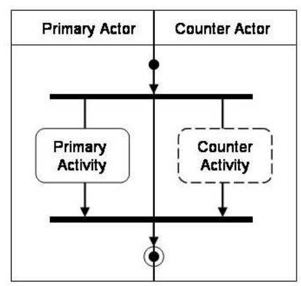
Context



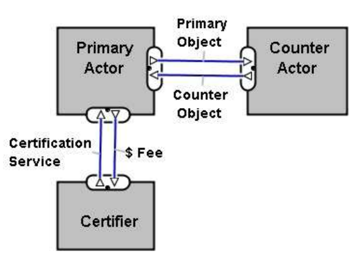


Problem





Solution



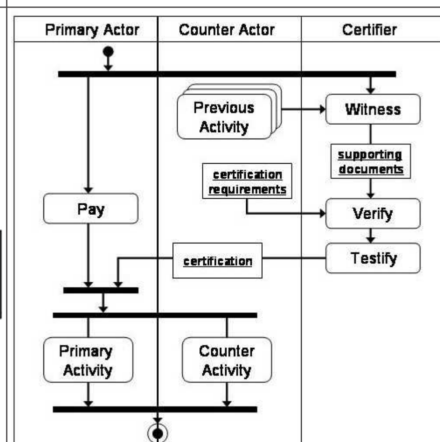


Table 12. Certification

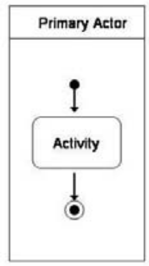
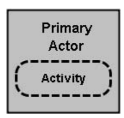
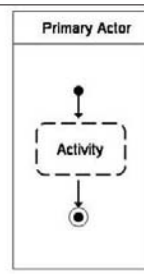
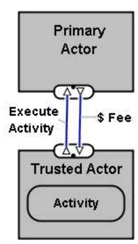
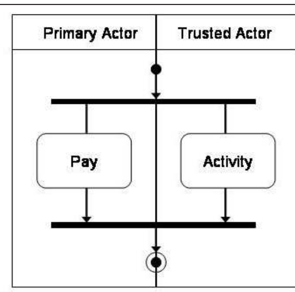
Delegation (Organizational Design Pattern)		
Context		
Problem		
Solution		

Table 13. Delegation (Organizational Design Pattern)