Information Technology as Coordination Infrastructure

R.J. Wieringa*

April 13, 2006

Abstract

Business information technology is traditionally viewed as information provision technology. In this view, organizations use their IT to implement databases that provide people with information when they want it. This view is persistent even though information provision is never an end in itself but always has the further purpose to support the coordination of activities of people. The role of IT as coordination technology became more prominent in the 1980s with the advent of network technology, that allowed activities across different businesses to be coordinated. This trend has accellerated since the growth of Internet usage, and today IT is used to support an increasingly varied range of processes performed by a variety of partners that do not all have a hierarchical relation to each other. This makes it difficult to analyze requirements for IT support and specify IT solutions: Business processes may not be well-defined, and interests of different businesses may clash. This report argues that to deal with this in requirements engineering and IT solution specification, business information technology should not be viewed as IT support for business processes but as IT support for the coordination of activities in one or more businesses. We will identify three basic coordination mechanisms, namely coordination by price, by management, and by shared norms, and for each of these mechanisms, we will identify requirements for IT support. The advent of flexible and standardized networking technology has facilitated the creation of novel coordination mechanisms within these three general paradigms, and we will give an inventory of generalized coordination mechanisms made possible by current IT. Finally, we will draw conclusions for requirements engineering methods for IT support for each of the coordination mechanisms identified by the framework.

*Department of Computer Science, University of Twente, P.0. Box 217, 7500 AE The Netherlands. Email: roelw@cs.utwente.nl

Contents

1 Introduction

1.1 The problem

In the 1960 and 1970s, information technology (IT) was used by organizations, logically enough, to provide people with information. To determine the requirements for IT support, business processes were analyzed, information needs identified, and information systems were specified that provide the information required at various places in the business processes. Starting from the 1980s, business processes were also analyzed to identify process support opportunities, and solutions in the form of workflow management or cooperative work support were defined. It is not a coincidence that process support systems arose with the advent of network technology: Business process support is always the coordination of activities of different actors in a business, and these actors need to be connected in an IT network in order to provide process support.

An important characteristic of the requirements engineering process in all these cases is that it takes place within one business. In other words, there is, ultimately, one source of authority to make design decisions: the business. Each business is a hierarchy with a single highest point of authority to which other must report, and it is that highest point of authority that in the end authorizes the decisions made in the business. There is a single business mission, an overall strategy, shared business processes, problem-solving teams, decision rules, etc.

Starting with Electronic Data Interchange (EDI) in the 1980s, computers where additionally used to increase the effectiveness and efficiency of transactions across business partners. This trend accelerated in the 1990s with the advent of e-commerce. Requirements analysis for networked systems faces the problem that there is no single decision-making authority for the system, and consequently, agreement on requirements may be hard to reach. There is no overall strategy, partners in EDI only optimize with respect to their own goals and not to any overall goal —assuming that there would be any—, there may be ambiguous processes, there is no shared problem-solving structure, etc. As a result, requirements negotiations may be interminable. This was already apparent in EDI systems development, which is notorious for its interminable negotiations among participating businesses. And the demise of the e-commerce hype made visible that in networked business, there is no magic either. One cannot translate a smart business idea into software and start operating without working out a win-win situation for all actors involved, and this involves negotiating an agreement with these partners.

To develop business information technology, we should drop the idea that this technology is used for a single business process in one business and incorporate theories and techniques that deal with multiple viewpoints and interests at the outset. However, the idea that requirements originate from a hierarchy is still pervasive in current requirements engineering methods. Textbooks in the area suggest that requirements are there to be elicitated [16], that problems wait patiently to be analyzed [12], and that there is a zoo of specification techniques to be used once requirements are found [35].

In this report we approach requirements engineering from a coordination point of view. Coordination by management hierarchy is just one coordination mechanism, and we will identify many more. Each of these coordination mechanisms imposes particular requirements on IT support, that we will analyze in this report.

We are concerned with the coordination of *business* activities. However, businesses

consist of people and they have customers that may be other businesses or people; and all people use software to accomplish tasks. Hence, the actors to be coordinated may be businesses, people, or software. Coordination between businesses has important differences with coordination between people, mainly of a legal nature, and these differences are relevant for the use of information technology. However, here we will investigate common features across all these different situations and draw conclusions from those common features about the use of information technology.

1.2 Related work

In this report I combine ideas from several areas. I start from a distinction between transactional and relational contracts that was introduced by Macneil [19, 20] in the 1970s in a thorough analysis of legal contracts. I combine these ideas with the concept of coordination introduced by Malone and Crowston [21] in the 1990s and apply them to the distinction made in economic sociology by Ouchi in 1980 and by Powell in 1990 between communities (called by various names, such as "clans" and "networks") and markets [27, 29]. Communities are relational structures coordinated by shared norms and values. Markets are transactional structures coordinated by price. Management hierarchies are structures coordinated by management authority [14].

I use the basic ideas from transaction cost economy introduced by Coase in 1937 [5] and worked out by Williamson [37, 38] in the 1970s and 1980s to characterize differences between the three basic coordination mechanisms of markets, hierachies and communities. Finally, I use the analysis of outsourcing by Clemons, Reddi and Rows [4] to describe the impact of IT on these basic coordination mechanisms. This leads to the classification of coordination mechanisms presented in figure 3. This table is a hypothesis to be further validated by analysis of empirical data.

The framework for coordination mechanisms is used to define a requirements engineering approach that borrows elements from Jackson's problem analysis [12] and my own systems engineering approach [36]. The novelty of the approach described in this report lies in the particular combination of ideas known from different areas and their application to requirements engineering for IT support.

1.3 Structure of the report

In section 2, we approach the topic of coordination by defining two extreme coordination mechanisms, namely coordination by relationships and coordination by transactions. These are ideal types[1] that do not occur in reality in pure form. Actual coordination mechanisms are a mixture of these two.

In section 3, we analyze the different properties of these coordination mechanisms with respect to planning and replanning of coordinated activities. We also introduce the notion of contract as coordination specification and relate this to coordination by relationships and coordination by transactions. This leads to the two concepts of transaction contract and relation contract. Again, these are ideal types that occur in reality only in mixed forms.

Section 4 then introduces the three coordination mechanisms based respectively on price, management authority or shared norms, and characterizes these by three variables known

[1] An ideal type is a concept that describes one aspect of reality and ignores other aspects. This helps us to understand underlying regularities that would otherwise not be visible.

from transaction cost theory: Predictability, trust and switching cost.

Section 5 discusses IT support that can be given to these three basic coordination mechanisms. IT also facilitates novel kinds of coordination, which are placed in a framework for coordination mechanisms in section 6. In section 7 we discuss the approaches to requirements engineering for IT support of each of the three major coordination mechanisms. This makes clear that RE approaches should be quite different. For example, if an RE approach to hierarchical coordination were used for market support, it would miss important requirements, which is a recipe for failure. Section 8 winds up the report with a summary and an indication of further work.

1.4 Terminology

In this report we use the following terminology.

- A **commodity** is a product of which all properties relevant for the consumer can be completely described and which, once described, can be produced in large quantities. A consumer can choose between commodities based upon a description of them and their price.
- **Coordination** is the execution, monitoring and (re)planning of dependent actions by different actors.[2]
- The term **product** will be used in this report to indicate material goods or immaterial services.
- A **norm** is a standard against which entities or activities are measured. For example, *Webster's* defines a norm as a principle of right action binding upon the members of a group and serving to guide, control or regulate proper and acceptable behavior [23]. Other examples of norms are the standard meter in the MKS system and the norm set by management for this week's production.
- A **value** is a utility. A phenomenon is of no value if no demand for it exists, and it is of negative value if there is demand for it not to exist. The value of a product determines its price to a certain extent but not completely. Prices only exist with reference to exchanges, but from the moment we want a thing we value it, and once we possess it, it can still have value for us [32, page 281].

2 Transactional and Relational Coordination

The basic coordination mechanisms of transactions and relationships have been analyzed in the 1970s by Macneil [19, 20] in an analysis of legal contracts. The idea is simple: A contract is an agreement between actors to coordinate their activities. So to classify contracts, one must classify coordination mechanisms. Macneil found two basic coordination mechanisms, namely coordination by transactions and coordination by relationships. We will define these mechanisms next and then summarize their properties.

[2]This is more specific than Malone's defintion of coordination as the management of dependencies between activities [21].

There are many different kinds of transaction, ranging from anonymous transactions on a spot market or stock exchange market, to informal exchanges between friends or members of a family. To emphasize that we are defining an ideal type, that only occurs in reality as part of a more complex coordination, we define the concept of a discrete transaction. A **discrete transaction** is an exchange of products and money between actors that have no relationship to each other whatsoever and that is separate from any other relationships to others that the actors currently have. Having no relationship to each other here means that the actors do not have a relationship now, did not have one in the past and will not have one in the future. Transactions on a sport market and on the stock exchange are usually discrete.

The concept of a discrete transaction is defined using the more primitive concept of relationships between people. A discrete transaction is an exchange between actors that do not have a relationship other than engaging in the transaction. We will not define this more primitive concept of a relationship except by giving examples. Example relationships are kinship relations, the relationship between friends, between husband and wife and between parents and children, between colleagues at work, between employee and employer, between participants in a game, between members of a professional community, between long-standing business partners, etc. Note that a relationship connects people. In a derivative sense, relationships can also connect businesses, because businesses consist of people that can have a relationship. This is different from discrete transactions. A discrete transaction can be performed by business partners or even by software actors; relationships as intended here can only exist between people.

Within all possible relationships between people we single out primary relationships as a special ones. A **primary relationship** is a relation between people in which each participant is irreplaceable and participates as a whole. Because the participant is irreplaceable, the relationship cannot be transferred to other persons; because the person participates as a whole, there is no limit to the aspects of life that the relationship can cover. A primary relationship constitutes part of the identity of the participants. Examples of primary relationships are those between parents and children, between husband and wife, and between close friends. Examples of non-primary relationships are those between employer and employee, between business partners, and between members of a profession. In non-primary relationships, participants can be replaced; the relationship can be transferred to other persons (e.g. by delegation); and the relationship covers only a limited part of the life of each participant. Non-primary relationships are not part of the identity of any participant.

The primary relationship is an ideal type that only occurs in reality only in a diluted form. Even parents and children do not talk about every conceivable topic, thereby violating the wholeness property of primary relationships; and even business partners may develop a close friendship, thereby violating the replaceability characteristic of non-primary relationships.

Primary relationships and discrete transactions can be contrasted as shown in figure 1. In a discrete transaction, participants are essentially anonymous. If any actor knows the identity of any other actor, then this is coincidental and not important for the transaction. Participants can be replaced without changing the identity of the transaction. The identity of participants in a relation, by contrast, is important. Participation in the relationship defines part of their identity, and conversely replacing a participant changes the relationship.

Related to this, participants in a discrete transaction are a means to a purpose, namely to perform an exchange of products against money. By contrast, participants in a pri-

	Discrete transaction	Primary relationship
Identity of participants	Unimportant	Essential
Purpose	To acquire products	To continue the relation
Communication	Concerns the exchange only	About any topic

Figure 1: Primary relationships and discrete contrasts contrasted.

mary relationship are not a means to a purpose. They participate because of their unique individual properties. One of the purposes of the relation is simply to continue the relation.

Because actors participate in a relationship as a unique whole, communication between participants in a relation is wide-ranging. There is no aspect of the participants' life that cannot be discussed in a primary relationship. Communication between participants in a discrete transaction, on the other hand, is purely functional. It concerns the exchange only.

To repeat the point once more, the concepts of discrete transaction and primary relationship are ideal types that occur in various mixtures in reality. For example, even when buying oil on a spot market, there is a shared language in which oil is described, there is mutually accepted money, a shared system of property rights in which each party accepts the transfer of property in the transaction, a shared classification system of types of oil, etc. In other words, the participants must participate in a relational community of shared norms in order to perform the discrete transaction. However, they do not need to have a primary relationship to perform the discrete transaction. Conversely, even in primary relationships, transactions occur. Parents may count the number of times their children do the dishes and even close friends may keep track of the number of visits to each other's homes. To the extent that this happens, shared activities are treated like monetary exchanges.

The ideal types of discrete transaction and primary relationships are useful to understand coordination mechanisms. Before we turn to that, we list a number of additional properties to highlight the differences between coordination by discrete transactions and by primary relationships.

The following differences between discrete transactions and primary relationships have consequences for IT support.

- **Duration.** Discrete transactions are performed quickly. Contact during the transaction is short. By contrast, primary relations need time to prepare and once they exist, they usually exist for a long time. Thus, any relationship has a history and a possible future. Activities coordinated in a relationship have a tradition, consisting of shared values and norms, and they are often coordinated in agreement with this tradition.

- **Start and finish.** A discrete transaction has a crisp start and a crisp finish. It starts by clear agreement and finishes by clear performance. Relations start gradually, because people need time to get to know each other, and they end gradually, because they also need time to alienate from each other. To participate in a relational community, people have to learn the tradition, the shared values and norms of this community.

- **Cardinality.** Discrete transactions preferably involve two participants. Inclusion of a third could introduce relational aspects that would destroy the discreteness of the

transaction. Primary relationships usually have many participants. These all share the tradition associated with the tradition.

- **Existence and identity.** A discrete transaction exists as long as actors participate in it, and ceases to exist when actors do not participate anymore. The identity of the transaction is determined by the exchange that takes place. A relation, by contrast, may exist among a number of actors before a particular actor starts participating in it and it may continue to exist among the remaining actors when an actor exited from it. Its identity at any point in time is determined by the participants in it, but this may evolve slowly in time. The relationship defines a culture that may accept new members but that maintains its identity independently from any particular member.

We can immediately draw some conclusions for IT support. IT support for discrete transactions needs to provide an atomicity guarantee and a possibility to clearly demarcate the start and finish of a transaction. Communication partners may be anonymous; they may even desire anonymity; and they may act on behalf of others. By contrast, IT support for relationships needs to provide a permanent communication infrastructure through which groups can communicate. People must be able to join and leave these groups while the communication network continues to exist. The community defined by the relationships must announce the identity of each participant to other participants and anonymity is not an option.

3 Coordination specification

3.1 Planning

Discrete transactions and primary relations have contrasting properties with regard to planning and specification. This is important for coordination support, because to support coordination, the coordination must be specified explicitly.

- **Specifyability.** A discrete transaction can be specified in advance completely. The specification describes what is exchanged when, and under what conditions, and what the value of the exchanged products is. It also specifies what must happen when something goes wrong. For each possible exception, a handling procedure is defined in advance. By contrast, in a primary relationship, none of the activities that participants will perform are specified in advance. In many cases, participants themselves may only be able to describe what happened after the fact, not beforehand; and in many cases the activities were so complex or chaotic that there is no agreement among participants about what happened; and neither may there be a need to agree about it.

- **Predictability.** Discrete transactions are planned beforehand and then executed as planned. All contingencies must be specified beforehand, as well as all responses to these contingencies. The exchange is predictable, precisely because the transaction itself is inflexible; or, more precisely, all flexibility can be foreseen and is specified in advance. Any planning for flexibility must take place before the transaction takes place and this planning activity is not part of the transaction itself. Relations, by contrast, are flexible. They are too complex to plan completely before they start.

Planning takes place as part of the relation and often is performed jointly, by several partners in the relation.

- **Performance ambiguity.** What is exchanged in a discrete transaction is commodified. This means that its properties are known in advance and can measured, and its value is known. Performance ambiguity is low: It should be clear to all partners who did and who did not deliver what was promised. In a primary relation, by contrast, no explicit exchanges take place. Partners engage in shared activities that may or may not not be planned in advance and that are not intended to distribute a measured and specified mount of value to the participants. It is very ambiguous, if not meaningless, to try to determine afterwards who contributed what value to results that are produced. It is a typical property of group work that attributing results to individuals is not possible and often undesirable because it violates group spirit.

- **Benefits.** A discrete transaction is a reciprocal shifting of valued products that is intended to be beneficial for both actors. Relations, on the other hand, are value-creating activities in which all participants share the burdens and benefits. Benefits partly return to the party that helped creating the benefit.

- **Conflict-resolution.** When trouble occurs in a discrete transaction, the transaction is rolled back and if that is not possible, damages are claimed according to procedures specified in advance in a contract. In a primary relation, trouble is expected as normal aspect of life and when it occurs, partners solve the problem together. The purpose remains to continue the relation rather than to terminate it and claim damages. Claiming damages would signify the end of the relation.

These differences all concern the way activities are coordinated: By planning in advance or by joint problem-solving when the need arises. A discrete transaction is a coordinated set of activities in which an exchange takes place. The exchange is specified in advance and executed as specified. The specification includes the expected performance of all partners as well as the ways in which non-performance can take place, plus the expected responses to non-performance. The basic coordination mechanism is *price:* Each participant in a discrete transaction puts a value on the exchange and is willing to exchange something for this value. In a market of purely discrete transactions, the exchanged commodities are completely described and their price is the only remaining characteristic that actors need to know in order to select the discrete transaction they want to engage in.

Primary relationships provide a context for the coordination of activities that are not specified in advance but for which shared norms and values exist. When kinship members, friends, or members of a profession need to coordinate their activities, there is a tradition of cultural norms, shared experience, shared values and shared goals that allows participants to coordinate their activities without explicit specification in advance. When the need arises, specifications of joint behavior may be produced that are grossly incomplete because they supplement implicit agreement about what should be done anyway. The basic coordination mechanism is a collection of *shared norms:* Each participant trusts each other participant to share these same norms and this allows them to coordinate their activities according to these norms. Participants trust other participants whose identity they know, and whom they know to participate in the relational community in which these norms are shared.

3.2 Contracts

A **contract** is an agreement between two or more actors to coordinate their activities [19]. Contracts may be specified in a written text, or they may be made orally without any written proof. Contracts may be legally binding or legally void. And a contract may contain or imply agreements that none of the partners may be aware of. For example, law or tradition may prescribe certain elements of the agreement that were unknown to the contracting partners at the time they struck the contract, but that are binding anyway.

Corresponding to the two basic coordination mechanisms there are two contracts, that we will call discrete and relational. A **discrete contract** is a specification of a set of coordinated activities among partners in which products and money are exchanged in a discrete transaction. Because a discrete contract describes a discrete transaction, it has the following properties.

- A discrete contract treats as irrelevant the identity of the parties to the transaction.
- It commodifies as much as possible the exchanged good or service.
- It limits as much as possible the sources to be considered in determining the activities that are to take place. As far as possible, these are all specified in the contract. In discrete contracts, possible secondary sources, such as an umbrella contract, the law or standards, are avoided.
- But if secondary sources are to be considered when determining what activities have been agreed to take place, there is a priority order among them. The order, from high to low, is this:
  - Written specifications;
  - Orally communicated specifications;
  - Information communicated nonlinguistically (e.g. by behavior);
  - Communicated circumstances when agreement was struck (e.g. time and place of agreement);
  - Noncommunicated circumstances when agreement was struck (e.g. tradition).
- The content and consequences of the transaction are treated as atomic.[3] This means that all promised consequences of the transaction are treated as if they were present at the start of the transaction. So the transaction is treated as an atomic activity in which all its effects are present immediately.
- A discrete contract assumes that at all parties to an exchange start the exchange with full consent as specified in the initial contract. All parties are assumed to be aware of all consequences of the contract and to agree to this.

A **primary relational contract** is a specification of a relational context for performing future activities. Just as any other contract, a primary relational contract may be written or oral, it may be legal or illegal and it may contain elements not thought of by the contracting partners. A primary relational contract is somewhat like the hypothetical social contract of

[3] Macneil uses the term *presentiation.*

the 18th century philosophers. As a specification of future activities, a primary relational contract is grossly incomplete because it describes a shared background for coordinating activities without specifying the coordination itself. The features of primary relational contracts mirror those of discrete contracts, and they follow from the properties of primary relationships.

- A primary relational contract identifies the partners of the relational community being defined.
- It does not specify any particular exchanges between the partners. Exchanges may very well take place, but they are not specified in a primary relational contract.
- The contract allows secondary sources outside the contract specification to determine what is actually to be done.
- The priority order of sources to be considered when determining what activities have been agreed to take place, is the reverse of that for discrete contracts:
  - Noncommunicated circumstances when agreement was struck (e.g. tradition);
  - Communicated circumstances when agreement was struck (e.g. time and place of agreement);
  - Information communicated nonlinguistically (e.g. by behavior);
  - Orally communicated specifications;
  - Written specifications.
- It is unknown which activities are going to be coordinated. It does not make sense to indicate particular results that must be produced by the primary relationship.
- The contract expresses agreement with a set of shared norms and values, without expressing willingness to engage in any particular joint action whatsoever.

In relational coordination, written contracts are the least important source to find out what activities should take place. Oral agreements have precedence, and the traditional way of doing things has highest precedence.

Discrete contracts and primary relational contracts are again ideal types. Rarely is a contract a fully complete specification of coordinated activities. And rarely is a contract merely a specification of an intention to cooperate under a certain tradition of cooperation. Actual contracts specify some activities to be coordinated, and also presuppose some relational community within they will take place. However, some contracts, such as spot market contracts, are clearly at the discrete end of the spectrum and other contracts, such as marriage contracts, are clearly at the relational end. In all cases, the contract is a description of the type of coordination activity that is to take place. And in actual cases, the priority order of sources to determine what should actually happen is not so clear cut as suggested by the two listings given above.

4 Coordination Mechanisms

4.1 Markets, communities and hierarchies

A market is an infrastructure for coordination by price, and a community is an infrastructure for coordination by shared norms. All actual coordination mechanisms combine these two. Of all possible coordination mechanisms, management hierarchy has received a lot of attention and we will treat this as third coordination mechanism. It will become apparent later that there are many other kinds of coordination mechanisms.

- In a **market,** coordination takes place in discrete transactions. In each transaction, products (goods or services) and money are exchanged. Because the description of a commodity is known to the user, the remaining item of information used for coordination (trading) is *price,* which for commodities gives information about the scarcity of the product [30]. Examples of markets are spot markets, the stock exchange, auctions, supermarkets, grocery stores, malls, weekly open air markets, and your local hardware shop. Some of these, such as spot markets and auctions, are discrete markets because all transactions occurring in them are discrete. Most other markets have some relational aspect. For example, you do business with your local hardware shop based on mutual trust. If the business owner knows you personally, and knows you share some norms that govern your and his activities, then you may return goods that you do not need and you may postpone payment to a later time.

- In a **community,** activities are coordinated relationally, i.e. based on shared norms. The coordinated activities are a shared burden but also benefits produced jointly are shared without being allocated in a measured way to participants. Examples of communities are kinship networks, ethnic networks, professional networks, "old boys" networks, regional economies, and even criminal networks. Some of these, such as kinship networks and ethnic networks, are primary, but most are nonprimary because they contain transactional elements. For example in regional economies, people do business with each other based on trust but in the long run, there is expected to be a measured reciprocity in exchanges.

- In a **hierarchy,** activities are coordinated by management. There is an asymmetrical relationship in which some actors, called employees, have agreed to obey management oversight by another actor, called their employer, but they agreed this against a price: their salary. Employees have an incomplete long-term relational contract with their employer that states the general content of their work and indicates a shared goal, but does not specify in detail all activities to be performed. Activities are coordinated on a day-to-day basis by management. There is an asymmetric relationship in which the employee *reports* to his or her manager and the manager is *accountable* for the employees reporting to him or her [14]. This accountability relationship is absent in markets and it is diffused in communities. Examples of hierarchies are businesses, schools and government institutions. Hierarchies have community and market properties to varying degrees. For example, in a film production company, hardly any activity can be specified in advance in detail, as is usual in communities, but in other hierarchies, such as job shops, most activities can be specified in advance, as is characteristic of discrete transactions.

The importance of hierarchies was first pointed out by Coase [5], who showed that to produce a product, we must choose a position between two extremes. One extreme is a market, where to produce a product, an actor will contract with other actors to perform the required activities. This reduces production costs for the actor to zero, because all production is performed by others and the actor merely coordinates these activities using contracts. But transaction costs will accordingly be very high.

The other extreme is to create a network of non-primary relationships called a *business* and introduce a management hierarchy to coordinate activities in the business. Coordination by management differs from coordination by shared norms because it is centralized and is done by communicated instructions rather than by implicit norms. In the extreme case, the business integrates all required labor and capital in its own hierarchy, so that all production activities are performed in-house. This reduces transaction costs to zero but production costs will be very high.

Assuming economic rationality, actors will choose a point between these two extremes. Some activities will be outsourced, reducing production costs and increasing transaction costs. Other activities will be performed in-house, increasing production costs and reducing transaction costs. Under the assumption of economic rationality, the position chosen will be one in which the sum

production cost + transaction cost

is low. We will say that the *efficiency* of a coordination mechanism is high when this sum is low.

4.2 Predictability, trust and switching cost

We now have identified three coordination mechanisms, namely coordination by price, coordination by shared norms, and coordination by management, and identified three infrastructures for this: markets, communities and hierarchies. Transaction cost economy [37, 38] and economic sociology [27, 29] provide concepts to understand the choice between these three coordination mechanisms. The three operative variables are predictability, mutual trust and switching cost.

- **Predictability** is our ability to describe the future. Predictability is a function of our cognitive limitations and the complexity of future events. In a discrete market, transactions can be completely described and so actors can conclude complete contracts. And once a transaction is contracted for, they can be certain about what will happen in a transaction. If predictability of future activities is high, performance ambiguity of these activities is low and vice versa.

- **Trust** is accepted vulnerability to an actor's possible but not expected ill will [2]. Mutual trust is high in communities, because actors depend upon unspecified activities performed by others. Trust disappears in the face of *opportunism,* which is self-interest seeking with guile [38]. Opportunists choose to perform activities in their own interest even if they promised to do otherwise, they intentionally present information they don't believe is true, or intentionally withhold information that is relevant for partners in the exchange.

	Markets	Hierarchies	Communities
Coordination mechanism	Price	Management	Shared norms
Predictability	High	Low	Lower
Performance ambiguity	Low	High	Higher
Mutual trust	Low	High	Higher
Opportunism	High	Low	Lower
Switching cost	Low	High	Higher
Asset specificity	Low	High	Higher

Figure 2: Differences between markets, hierarchies and communities.

- **Switching cost** is the cost of replacing an actor in a cooperation by another actor. It is directly related to *asset specificity,* which is the specialization of assets with respect to its uses. For example, interactions with a particular actor may require specialized equipment, or require transaction-specific knowledge from employees, or may require assets to be located at one geographical place. Asset specificity, and therefore switching cost, is low in discrete markets, because discrete transactions can be performed with any actor at all that offers the required products. In relational communities it is high, because relationships involve a a build-up of relation-specific knowledge and may involve specialized equipment dedicated to particular partners in particular locations.

Note that predictability is a cognitive factor that is essential for discrete markets because without predictability there cannot be discrete contracts. Trust is a moral factor that is essential for communities because without trust there cannot be coordination based on incompletely specified activities. Asset specificity, and thus switching cost, is an economic factor that concerns the investment by an actor in specialized assets. Figure 2 summarizes the values of these variables for the three coordination infrastructures.

If transactions can be specified completely, then they can be specified in discrete contracts and we can coordinate exchanges in a market by means of price. However, if they cannot be predicted at all, coordination by shared norms is an option. Shared norms can substitute for completely specified actions. This means that switching costs are high because there will be a significant investment in personal relationships and specialized assets dedicated to particular relationships. This is efficient as long as mutual trust is high, because otherwise opportunistic partners may switch to other partners if they perceive this to be in their self-interest.

If mutual trust is low but asset specificity is high, then a business can reduce risk by incorporating potentially opportunistic partners. In other words, it will buy these partners and integrate them as business units into its own hierarchy, making switching impossible for them. This will improve efficiency and replace coordination by price with coordination by management. The same holds for market transactions that require asset specificity: the risk of opportunism can then be reduced by incorporating business partners within one hierarchy. In both cases, a situation of low trust is replaced by one of higher trust.

There are detailed economic analyses of the mechanisms why economically rational actors choose one of these coordination mechanisms [4, 37, 38]. It is not the purpose of this report to rehearse this. We will use the characterization of coordination mechanisms given so far to identify IT support for these mechanisms in the next section. This will lead, in the

section after that, to an identification of refinements of our classification of coordination infrastructures.

5 IT support

5.1 IT support for coordination in markets

IT support for markets must provide functionality for coordination by price in a context where transaction partners initially do not know each other, and may never get to know each other. Transactions are completely specified in advance, and performance can be monitored and evaluated quantitatively. This leads to the following list of IT support functions for coordination by price [1, 3, 15].

- **Matching buyers and sellers.**
  - Advertising a product.
  - Searching for a product.
  - Aggregating product supply or demand
  - Gathering product information.
  - Negotiating a price.
  - Specifying a contract.
- **Transaction execution.**
  - Supporting logistics.
  - Supporting or performing payments.
  - Replacing trust, e.g. authentication or credit rating.
  - Executing a contract (if the contract is executable by computer)
  - Monitoring transaction execution.
  - Evaluating the monitored execution against the contract.

Each of these functions follows from the requirements that anonymous actors coordinate by performing transactions on a discrete market. To be able to do that, actors must know what is being asked and offered, they must match and negotiate a price, they must specify the results of this in a contract and then execute the transaction as contracted. Because partners may be anonymous, execution must be monitored and evaluated.

Examples of infrastructures providing some of the the above support functions are e-markets, e-auctions, e-malls, and web shops. Some actors specialize in providing generic services that can be used in all these infrastructures, such as payment or authentication. As explained earlier, we will view all of these infrastructures as markets in which activities are coordinated by price. One precondition for any of this to be efficient is the presence of a *standardized* infrastructure for e-markets. Without standardization, we would not have a market but a hierarchy with a local, business-internal "standard"; or a community with a a multitude of local habits and relation-specific ways of doing business. Either way, switching costs would be high. The hallmark of e-markets is that switching costs are low.

5.2 IT support for coordination in communities

Coordination by shared norms requires a totally different infrastructure. Actors coordinate in an ad hoc manner when the need arises, and then may not specify in advance what they are going to do. They are aware of each other's identity all the time and know what the capabilities of each actor are, and share norms of cooperation that may not all be explicitly specified. Each actor evaluates activities by these shared norms. Examples of IT support for this way of coordination are chatboxes, meeting rooms, email, shared workspaces, collaborative authoring tools, conference paper reviewing systems, videoconferencing systems, etc. Each of these systems supports one particular type of collaborative activity, such as sending email, chatting, changing a document in a shared workspace, reviewing a conference paper, etc. This leads to the following list of functions to be provided by an infrastructure for coordination by shared norms.

- **Establishing a shared context.**
  - Identifying an actor to the group.
  - Creating groups.
  - Establishing shared norms for a group. These are not all shared norms, and they may evolve, but they do determine a group identity.
  - Allowing an actor to leave a group.
  - Phasing out a group.
- **Coordinating activities.**
  - Signaling occasions that require coordinated actions.
  - Signaling that an activity took place, so that actors may decide to coordinate their own activities with it.
  - Performing joint activities.
  - Establishing a shared view of the activities to be coordinated and of their results.
  - Maintaining a log of joint activities.

All of this functionality presupposes that there is a group of people that act *in the awareness that they are a group.* This in turn presupposes mutual trust, relation-specific assets and all other community aspects mentioned earlier. The functions listed above then support coordination activities performed in this group awareness.

5.3 IT support for coordination in hierarchies

Turning to coordination by management, we find the traditional information system functions described eloquently by Galbraith [8]. In a hierarchy that operates in a totally predictable environment, all processes can be specified in advance and workers coordinate their activities according to fixed process descriptions. When unpredictability increases, the number of exceptions to be referred to management increases and some coordination mechanism other than process specification must be sought. One of these mechanisms is the creation of communities of actors that coordinate their activities in an ad hoc manner, as described above. Another mechanism is the introduction of information systems that

allow actors lower in the hierarchy to make decisions themselves, thus reducing the number of decisions to be made by higher management; or by introducing information systems that bring information directly to the point of decision-making, speeding up the decision-making process in the face of exceptions.

In the 1980s, an additional infrastructure to improve coordination within a hierarchy arose, namely workflow management technology. Using workflow management systems, managers can specify more complex processes and can monitor the progress of cases, employees can coordinate their tasks more efficiently, and workloads can be managed more efficiently. The IT support functionality required in hierarchies is accordingly the following:

- **Data management support.** This includes operational information systems that collect, store and distribute operational data, as well as of management information systems such as data warehouses.
- **Process management support.** This includes workflow management, case handling and document management.

Note that here we have the traditional view of information systems as support for coordination by management in hierarchies [7].

5.4 The impact of IT on coordination

The Internet provides us with two features not previously present in IT support, namely relatively cheap monitoring capability and a relatively cheap standardized communication infrastructure. As explained in the next few paragraphs, this makes new coordination mechanism possible.

Maladaptation. One of the costs of coordinating by price is that a partner may not perform as promised, or that a contract we struck turns out not to be as profitable for us as expected [38]. This causes a cost whichever way we turn:

- Continuing the agreement as it is currently performed is costly.
- However, haggling out a solution is costly too. It uses resources that we could have spent in a more efficient way.
- If haggling does not yield a satisfactory answer then we must either take out loss or invoke an arbiter, which comes with a cost too.
- To back up our actions in case of maladaptation, we need a governance system consisting of regulators, a legal force and arbiters. Running such a governance system is costly too.

The sum of all these costs are called *maladaptation costs* by Williamson. We cannot avoid all of these maladaptation costs but we can reduce some of them by monitoring. Current communication technology allows us to monitor a partner during execution of an agreement, which reduces the risk of malperformance in the first place. We can also ask a third party to monitor both and we can collect performance data, which allows us to select reliable partners in the long run. All of this makes the market more efficient.

Outsourcing. Improved monitoring capability also reduces the risk of outsourcing activities to a small number of partners. Outsourcing can be efficient because the organizations to which activities are outsourced, can use economies of scale and of specialization to achieve greater efficiency. In addition, outsourcing to a smaller number of long-term partners allows increased product complexity, without increasing search cost. However, this requires a long-term investment in relation-specific assets and this in turn creates the risk of opportunistic behavior of long-term partners, who may leak information to competitors or who may not produce the best possible product because they are assured of a customer anyway. Traditionally, these were reasons for a business to integrate these long-term partners as business units in its own hierarchy. Improved monitoring reduces these risks and takes away the need for hierarchical integration [4].

The fundamental transformation. An actor may lose bargaining power simply by executing a transaction. For example, a supplier may have made large investments to do business with a particular customer, and if these investments are not usable for other customers, the supplier is stuck to this customer and has lost bargaining power. Or a business may have built up detailed knowledge of a partner, that can be only acquired by years of interaction, making it costly to switch to another partner that does not have this knowledge. Simply by doing business, a situation of large numbers, in which an actor can negotiate with a large number of other actors that are more or less interchangeable, is effectively reduced to a situation of small numbers, in which bargaining is nearly impossible. Williamson [38, page 61] calls this the *fundamental transformation.*

The classical answer to this risk to vertically integrate with the partner, preventing this partner to switch to other customers. This replaces the cost of transactions by the cost of hierarchy. Standardization is however another way to avoid this risk, without resorting to hierarchical integration. The more interactions with business partners are based upon standard hardware and software, the more interchangeable these partners remain and the less a situation of large numbers is transformed into a situation of small numbers. Determination of *de facto* standards is one of the major rules of the information economy [31].

6 A Inventory of Coordination Mechanisms

Reduced transaction costs, increased communication speed and connectivity, risk reduction by monitoring and standardization all facilitate the rise of new forms of organization networks. In figure 3 we extend our simple three-way classification of coordination mechanisms by including a number of novel coordination forms that have been facilitate or at least stimulated by IT.

Figure 3 contains the three coordination infrastructures identified so far, namely hierarchies, markets and communities. For each of these, there are many variants, one of which has been listed in the table.

The simplest variation is that between **external markets,** which are discrete markets of actors that are not hierarchically related, and **internal markets,** which are markets within a hierarchy. An example is the Swedish firm ABB, in which business units trade products against market prices [24]. Because these business units are part of one hierarchy, mutual trust is higher than in external markets. Business units can also do business with external partners.

	Generalized hierarchy		Generalized market		Generalized community	
Coordination mechanism	Hierarchy	Star network	External market	Internal market	Value network	Group
Management authority	X	X				
Price			X	X		
Shared norms					X	X
Switching cost	high		low	low	high	high
Trust	high	high	low	high		high
Specifyability of activities			high	high		low

Figure 3: A framework for coordination support. The columns indicate coordination infrastructures, the rows indicate coordination mechanisms. A cross in a cell of the upper three rows means that this is the dominant coordination mechanism for that infrastructure. An empty cell in the lower three rows means that this variable can have any value for this infrastructure.

A more complicated variation is the **star network,** which is a community of actors managed by a central authority. This is a community because it is a relational structure, but coordinated by a hierarchical management structure. Here are some examples.

- Networks of professional companies may be organized by a central coordinator on a project basis. For example, in the construction industry, a central construction firm coordinates many subcontractors in a building project. In the publishing industry, a publishing house coordinates a large number of authors, agents and editors. In the film and recording industry, companies do the same with artists, directors, camera persons, etc. [29]. These communities consist of independent companies that have other activities besides contributing to this community, but they are managed by a central company, at least for the duration of the project. Participants in the star network are valued because of their professional abilities. However, usually actors are not so unique that cannot be replaced by another—a market-like property. On the other hand, mutual trust is high—a community property. The activities to be performed may be fully standardized, which would help to reduce switching cost, but in many cases the project is too chaotic for that and coordination is not done by contract but by on-the-spot negotiation. Oversight is maintained by central management, which makes it a hierarchy.

- Donnelley and Sons is a communication services company that manages educational book production on behalf of publishers. For complex projects they ask for bids from graphics companies on behalf of their customer and do this for many projects for many customers in parallel. This is done using a web-based IT infrastructure that keeps tracks of projects, customers, bids, negotiations, deadlines and agreements [34]. This is an example of an IT-facilitated star network in the publishing industry.

- IKEA has mobilized suppliers world-wide to produce components for the furniture that it designs and sells [26]. The furniture is transported and assembled by its customers. IKEA offers to assist its customers in doing so but this too is outsourced to independent companies and customers have the choice performing these activities themselves. IKEA also helps its suppliers to bring their production process to high standards, by means of technical assistance, leased equipment, and advice. It will also help them find raw materials and introduce them to new business partners, which will bring financial health to the supplier and ensures a steady high quality influx of component from the supplier. IKEA coordinates this network through an IT infrastructure in which each cash register in any of its stores updates warehouse information system as well as operational headquarters in Sweden in real time. This star network is based on mutual trust, but any shared norms for processes and products are installed by IKEA as in any management hierarchy. Products are precisely specified and processes are specified in advance. Switching costs for IKEA would be high due to investments in suppliers, but this is balanced by mutual trust and the creation of a win-win situation for all actors in the network.

- Nike designs and markets sports shoes and coordinates a large number of suppliers in the Far East that produce these shoes according to specification [24]. These suppliers also work for other designers so that their technical competence is enhanced. This reduces switching costs for all parties concerned. Cooperation is on a permanent basis but Nike will not hesitate to switch to another supplier when this is cheaper–which is a market property. Mutual trust is high, a property of communities and hierarchies, and specifyability of activities is very high—a market property.

- Dell and Cisco outsource production activities to others and have implemented low transaction costs for their customers and low switching costs with respect to their suppliers. If they find a cheaper supplier, they will select it [33]. They are virtual organizations, which is a limiting case of a star network. A **virtual organization** is an organization that specified its business processes independently from any resources that perform these processes, and dynamically switches to other resources when this is more efficient [25]. In the extreme case, we are back to the single actor in a discrete market, that delegates all production activities to others and who creates value by coordinating the work of others. It is not the same as a market, because a resource has relationship to the virtual organization, at least till the end of the process for which it is contracted.

Some of these communities exist without using IT but most of them could only exist using an IT infrastructure. They take advantage of reduced transaction costs, increased monitoring capacity, and standardization. Typical functionalities are collaboration support, negotiation support, and tracking of interactions. These functionalities are absent from the traditional IT support for hierarchies listed earlier in section 5.3. Requirements engineering for such infrastructures should take the characteristics of the coordination mechanism into account.

Moving to the right-hand side of our framework, we encounter **groups,** which are relational communities as defined earlier in this report. We call them groups here to emphasize that they are communities of *people,* not of businesses or software. Here are some examples.

- Kinship networks coordinating a family meeting;

- Geographically distributed programmers working on a GNU software product;
- Engineers in one or more companies collaboratively working out a new product design;
- Professionals in competing companies discussing technical topics of mutual interest;
- Scientists organizing a conference;
- Insurance experts in a large company discussing hard cases;

In all these cases, activities are roughly known in advance but not specifyable in detail. Coordination is by shared norms determined by tradition, the profession, shared education, etc. Members of the group trust each other and people who participate do this as unique individuals. Because each participant contributes to the community, the value of participation grows the more people participate in the community. Community connectivity is high: Each member is connected to each other member.

In an institutional variant, it is organizations who form a community. A **value network** is a set of companies that jointly responds to a need that none of them can solve on its own, Coordination is however not done by management hierarchy but by local adjustments and negotiations. Here are some examples.

- In the South of Germany, small and medium-size firms with a specialty in some aspect of textile manufacturing cooperate to produce textiles. The network includes small craft industry as well as research institutes, vocational training centers, consulting firms, and marketing agencies [29]. The more specialized a firm is in this network, the more it depends for its success upon the success of other firms. Key technologies are developed in a collaborative manner. The network is structured to strengthen the social structure in which textile firms are embedded and to encourage cooperative relationships that attenuate the destructive aspects of competition. There is no central management hierarchy such as in star networks. Coordination is done by shared norms. Switching costs are high but this is offset by mutual trust.
- Fraud control in the Netherlands involves hundreds of organizations, including the national police force, regional police forces, the auditing department of the ministry of economic affairs, the auditing department of the ministry of agricultural affairs, the tax department, the fiscal intelligence service, the social service, and many others. When a suspected case of fraud is investigated, it is unpredictable which of these organizations will be involved, information is difficult to get and in general activities are unstructured. To facilitate investigations, an interorganizational information system has been implemented containing indexes to information present in these different organizations [10]. There is no central authority in all these organizations and products as well as procedures are often ill-defined and changeable. Organizations collaborate by an extensive system of ad hoc agreements and informal evaluations. Switching costs are high—each actor in the network is unique. Each actor depends upon others for its success, but at the same time each actor tends to optimize its decisions with respect to its own goal rather than with respect to an overall network goal.
- The port of Rotterdam contains a network that connects transporters, ship owners, ship brokers, stevedores, insurance companies, the customs department and many other parties that all work together to respond to a shipping request by shipping the

goods to the desired destination. Intermediate products and communication processes are well-defined. It is not predictable in advance for each case what activities will be performed, and there is no central management hierarchy. Nevertheless, it is in the interest of each actor to participate in the network. This is an example of a **value chain,** which is a network of companies coordinated by mutual agreements, in which processes and products are well defined from customer demand to customer response. A value chain may be a simple pipeline, but it may in general have any topology of cooperating actors. A crucial feature is that the process from demand to response is well-defined.

In these communities, switching costs are high because each actor has its own individual contribution. For each actor, it is not easy to find another actor that can provide the same services. Mutual trust may be high or may be virtually absent. For example, organizations in the legal value network may be fighting each other to defend their own right of existence and their own budget, but at the same time they need each other to do their job. Decisions are made according to the garbage can model in which problems, solutions, actors and decision occasions meet at particular points in time governed by the social context, social structure and the importance of decisions [6]. The outcome is at most boundedly rational. To the extent that mutual interest is understood or there is some form of central coordination, as in the port of Rotterdam, decisions are made in a more rational manner.

Specifyability of activities to be coordinated may be low or high. In regional economies, activities may be subcontracted so they are specifyable. Informal social contacts in the community will take care of the part not specified in advance. In other situations, such as fraud investigations, specifyability of activities is extremely low and in others, such as in the port of Rotterdam, it is very high.

These examples should make clear that there are many intermediate and mixed forms of coordination in addition to those listed in figure 3. However, figure 3 does list a number of mechanism and variables that can be used to understand any given coordination mechanism. It can serve as a starting point for requirements engineering for IT support, discussed next.

7 Requirements Engineering for Coordination Support Technology

In any given coordination mechanism, one will find a mixture of the basic mechanisms discussed above. To analyze requirements for IT support of coordination, the analyst should consider the different mechanisms as aspects of one situation and analyze each of these aspects in isolation, combining the results later.

We view requirements engineering as the problem analysis task in a practical problem-solving cycle, where problem-analysis is *theory-building.* A practical problem is a difference between what we perceive a situation to be and what we would like it to be, and to analyze a practical problem we must build a theory of the problem domain, which provides answers to several questions.

- What are the *phenomena* of the problem domain? Which events occur, how often do they occur, what entities are affected by the events?
- What are the *regularities* ("laws of nature") according to which the domain evolves?

These regularities must be used to motivate any solution that we come up with. A solution is introduced in the problem domain because it is expected to cause a certain effect; and this causation is backed up by regularities, i.e. by the laws of the problem domain. If the problem domain is physical, then the regularities are the laws of physics. If it is social, then the regularities include generalizations about the behavior of people, business procedures, laws issued by governments, etc.

- What are the *values* that make the phenomena problematic? The value of a phenomenon is its utility, expressed in quantitative or qualitative terms. Each problem domain has a value system, which is a set of values that indicate what is good and what is bad, what has a positive and what a negative utility [11]. To say that a phenomenon is problematic is to say that according to the value system of the problem domain, it has a negative value.

- Who are the *stakeholders* that have these values? This introduces the possibility that different stakeholders have different values.

- What are the *goals* of these stakeholders with respect to the problem domain? Goals are desired situations. Again, different stakeholders may have different goals.

- What are the *obstacles* that prevent the stakeholders to achieve these goals? Any solution must do something about these obstacles.

- What are the *constraints* within which a solution must be achieved?

The answers to these questions guide the problem-solver to specify a solution, analyze it, and evaluate the specified solution. In this report, **product requirements** are desired solution properties specified by the problem-solver. **Domain requirements** on the other hand are the goals of the stakeholders with respect to the problem domain. Requirements for coordination support are thus desired properties of products that support coordination. But these product requirements are motivated by domain requirements, which are the desired coordination mechanisms to be realized in the domain.

The analysis in the following sections merely scratches the surface of a complex topic. The analyses should be viewed as an indication of what can be done and as a programme for further research.

7.1 Requirements engineering for supporting coordination by management

The problem domain is here a hierarchy within a single business or across businesses (a star network). For each of the domain aspects, there are some generic questions to ask.

- *Phenomena.* This includes what people in the hierarchy actually do: Tasks performed, services provided to customers, markets, customers, distribution channels, information used and produced, the meaning that this information actually has, exceptions that actually occur, they way people respond to this, volume of data handled and frequency of events, etc. For a star network, this includes the actual network structure and the phenomenology of communications between members of the network.

- *Regularities.* This includes centrally defined work procedures (that do not need to coincide with what is actually done) and exception-handling rules, formal dictionaries in use, formal agreements between members of a star network about communication and logistics, communication standards in use, etc.
- *Values.* This includes the desires of stakeholders that makes the phenomena problematic, and the priorities and urgency that makes it important to solve certain problems before others.
- *Stakeholders.* This includes the business structure, relevant business units, reporting and accountability structure, etc. For star networks, it includes identifying the business actors and their role in the network.
- *Goals.* In a hierarchy, this includes the business mission, business strategies, targets defined by management to improve the situation, opportunities to be realized, etc. In a star network, it includes the goals of all business actors
- *Constraints.* This includes the business strategy, constraints arising from law and labor relations, etc.

Classical requirements engineering methods from ISAC [17, 18] to Information Engineering [22] approach information systems requirements through one or more of these aspects of the problem situation. Requirements engineering for business process support usually focuses on methods for process modeling [13, 28].

7.2 Requirements engineering for supporting coordination by price

Requirements engineering in this case is oriented toward finding out how products are exchanged.

- *Phenomena.* These include processes for matching buyers and sellers and for executing the transaction as they currently take place, volume and frequency of trade, etc.
- *Regularities.* These include regularities in consumer behavior found by marketing people, trade procedures, legal requirements, etc.
- *Values.* This includes the distribution of costs and benefits deemed desirable by each stakeholder. Different stakeholders may have a very different view of this.
- *Stakeholders.* This is a description of the business actors and the means of transport and communication that connect them.
- *Goals.* These can be represented by a value model [9] that represents the flow of monetary value in transactions in the desired situation.
- *Constraints.* This includes constraints arising from technology, constraints imposed by law and international relations, etc.

Coordination by price may be combined with coordination by management in a star network and with coordination by shared norms in a value chain. Nevertheless, to understand the problem domain, the requirements engineer should try to understand these different aspects separately before building an overall theory of the problem domain.

7.3 Requirements engineering for supporting coordination by shared norms

In this case we know that there are unstated shared norms that govern coordination. Problem analysis must identify these shared norms.

- *Phenomena.* This includes the activities to be coordinated, their frequency, actors involved, etc. In a value chain, these activities can be described completely.
- *Regularities.* This includes shared norms that govern coordination. These may be rules for behavior in particular circumstances, guidelines for performing an activity, rules for involving certain actors in certain activities, and for involving them in a certain precedence order, the semantic codes used to interpret the meaning activities, etc. In a value chain, it includes a description of the business process performed by the actors in the chain.
- *Values.* This includes the traditional values in this community, burdens and benefits shared by group members, professional values etc. that make certain phenomena problematic. In groups, there may be performance ambiguity, so it may be hard to get an unambiguous description of the value system; the value system itself is likely to be ambiguous.
- *Stakeholders.* This includes a map of all actors and the roles they play. In a value chain, it includes a description of the actors that perform tasks in a chain and of their role in the value-adding process.
- *Goals.* This includes goals shared by the group. If there is no consensus, it is important to find out which actor supports which goal.
- *Constraints.* Norms and values imply constraints on possible solutions. Other sources of constraints are the law and tradition.

Since in a group, activities cannot be specified in advance, and in all communities, goals may not be clear but problems are, it is likely that support for coordination by shared norms is implemented in an evolutionary manner.

8 Conclusions

In this report we analyzed three groups of coordination mechanisms for which IT support can be provided. These mechanisms mix transactional and relational coordination to various degrees. IT support for these mechanisms facilitates the introduction of novel coordination mechanisms. We defined a framework for coordination within which all these mechanisms can be described and used this framework as a basis for requirements engineering approaches. The three basic coordination mechanisms are three top-level problem classes for which different IT solutions can be sought, and the three RE approaches are ways to analyze these three domains.

In practice, the coordination mechanisms will occur in mixed forms, so that in practice, we will have to combine the different RE approaches. However, we claim that the omission of an approach where this coordination mechanism is present is a recipe for failure. If we

would use a classical RE approach for management hierarchies to markets or communities, we would miss important aspects of the problem domain, which would lead to inappropriate IT support.

In future research, we will investigate case studies and perform action research to analyze and further elaborate the requirements engineering approaches for supporting coordination price-based and norm-based coordination.

References

[1] S. Angelov and P. Grefen. B2B e-Contact Handling –A Survey of Projects, Papers and Standards. Technical Report 01-21, CTIT, 2001.

[2] A. Baier. Trust and antitrust. *Ethics*, pages 231–260, January 1986.

[3] Y. Bakos. The emerging role of electronic marketplaces on the internet. *Communications of the ACM*, 41(8):35–42, August 1998.

[4] E.K. Clemons, S. P. Reddi, and M.C. Rows. The impact of information technolgy on the organization of economic activity: The "move to the middle" hypothesis. *Journal of Management Information Systems*, 10(2):9–35, Fall 1993.

[5] R.H. Coase. The nature of the firm. *Economica, new series*, pages 386–405, November 1937.

[6] M.D. Cohen, J.G. March, and J.P. Olsen. A garbage can model of organizational choice. In J.G. March and R. Weissinger-Baylon, editors, *Ambiguity and Command: Organizational Perspectives on Military decision Making*, pages 311–336. Pitman, 1986.

[7] G.B. Davis and M.H. Olson. *Management Information Systems: Conceptual Foundations, Structure, and Development*. McGraw-Hill, 2nd edition, 1985.

[8] J. Galbraith. *Designing Complex Organizations*. Addison-Wesley, 1973.

[9] J. Gordijn and H. Akkermans. Designing and evaluating e-business models. *IEEE Intelligent Systems*, 16(4):11–17, 2001.

[10] J.H.A.M. Grijpink. *Keteninformatisering: Met Toepassing op de Justitiële Bedrijfsketen*. SDU Uitgevers Juridisch en Fiscaal, 1997. Ph.D. Thesis, Technische Universiteit Eindhoven.

[11] A.D. Hall. *A Methodology for Systems Engineering*. Van Nostrand, 1962.

[12] M.A. Jackson. *Problem Frames: Analysing and Structuring Software Development Problems*. Addison-Wesley, 2000.

[13] M.A. Jackson and G. Twaddle. *Business Process Implementation*. Addison-Wesley and ACM Press, 1997.

[14] E. Jacques. In praise of hierarchy. *Harvard Business Review*, pages 127–133, January-february 1990.

[15] S. Kaplan and M. Sawhney. E-hubs: The new B2B marketplaces. *Harvard Business review*, pages 97–103, May-June 2000.

[16] G. Kotonya and I. Sommervile. *Requirements Engineering: Processes and Techniques*. Wiley, 1998.

[17] M. Lundeberg, G. Goldkuhl, and A. Nilsson. A systematic approach to information systems development - I. Introduction. *Information Systems*, 4:1–12, 1979.

[18] M. Lundeberg, G. Goldkuhl, and A. Nilsson. A systematic approach to information systems development - II. Problem and data oriented methodology. *Information Systems*, 4:93–118, 1979.

[19] I.R. Macneil. The many futures of contracts. *Southern California Law Review*, 47:691–816, 1974.

[20] I.R. Macneil. Contracts: Adjustment of long-term economic relations under classical, neoclassical and relational contract law. *Northwestern University Law Review*, 72:854–905, 1978.

[21] T.W. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, 1994.

[22] J. Martin. *Information Engineering*. Prentice-Hall, 1989. Three volumes.

[23] Merriam-Webster Inc. *Webster's ninth New Collegiate Dictionary*, 1988.

[24] R.E. Miles and C.C. Snow. Causes of failure in network organizations. *California Management review*, 34(4):53–72, 1992.

[25] A. Mowshowitz. Virtual organization. *Communications of the ACM*, 40(9):30–37, September 1997.

[26] R. Normann and R. Ramírez. From value chain to value constellation: Designing interactive strategy. *Harvard Business Review*, pages 65–77, July–August 1993.

[27] W.G. Ouchi. Markets, bureaucracies, and clans. *Administrative Science Quarterly*, 25:129–141, March 1980.

[28] M.A. Ould. *Business Processes: Modelling and Analysis for Rengineering and Improvement*. Wiley, 1995.

[29] W.W. Powell. Neither market nor hierarchy: Network forms of organization. *Research in Organizational Behavior*, 12:295–336, 1990.

[30] M.C. Sawyer. The nature and role of the market. In C. Pitellis, editor, *Transaction Costs, Markets, and Hierarchies*, pages 20–40. Blackwell, 1993.

[31] C. Shapiro and H.R. Varian. *Information Rules. A Stretegic Guide to the Network Economy*. Harvard Business School Press, 1999.

[32] C.J. Smith. *Synonyms Discriminated*. G. Bell and Sons, Ltd., 1926.

[33] S. Tully. The modular corporation. *Fortune*, pages 52–56, February 1993.

[34] B. Welty and I. Becerra-Fernandez. Managing trust and commitment in collaborative supply chain relationships. *Communications of the ACM*, 44(6):67–73, June 2001.

[35] R.J. Wieringa. *Requirements Engineering: Frameworks for Understanding*. Wiley, 1996.

[36] R.J. Wieringa. *Design Methods for Software Systems: Yourdon, Statemate and the UML*. Morgan Kaufmann, In press.

[37] O.E. Williamson. *Markets and Hierarchies: Analysis and Antitrust Implications*. The Free Press, 1975.

[38] O.E. Williamson. *The Economic Institutions of Capitalism*. The Free Press, 1985.