December 2005

Designing Control Mechanisms for Networked Enterprises: The Internet Radio Case Study

Jaap Gordijn
Vrije Universiteit Amsterdam

Vera Kartseva
Vrije Universiteit Amsterdam

18th Bled eConference

eIntegration in Action

Bled, Slovenia, June 6 - 8, 2005

Designing Control Mechanisms for Networked Enterprises: The Internet Radio Case Study

Vera Kartseva, Jaap Gordijn, Yao-Hua Tan

Vrije Universiteit Amsterdam, The Netherlands
vkartseva@feweb.vu.nl, gordijn@few.vu.nl; ytan@feweb.vu.nl

Abstract

In a network of organisations the design of appropriate control mechanisms is important to prevent and detect opportunistic behaviour of the members of the network. In most cases, control mechanisms can already be seen in the business value model, because most controls add new exchanges of economic value between enterprises. However, controls encompass also operational aspects, not covered by business value models, but which are important for the understanding and operation of controls. We developed the e^3-value$^+$ methodology for designing inter-organisational control mechanisms, based on analysing value aspects of network organisations. We illustrate it with the case for the distribution of music tracks via Internet radio, where we apply the methodology to design a control to monitor whether Internet radio stations and Right Societies cleared the right amount of tracks. We present the control mechanism not only from a business value model perspective, but also from an operational perspective, thus showing that the control can indeed be implemented.

1. Introduction

In a network of organisations, the design of appropriate control mechanisms is important to prevent and detect opportunistic behaviour of the members of the network. There have been studies on inter-organizational controls in business research, mainly with objectives to *explore* and *explain* control mechanisms (see e.g. for an overview [3]), however they do not address the *design* of these control mechanisms. On the other hand, research on the design of networked business models ([1], [10], [11],[12]) concentrates mainly on *economic value* aspects, and neglects the *control* aspects.

In this paper we propose a four-step methodology to design inter-organisational control mechanisms, based on analysing the *objects of value* that are exchanged by enterprises, forming network organisations. To analyse these objects, we employ the *e^3-value* approach [6]. In earlier work [8], [9], we have extended this *e^3-value* approach for modelling inter-organizational control mechanisms (*e^3-value$^+$*). A motivation to use a value-based approach as a starting point for inter-organisational control design is that many controls *themselves* have a strong value component. First, controls should act as a kind of safe-guards for the proper exchanges of economic value objects between enterprises. Second, control mechanisms are often themselves commercial services that create new exchanges of economic value objects. For instance, the Letter of Credit procedure is a control mechanism that is also a commercial service offered by banks to ensure that a seller gets paid for the products he delivers (both value exchanges) [7].

However, controls cannot be considered only from a value perspective. For example, the Letter of Credit procedure is implemented using various specific inter-organizational business processes between banks, sellers, buyers, and shippers; e.g. the exchange of all kind of evidentiary documents such as the Bill of Lading [7]. In this paper we focus on the role of IT to implement a design of a control mechanism. Hence, the main contribution of this paper is to show how an abstract design of a control mechanism, based on the economic value objects exchanged by enterprises, can be transformed into an actual implementation of this control mechanism.

To illustrate our approach, we use a case study from the area of Internet Radio. This is a new online service for which suitable control systems still have to be developed. The second author has an extensive experience in developing business models for Internet Radio, and the information provided here reflects the state of the art in the development of control mechanisms for value exchanges in Internet Radio.

This paper is structured as follows. In section 2 we explain the *e^3-value* methodology. This methodology allows for modelling the exchanges of value between enterprises as an *ideal* network of enterprises. 'Ideal' refers to the assumption in *e^3-value* that every enterprise keeps its promises; in other words: no enterprise will cheat. A, by definition ideal, *e^3-value* model provides a good starting point for inter-organizational control design, since all decisions concerning *who* offers *what* of value and requests *what* in return are already taken. In section 3 we present *e^3-value$^+$*, an extension to *e^3-value* to address the design of control mechanisms. The *e^3-value$^+$* models assume *sub-ideal* behaviour of an enterprise: sometimes they will show fraudulent behaviour (e.g. not delivering a good while a customer has already paid for it). In section 4, using the Internet radio case, we present a step-wise approach for designing control mechanisms, for which we use *e^3-value$^+$* to describe the business requirements for control mechanisms, and proceed with designing operational aspects of the control mechanism using the encryption technology. The paper ends with conclusions in section 5.

2. Ideal *e^3value* Models

A first step in developing controls is to understand the exchanges of economic value objects between enterprises. It is these exchanges that are subject to controls. We call models ideal, if it is assumed that all economic exchanges agreed between the business partners will indeed be carried out. This is called the *Principle of Reciprocity*. The *e^3-value* methodology [5],[6] provides modelling concepts for showing which organizations exchange things of *economic* value with whom, *and* expect *what* in return. The methodology has been previously applied for analyzing business scenarios in a series of case studies including media, news, banking and insurance, electricity power, and telecommunication companies to design value models of network organizations [6]. We briefly describe the concepts of the *e^3-value* methodology using a simple example. In Figure 1 a buyer obtains goods from a seller and offers money in return. According to the law, the seller is obliged to pay the value-added tax (VAT). This can be conceptualized with the following *e^3-value* constructs:

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

Value Object. Actors exchange value objects, which are services, products, money, or even consumer experiences. The important point here is that a value object is *of value* for one or more actors. *10 Boxes of DVDs* and *payment* are examples of value objects, but *legal compliance* to pay tax is also a value object.

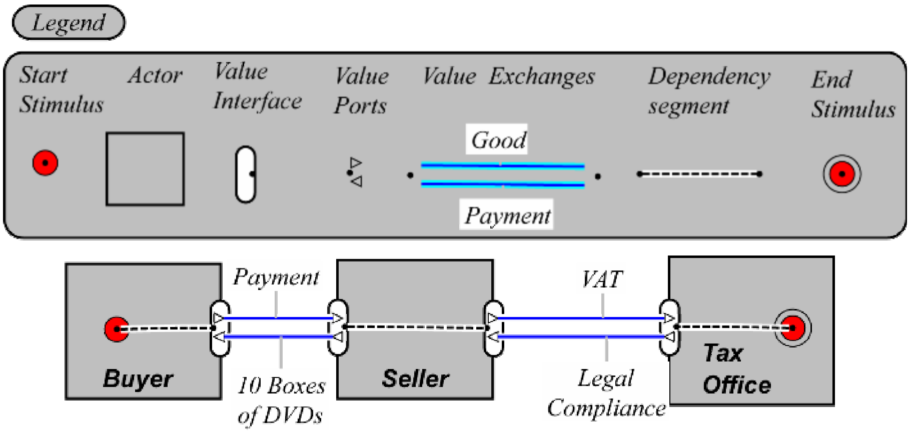

Figure 1: e^3-value model of a Purchase with Tax payment

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

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

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

With the concepts introduced so far, we can explain who wants to exchange values with whom, but we cannot yet explain what happens in response to a particular end-consumer need. For this purpose we include in the value model a representation of *dependency paths* (based on [2]) between value interfaces. A dependency path connects the value interfaces in an actor and represents triggering relations between these interfaces. A dependency path consists of dependency nodes and segments.

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

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

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

3. Sub-Ideal *e^{3}value*$^+$ Models

In designing control mechanisms it must be considered what could go wrong in the value model [14], [16], [18]. The value model can be in two states: (1) actors act in a way the *e^3-value* model prescribes, which further is referred to as an *ideal situation*, or (2) actors violate some prescriptions of the value model, which is referred to as a *sub-ideal situation*. An ideal situation can be described by an *e^3-value* model; for the description of a sub-ideal model we introduce *e^3-value*$^+$, which is an extension to *e^3-value.* In this section we describe extensions implemented in *e^3-value*$^+$.

3.1 The Violation of the Principle of Reciprocity

The first extension relates to the *principle of reciprocity*. The principle of reciprocity in *e^3-value* models (see section 2) states that an actor is only willing to exchange objects via all ports of its value interface, or none at all. This excludes the possibility of an exchange of a single value object, which, however, is possible in *sub-ideal situation*, for example, when one of the actors violated the agreement of reciprocal value exchange.

Figure 2 is a sub-ideal *e^3-value*$^+$ model, which shows various types of violations of the ideal model in Figure 1. The first type is the **exchange violation**. These violations are represented by value exchanges 2, 3 and 6. These exchanges have one of the value objects not delivered, which we call an **empty value object** (e.g. *No Payment, No Goods, No VAT, No Legal Compliance*). The corresponding exchanges are called **non-executed value exchanges** and are marked with dotted lines, and corresponding empty value objects take a different name, which starts with negation “No”. The second type of violation is the **object violation.** In value exchange 4 is modelled that the buyer obtained only 2 boxes of DVDs and the other 8 boxes with CDs. This situation reflects object violation: the exchange was actually done, but the value object exchanged was different from the ideal model. In case of object violation we distinguish **incorrect value exchanges** and **incorrect value objects**; incorrect value exchanges are marked with dotted lines, similarly to the representation of non-executed value exchange; incorrect value objects are assigned a different name.

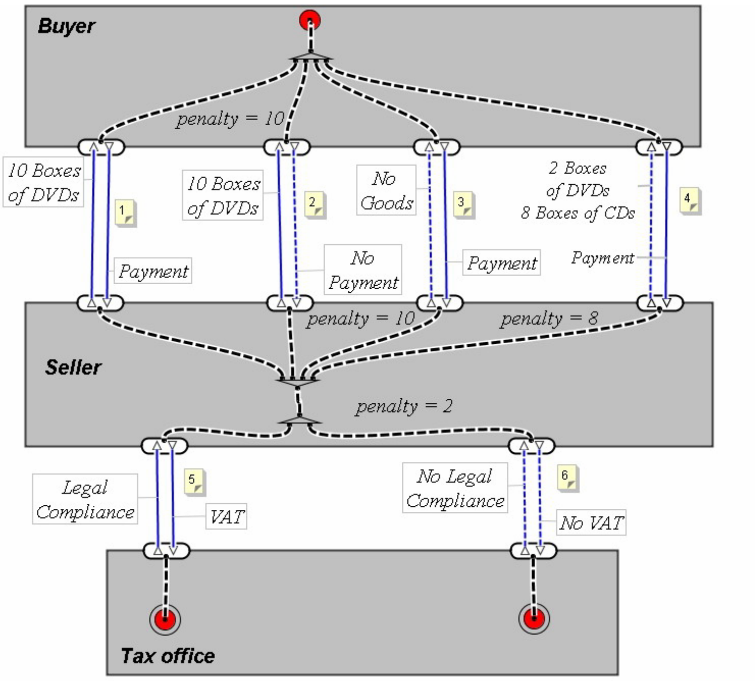

Figure 2: Some sub-ideal scenarios modelled with e^3-value$^+$

Due to these changes, in an *e^3-value*$^+$ model we distinguish **sub-ideal** and **ideal paths**. **Sub-ideal paths** are dependency paths in the value model that go through at least one "dotted" value exchange with empty or incorrect object. Hence, an **ideal path** has no segments that are connected to an incorrect or non-executed value exchange.

3.2 Modelling Control Mechanisms with Incentives and Penalties

Control mechanisms should motivate actors to take the ideal path rather than a sub-ideal path. For this purpose, the designer of a control mechanism should be able to identify what exactly is the sub-ideal behaviour, of which actor, and how severe this violation was with respect to other possible violations. For this purpose we introduce penalty weights. As in [13] and [17], penalty weights represent fines, which an actor gets if he does not behave as specified in an ideal *e^3-value* model.

Penalty weights are assigned to scenario segments connected to an interface of the *responsible* actor with a non-executed or incorrect value exchange. Penalty weights represent fines: the more severe the possible violation of the actor, the higher the fine. The segment connected to an interface with ideal value exchanges is assigned a zero penalty. In such a way, the designer of the control mechanism can identify what actor performed the violation.

To explain the method, we assigned in Figure 2 penalty weights to violating parties. Zero penalties are not modelled explicitly. To keep things simple, we use in this model absolute numbers indicating a preferential ordering. In exchange 2, 'no payment' we assume a violation by the buyer; therefore the buyer's scenario has a penalty of 10. In exchange 3 we assume the seller is responsible for not delivering goods, and is assigned a penalty of 10. In exchange 4 the seller delivered an incorrect value object, and is assigned a penalty of 8. In value exchange 6 we assume that the tax office cannot violate (thus, if the VAT was paid the legal compliance is always granted), therefore, the scenario segment of the seller is assigned a penalty of 2.

4. Internet Radio Case

We use a case study to present the four-step methodology for designing inter-organizational control mechanisms. The first step is the design of an ideal value model, using *e^3-value.* In the second step is the analysis of the control problems using *e^3-value$^+$*. The third step is the design of control mechanisms at the value level using *e^3-value$^+$*. In the fourth step the control mechanism is implemented at the operational level. The purpose of the case study is twofold: (1) to explain the methodology and (2) to enrich *e^3-value$^+$* with extra constructs necessary to model and implement control mechanisms.

4.1 Case Study Description

The case study is about free Internet radio. With 'free' we mean that listeners do not pay for listening to the radio. Many other options exist including pay radio, but we focus on free radio only. To broadcast a radio stream, an Internet radio has to obtain the rights from right owners such as performing artists, producers, text writers and songwriters. In Europe, the relevant right is the *right to make public*, defined in the upcoming European law [4]. Similar law (the DCMA) exists in the USA. Other opinions on rights (and even their relevance) exist, but we focus on the current situation on the right to make public and the consequences on controls. The case study focuses on clearing the right of making public. Roughly, if music is played outside the private environment, and listeners cannot select the tracks (but only the stream), one has to pay for 'making public'. In Europe, such rights are cleared by Intellectual Property Right (IPR) societies. These societies exist for many right holders and many different rights. IPR societies pay (repartition) fees for such rights to right owners.

Step 1: Ideal Value Model

A first step in designing controls is to construct an ideal value model. This ideal model for Internet radio right clearance has been developed in cooperation with one of the Dutch IPR societies during an earlier research project (see http://obelix.e3value.com). Note that, when we talk about the ideal model for Internet Radio here, we do not claim that the regulation of the Dutch IPR societies itslef is the best solution for Internet radio. The term ideal value model only indicates that the model represents that the economic exchanges between the different organizations are completely in accordance with the regulation of the Dutch IPR societies. For reasons of simplicity, we use a concise ideal value model here, as shown in **Figure 3**. In this model we only show details used in the further control mechanism design, but we do not show such components as, for example, advertisers, or an infrastructural component *Internet Access,* which listeners needs to receive Internet radio.

As **Figure 3** shows, in order to listen to a track broadcasted by an Internet Radio Station (IRS), a Listener first needs a track (or a stream) from the IRS. This IRS delivers the track for free. In return for a track playback, an IRS obtains *audience* to the track. Audience is of interest for the IRS, because audience attracts advertisers, which are the main source of revenue for an IRS (advertisers are not modelled here for the reasons of simplicity). IRSs use right societies for clearing the right to make the music public. Such societies operate on behalf of right owners, and offer stations the service to clear rights for a large group of right owners, say all Dutch right owners, and get a fraction of collected fees for their services. It is important to understand that in **Figure 3**, the exchanges between the IRS and Right Societies are on a per track per listener basis; a stream has been decomposed in its tracks via the AND fork at the Listener, and each listener finally results in exchanges between the IRS and right societies.

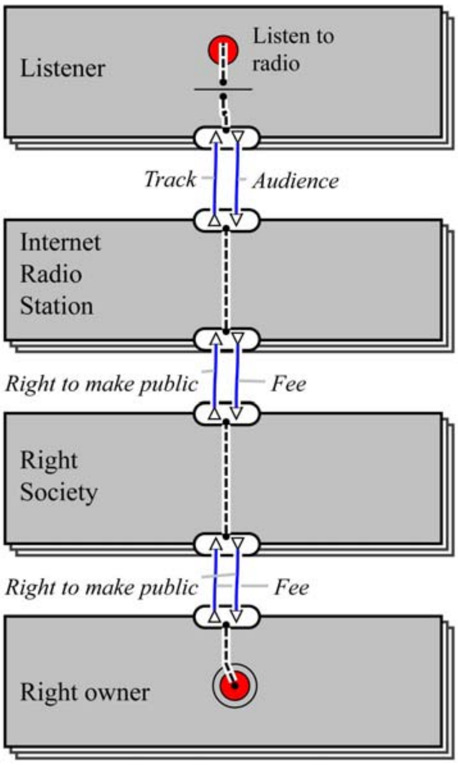

Figure 3: Ideal value model for free Internet radio

Step 2: Modelling Sub-Ideal Paths

Figure 4 presents the various sub-ideal paths, related to control problems that have to be solved. There are two possible ways of sub-ideal behaviour of actors: (1) the rights for a played track for a specific listener may not be obtained *at all* or (2) the rights for the *wrong track* (e.g. track B instead of the played track A) may be obtained. If the first situation occurs, a right owner will *not* be paid at all, and in the second situation the wrong right owner *is* paid (e.g. a party who created track B, not the played track A). Both the IRS and right society may expose sub-ideal behaviour; the Listener has no principal interest to do so.

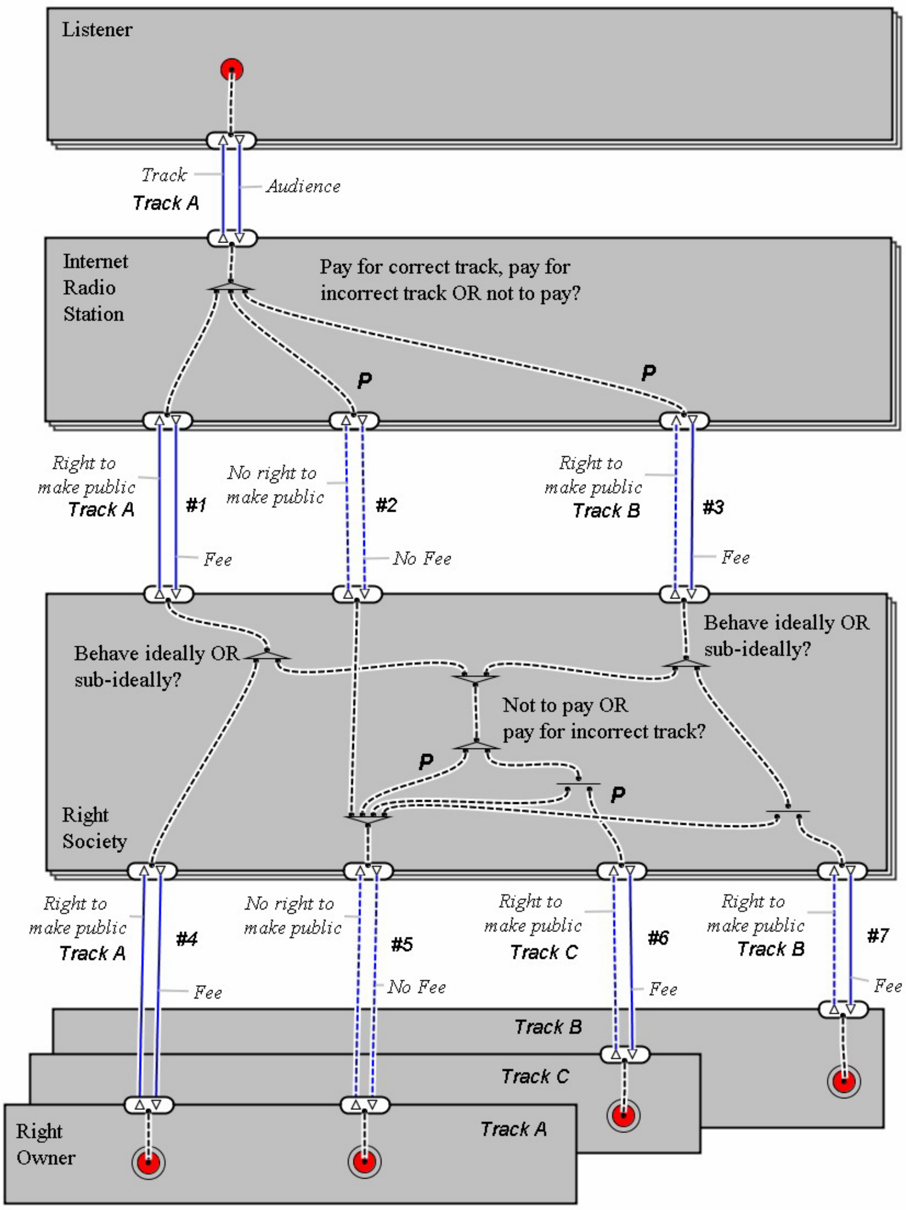

***Figure 4**: Sub-ideal value model for Internet radio*

Internet Radio Station

In **Figure 4** the Listener obtains from the IRS a track, part of a radio stream (*Track A*). Ideally, the IRS pays the right society for the right to make Track A public (*value exchange #1*). Not paying for the track at all is modelled with the “dotted” non-executed exchange *#2*, exchanging an *empty* value object. Obtaining the right for another track (*Track B)* than the one being played is modelled with the incorrect exchange *#3*. Here, the IRS purchases rights for the incorrect track; this is modelled with the *incorrect* value object *Right to make public* with label *Track B*, and the dotted line-marked incorrect value exchange. To model the last sub-ideal situation, it was necessary to add additional control information about tracks. Every right is associated with a specific track, therefore we *label* all the rights with the corresponding track. Thus, *Track A* stands for the original track (also exchanged between the listener and IRS), and other labels (e.g. *Track B*) refer to other tracks, but not the original one.

Right Society

Sub-ideal behaviour is tied to a specific actor. So, even if the IRS reports an incorrect track for clearance (*Track B*), the right society still can behave ideally or sub-ideally. If the IRS behaves ideally (value exchange *#1*), the right society has a choice to behave ideally or sub-ideally. If the right society chooses at the OR fork the ideal left path, then it leads to the ideal value exchange *#4*, so that the right owner of the played track is paid. If the Right Society makes a choice to behave sub-ideally and chooses the right path of the OR-fork, it leads to another OR-fork, and the Right Society has again a choice of two sub-ideal paths: not to pay at all or pay to another right owner. The path executed in case the right society does not pay at all, is the left path at this fork. It leads to the value exchanges marked as *#5*, consisting of the *non-executed* value exchanges *no rights* and *no fee*. If at the OR-fork the right society decides to pay to an incorrect creative party, then the most-right path of the OR fork is executed, which leads to the AND-fork, and then to the execution of the value exchanges marked *#5* and *#6*. The value exchange marked *#6* consists of an exchange with an *incorrect* object *Rights to make public* with label *Track C*, and the corresponding value exchange for the fee paid.

In case the IRS paid for the rights for the incorrect track (track B instead of track A), the right society also has a choice to behave ideally or sub-ideally, which is modelled with the OR fork on the path leading from the value exchanges annotated *#3*. The sub-ideal path (left path of the OR-fork) leads to the same OR-fork as was explained before. If the ideal path is chosen (the right path of the OR-fork), it leads again to an AND-fork, and to value exchanges annotated *#5* and *#7*. An interesting case is that the value exchanges annotated *#7* are also sub-ideal, however, this sub-ideality is caused by the IRS and not by the right society. This distinction is modelled using penalty weights. Penalty weights marked with letter P are assigned for different cases of violation, identifying actors responsible for violation. In this case, the right society does not get a penalty, because the violation was done by the IRS, which gets a penalty at the segment leading to value exchange #3.

Step 3: Value-Based Design of the Control Mechanism

Now that we have analyzed the control issues in the ideal value model, we can design controls that should monitor or even prevent execution of sub-ideal paths. A control mechanism may address a sub-ideal path in two ways: first, a control mechanism may *detect* a sub-ideal path execution, second, the control mechanism may *prevent* sub-ideal path execution. In this section we focus on modelling a specific *detective* control mechanism that can be used to assess whether all tracks are cleared. The key idea is to add a new right, being *the right to listen to music* (see Figure 5) to be obtained by the listener and it is impossible for the listener to listen to the track unless he has obtained this right. In section 0 we will show how to implement the control such that the listener cannot listen without obtaining this right, using encryption technology. In Figure 5, we suggest that the right society distributes the right to listen to the listener. Similarly to the right to make public, the right to listen is associated with a specific track at a specific point of time, therefore the label *Track A* appears near the corresponding value object.

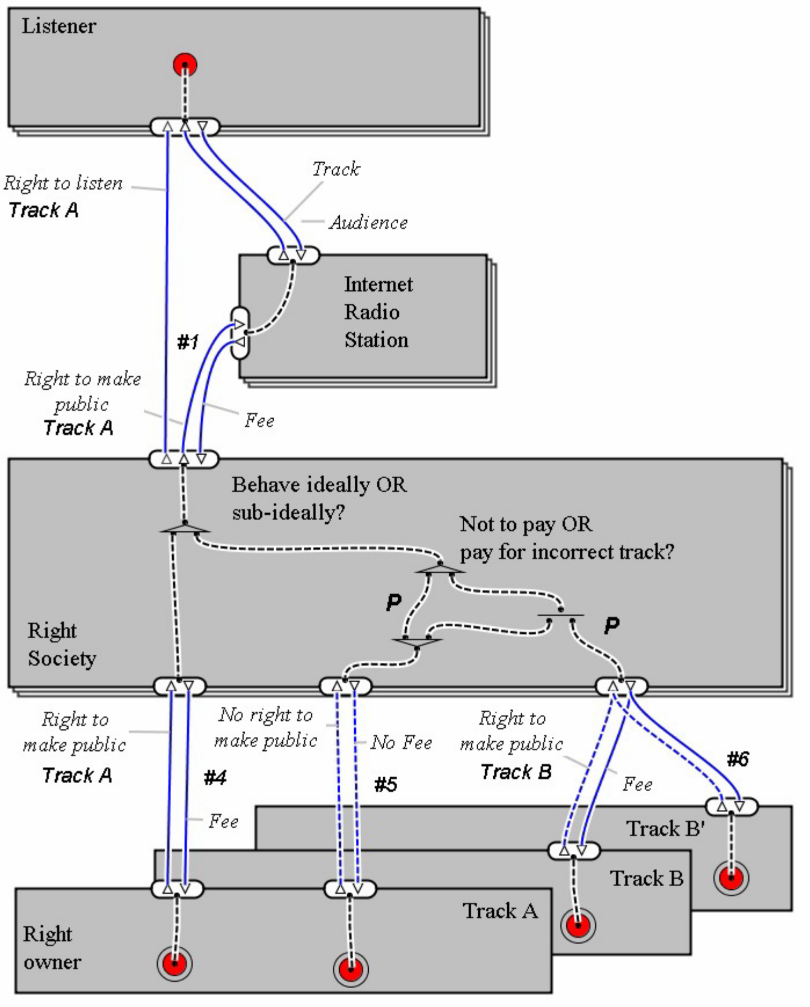

Figure 5: Control of IRSs

Due to this new value exchange, the right society can reconcile the number of rights to make public issued to the IRS with the number of rights to listen requested by listeners. This reconciliation is provided by the right society's value interface with three ports, which requires that the number of objects "Right to listen" equals the number of objects "Right to make public", and equals the number of object Fee exchanged. Assuming that the exchange of the right to listen for a specific track is guaranteed, the sub-ideal exchanges #2 and #3 between IRS and right society are detectable, and therefore they are removed in Figure 5. However, the sub-ideal exchanges *#5* and *#6*, caused by a cheating right society still remain and are not targeted by these controls.

To eliminate the remaining sub-ideal exchanges, we need the reconciliation to be executed by a party *different* than the right society. Ideally, the remaining party that can provide the right to listen is the right owner. In Figure 6 the right to listen is issued to the Listener by the right owner. The reconciliation of issued right to make public can be performed now at the three-port value interface at the right owner. Thus, if for a specific track the number of the exchanged rights to listen does not equal the number of rights to make public, such situation is detectable.

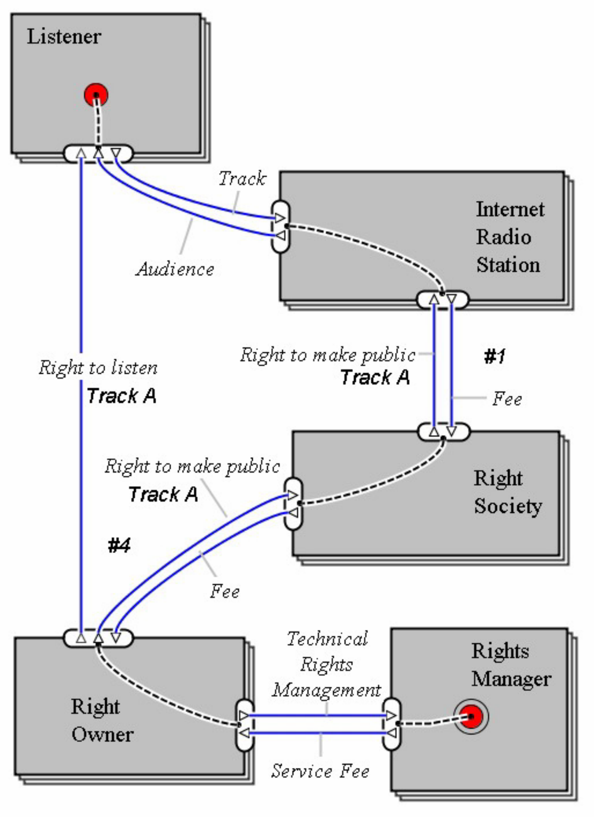

Figure 6: Introducing a trusted third party to control both IRSs and right societies

Additionally, we introduce a new actor *rights manager* who does the technical rights management on behalf of the right owner. Actually, the right manager has to *guarantee the exchange of the right to listen*. This is done at the operational level, described in the next section.

Step 4: Implementation of the Control Mechanism

In the previous section, we proposed to add an additional right: the right to listen to a music track. So, the listener should obtain *both* this right and the stream of tracks. This is expressed by the value interface of the Listener. How to guarantee that the semantics of the interface (exchange all objects, or none at all) hold? We can do so by using encryption technology (see **Figure 7**). Note that this figure is not an e^3*value*+ model, rather it is more similar to an UML collaboration diagram [15]. Arrows indicate messages that are exchanged between actors (boxes). Numbers next to the arrows indicate a time sequence. Boxes with an "E" denote an encryption operation, whereas "D" stands for decryption. This solution 'translates' the rights to a crypto graphical key issued to the various parties.

We distinguish three parties: The listener, the IRS (as in the value model) and the right issuer. The Right Issuer can be a right society, a right owner, or an organization operating on behalf of these.

In advance the right issuer and the IRS have agreed on an encryption key, K_{IRS} (message -2). The same holds between the right issuer and a listener, they agreed on $K_{Listener}$ (-1). *How* these keys are exchanged falls outside the scope of this paper, but one possibility is that these keys are stored on a smartcard, which is issued by the right issuer to the listener and the IRS, respectively. The assumption that these keys are distributed in advance is denoted by the "-" sign in the figure.

Both the IRS and the listener have at their premises a so-called *secure computing and storage device*. Such a device is tamper-proof and is trusted by the right issuer. In practice, a secure device may take the form of a smartcard, but it is also possible to implement such a device in a software component (in general a smartcard is more tamper-proof than software code). It is important that the listener and the IRS have no access to this device (without damaging it). The keys, K_{IRS} and $K_{Listener}$ are stored on the secure devices of respectively the IRS and the listener. So, although the IRS and the listener physically have the keys, they cannot read the keys because they are stored inside a tamper-proof device.

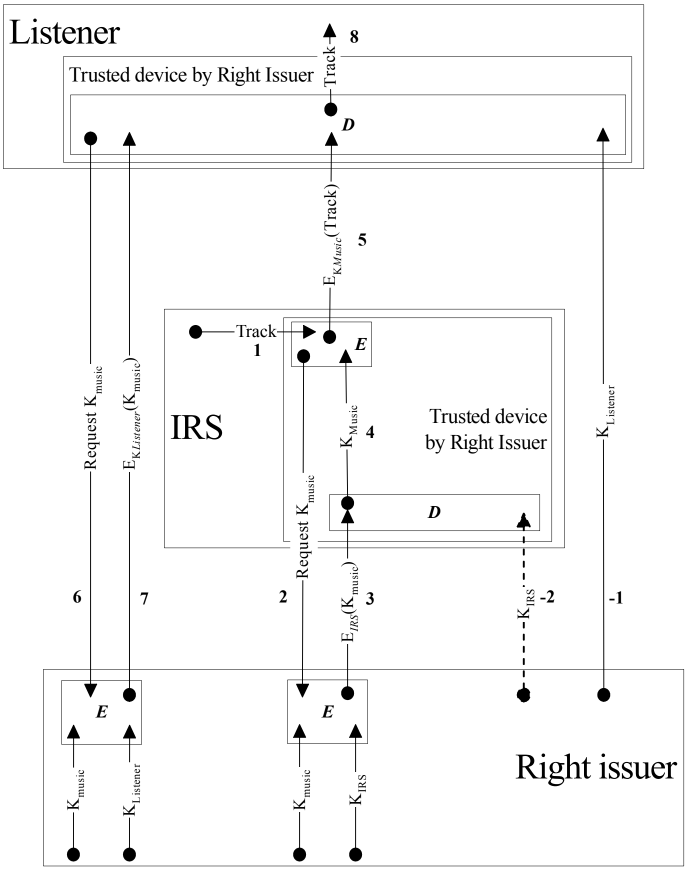

Figure 7: Use of encryption technology for track counting

If the IRS broadcasts a track (1), the secure device of the IRS first requests a key, K_{music} (2) This key is later on used to encrypt a music track such that to listen to a track, a listener *must* obtain a key to decrypt the track. This K_{music} is issued by the Right issuer and encrypted with K_{IRS} (3), a secret shared by the right issuer and the IRS. Consequently, no one can read K_{music}, even the IRS can not read the key. The encrypted music key (denoted by $E_{KIRS}(K_{music})$) is decrypted by the secure device of the IRS (denoted by D_{KIRS} ($E_{KIRS}(K_{music})$)), resulting in the plain key K_{music}(4). This key is used by the secure device of the IRS to encrypt the track (E_{Kmusic}(Track) (5)). Finally, this encrypted track is broadcasted and received by each listener.

To listen to the track, the secure device of the listener should decrypt the encrypted track. For doing so, the listener's secure device needs to obtain K_{music}. So, the device requests this key from the right issuer (6). *This request is logged by the right issuer for counting purposes. The right issuer compares the number of requests with the number of tracks reported by the IRS.* The right issuer sends in return the music key, encrypted with earlier agreed key of the listener ($E_{KListener}(K_{music})$) (7). The listener's secure device decrypts this message and uses the obtained key to decrypt the track, and plays finally the track (8).

This control mechanism does not yet implement all aspects of the control mechanisms designed at value level (see **Figure 5** and **Figure 6**). Namely, because K_{music} is not bound to a *specific* track (modelled with value object labels *Track A*, *Track B* etc.), this solution is not able to guarantee that the *correct* track is cleared, it only checks that *a* track is cleared. Thus, the IRS can still execute sub-ideal path #3 (see **Figure 4**). How can we prevent that an IRS combines a series of tracks into one track and offers this one combined track to its secure device for encryption? There are some solutions possible. First of all, the secure device can intelligently detect change of tracks. Such technology is successfully used, e.g. to remove commercials from a video stream. Second, the right issuer's computer can listen to stream of tracks broadcasted by the IRS and do intelligent track detection. The detected tracks can then be compared to the reported tracks. Using time-stamps, detected tracks can be bound to the logged tracks per listener.

5. Conclusions

The most important contribution of this paper is that we showed how to model controls from two perspectives; (1) the value exchange perspective and (2) the operational perspective, here in terms of a cryptographic implementation. The example shows that it is really important to distinguish the two phases. To some extent this is similar to modern methods in information systems developments, which starts with an abstract requirements analysis of the system, which is stepwise refined into a functional specification. Essentially, what we argue for is a similar approach for the design of controls. The first step of control design should be high-level, and abstract from implementation and operational details. At this stage the main issue is to identify the economic interests of all the partners involved in an economic exchange, or even a larger network organization. Based on this value perspective analysis of each of the partners, the next step should be to jointly identify the possible wrongdoings by the various partners, the so-called sub-ideal paths. Only after this has become clear for all the partners, one should go to the next step and design the operational details of the control mechanism. The Internet Radio case study clearly indicates how this stepwise methodology can be applied. Also the case study clearly shows that the step from designing an abstract control system to a concrete implementation is far from trivial. In future research we will study in more detail the underlying guidelines that help the designer of a control system to make the transition from the design of an abstract control systems to the design of an operational implementation of this system.

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

References

[1] Alt, R. and Zimmerman H.D. Preface: Introduction to Special Section – Business Models. *Electronic Markets*, 11, 1 (2001), 3-9.

[2] Buhr R.J.A. Use case maps as architectural entities for complex systems, *IEEE Trans Softwtware Engeneering*, 24, 12 (1998), 1131–1155.

[3] Dekker, H.C. Control of inter-organisational relationships: evidence on appropriation concerns and coordination requirements. *Accounting, Organisation and Society*, 29, 1 (2004), 27-49.

[4] EU Directive 2001/29/EC of the European Parliament and of the Council of 22 May 2001 on the harmonisation of certain aspects of copyright and related rights in the information society.

[5] Gordijn, J. *Value-based Requirements Engineering - Exploring Innovative e-Commerce Ideas*, PhD thesis. Vrije Universiteit Amsterdam, 2002, http://www.cs.vu.nl/~gordijn/.

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

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

[8] Kartseva, V. and Tan, Y.-H., Towards a Typology for Designing Inter-Organisational Controls in Network Organisations. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (CD/ROM),* 3-6 June 2005, Computer Society Press, 2005.

[9] Kartseva, V., Tan, Y., and Gordijn, J. Developing a Modelling Tool for Designing Control Mechanisms in Network Organisations, *Proceedings of 17th Bled International e-Commerce Conference*, 2004.

[10] Morris, M., Schindehutte, M., and Allen, J. The enterpreneur's business model: toward a unified perspective, *Journal of Business Research*, 58 (2005) 726-735.

[11] Pateli, A.G. and Giaglis, G.M. (2003): A Framework for Understanding and Analysing e-Business Models, *Proceedings of 16th Bled Electronic Commerce Conference*, Bled, Slovenia, June, 2003.

[12] Petrovic, O., Kittl, C., and Teksten, R.D. Developing Business Models for e-Business*, Proceedings of the International Conference on Electronic Commerce*, Vienna, Austria, October 31-November 4, 2001.

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

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

[15] Rumbaugh, J. and Jacobson I. and Booch G. *The Unified Modelling Language Reference Manual.* Addison Wesley Longman, Inc., Reading, MA, 1999.

[16] Starreveld, R.W., B. de Mare, and E. Joels. *Bestuurlijke Informatieverzorging (4th edn, Deel 1)*, Alphen aan den Rijn: Samsom, 1994.

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

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