Communication
in the
Building Industry

a strategy for implementing electronic information exchange

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van
de Rector Magnificus, prof.dr. J.H. van Lint, voor
een commissie aangewezen door het College van
Dekanen in het openbaar te verdedigen op
woensdag 17 januari 1996 om 16.00 uur

door

Bauke de Vries

geboren te Velp

Acknowledgements

The research that is presented in this thesis was carried out during a period of five years. In this period I spent about half my time on the research, the other half I was occupied by my employment as automation manager of the Department of Architecture and Building at Eindhoven University of Technology. Hereby I would like to thank both the department and the computing centre of the university (which is my employer) for giving me the opportunity of doing a Ph.D. I am convinced that the knowledge I gained by my research about Information Technology will prove to be very useful in my function as automation manager as well. However, I have refrained from several tasks that I consider to be my responsibility the last five years. An important part has been taken over by Erik te Nijenhuis whom I wish to thank very much for this. The other part I hope to catch up very quickly.

Two companies of the Dutch building industry greatly contributed to my research, namely Archirectenbureau Van den Pauwert and BetonSon. By being allowed to have a close look at ‘real’ building projects, I was able to verify my ideas and my results. Hopefully they profited by my knowledge as well.

I would like to thank Lou Somers from the department of Mathematics and Computing Science for spending so much time trying to understand all the peculiarities of the building industry. Although we originate from completely different backgrounds, I had a very pleasant time discussing IT matters. I can at least say my knowledge about formal specification methods was greatly improved in the past few years. I thank my supervisor Harry Wagter, who prevented me from drifting too far away from building practice into abstract models.

Finally, I thank my parents for their faith in me and I thank my wife Sylvia who allowed me to spend many evenings on my thesis instead of on my family. Although I know I can never make up for this loss, I promise her to take great care of her and my children in the future.

Communication in the building industry

Contents

1 Introduction

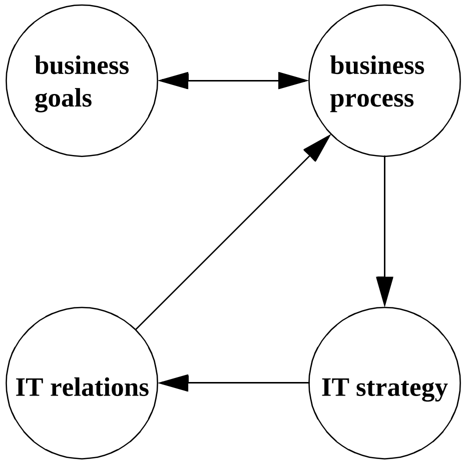
Information Technology in general and electronic communication in particular influence organizational structures. New communication media will change communication processes and business processes. To be able to analyze the influences of the new communication media, a clear view is required of the information flow and the information contents during a building project. Given this view, the question can be answered whether the business process and business organization fit the actual information needs. From the information flow frequency and the information contents, the most appropriate storage structure and transfer medium can be determined. The goal of this thesis is to create a formal description of the information exchange process during a building project, to provide the clear view as stated above.

1.1 Background

‘By linking actual operations across company lines, by making it possible for companies to compete in fields once regarded as alien, extra-intelligent networks break up the old specialization, the old institutional division of labor.’ ‘Its growth is causing new struggles for the control of knowledge and communication, struggles that are shifting power among people, companies, industries, sectors, and countries.’ [Toffler 1990]

Information Technology has become of great influence on organizational structures. Present organizational structures have their foundation in the industrial revolution. The dominating structure has a layered pyramidical form. A hierarchy of functions that are carried out by supervisors and workers, is well suited for manufacturing activities which are broken down into small tasks. Information Technology stands for a range of new technologies such as: electronic computing, electronic design, electronic communication and electronic archiving. After the introduction of IT in the nineteen sixties, the first implementations mainly took place in electronic computing and archiving. The pyramidical organizational structure can still be found in many companies. In today’s companies there are four main factors that influence the organizational structure: the business goals, the business process, the IT strategy and the IT relations [Wagter 1995].

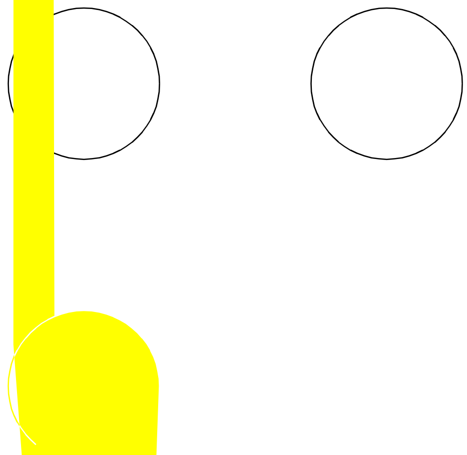
FIGURE 1. Traditional organizational influences



IT strategy needs to be based on business goals and the required business processes. Traditionally IT strategy is fully determined by the business process. IT has invoked new relationships between companies. Short command lines between process parts allow for decentralization of organizations. In the organization a balance must be found between overall guidance and decentralized responsibilities.

IT allows for process integration, because information becomes available as soon as it is generated and to everyone that is authorized. On the other hand, customer demands lead to more complex products in greater varieties, which implies adding more functions in the organizational pyramid. The new possibilities of IT and the stronger customer demands have raised questions about the traditional organizational structure.

FIGURE 2. New organizational influences



A new stage becomes a[illegible] when IT strategy directly influences the business goals. IT is an enabler [illegible] business goals when new products can be introduced on the mark[illegible] using IT relations, but as a result of IT or as a part of IT [Hammer 19[illegible]

Electronic communicati[illegible] an organization and between organizations as a part of the IT strategy [illegible]ge:

- The speed of information exchange.
- The frequency of information exchange.
- The contents of the information.
- The information flow.
- The business products.
- The business process.
- The relations with other organizations.

Communication takes place between organizations and between people that work in those organizations. Electronic communication can take place between computer systems, partly or totally replacing human communication. It does not seem very strange that there are similarities between communication between computers and between humans. In the next two sections we will take a quick look at the literature on this subject.

1.2 Human communication

Cherry gives the following definitions of human communication: “The establishment of a social unit from individuals, by the use of language or signs”, and “The sharing of common set of rules, for various goal-seeking activities” [Cherry 1966].

The first definition shows us that communication isolates a group of people from the rest because of the fact that they can understand each other. The second definition says that a set of rules is used to ensure a certain behavior of the individuals. The result of the communication is the start of an activity which evokes a new situation.

Signs can have different forms such as: speech, writing, pictures, drawings etc. The theory of signs [Carnap 1939] distinguishes three levels of abstraction:

1. Syntactic: signs and there relations to other signs.
2. Semantic: signs and their relations to the “outside world”: things, actions, relationships, qualities (designation).
3. Pragmatic: signs and their relations to users (psychology).

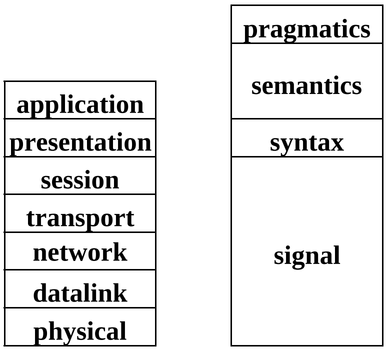
Language and signs are tools used to express reality. Syntax and semantics concern the rules that must be obeyed by the users. Pragmatics concerns the perception, recognition and interpretation of information by the individual depending upon his or her experiences.

Communication is set up to exchange information. One of the dictionary definitions of information is: “Communication of instructive knowledge”. Here we find that the information encloses knowledge and that it instructs the receiver to do or to learn something. Reasoning and calculation use data to produce new knowledge. Data are the facts that are taken for granted by its users.

1.3 Communication technology

A well known architecture for electronic communication is the Open Systems Interconnection model of the International Standardization Organization [ISO 1984]. This architecture consists of seven layers. Each layer is defined by a set of functions. A layer only has interfaces with its lower layer and its upper layer. A layer uses functions from the lower layer and supplies functions to the upper layer. Comparing the OSI architecture with human communication results in the following diagram:

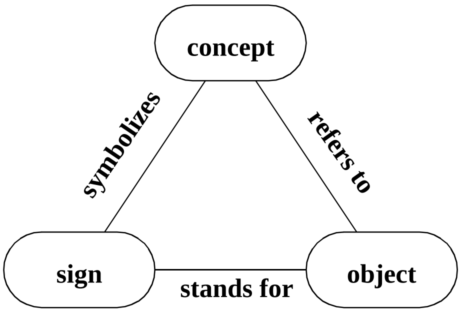
FIGURE 3. OSI architecture compared with theory of signs



To draw a comparison with the ISO architecture, the theory of signs has been extended with an extra layer: signal. Signals are the sound that reaches the ear and the light the reaches the eye. The functions of the presentation layer are identical to the function of syntax in speech and documents. Semantics can be implemented in the application. Additional semantics can be added by the human that uses the application. Evidently pragmatics is outside the OSI architecture.

A concept from database modelling theory is the distinction between the conceptual schema and the information base [ISO 1985]. Conceptual schemata and information bases are used to model the reality we want to communicate. The conceptual schema is an abstraction of the information base. Concepts are abstractions of physical or abstract objects in reality. Information is an expression of reality, expressed by means of signs (as stated in the previous paragraph). Concept, sign and object are related in ‘the meaning triangle’ [Sowa 1984].

FIGURE 4. The concept, sign, object - triangle



1.4 Communication in the building industry

In the building industry a great variety of signs is used to express the design or the product and to express how we wish the reader of the signs to act. A choice for a specific communication method depends on the goal the user wants to reach. Examples of various methods and goals are:

- A cardboard model for experiencing space in building elements and between building elements.
- A perspective drawing creating an image of the product from a certain viewpoint.
- An extensive document describing a building product in every detail.
- A video showing a realistic image of the building product from various viewpoints.

In the design process, form representation and form interpretation play an important role. In most cases the form that is presented is an abstraction of reality and it will hide information that is not yet known or not yet regarded relevant. A great part of an architect's education consists of learning to understand and to use these abstraction mechanisms. The correct interpretation of building design is a skill that greatly depends on the experience of the 'reader' of the design. The abstraction mechanism varies with the problem that is to be solved. For that reason an architect uses other signs to describe the design than a building contractor. Sign transformation is part of the interpretation process. It not only means replacing one sign with another but also combining signs and splitting signs into new ones. Of course education serves as a common background necessary to bridge the gap between the disciplines.

Forms are materialized during the design process. Building objects are often referenced by their form (e.g., layer) or by their function (e.g., isolation) and sometimes by their behavior (e.g., buckling). In the design stage form is the primary approach, function and behavior are often attached to the form of an object. In the building stage building objects are referenced by their material aspects and the way they are constructed. After realization of the building during the maintenance stage, building objects are referenced by their function and material aspects and sometimes by their behavior.

Buildings are usually realized by a group of building companies. The complete task of creating a new building product is decomposed into subtasks. Each company has a view about the contribution of the other companies. To take care of the continuation of the building process, one of the companies takes the responsibility so that activities are executed in a proper way. In the design and building process companies communicate with each other as long as the building project lasts.

1.5 Communication problems

Communication changes rapidly along with the introduction of new be made explicit to allow for the control of concurrently running processes. A phenomenon that is very special for the building industry is that the group of companies changes with every new project. This implies that agreements about communication only last during one project.

If processes are running concurrently special precautions must be taken to be sure that no one is using information that is not up-to-date. Much time is lost when activities have to be redone because new information is received while working on the job. If changes are made afterwards without restarting the activity there is a great risk of not taking all the consequences of the changes into account.

There is of course a global understanding about how to communicate, but the list of the communication media has been extended with new ones such as telefax, floppy disk, tape, electronic network, etc. It should be possible to chose an option from the available list that makes the best fit to the demands at a certain time. Communication media are often entangled with communication protocols. To allow flexible use of new media there should be a clear distinction between them.

A complicating factor every design discipline faces is that the subject of communication is never the same. Every building project is unique and within a project the design and construction evolve in time. Therefore, project independent agreements can not contain very detailed information. The project dependent information is specific for the building object at hand. Still there is the need to have a common understanding about the building object at every point in time to be able to communicate about it.

Communication between building companies and internal company communication are based on different data structures. Companies explicitly or implicitly use a data structure to represent the information that has been received from other companies or that has been produced by themselves. A data structure reflects the way information is ordered to serve a specific goal such as designing or production. This implies that every company has a unique data structure since each of them will try to offer a unique set of services and/or products. For that reason mapping of information is unavoidable. The message that is exchanged must be understandable by both the sender and the receiver so it will contain a set of definitions that they both agree upon. Likewise language leaves room for different interpretations, it takes a lot of effort to find the common language that will serve only one interpretation. After that, mapping rules will take care of the translation into the foreign data structure. It is evident that the less mapping rules that are necessary the easier the communication can be established and maintained.

1.6 Aim of the research and outline of the thesis

Looking at the communication problems there is a need to describe the communication process itself in an explicit way. This is necessary to get a clear view about what the actual information is that is required to have at one’s disposal at every stage during the building project. The description must create a common background for all companies involved in a certain building project. The common background should be well defined and should provide a unified interpretation. The communication description is project dependent so as to incorporate the demands of every unique building project. As a result of the process description the information exchanged becomes readily available. Both the process and the information description should offer enough criteria to make a choice between the communication media that can be used. Since communication media change in time the descriptions should not be communication medium dependent. Thus, the aim of the research is as follows:

> *Create a formal description of the information exchange process during a building project.*

The process description should allow activities to take place concurrently and to analyze their interdependency. This demand increases the complexity of the description enormously. A formal description language is necessary to make the control of the flow of information possible. The main concern is to prevent the use of outdated information at any point during the process. If the information flow is controllable then changes in activity requirements can be studied to see what the influence is on the communication process and on the building process.

The structure of the thesis is as follows. Section 1 discusses the background and the aim of the research. Section 2 gives an overview of the communication media and their characteristics as they are used nowadays in a building project. In Section 3 the boundaries of the research area are defined. Within these boundaries a selection of other research activities are analyzed. Section 4 deals with general aspects of communication process modelling and the special aspects of the information exchange process during a building project. As with the previous section, Section 5 deals with general aspects of information modelling and the special aspects of the information being exchanged during a building project. In Section 6 the results of the process analyses and the

2 Communication media

People that are employed in the building industry have a long tradition in using text and drawings to express their thoughts. A logical first step to get more insight is to analyze the current way of communication. Although there are more communication media such as the card board model and the video, the research is focused on current text documents and drawings. The explanation for this choice is that these two are often used as contractual documents, so the information that is described should meet a certain quality level. More precisely we will look at which information is represented and how it is structured. Meanwhile, an explanation is given about which representation form is chosen under a given circumstance and for what purpose.

2.1 Drawings

Special techniques and conventions are used when making a drawing. The first problem that has to be conquered when starting to draw on paper is how to represent a three dimensional object on a two dimensional medium. Many techniques have been invented to help solve this problem, such as cross-sections and perspective views. Still it takes a lot of interpretation effort to create the 3D model from the 2D information. It is up to the designer to be aware of all possible intersections of the objects in space. Complex connections are often worked out in more detail to see if the design is practicable. The meaning of the drawn entities is fully defined in the common background, that is the education of the designer or engineer, while ‘reading’ the drawing. Symbols on drawings are used by every discipline, but each of them has its own conventions. Sometimes text is added to the drawing to give a more detailed description. Because of the density of the drawing information, designs are often split into two or more drawings. Each of these drawings usually contains information for a specific purpose (e.g., foundation) [SBR 1990].

Most CAD systems are drawing systems with extra features. When CAD systems were introduced to create drawings, the drawing techniques did not change except that it was done on a screen. Soon the extra possibility of attaching additional information to the CAD drawing was introduced. A very popular way to divide or structure information is the use of layers with a layer name that expresses something about the contents. Lots of mostly national committees have worked on defining useful layer names so that these names could serve as a parameter for other application programs. CAD systems facilitate the making of library elements with unique names. A library element is in fact a small drawing. Such a library element can be referenced from different points in the main drawing. This facility is mainly used as a drawing feature to enhance drawing speed. It is rarely used as a feature to create building objects with a unique identification so that they can be referenced individually by other applications. Building object libraries can be developed or bought to add to the CAD system.

Additional information about the building objects can be enclosed in the drawing. A traditional way to add material aspects to a drawing is the use of hatch patterns. A separate short-list on the drawing explains the meaning of these symbols. A more advanced method to add information is the attachment of parameters to drawing entities. These parameters can be given a certain value for each occurrence of that drawing entity. Such a CAD system offers the possibility to extract the parameter values, so that they can be used as input for other applications. Some CAD systems offer the possibility to store the parameter values in a database system. In that case the data can also be retrieved through for instance SQL.

The legend offers project management information. Drawing names often indicate something about the contents of the drawing and about the stage of the building process (e.g. preliminary design). Normally drawings are subsequently changed and each new version gets a unique number. The project name and the person who created or controlled the drawing are noted on the drawing.

Dimensions are not part of the design but are nevertheless enclosed in the drawing. It is common practice to write down the dimensions of object separately on the drawing. This is done for the convenience of the ‘reader’ of the drawing but is not strictly necessary. A CAD system is able to retrieve dimensions from the drawing data. Despite of that, the most significant dimensions are usually appended to an object so that other dimensions can be derived, even in CAD drawings. The design is drawn at a specific scale that is stated somewhere. As a building practice rule it is not permitted to measure dimensions from a drawing. Only the dimensions that are plotted on the drawing are valid. This rule allows alterations in the dimensions without changing the design and stems from the time when handmade drawings were very time consuming. Evidently such an inconsistency is likely to be the cause of errors.

Not every part of a building design is usually drawn, because there is often a lot of repetition of building elements in one building. Such building elements are worked out in detail once and are referenced at all other places. Drawings are always created for a specific goal. The level of detail and thus the drawing scale suits the purpose of the drawing. Within a drawing only those parts are worked out that are characteristic for the complete building structure. Usually these are the places where building objects meet. Sometimes building objects are not drawn because it is easier to describe them in words.

2.2 Documents

Lots of documents are exchanged within the design and construction process such as: order, invoice, product information, invitation to tender, bill of materials, delivery schedule, etc. In this set the invitation to tender is of specific interest, because it contains a description of the final building design and it is used as a contractual document. So we will look at this document in more detail which consists of a set of drawings and a textual documents.

Traditionally the invitation to tender is a description of how the building is to be constructed and which materials will be used. Whereas the drawings mainly contain technical descriptions of the building objects, the textual part describes the work to be done to create the objects or to put them in place, and in what order. The latter is necessary to ensure a certain quality level of the building realized that depends on both the material that is used and the way the building is constructed. Also, the temporary equipment that is to be used during the construction is part of document. To reduce the risk that the result is not what the designer expects as much as possible, the construction description can become very technically detailed. The drawback is that the building company is very limited in choosing a solution for a construction that it considers to be the best. There is a tendency to describe the functional aspects of a building object and to leave the way it is constructed to the building company. Both description methods - technically detailed and functional- can be used within one document.

Since not every object that is to be part of the building is drawn, the textual part of the invitation to tender contains descriptions of all materials to be used. If applicable a drawing will be referenced from the textual document to give more information about geometry and location. Much material that is used while constructing the building object is implicitly mentioned as part of a specific job, but not explicitly drawn in the design. Some parts of the design are left open to be filled in during construction.

In the construction process a lot of checkpoints are built in to see whether the desired quality level is reached or not. The document states for what building object or construction task approval is needed and from whom. The location of these checkpoints is determined by the person that made the invitation to tender. One purpose is to offer the principal the opportunity to explicitly approve parts of the building project. Another purpose is that approval is legally required for some building products from special authorities.

While describing the working parts and the materials to be used, references are made to standards whenever applicable. Standards contain criteria that should be met by the construction due to building regulations. Sometimes documents are referenced that are not official building regulations, but add extra criteria that should be taken into account. In the case of material descriptions, numbers and codes of specific buildings objects are referenced as used by the company that produces them.

In the document a lot of terms are used that are specific for the disciplines that are involved. It takes quite some building and construction knowledge to make the correct interpretation. Within the complete context of the design, the risk of making a mistake by a building engineer is limited, but there is no such thing as a normalized terminology for building description.

A large part of the invitation to tender is not concerned with the building as such but with legal and commercial affairs. These regulations deal with circumstances when problems occur during the construction or with the payment.

Summarizing, the invitation to tender is used for several purposes:

1. Tendering.
2. Description of the building construction.
3. Juridical description of responsibilities.

Although these purposes share a great deal of the same information it should not necessarily be in the same document. In most circumstances only a very small part is actually used as input by the participant that receives it.

3 Research inventory

Communication between companies in general and in the building industry in particular is a research item at many research institutes and for many international committees. Every research takes its own point of view on the communication problem. To be able to compare the various outcomes of the researches a research framework has to be defined first. Afterwards the different results are summarized by means of a group of questions.

3.1 Research area

Communication as a phenomenon will be restricted in some senses to create a clearly defined environment. The research will deal with the complete life cycle of a specific building *product*. A building *project* usually covers a part of the life cycle of a product. The *start state* of the building product that exists at the beginning of the project and the *end state* that should be reached when the project is finished is explicitly described. The building product will pass intermediate states between the start and the end state. A building product can be described as an assembly of building *objects*. In the building project a fixed and limited number of *participants* are involved. Each of them is capable of executing a unique set of activities. An *activity* modifies and/or adds building objects. During the building project information is exchanged between participants in order to execute activities in such a way that the building product is created in accordance with its defined end state. The research is focused on describing the information exchange process explicitly. The information exchange can take place between each pair of participants and at any moment during the project time.

The following considerations can be made according to the formulated framework:

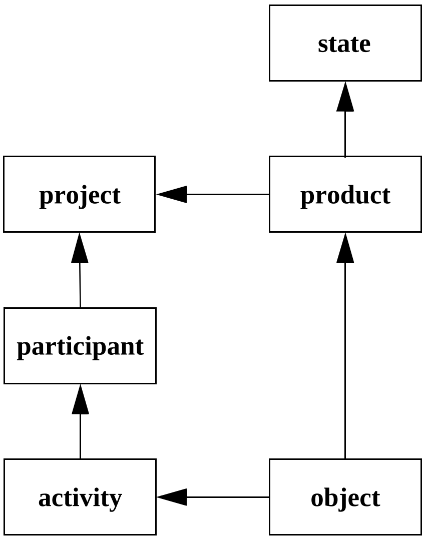
- Describing the state of a product at the end stage is troublesome since at the start of a design process the product to be created is still being developed. On the other hand, there must always be ‘some idea’ of what the result of the project should be. This problem will be taken into account when we are developing the description of the information exchange process.

- Concurrent execution of activities can happen within one project but also between two or more projects. The latter has to do with scheduling of projects by a participant. The research is focused on the consequences of concurrences of activities that are executed by all participants within one project.
- The use of resources by activities is dealt with implicitly. Objects that are used in activities are specified in the same way as the building objects that are modified and added. In building practice equipment, labor and consumed material are considered a resource. Note that consumed material (nails etc.) also belong to the product.

3.2 Related research

Related research projects are analyzed within the formulated framework [Figure 5]. The graphical representation of the framework and of the data models that are adapted from the research projects is based on functional object modelling [Hee 1994]. A formal definition of the graphical conventions and a comparison with other graphical conventions can be found in Appendix A.1.

FIGURE 5. Research framework



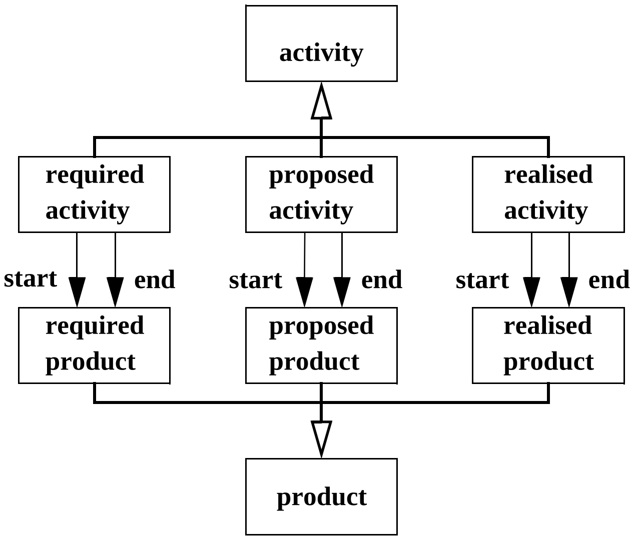
A selection of the work of standardizing committees and scientific researchers is presented. The selection is arbitrary in the sense that the fact that a project is presented or is not presented is not a qualification. The purpose of the presentation is to give a short overview of work from the past and of recent work. Each project covers part of the aspects that are considered relevant within the communication context. First, all projects will briefly be introduced in alphabetical order and limited by the framework. Afterwards a summary will be given by answering the following questions:

- How is the building project described?
- How are participants described?
- How are activities described?
- How are building objects described?
- How are building products described?
- How is the state of a building product described?
- How is the information exchange process described?
- How is multiple occurrence of the same building object controlled?

3.2.1 BPM

The Building Product Model presented in the thesis of G. T. Luiten [Luiten 1994] is focused on the activities of a constructor. An activity is defined by its name and is related to a building product by a construction state. Products can be described in three states: required state, proposed state and realized state. Product states will move between required and proposed until finally the realized state is reached. The product descriptions of a product in different states facilitate the verification of the design according to the requirements and of the physical product according to the design. Activities are specialized into subclasses that are analogous to the product states. In each state a start and end relationship is defined between activity and product.

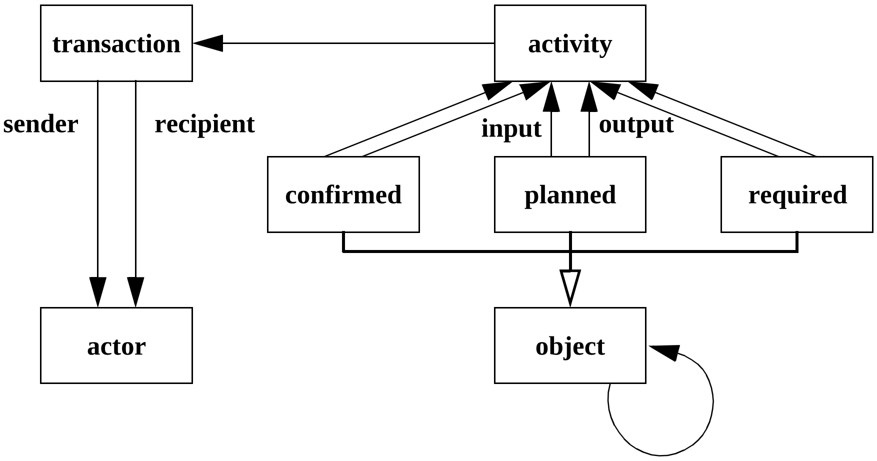
FIGURE 6. Adapted from BPM



3.2.2 BTMS

A Business Transaction Management System as defined by Hofman [Hofman 1994] is a conceptual model of an interorganizational information system. One of the parameters of a BTMS is a set of procedures. A procedure is a specific ordering of activities. An activity is defined by uniquely identified input objects and output objects. A transaction is a sequence of messages. A transaction (like a message) can be incoming or outgoing. The role of participant can likewise be that of a recipient and a sender. Objects can remain in three availability conditions: required, planned and confirmed. The availability condition specifies the status of objects at a certain place and time. Objects can be decomposed into other objects.

FIGURE 7. Adapted from BTMS



3.2.3 Classification and Coding

Classification and coding committees have mainly been concerned with classifying building products or building parts (and not with building objects). Depending on the purpose of the classification method, the class hierarchy and the object descriptions differ [Bouman 1973]. Often classification methods use a certain coding method to identify classes in shorthand. A classification method that has had a major influence in the building industry is the SfB (Samarbetskommittén för Byggnadsfragor) system [CIB 1973]. It consists of three tables: (1) elements, (2) constructions and (3) material and other resources. Each of these tables consists of a list of object descriptions. Classifying a building object is done by choosing a description - if possible - of each of the tables and combining the corresponding codes into a new classification code. A distinction is made between products that are made on the building site and building products that are purchased. The problems that arise with this classification method is that although there are a great number of description combinations possible, it offers classification of building objects on a global level only.

Classification methods are not developed for a detailed product description. It is almost impossible to define a classification method that leaves no room for discussion when classifying a building object. In other words, there is no guarantee that two different persons participating in the same building project will classify a building object to be an instance of the same object class. Classification methods such as the SfB system are not meant for transferring information in a clearly interpretable and consistent way, but as an ordering mechanism. That is why these methods do not serve the purpose of product or work description in a certain stage of development very well.

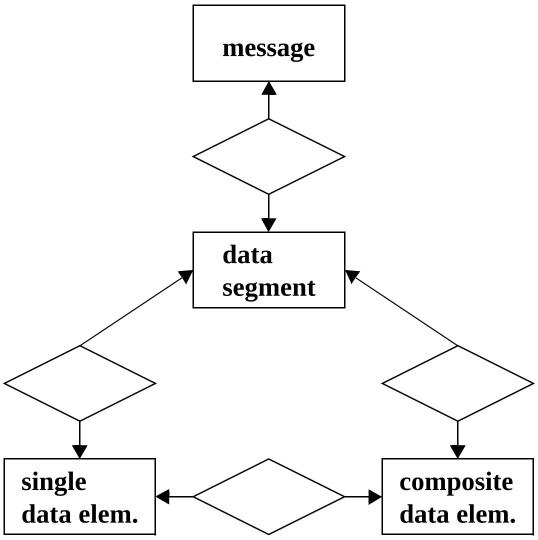
The invitation to tender is not really a classification system, rather it structures text in a specific way. It contains a very detailed description of a building design. Modern applications (such as STABU [STABU 1990] in the Netherlands) generate such a document by combining pieces of text that describe a building element and/or a working section. These descriptions are coded and contain part of the complete object description. A description part usually consists of a set of attributes grouped together in a certain context. Some attributes have a predefined attribute value, some can be filled in while creating the document. The invitation to tender contains a work description but the logistic aspects of the work sections are considered only incidentally. The description method is not suitable as a classification system, because attributes cannot be selected independently from their context.

Recent work by classification and coding committees [ISO/TC59 1993] shows that they are moving to a generic product model for building objects. Product descriptions should adapt to this generic model.

3.2.4 EDIFACT

An EDIFACT message based on the Electronic Data Interchange for Administration, Commerce and Transport standard [ISO 1988] can contain planning information, financial information, product information, etc. A standardized EDIFACT message (or United Nations Standard Message) is defined using the EDIFACT syntax and the objects from the EDIFACT Data Segment Directory. A Data Segment consists of a set of single element and/or composite data elements. A single data element contains one attribute. A composite data element consists of a set of single data elements. Each attribute is a well defined “subject of interest” with a type definition in accordance with the EDIFACT syntax. Some of the attributes have a limited value range. This range is defined as a set of values called a Code list. Code lists are standardized by e.g. the International Standards Organization. The information exchange process (or message scenario) is described as a set of messages. Not part of the EDIFACT standard but very common in practice is the definition of an Interchange Agreement among the companies that will use the EDIFACT messages. Such an Interchange Agreement consists of a juridical part, a technical part and a message scenario. The juridical part deals with subjects such as security, archiving, proof, duration of the agreement, etc. The technical part describes which (part of the) standardized EDIFACT messages will be used.

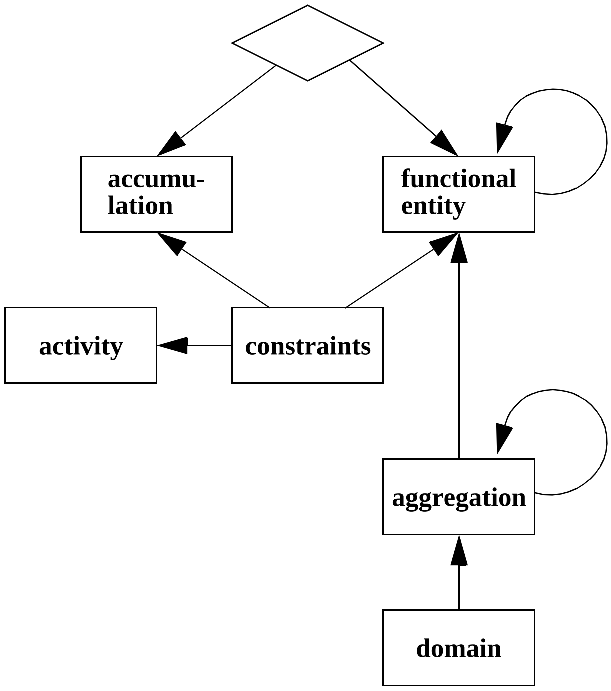
FIGURE 8. Adapted from EDIFACT



3.2.5 EDM

EDM (Engineering Data Model) is a data model developed to store design information [Eastman 1991a]. Therefore it uses three primitives: (1) domains, (2) aggregations and (3) constraints. A domain is a set of possible values. An aggregation is a set of domains and aggregations. A constraint is an expression in some constraint language that evaluates to true or false. An object as defined in Section 3.1 is called a functional entity and is composed of other functional entities, aggregations and constraints. Assemblies of functional entities are called accumulations and are defined as sets of constraints over a set of functional entities. A product description as defined in Section 3.1 can be regarded as an accumulation that reflects the current state of development. EDM allows partial integrity which means that not all constraints need to be satisfied. Changes in the design are effectuated by so-called transactions. A transaction alters specific integrity conditions on a set of functional entities and/or accumulations. Integrity conditions are preserved by sets of constraints. Transactions can take place concurrently if the integrity conditions of a functional entity or accumulation are not violated during the transactions [Eastman 1991b].

FIGURE 9. Adapted from EDM.



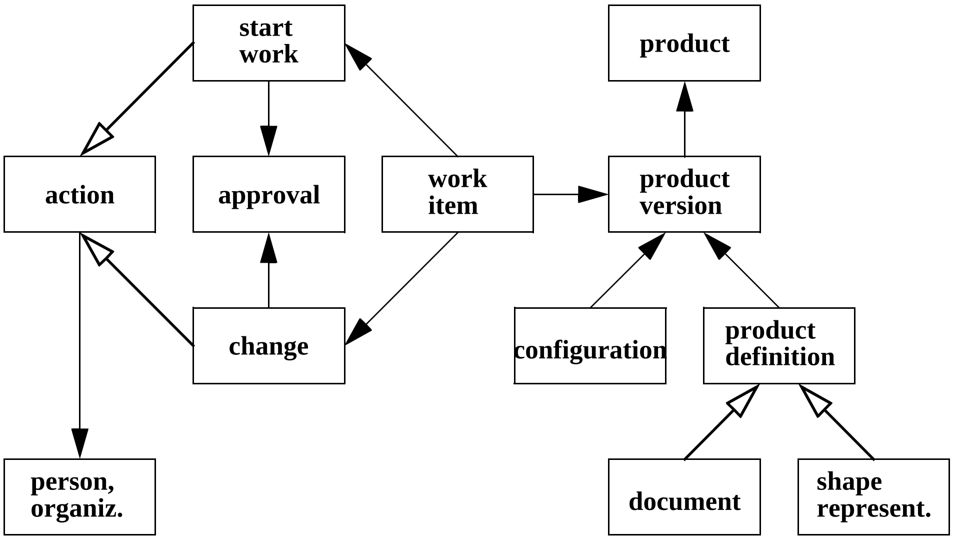
3.2.6 STEP

The international STandard for the Exchange of Product model data [ISO/TC184 1993b] organization is developing a series of product models officially known as the ISO 10303 standard. Each of these product models is called an application protocol. The objective of STEP is to provide a mechanism capable of describing product data throughout the life cycle of a product, independent of any particular system. STEP development aims at data communication and data sharing between companies (e.g. building company, car manufacturer, material supplier, etc.) The application protocols that are available now and that are concerned with either communication or the building industry are:

- AP 203: Configuration Controlled Design [ISO/TC184 1993a]
- AP 225: Building Elements Using Explicit Shape Representation [ISO/TC184 1994]

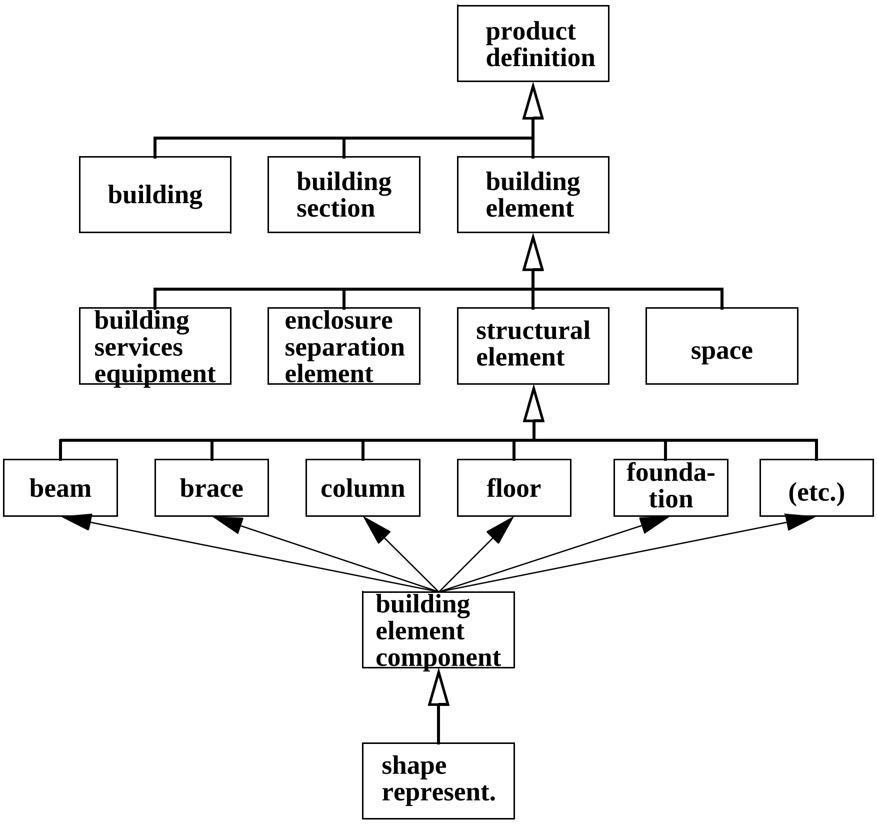
AP 203 specifies the standard that is used to exchange product definitions with three-dimensional shape representations and the data that define and control the configurations of those product definitions. This part is solely concerned with the design phase of the product life cycle. Only the designs of the mechanical parts and assemblies can be specified by this standard. The primary focus of AP 203 is the data that control the tracking and management of the product. A product will have at least one version. Each version has one or more product definitions and, optionally, a shape representation. Product definitions may refer to documents. Configuration of the product is defined by work items. Works items describe the actions that must be performed upon the product. An action is executed by a person or organization. There are two types of actions: starting new work for product definition and changing an existing product definition. New work and changes must be approved before the end state of a product definition is reached. In this data model the absence of a description of the product structure is quite striking. A product is merely described as an assembly of configuration items without additional relationships.

FIGURE 10. Adapted from STEP AP 203



AP 225 specifies a standard for the exchange of building design information between architecture, engineering and construction (AEC) application systems and related systems using explicit 3D shape representations. On the highest level of the hierarchy we find the building, the building section and the building element, which are subclasses of the product definition from AP 203. The building element is specialized into four subclasses. One of these classes, namely the structural element is in return a generalization of a large group of building objects that form the leaves of the specialization tree. Each of the building objects is related to a shape definition via a generic building element component. The structure of AP 225 looks like a high-level classification tree for building objects. Commonly used building objects are part of the tree; it is doubtful that all possible existing and future object descriptions will be incorporated in the standard. Functional relationships between building elements, such as column X is bearing slab Y, are not part of the model.

FIGURE 11. Adapted from STEP AP225



3.3 Summary

- How is the building project described?

  With EDIFACT and BTMS the interchange agreement and the transaction are used respectively to describe which messages and activities are included in the process. A process however, is not similar to a project. Process agreements have a more long-term goal and span more than one project. In the building industry the span of an agreement among companies is usually not longer than one project.

- How are participants described?

  In STEP part 203 a person or organization can perform a set of actions. Likewise in BTMS actors are linked to a transaction that defines the set of activities that will be executed.

- How are activities described?

  Activities are described as a set of work items that have to be executed upon a building product or building elements in the STEP parts. In EDM activities act upon constraints, but are not explicitly defined. BPM describes an activity by the start and end state of a product. Likewise BTMS describes activities by a set of input and output objects.

- How are building objects described?

  Classification and coding methods (such as the SfB system) and STEP part 225 offer a hierarchy of building object classes on a global level. Building objects are defined by their name and they have additional properties. EDIFACT, EDM and BTMS allow for the construction of nested objects, but do not specify them in detail.

- How are building products described?

  STEP part 225 defines a building product as a set of building objects each with its properties and shape representation. Functional relationships or constraints between building objects are not part of the model. BPM does not specify the product structure, whereas EDIFACT and BTMS make no distinction between objects and products. EDM defines a product as a set of constraints over a set of objects.

- How is the state of a building product described?

  States relate activities to the product in BPM and BTMS. A state is defined as a subclass of the product. In STEP part 203 a product has a specific version. The activities that act upon the product version are in a certain state of approval.

- How is the information exchange process described?

  In general a process can be described as moving from one state to the next. In this sense the STEP parts and BPM offer a kind of process description. BTMS describes a transaction process by procedures that in return are defined as sets of activities that will be executed by a participant.

- How is multiple occurrence of the same building object controlled?

  In the complete information exchange system only the most recent occurrence of an object in the system is valid. Multiple occurrences of an object can occur when two or more activities act upon an object simultaneously. In EDIFACT and BTMS consistency of information is controlled by a predefined message sequence and message definition. In the other models approval to move on from one state to the next is controlled from outside the system.

4 Process analyses

The aim of the research is to offer a formal description of the information exchange process for a certain building project. Therefore, the characteristics of the design and construction process in the building industry have to be analyzed first. To describe these characteristics three accompanying formalisms are used called: state transitions system, functional object model and timed, colored Petri nets [Hee 1994]. An explanation of the graphical representations of the formalisms can be found in Appendix A.1.

4.1 Evolution of information

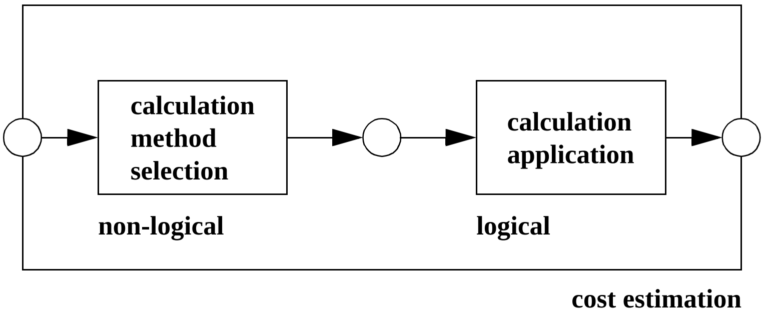
During the life cycle of a building product, the product itself evolves and so too the information about the building product. Evolution comprises changes and an extension of the information structure and of values of the data elements. There are two main approaches to describing the evolution process:

1. Describing the relation between start and end state of the product in a formal way.

   Formal process descriptions can be used for activities in which mathematical expressions relate input to output: *logical activities*. For some building activities mathematical expressions do exist relating input values to output values, take for instance a finite element analysis of the structural behavior. Less obvious are mathematical expressions for relating input information structure to output information structure. Changing the information structure of a building product is exactly the part of the building project that needs human interference in most situations. At the very least the final responsibility about products or product parts that have been changed belongs to a person. Informal process descriptions which are not merely the result of logical reasoning are called *non-logical activities*. Especially design activities are non-logical activities.

   Logical activities are advertised as business activities by a company if they require special resources such as cranes, labor etc. Non-logical activities are advertised as business activities if they require special knowledge, such as architectural knowledge, etc. Non-logical activities (e.g. cost estimation) may be decomposed into other non-logical activities and logical activities (see Figure 12).

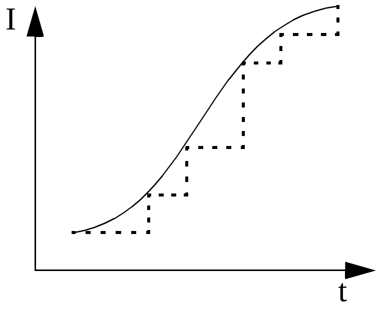
FIGURE 12. Example of decomposition of business activities



2. Describing the occurrences of the product in a finite number of states.

   Instead of describing the actual process, the result of part of the process called a *state* is described. The actual process is approximated by a set of states, moving from one state to the next. Normally the amount of information will increase during the building project. The production of information (I) in time is described by a sequence of states: <s_i,t_i> (Figure 13). This method is often used if it is not possible to formally describe the state transition between two states. Therefore, it is an appropriate method for processes with human interference such as design processes. Systems that describe dynamic behavior as a sequence of states are indicated as *(state) transitions systems.* This concept is extensively worked out by Booch [1994] and Rumbaugh [1991]. Transition systems can be accommodated with stores to preserve persistent information [Hee 1994]. Storage of information at companies plays an important role in maintaining consistency of all information that is available at a certain time in the complete information exchange system.

FIGURE 13. State transition

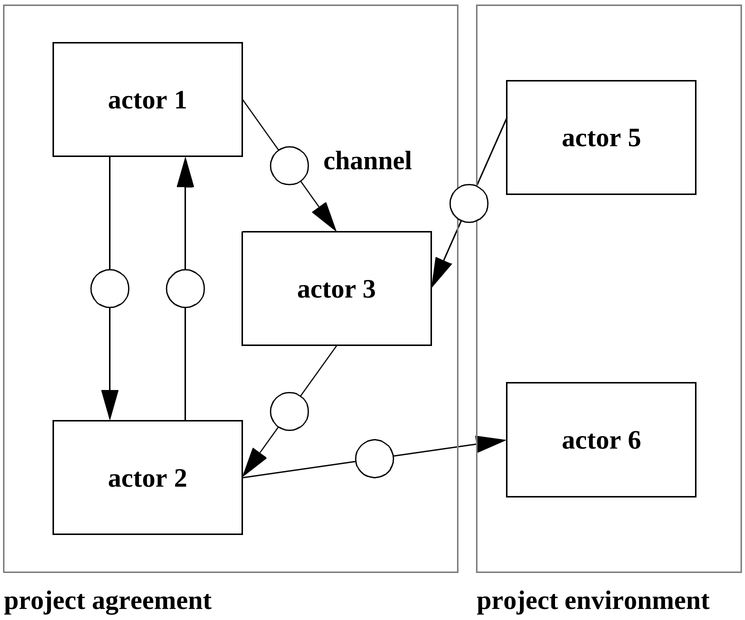


Combining both process description approaches is possible if the state transitions can be formalized. In the case of logical activities a formal description is available. At the lowest level of decomposition a process description will normally consist of logical and non-logical activities. If a logical and a non-logical activity are aggregated to a higher level activity, the aggregated activity is a non-logical activity. Thus, high-level process descriptions such as the information exchange process mostly solely contain no-logical activities as processors in a state transition system.

4.2 Actor models for information exchange

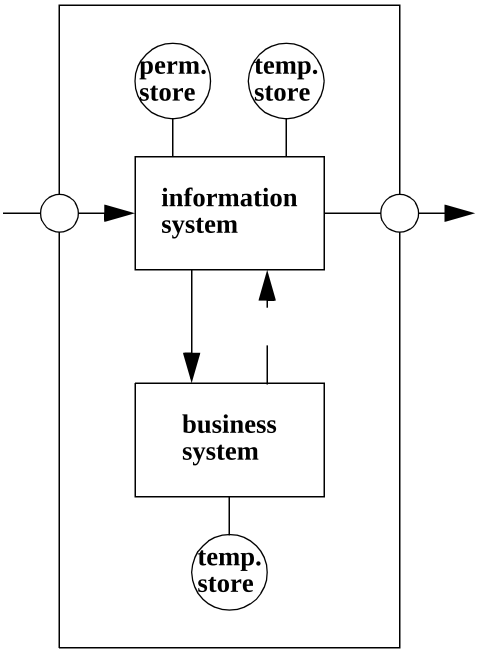
The information exchange process will be modelled as a state transition system. A state transition system can be described by a Petri net using actors, channels and stores as the basic entities. *Actors* communicate via the channels that interconnect them. In a building project the actors in the transition system are the building companies (e.g. architects, contractors, etc.) that are involved. The *channels* are the communication media they use for exchanging information. *Stores* are used by actors to preserve information during the life cycle of the information system. Usually the government, local authorities and small companies that play a minor role are considered as the environment (Figure 14). To create a clear definition of which of the actors is a company and which is not, it is assumed that all companies that are an actor in the building project have an agreement during the project. Companies play different roles in different projects. Large companies embody various disciplines such as design, management, financing, etc. A *participant* is defined as the role a company plays in a specific project. An *agreement* defines a contractual relationship among participants about the contribution of each of them to create a building product. In other industries contractual relationships exist for a long time between two or more participants. It is a specific characteristic of the building industry that agreements only last for one project. For this reason the building project is chosen as the scope of the information exchange description.

FIGURE 14. Example of a state transition system for a building project



A participant’s system can be divided into an *information system* and a *business system*. The information system supports the business system by producing information that is needed to control the business system (e.g. the structural design that is needed to produce the concrete floor). In return, the information system retrieves information from the business system (e.g. the status of the floor production). Some participants have information production as their primary business (e.g. architect). Both the information system and the business system need stores to store information and products respectively. The storage of physical products is normally temporary because these products will become part of a building. The storage of information can be *permanent* and/or *temporary*. At the start of a project the temporary store is empty and the permanent store contains information that is not project dependent (e.g. material aspects of a building part). Part of the temporary information can be made permanent if the participants expects that he can reuse it in another project in the future. The information system will be connected to a temporary store and to a permanent store. Participants that produce physical products only have a temporary store connected to their business system [Figure 15, Participant system A]. Participants that produce information have a temporary store and a permanent store connected to their business system [Figure 15, Participant system B].

FIGURE 15. Participant system A and B

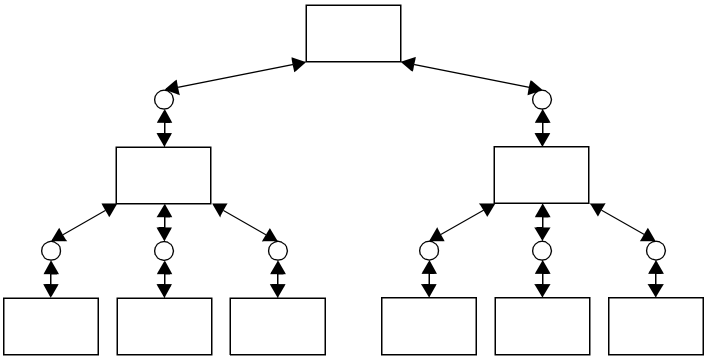


Since the research is focused on information exchange, the exchange of physical products is neglected. As a result Participant system A in Figure 15 has no input and no output channel connected to the business system. In Participant system B the consumed information must be split into a part for the information system (e.g. delivery date) and a part for the business system (e.g. drawings).

The way participants communicate depends on the way the building project is organized. Analogously to the three main project organization types that are recognized in the Dutch building industry [SBR 1992], three communication topologies can be distinguished:

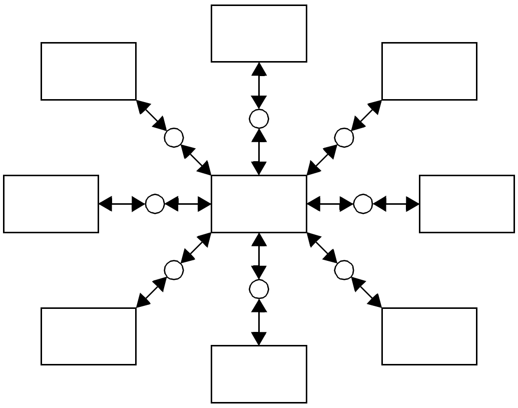
1. Tree
2. Star
3. Black box

FIGURE 16. Tree topology



A description of a communication pattern that fits the tree topology could be as follows: The principal is the root of the tree. He[1] asks the architect to design a building in accordance with his requirements. The architect will consult a structural engineer for the structural design and he will consult a HVAC (Heating, Ventilation and Air-conditioning) engineer to design the installations. When the principal is satisfied about the design he will ask building constructors to calculate the building cost. He will choose one of them (usually the cheapest) to construct the building. The building contractor will ask other (sub)contractors to do part of the job such as pile driving, brick laying, mechanical engineering, etc.

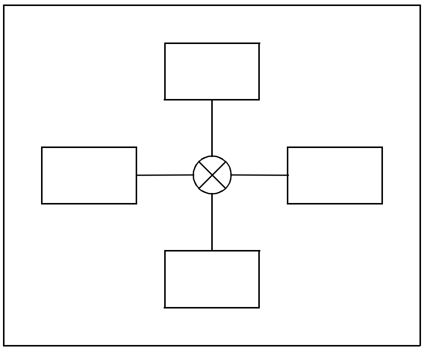
FIGURE 17. Star topology



[1] For readability reasons persons are considered to be males.

A description of a communication pattern that fits the star topology could be as follows: The centre of the star is the building manager. The principal tells him what kind of building he has in mind and leaves the rest up to him. The building manager asks a team of experts (e.g. architect, structural engineer, HVAC engineer, building constructor) to be members of the design team. When the principal is satisfied about the design the building manager calculates the building cost. Then he will search for constructors that will do the job or part of the job for a ‘fair’ price. During the construction process the building manager takes care of the construction management.

FIGURE 18. Black box topology



A description of a communication pattern that fits the black box topology could be as follows: The black box is a symbol for a large building company that covers the complete design and construction process. The building company will finance, design and build the building. The company consists of a group of small companies that work together permanently and share each other’s information. After the building is realized it will be sold to someone else for investment or for housing.

In building practice a building project hardly ever completely fits one of the three topologies; it will usually be a mixture of them. Moreover, the communication pattern can change during the project. The relationship between participants is considered during the project time. It is of course possible that participants have a more permanent relationship that lasts throughout the project time. Collaboration of participants influences the project organization and the communication pattern.

4.3 Information exchange

Information exchange can take place in two ways: information sharing and information transfer.

Information sharing requires that the participants share each other’s stores. This seems the most straightforward solution for the communication problem. In the early years of data exchange there was a major attempt in the Netherlands to create a comprehensive data model and process model for building design and construction [IOP 1989]. However, the building practice was not very eager to adapt to this model for many reasons. One of them is that each participant wants to be flexible in changing his information system and he will not allow himself to be dependent on a general model. Another reason is the question who is responsible for the authorization management of the shared data. If information sharing is possible between two or more participants, then this group of participants is regarded as one participant with one shared information store. This situation corresponds to the black box topology.

Information transfer means that the output of the information system of one participant is the input for another participant. The information that is exchanged is called a *message*. *Information* is the meaning of data within the context of the system that uses it. This implies that the information changes if the context changes. (Note that the definition of information that is stated in Section 1.2 is slightly adjusted to information technology usage.) When importing the received information the following situations can occur:

1. The recipient consumes the information and uses it for his own activities. This implies that the data structure of the recipient is part of the data structure of the sender. In fact this is the same assumption as needed for data sharing that has been mentioned in the first paragraph of this section.
2. The recipient translates the information into the format that he uses for his own activities. The knowledge that is necessary to perform the translation is available through human interpretation or is made explicit by a set of rules that can be interpreted by a computer. Translation includes interpretation of the information and mapping the underlying data structures. Interpretation adds semantics from the background of the recipient. Mapping transforms the message data structure to the recipient specific data structure. Because mapping of non-equivalent data structures normally results in loss of information, extra information needs to be added by the recipient.

Much research has been spent on creating a general model that acts as the neutral exchange format [ISO/TC184 1993b]. In the most ideal situation software venders will equip their packages with export and import facilities to this neutral format. Even then loss of information is unavoidable, because there will be differences between the application data structure and the neutral data structure. The question is whether the software developers are willing to adapt to a general model and lose their flexibility in making changes to the data structure of their software application.

Data exchange can be divided into the exchange of the data structure and the instance of the data structure, also indicated as the data values. A data structure can be described using the concepts from object-oriented technology, namely objects, object classes and relationships. These concepts are extensively worked out by for instance [Booch 1994], [Rumbaugh 1991][1] and [Hee 1994][2]. Whereas a new message will always contain new data values, this need not be the case for the data structure. To transfer the data structure there are two possibilities:

1. Join the data structure with its instance in each message.

   This option prevents each participant from keeping data structures in store. Nevertheless, it is not a very attractive option because it causes a lot of overhead. In the case of identical data structures distribution is only necessary if there are changes in the general data structure. In the case of different data structures the source and the target data structure will be fixed for a period of time to keep the mapping rules intact. Thus, in both cases transferring the data structure with each message is unnecessary.

2. Distribute the data structure beforehand.

   To be able to implement the general data structure or to implement the mapping rules the data structure will be distributed separately from the data values. Updates of data structures will have a low frequency because of the investment in “learning” new data structures.

Mapping of data structures is only possible if data structures are available beforehand. To be able to capture this process in a set of mapping rules, the data structure of the message to be received and the target information system must be described formally and remain constant.

4.4 Data structure mapping

The mapping situations that can arise if two non-equivalent data structures are mapped have been investigated. Mapping of two data structures means mapping of object classes and relationships. Each of them can be identical or different if

[1] Rumbaugh distinguishes links and associations instead of relationships.

[2] Van Hee distinguishes two types of object classes: simplex classes and complex classes.

one compares the source system and the target system. The investigation assumes that all information is available in the source system and that the target system is capable of importing all information.

TABLE 1. Mapping situations

	object class	relationship
1	identical	identical
2	different	identical
3	identical	different
4	different	different

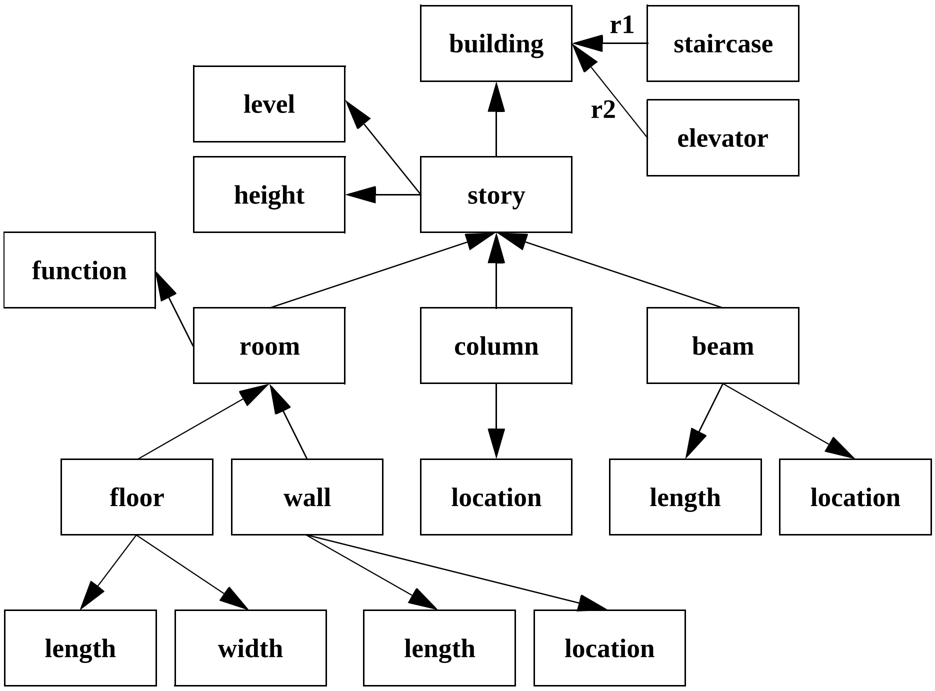
Mapping situation number 1 refers to the ideal circumstance in which two information systems have identical data structures. The communication topology that fits to this situation is the star topology. There is a central organization that prescribes the data structure to be used by the participants. Each participant looks after the adaptation of this data structure. The central organization can be a participant in the project or an actor in the environment. If it is a participant then in fact he imposes his data structure upon the other participants. This situation occurs in countries such Japan with a few very dominant companies in the building market. If the central organization is an actor in the environment then one could think of international standardization committees such as ISO. In that case the tree topology communication pattern also can take advantage of the availability of a common data structure. Standards that have emerged in this field are EDIFACT messages [EDI 1994] and STEP application protocols [ISO/TC184 1994].

Mapping situation number 2 refers to the circumstance in which the relationships are identical but the object classes are not. This means that all building object classes exist in the target system, but at least some of them have a different name. In the building industry each discipline uses its own vocabulary to express their thoughts; often different nouns are used for the same object. If the object characteristics are identical, then they can be regarded as synonyms. This type of mapping could easily be solved if there was a dictionary of building objects. The communication topology that fits this situation is the star topology. There is a central organization that describes the data structure to be used by the participants, but the participants are free to use their own vocabulary. This situation may occur between participants that strongly agree in their view on the building product, e.g. a structural engineer and a firm manufacturing concrete floors.

Mapping situation number 3 refers to the circumstance in which the object classes are identical and the relationships are not. This means that all building objects in the target system can be identified by name, but at least part of the relationships are absent. Often missing relationships are added during the interpretation process. If there are still relationships missing after that extra knowledge must be obtained from an expert to fill in the missing relations. The communication topology that fits to this situation is the tree topology. There is an actor in the environment that prescribes which objects are to be used, but not the data structure. This situation arises from the use of standards with object definitions. Especially relationships appear to be very dependent on the specific view of each participant in the building product [Tolman, et al. 1989].The central organization in this mapping situation are national or international committees such as EDIFACT [ISO 1988] and STEP [ISO/ TC184 1993b].

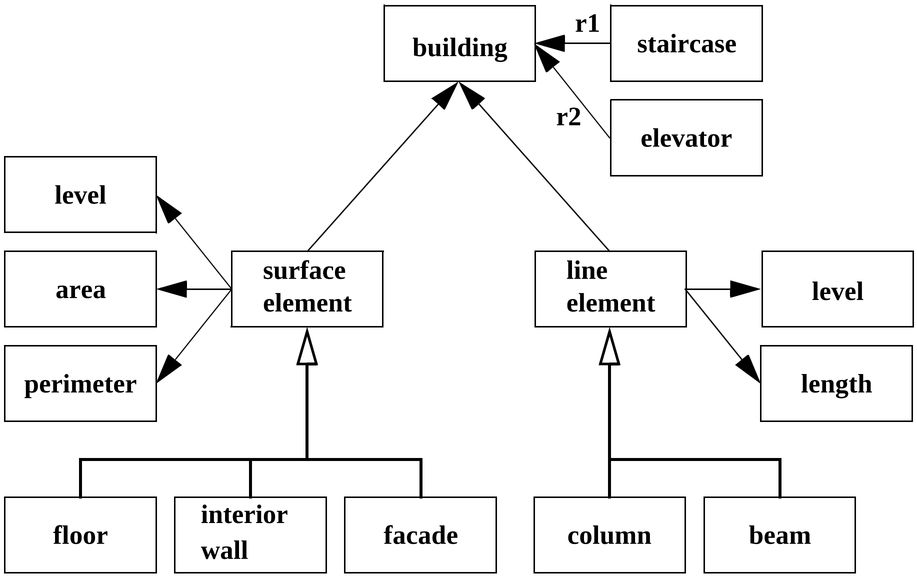
Mapping situation number 4 refers to the circumstance in which the object classes are different and the relationships are different. Reaching an agreement about the object classes that do not belong to the intersection of set of object classes of both data structures is usually very hard. Expert knowledge is needed to introduce the missing object classes and relationships. This mapping activity comes close to the traditional human interpretation of text and drawings to make it suitable for its own usage. Mapping can be made explicit by defining it as a set of mapping rules. Since object classes and or relationships are added, the mapping rules will be different for the reversed direction. The communication topology that fits this situation is the tree topology. If there is no actor in the environment who is able to impose a general data structure, every participant will stick to his own data structure. Because situation 4 occurs most frequently nowadays, an example of the mapping of two different data structures will be explained in more detail.

FIGURE 19. Source data structure



The source data structure of Figure 19 describes a preliminary design of a small office building. A simple line drawing of each story is available including the ground floor. The primary focus of this representation is the area of rooms, their dwelling function and their location.

FIGURE 20. Target data structure



The target data structure describes the view of the quantity surveyor on the design. To be able to calculate the building costs at an early stage, he compares the design with a similar, already realized, building. To this purpose he translates all building elements into surface elements and line elements except special facilities such as elevators and staircases. The reference building is also translated into surface and line elements. The total cost is known for each reference element group such as the floor, interior wall, facade, column and beam. By comparing the area, the perimeter and the length of the design and of the reference building, an approximation of the construction cost can be calculated. The level of an element determines whether it is a roof construction, a foundation or another construction. This is an important fact because it has a great influence on the costs. Mapping the source data structure onto the target data structure requires the following steps:

1. Mapping of the equivalent objects: building, staircase and elevator.
2. Mapping of the equivalent relationships: r1 and r2.
3. Area and perimeter are calculated from the length and width of the source floor and mapped to a surface element. The surface element gets a specialization relation with the target floor. The source level of the story of the room that the floor is part of is mapped to the target level of the surface element at hand.

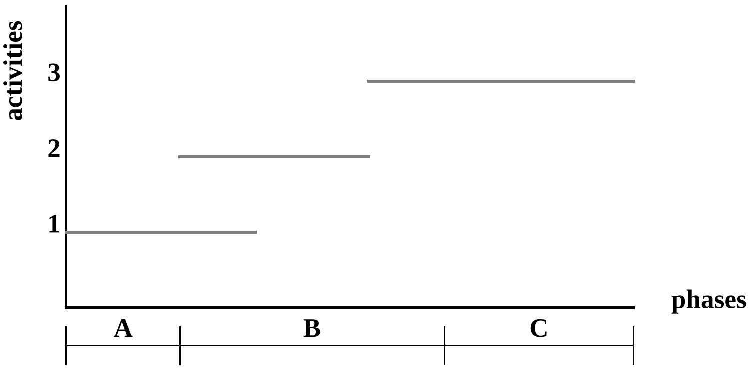
4. Walls must be divided into interior walls and facades. To solve this problem a procedure will be executed that searches for walls that are part of only one room. All these exterior walls together on one level will form a facade. All other wall are interior walls. Area and perimeter are calculated from the source length and the story height. The relation mapping is analogous to the floor.
5. A source column is mapped to a line element. The line element gets a specialization relation with the target column. The target length is derived from the story height. The story level is mapped to the target level of the line element at hand.
6. Mapping of the beam is analogous to the column. The only difference is the mapping of the beam length to the line element length.

4.5 Building process model

Traditionally the building process is described as a sequence of *phases*. A very global process description that is valid for any type of product development recognizes three phases: analysis, design and construction. For a building process these phases are divided into subphases and often carry names that describe the output of that phase, such as preliminary design. The output document of a phase can be a contractual document. When the document is approved then the process moves on to the next phase. Often approval also means payment for the work that has been done. The phase definitions for the building process that can be found in literature [SBR 1992] are useful as a general description of the building process. To describe a specific building project a more explicit definition of the information flow between the phases is necessary to check the state of the building product.

A building product is regarded as the result of a set of activities [Figure 5 on page 16]. Activities can take place concurrently and can be part of one or more phases [Figure 21]. Building process models are a generic representation of building projects. The next section will continue the analysis at building project level.

FIGURE 21. Phases-activities



4.6 Building project model

Models of a building project that encompass the characteristics of a specific building project are not available in literature. On the other hand, it seems obvious that the actual information exchange process depends on the project characteristics. For that reason a formal description of the information exchange process of a specific building project is developed in this section, using the Petri net methodology [Hee 1994].

A building project is defined by:

1. The information about the building product at the start of the project.
2. The information about the building product hat has to be produced.
3. The information that is exchanged between the participants during the building project.
4. A set of participants that have an agreement.

Information is always exchanged in discrete packages which are called *messages*. The *message flow* in the building project model has the following characteristics:

- Sending a *new messag*e will always result in a *reply message.*

  The reason for this condition is to prevent a message from ending up at a 'dead end'. To be able to control whether a network of participants is not in a pending state this feed-back mechanism is necessary. As a result of this condition the sender that starts the process is also the receiver of the last reply message.

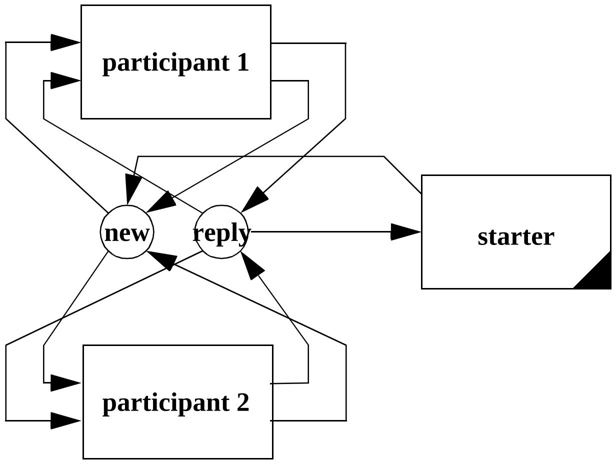
- A message is addressed to one receiver.

  This condition excludes the possibility of broadcasting a message to an anonymous group of receivers. In the model we deal with a set of participants as potential senders and receivers of messages that is known beforehand.

A communication structure is created by connecting the participants using channels. The information system of a participant has two input channels and two output channels; one for each message type: new and reply. Each participant may communicate with any of the other participants. Internal information flow inside a participant can be made explicit by allowing a participant to send messages to himself. To achieve this communication structure, there are two solutions, namely (1) interconnecting all participants and (2) connecting all participants to a shared channel. In the first case addressing a message means choosing the right output channel. Let N be the number of participants, then N^2 is the number of channels per message type. In the second case addressing must be part of the message. Every participant will only "listen" to the message that is addressed to him. In our message exchange system the second solution is used, because the overall structure is less 'crowded' with channels than in the first solution. Whether this structure resembles real practice or not is not of interest at this point. To control the information flow afterwards, it is necessary to keep the address with the message.

A special participant called *starter* is required to start the information exchange process. The starter starts the project by sending a message to one of the participants containing the project dependent information about the building product that is available up to that point. The starting message also contains a description of the building product as it should be realized at the end of the project. A secondary reason for the existence of the starter is that a participant cannot generate messages autonomously. The message flow is modelled as a chain of new messages and reply messages with a start off and an ending at the starter. The assumption behind this concept is that participants always act as a reaction to another participant. For the design and construction process this seems reasonable. The only exception to the rule is the participant that starts the process. That particular participant is triggered by the 'starter' to start the project.

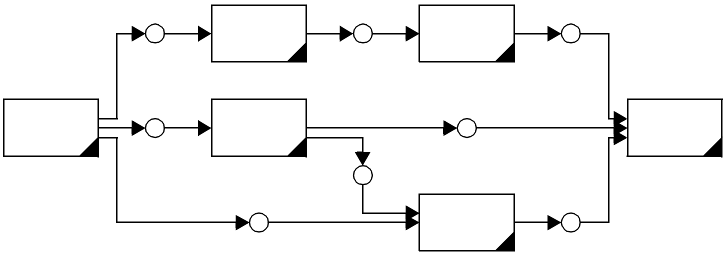
FIGURE 22. Example of project process model



A participant is defined by unique set of activities [Figure 5 on page 16]. Activities in the building project model are subsystems of the information system of a participant. If activities are connected by channels, a special type of transition system is created called an *activity network*[1]. The characteristic of this system is that a channel is connected to one subsystem at most on one side and one subsystem at most on the other. The *execution* of activities is the business of a participant. For each specific project a specific set of activities is selected and ordered into an activity network. An *activity* is defined as the smallest unit of work that can be advertised by the participant, because each activity should (in principle) be able to communicate with other activities inside or outside a participant. This requires that there is a clear definition of the required input, the produced output and the execution time. Input and output of an activity are formally described by their data structures. Two activities are the same if their input and output definitions are the same. The input and output channels that are connected to an activity can only transfer information that matches the input data structure or the output data structure respectively. A participant can create activity networks from his own set of activities and the activities that are provided by other participants. The role a company plays varies per project. In the model the company's role is played by a participant. The *role* in a specific project is defined by a subset of the total set of activities that can be provided by a company. In an activity network, an activity is always triggered by one or more other activities, it cannot trigger itself. This behavior complies with the condition that participants do not send messages autonomously.

[1] In Petri net literature indicated as a marked graph.

FIGURE 23. Activity network



Cycles do not exist in the activity network since repeating the same activity does not add new information to the existing information. Activity definitions consist of project independent data structure descriptions. Whereas data structure instances may change or increase during an activity repetition, the data structure itself will not. After the execution of an activity the universe of discourse has changed and thus another activity with a different input definition is required to continue the process. This condition seems to be in conflict with the fact that a design process is often considered an iterative process. There are two approaches to this problem:

- Keep the iteration within an activity.

  The activity definition does not describe the process within the activity. So there is freedom in doing the job in a linear or an iterative way.

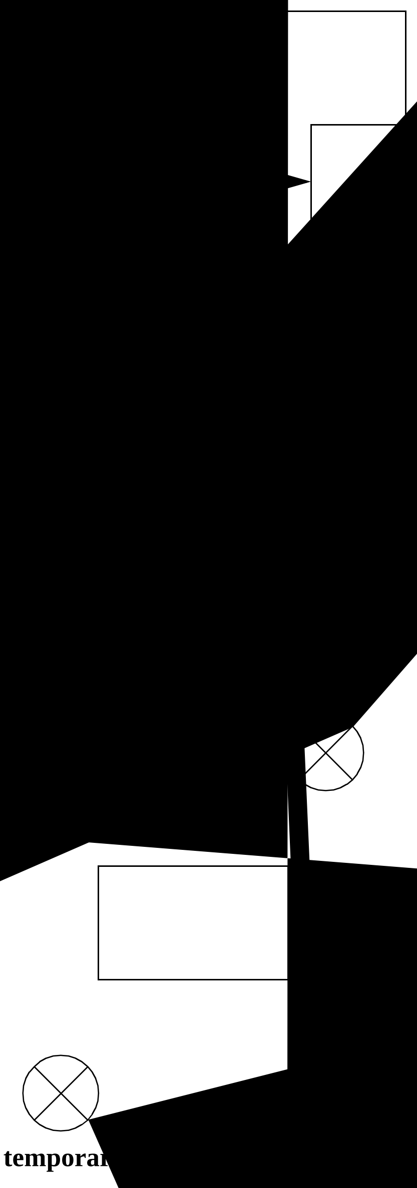
- Decompose the iterative process into a non-iterative one.

  After fulfilling an activity new information must be available otherwise it would have been completely useless. In other words: every iterative activity that is decomposed in enough detail will become non-iterative.

In the production process, iteration cycles are determined by the availability of resources. Activities use resources such as cranes and brick layers to perform their task. *Resources* are abstract or physical objects that do not become part of the building product. Activities can share resources, and thus there is a dependency in the execution of these activities [Figure 24]. For instance when only one crane is available at a building site and it is needed for lifting floors and for lifting roof panels, than concurrence of these activities is hindered by the lack of resources.

[redacted]

Process analyses [redacted]



[redacted] ed.
ne of
share the
ore.

[redacted]

The permanent (product) store contai [redacted]
considered to be of value for the future [redacted]
information deleted or updated. Perman [redacted]

indicated as project independent information because of the usage in more than one project. Information that is stored in the permanent store is:

- information about the end state of products

  The end-state information is kept for several reasons. Projects can be continued in the future. Realized products can serve as examples of a participants capabilities. Information of realized products can be reused in newly designed and constructed products. Information of realized products serves as a reference to make estimations for a product that is to be developed.

- information about building regulations

  Building regulations can be regarded as a set of constraints that have to be adhered to. Realized products make references to the building regulations that are applicable at the time of evaluation. Products that are still under development make references to the current building regulations. Part of many activities is conformance checking of the design against the building regulations [Waard 1992].

The permanent store requires a general model for storing long term information in a very broad sense. Three different types of actors are involved: participants (architects, constructor, HVAC engineer, structural engineer, etc.), suppliers (of roof panels, concrete floor, doors, windows, etc.) and the central or local government. The building project model is limited to the project time, so the subject of the permanent store will not be extended further.

<u>The temporary (product) store</u> stores information that is generated or received during the project. Only the most recent information about a building product is kept in store. Not all information needs to be part of the building product in its end state (e.g. product information from suppliers). Information can be added, updated and deleted. When a participant wishes to be able to reproduce the output of an activity network at any time in the future then he has to copy every new or updated object from the temporary store to the permanent store. Then the temporary store can be discarded since all information is immediately put into the archives and thus the permanent store contains all information.

<u>The product status store</u> is used for monitoring and controlling the design and construction process within a participant. Products are defined by a start and end state for the complete project. During the development of a product, the product’s objects pass a sequence of states (e.g. reviewed, approved, etc.). The set of states will differ for each participant. The next activity in an activity network can only be started when all required objects have reached a certain state.

The (permanent) message store stores all information of the messages that are received and that are sent. Receipt of a new message will result in the start or continuation of an activity. The message contains the name of the activity and the required input. When an activity is started, the message store is inspected to retrieve the input information that is needed. After the execution of the activity the message store is updated with the newly generated information. Completion of an activity will result in a reply message. While the project is running, communication can take place with other activities from other participants by sending and receiving messages. Every message that is sent, is uniquely identified by the activity name and the time stamp it gets at the time that it is sent. The message store exceeds the life time of a project, because the messages that are used in the communication process with other participants are also used as a sequence of states that can be traced back. This can be useful if a conflict arises between two participants about the information that has been exchanged at some time during the project.

The activity status store keeps track of the status of the execution of the activity network that was started in the project. The states that are distinguished are:

- Not completed.

  Upon receipt of a message it is stored with status “not completed”.

- Missing input.

  The status changes from “not completed” to “missing input” if not all input information is available in the temporary or permanent store. In this state messages will be sent to other participants to acquire the missing input.

- Executing.

  The status changes from “missing input” to “executing” if all information is available to execute an activity in the activity network. The activity will stay in this status during the execution time of the activity. An activity can be restarted when new information is received that is part of the input of an activity that is in the “executing” state. All participants that have an activity that is not in the “completed” state and use information that depends on the newly received information will get an abort message.

- Missing output.

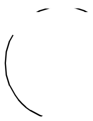
  The status changes from “executing” to “missing output” if not all information is produced that had been asked for in the message. In this state messages will be sent to other participants to have them produce the missing output.

- Comple

  The sta
  that ha
  sent b

- Abo

  The
  “co
  by
  h
  ‘



48

5 Information analysis

The information that is exchanged between participants during a building project can be divided into three parts: (1) product information, (2) the task description and (3) the envelope of the message. The current information representation is described in this section in order to create a basis for defining messages that are exchanged under perhaps different circumstances. While describing the exchanged information the conditions are pointed out that are required for consistent information transfer. These conditions are formulated assuming that the received information is interpreted without human interference.

5.1 Product information

Products are the output of the business system of a participant as we have seen when describing a participant. The research is focused on information exchange about products and not on the exchange of the products themselves. Products can be divided into physical products such as floors and abstract products such as designs. A distinction between the industries and for instance banking or insurance companies is that the activities that are performed always contribute to the creation, maintenance or demolition of a physical product. In object-oriented design there is a tendency to depict physical objects that can be found in the real world as the basic entities. This assumption seems quite natural but it imposes a strong structuring impact on information modelling. As a consequence project references such as project name and project location for example, cannot be modelled as attributes of a project object, which is an abstract object. Instead the building and the building ground are introduced as (physical) objects, and project name and project location respectively are the attributes.

5.1.1 Object types and aspect types

Before elaborating more on a modelling technique we will take a closer look at the way the architectural designer structures his information. Boekholt [Boekholt 1984] states that an architectural designer solves his design task by dividing the universe of discourse into a limited set of hierarchically ordered building objects. Building objects consist of two mutually exclusive sets: *physical objects* (e.g. floor) and *spatial objects* (e.g. kitchen). Building objects perform functions such as bearing loads or providing a cooking area. Designing can be considered as translating functional requirements into building objects. Lucardie [Lucardie 1994] puts forward the notion of functional object types. Objects are equivalent if they are equivalent in achieving a specified function.

So the possibility of attaching a *function* to an object appears to be an inevitable condition. Eastman [Section 3.2.5 on page 21] likewise centres his design model around a functional entity. Specifying functionality of an object is often not clearly separated from specifying the materialization of the object. Objects are usually described as a mixture of functional requirements and material characteristics. A functional description of a wall could be: a layer that offers resistance against thermal transport and acts as a water shield. These two functions can be achieved by a limited set of materials. The architect is free to choose from the available set. A mixed description of a wall is: a brick wall. The architect is limited to the set of brick walls, but still free to choose the size of the brick, its color and so on. The principal expects the architect to know what functions have to be fulfilled by the brick wall. These functions are not described explicitly; it is the responsibility of the architect that at least all building regulations are met. Each participant in the project is responsible for achieving a specific set of functions. *Activities are the processors in a transition system that translate functions into objects*. Although the principal describes a brick wall, he may leave the choice of brick open for discussion. If this should be so, it must become clear from the message the architect receives. Whether a message contains functional descriptions, partial functional descriptions or materialized descriptions of objects depends on the restrictions that are imposed upon the domain of possible solutions specified by the sender of the message.

Translation of functional specifications into building objects requires information about these objects. This information is stored in a set of *aspect types* (e.g. geometry, material) that is unique for each object type. An object has an object name. The important thing is that an object name does not refer to a constant set of aspect types. In other words: *object names can have different meanings during the project*. This outcome is in accordance with the fact that each participant will create his own view of a building object. The definition is dependent on the activities within the activity networks of two participants that will communicate with each other. Objects and aspects must not be confused with each other. An object floor with aspect material and aspect value concrete, describes a concrete floor. An object concrete, describes all concrete that is used. This example shows that the same noun can be used in very different senses in an object name and in an aspect value. Aggregate aspects (e.g. geometry) can be decomposed into other aggregate aspects and/or basic aspects (e.g. shape, location and shape representation). At the lowest level of decomposition the range of a basic aspect is a typed set. A typed set is for instance [x-coordinate: Q, y-coordinate: Q], where Q is the set of all rational numbers. A very complex type is a geometry description such as DXF [Autodesk 1992]. Some aspects are expressed in units (e.g. W/m^2). Aspects are the parameters that are used for evaluation of building products. To be able to compare objects (aggregated and basic) aspects must have a univocal definition. These definitions form a *normalized aspect type library* and should be part of the building regulations.

Some aspect types will have limited ranges for safety or durability reasons. These limited ranges are fixed by the government and can be maintained separately from the aspect types themselves. Conformance checking can be applied as a separate activity or as an integral part of a design application. In the information exchange process the conformance checking activity is described like any other activity in the building process.

To maintain consistency on the one hand and flexibility on the other, object names and aspect names should be chosen carefully. Building objects as used in natural language are often referenced by a combination of a location description or a material description or a function description and an object name (e.g., front door, brick wall, living room). Object names may not conflict with possible aspects. As a consequence references to material, location, function, etc. are not allowed in object names that are used in a formal description. However, many object names include a reference to a material (e.g. paint), location (e.g. roof) or function (e.g. piles). As a rule adjectives are forbidden and neutral object names are preferred. A *building object* is considered unique if it has a unique set of aspects. This definition of a building object allows redundancy of information since object names can reference aggregations (e.g. building) of other objects (e.g. rooms, walls, floors, etc.). A second appearance of redundancy is that aspect types (e.g. the heat loss) can sometimes be derived from other aspect types (e.g. the heat resistance). *Redundancy in a message is necessarily allowed simply because the recipient demands aggregated and derived information.*

A physical object can be in either of the *virtual* or *physical* state. As long as the object only exists in the CAD system, it is still virtual. Afterwards, when its counterpart is created on the building site, a new object comes into existence with the “physical” state. Because it is the same object, they will occupy the same space. If information about an object in both states is available, then checks can be made whether aspect values in the “virtual” state meet the corresponding values in the “physical” state. Measuring aspects of realized objects seldom occurs. During the maintenance phase however, design information is often updated with the actual information. Such an information system can be used for facility management of the building.

During the design phase often several design *alternative* is presented for a single design problem. Design alternatives can be considered specializations of a building object. The ‘root’ object only references functional aspects, whereas the specializations only reference geometrical and material aspects. The same approach towards describing the design problem can be found in [Boekholt 1984] and [Tolman, et al. 1989]. Activity execution generates results for each design alternative, unless selection of an alternative is part of the activity.

Identifying an object within the set of objects that are contained in a message can be realized in three ways:

1. Pointing out the object virtually.

   "Virtual" identification is done using a pointing device from a CAD system. By navigating through the 2D or 3D model the object that is looked for is selected by selecting its boundaries. Unfortunately, there is no link between the "virtual" and the "physical" representation of an object. There is no guarantee that the information about an object in the "virtual" state is consistent with the information in the "physical" state.

2. Tagging an identification to the object.

   A solution for the missing link between the virtual and the physical representation of an object is tagging an identification code upon the physical object and storing this code with the object description. This method is well known under the name of bar code. Bar codes are a very appropriate solution for identifying industrial products such as prefabricated doors, etc. The bar codes are usually tagged on the container of the product itself. Thus, the building product will lose its identification as soon it is in its place. For tracking down moving objects such as furniture, a permanent barcode is a suitable solution.

3. Selecting the object by a unique set of aspect values, the state value and information storage time.

   If a set of aspects values gives an exact match with the set of aspect values belonging to one object, then the object is uniquely identified. An object can have two appearances if both states are available. The set of aspects as well as the aspect values of an object changes during its life cycle. A message contains the most recent information available at the time of sending of the message. This information is valid as long as the object is not part of the output of an activity that is in the "executing" state (see Figure 26 on page 48).

An import aspect of a building object is the geometry description. Geometry is defined by a shape description and a location description. The preferred shape description depends on the kind of information that has to be transferred. Cross-sections of a building object are often used to describe the assembly and the material usage, views show how an object will look when it is finished. Shape descriptions are often abstractions of reality. For instance, a structural engineer will translate the architect's cylinders representing columns into line segments for the sake of the structural analysis. Unfortunately, designers and engineers rarely construct their models from three dimensional elements, which is a complicating factor in keeping the information consistent. Instead, the complete model is merely represented by cross-sections and views. It is up to the knowledge of the reader of the message to construct a three dimensional model from this information. On the other hand, it is obvious that in the early design stages three dimensional representations are simply not available. Walls for instance are represented by line segments just to concentrate on the area and location of rooms in the first place. The thickness and height of the walls will be entered later on in the design process. Apparently shape descriptions must permit two dimensional and three dimensional representations. One object can have more than one shape description, like for example a front view and a cross-section. Objects need not have any shape representation but just be listed as being present somewhere (e.g. nails).

To put shapes in their position, a location description is needed for each shape and a general reference point. A point somewhere on the building ground of which the position is exactly known by the town planning authorities could serve as a reference point. It is very important that all objects that have a shape description also have a location description that refer directly or indirectly to the same general reference point. This is an inevitable condition to enable applications to do checks on object interference problems. The implication of this condition is that cross-sections, views, etc. are actually located in the right place and position in space. Moreover, all shape descriptions must refer to a building object, otherwise they are meaningless. In current designing practice these conditions are hardly met by any design system.

Geometry representations such as Boundary Representation, Constructive Solid Geometry and Surface Modelling are often used in combination. It is evident that whatever representation method is used, the receiver of the message must be able to interpret it. In other words: a CAD application must be available to interpret the geometry representations of the objects. A major effort to create a CAD vendor-independent geometry representation is part 42 of the ISO 10303 (STEP) standard [ISO/TC184 1993c].

Apart from the information about building objects that is required to start the execution of an activity, the presence of physical objects at a specific time and location can also be part of the input definition of an activity. There are two circumstances in which physical objects are needed:

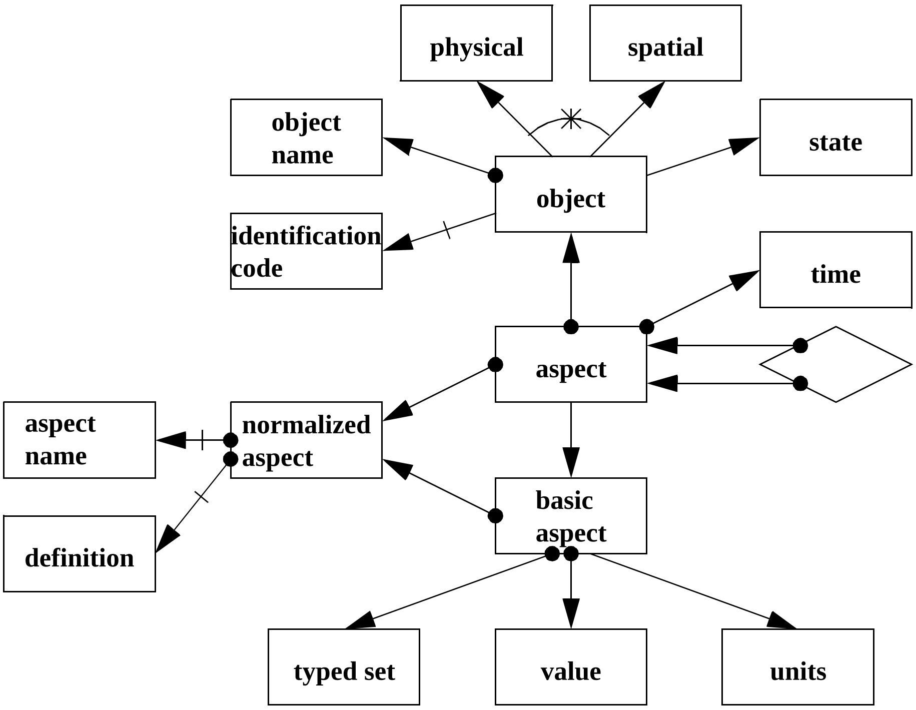
1. Use of resources.

   Equipment (e.g. a crane) that is required to perform a certain activity. The message that starts the activity must also contain information about the current location of the equipment. Before the activity is actually executed the availability of the equipment in the right spot can be checked.

2. Presence of realized objects.

   It is obvious that the construction of a building product takes place in a specific order. Physical objects (e.g. a foundation) must be in place before new activities (e.g. the placement of a floor) can be executed. In such cases the location aspect of such objects in the “physical” state are specified as an input object for the activity. The message that starts the activity contains the current information at the time of the message creation about the required realized objects.

FIGURE 27. Objects and aspects.



Most considerations about objects and aspects are presented in the functional data model of Figure 27. Central part is of course the object with one or more aspects associated to it. Note that the object can have any object name, whereas the aspect must be chosen from an aspect type set. Aspects are decomposed using the association entity. Every aspect has a time stamp indicating its creation or updating time.

5.1.2 Relationships

Relationships between object types are known from object-oriented technology, namely [Booch 1994]:

- Association

  The association relationship denotes a semantic connection. For instance: A column is bearing a floor. The verb “bearing” associates the column and the floor towards the structural behavior.

- Inheritance (generalization, specialization, is-a)

  An object type inherits the structure and behavior of the super object type. For example: The supertype of the object column describes the geometry and the material that is used; the subtype describes the location of a specific column.

- Has (aggregation, whole/part, decomposition, has-a)

  One object type is considered to be part of, or is physically part of another object type, e.g. reinforcement is part of a concrete column.

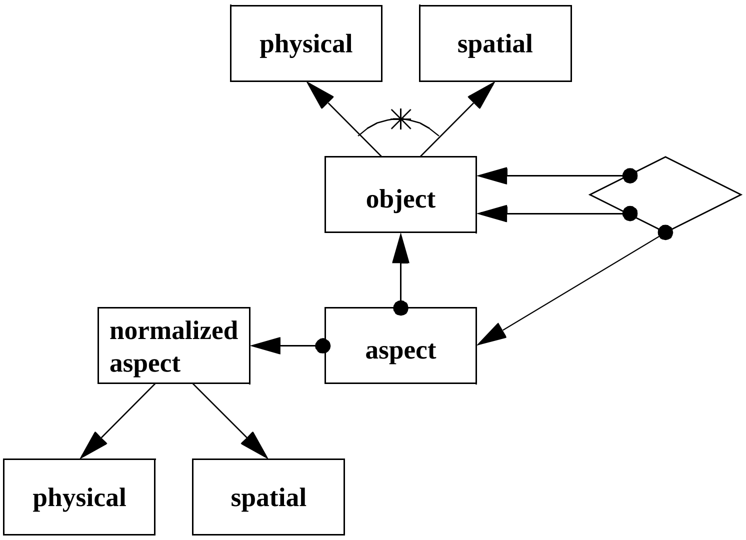
Relationships in a message at the logical level of data structure are used in a specific way to express more information about the building objects. The “inheritance” relationship and the “has” relationship are refinements of the “association” relationship. The “association” relationship is only used during object-oriented analyses and can be omitted for the description of the message data structure. The “inheritance” relationship is often used to store information in a more efficient way so that it takes less storage capacity and less computing power to retrieve information. A more fundamental use is the expression of design alternatives inheriting the functional aspects of the supertype. Inheritance used in this specific way keeps the product design as a whole. A less elegant method that is harder to maintain, is creating a separate message for every design alternative. The “has” relationship is really indispensable, because the receiver of a message containing this relationship could never deduce what the sender considers to be an aggregated object otherwise. Aggregation or reverse, decomposition is always used in a certain context. Saying that reinforcement is part of a column can mean that the reinforcement is “inside” the column. It tells us the location of the reinforcement in relation to the column. A second meaning can be that the reinforcement contributes to the structural behavior of the column, namely “bearing” the floor. Saying that the dining room is part of the living room means that the dwelling function of the living room also “contains” the dining function. In these examples “location”, “structural behavior” and “dwelling” are aspect types that are referenced by both related object types. In general one can say that decomposition of an object always takes place according to certain aspects. An inevitable condition is that the supertype and the subtype of the object share a common aspect type, which is the subject of the decomposition. The *decomposition with aspect reference* construction offers great flexibility in creating relationships with additional semantics to the (‘normal’) “has” relationship.

Object types are either “physical” or “spatial”. Decomposition of an object type into one or more other object types means that a “physical” object can reference “physical” objects and “spatial” objects. (E.g. a hollow brick wall can be decomposed into two layers of brick with space in between). Analogously a “spatial” object can be decomposed into “spatial” objects and “physical” objects. (E.g. a living room can be decomposed into dwelling space and furniture.) Looking at an object at a more detailed level often results in finding objects of both a “physical” and “spatial” nature. It depends on the sender of the message as to which view he will take towards a building object. Of course there is a difference in the set of aspect types that can be used for physical objects (e.g. the mass) and for spatial objects (e.g. the maximum number of persons).

Topology of a product description is very closely related to the geometry description. Topological conditions have two levels of abstraction:

1. The geometry description of an object. For instance a wall may be moved but not cut into parts. This kind of condition is entirely integrated in the geometry description.
2. A relationship between two objects. For instance a “column” is rigidly connected to a “floor”. In a message, the decomposition with aspect reference construction can be used for this purpose. An object type “construction” is introduced, and an aspect type “floor_column connection” that is referenced by all three objects. Furthermore a “decomposition” relation is introduced with a reference to the “floor_column connection” aspect type, from super type “construction” to both subtypes “floor” and “column”. From an instance of this data model can be deduced that the “floor” and the “column” share a common aspect, namely the “floor_column connection”. It is up to the sender of the messages whether he allows movement of the column while keeping the connection intact. Keeping the connection intact implies that there be no change in the value of “floor_column connection” aspect.

FIGURE 28. Relationships



Using functional data modelling in Figure 28, the object decomposition relationship is presented by an association entity. The association entity references an aspect from the normalized set of aspect types. Whereas an object is either “physical” or “spatial”, aspects can often be used in both circumstances but not always. A relationship always references an aspect that is common to the two related objects.

5.1.3 Constraints

Constraints can be imposed upon objects and relationships. Evaluation of a constraint requires aspect values. In general, a constraint is a boolean function with the aspects of an object or the aspect referenced by a relationship as its parameters. The function is expressed in some specification language. Constraints can be used among other things to:

1. Restrict the range of output values of an activity.

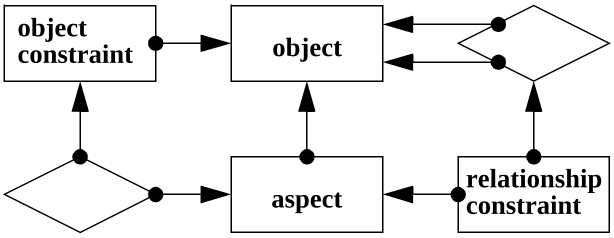
   An example of an object constraint is an aspect type that can be derived from other aspect types. The constraint contains the derivation formula which evaluates to false if the aspect values are not consistent. An example of a relationship constraint is that a door may be moved within a wall, but it may not get too close to the edge of the wall, otherwise the constraint evaluates to false.

2. Define the behavior of a product.

   If an architect moves a column, then he changes the structural behavior of the complete building. Structural behavior can be described using the finite element analysis for instance. If all geometrical and material aspects are known from the objects that the building product is made of, then the structural aspects of all objects and of the complete product can be calculated. The constraints are defined as a matrix, relating all structural aspects of each object to the structural aspects of all other objects [Alberts 1993]. Moving one object requires recalculation of the complete product to satisfy the constraints.

A message that contains restrictions of the output values is very useful in preventing the recipient from producing unrealistic, undesirable or impracticable output. Constraints will be rule of thumb formulas that often have come forward empirically and are generally known. Product behavior however, encloses a great deal of in-depth knowledge about a specific aspect. It is exactly the absence of that knowledge that is the reason for a participant to consult another participant.

FIGURE 29. Constraints



In the functional data model of Figure 29, an object constraint is associated to one or more aspects of a certain object. Each relationship constraint references an object decomposition relationship and a specific aspect. The relationship constraint in fact replaces the direct reference from the object decomposition to an aspect (see Figure 28).

5.2 Task description

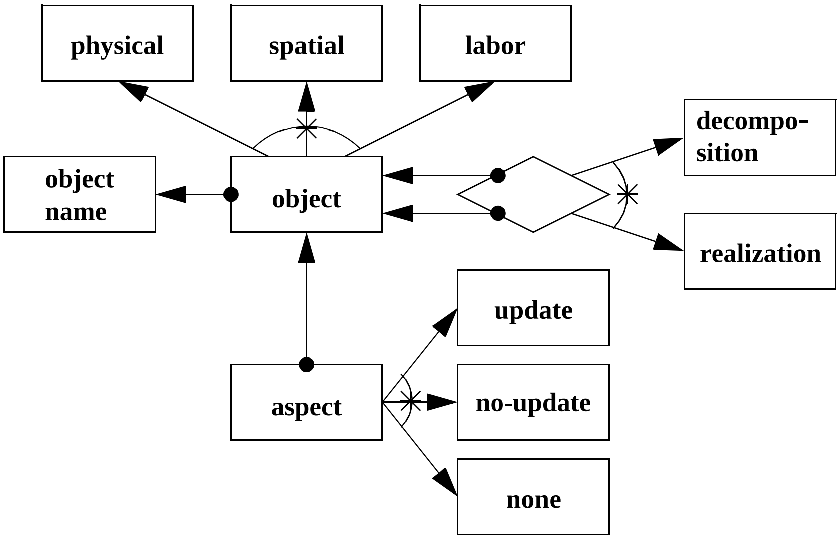
To create a new product, describing its end state is often not sufficient. The quality of a product not only depends on the materials that are used but also on way it is realized. Control over the realization is gained in two ways:

1. Cut the complete realization process into parts so that the result of each part can be inspected. Each process part is called a *task* and it is fulfilled by executing an activity. Tasks are defined by adding an *operation* to an object. More precisely, the aspects of an object get an operation attribute. A task is fulfilled to satisfy the lack of information at some point during the building project. One participant will search for another participant that is able to produce the required information or part of the required information. There are three operations: *update, no-update* and *none.* An aspect with an "update" operation requires that the aspect values be updated by the recipient of the message. An operation "none" means that updating is allowed but not required. "No-update" permits no changes of the aspect value. The result of a task that will be contained in the reply message, consists of new information. Continuation of a process does not always require new information. Sometimes the missing information can be acquired from another participant. If a participant is found who possesses the required information, then a *request* will be sent to retrieve the information (e.g. product information from suppliers).

2. Describe the realization process, not by defining the end state, but by making the labor (man or machine) explicit. For this purpose an extra object type category next to "physical" and "spatial" is introduced, namely *labor*. Moreover an extra relationship is introduced next to "decomposition", namely *realization*. The instance of an object type of the "labor" category is a building activity such as digging, painting, bricklaying, etc. These objects often have only one aspect, namely cost, or no aspect at all. They almost always have a "realization" relation with a building object of the category "physical" (e.g. painting is related to a wall). To create generic object names, references to material are not allowed. This condition occasionally causes problems because many building activities are strongly related to a certain material (e.g. bricklaying). Using "labor" to describe a task does not supply any information about the order of the building activities. The recipient of the message is free to choose the construction sequence that suits him best.

3. Describe the realization process using a formal language, for instance Petri nets [Jensen 1991] or State Transition Diagrams [Rumbaugh 1991] or EXPRESS-P [ISO/TC184 1995]. A task can be described as a process to be executed. The message model therefore must contain a formal data description as well as a formal process description. The recipient executes the process description as defined by the sender. This approach of fulfilling a task gives much power to the sender of the task and requires much flexibility of the recipient. A simple example of such a task description is the assembly direction that comes with some building products.

FIGURE 30. Task

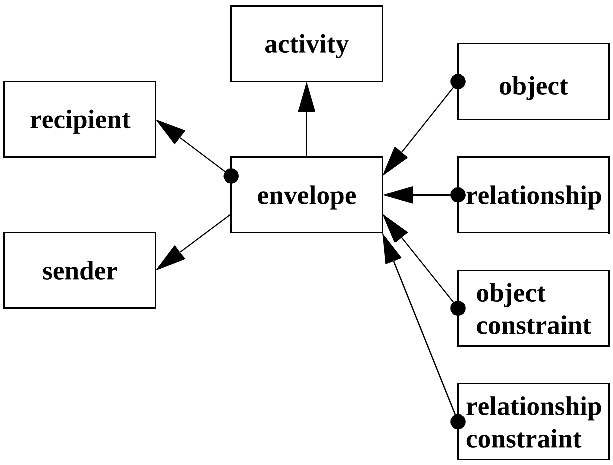


Since process description can evidently not be represented by functional data modelling, the third task description method is not presented in Figure 30. The association entity that relates two objects now can be of type “decomposition” or “realization”. Every aspect of each object in a “task” or “request” description will have one of the three operation attributes.

5.3 The envelope

Like a postman needs an envelope to deliver mail, an electronic message needs information to find its destination. Because every “new” message results in a “reply” there is also a reply address (see Section 4.6 on page 41). Messages must be uniquely identified within the project because they can activate concurrently running activities. Each of these activities will reference the information of the message that started them off. Additional information that covers aspects of the participant such as address, telephone number, bank account, etc. are also included. No information about any physical or abstract product should be listed here because this information is part of the contents of the envelope. “New” messages will also contain the name of the activity that will be executed by the recipient.

FIGURE 31. The envelope

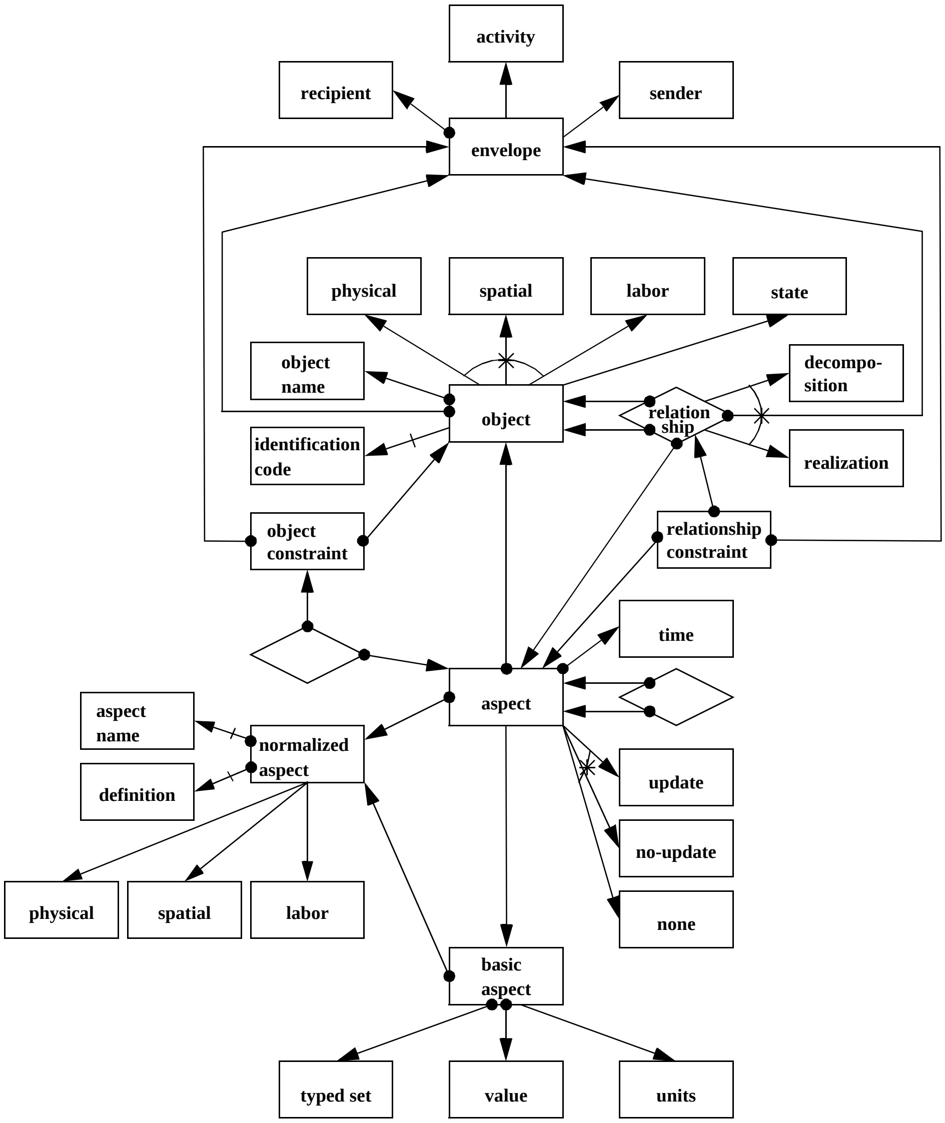


The functional data model of Figure 31 shows the envelope with its attributes and the contents of the envelope, the product model. The product model is represented by the basic concepts of object-oriented modeling, namely objects, relationships (between objects) and constraints (on objects and relationships). “Reply” messages do not contain the sender entity and the activity entity.

5.4 The message data model

Finally, all functional data model parts are integrated into one message data model. The result is presented in Figure 32.

FIGURE 32. Message data model

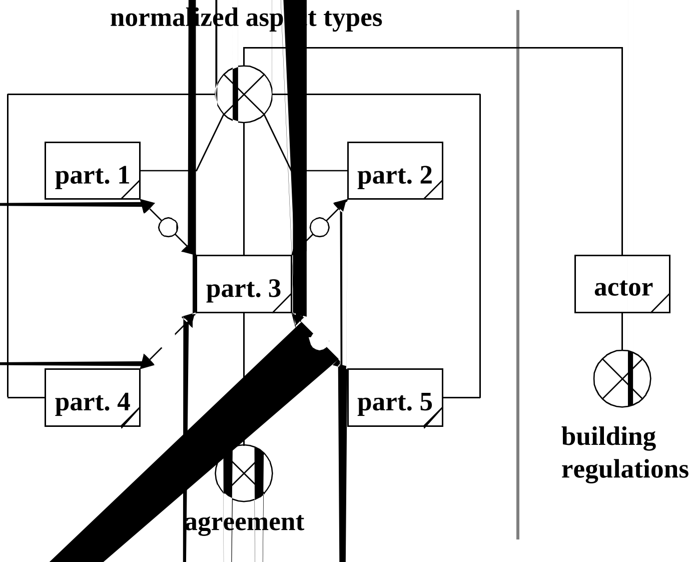


6 Message exchange model

In this section the concepts that have been introduced for defining the information exchange process and/or the information itself will be combined into one *message exchange model*. With this model it is possible to explicitly describe the information that is being used at any moment during the building project. Process and information definition are generic, because they should hold under all circumstances. Adapting the message exchange model to practice is done by instantiating the presented schemes to 'real' building processes and information packages. Examples will show what this will look like. To make a 'good' choice from the available communication methods, a classification proposal is made for messages and transfer media. Finally a development strategy is worked out for the introduction of the electronic information exchange.

6.1 Process and information integration

FIGURE 33. Pre-project process model



Before a project starts, an agreement has to be reached about the jobs that each of the participants will be responsible for. In the pre-project phase one of the participants will coordinate and negotiate about the jobs to be fulfilled. In Figure

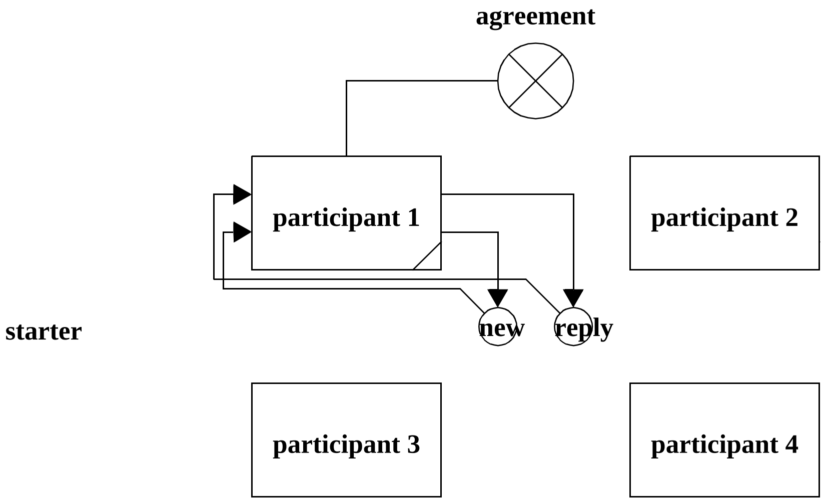
33 this role is played by participant number 3. He will send other candidate participants the product description in its start state and in its end state. Participants that are interested in participating in the project will reply with their activity definitions that can contribute to the creation of the building product. The coordinating participant then has to check whether a set of activities exists that can produce the building product from its start state to its end state. The check can result in the following conclusions:

1. The project does not reach the end state because of a lack of information. Solutions for this problem are:
   a. The missing information must be gained as a “request” from a participant that has to be added to the project.
   b. The missing information has to be produced as a “task” by a participant that has to be added to the project.
   c. A participant must be substituted by another participant that is able to continue without that specific information.
2. The execution time of a project exceeds the available time to create the building product. Solutions for this problem are:
   a. Activities have to be speeded up by adding more resources. Of course this will also effect the cost of the product.
   b. Activities must be replaced by completely different ones that can do the job in less time.

The final agreement is stored in the agreement store of Figure 33. Agreements are composed of activity definitions. Activity definitions are made up of objects and their corresponding aspects. Object names depend on the participants view. Aspect names are uniquely defined in a normalized aspect library. The maintenance of the aspect names and their formal definitions resides under the authority that is also responsible for the building regulations.

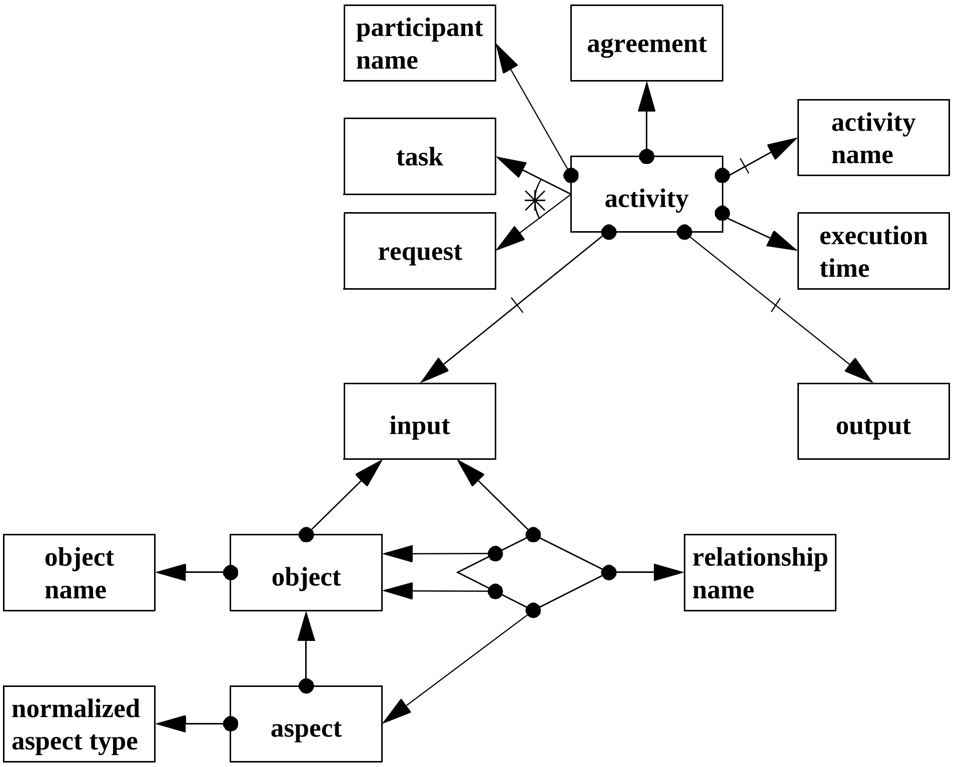
For the actual project, a communication topology (tree, star, black box) is chosen depending on the organizational structure of the building project [Section 4.2 on page 29]. The communication topology determines which mapping situations (object types and relationships: identical or different) are applicable [Section 4.3 on page 34].

FIGURE 34. Project process model: example of four participants (only one of them is connected)



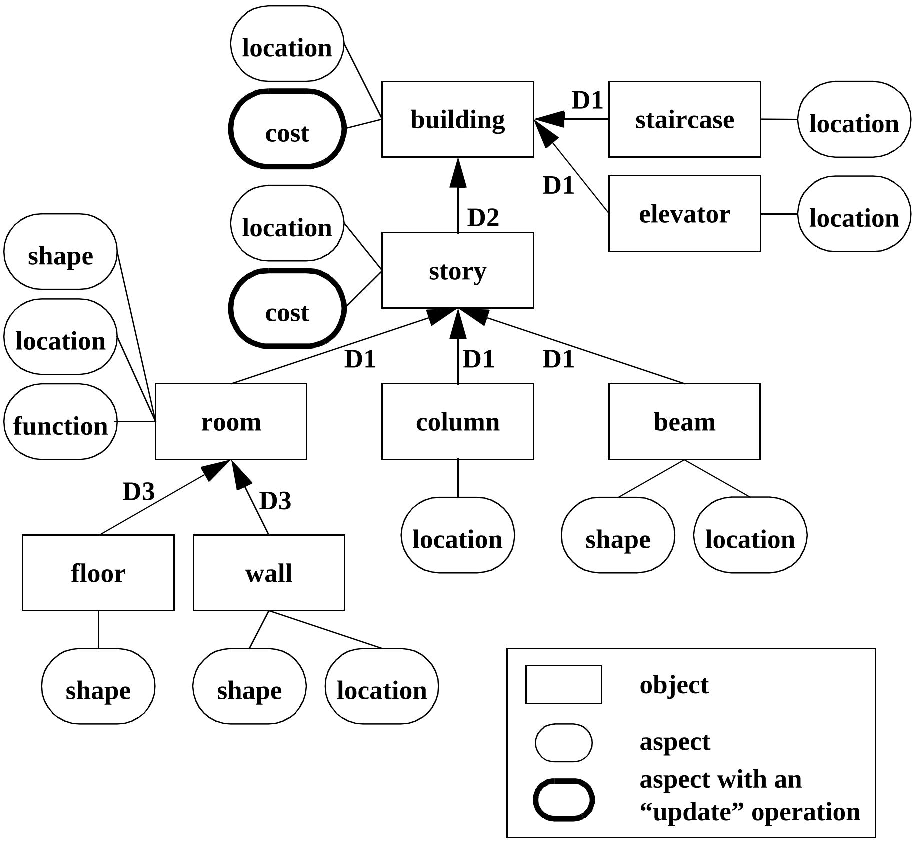
When the agreement is set the project can start. All participants are connected by four channels: one channel of type “new” and one channel of type “reply” on the input side and on the output side. The “starter” of the process sends a message of type “new” that contains the start state of the building product and the desired end state of the building product.

FIGURE 35. Agreement data model (the output part, which is equivalent to the input part, has been omitted)



The agreement in Figure 35 specifies the required input and the produced output for each activity. The output definition contains only objects, aspects and relationships that are updated or added. Aspect names are retrieved from the set of normalized aspects. Possible relationship names are: “decomposition” and “realization”. Every activity gets an execution time. Constraints are not part of an activity definition. Constraints reduce the possible set of solutions, but do not add inevitable information for execution of activities. The input and output parts of the data model are meta descriptions. They define how data models about building product information must be structured in the message exchange process.

FIGURE 36. Example of an instance of the message data model



As an example, part of an instance is shown in Figure 36 from the meta data model (see Figure 32 on page 62) for a message from an architect to a quantity surveyor (compare with Figure 19 on page 38). The entities from the message data model that are instantiated are: object, aspect and relationship. The relationships that have been introduced are:

- D1: Decomposition of the “location” aspect.
- D2: Decomposition of the “cost” aspect.
- D3: Decomposition of the “shape” aspect.

Simultaneously, the “location”, “cost” and “shape” aspects were introduced at the related objects.

An instance of the message example is presented in two tables. Table 2 for the objects part and Table 3 for the relationship part.

TABLE 2. Instance of the message example: objects

object name	nr	aspect name	type	aspect value	units	operation
building	1	cost	Q			update
building	1	location	$Q \times Q \times Q$			no-update
staircase	1	location	$Q \times Q \times Q$			no-update
room	1	shape	DWG	{room1}		no-update
room	1	location	$Q \times Q \times Q$	(10,25,0)	m	no-update
room	1	function	string	office		no-update
room	2	shape	DWG	{room2}		no-update
room	2	location	$Q \times Q \times Q$	(15,25,0)	m	no-update
room	2	function	string	office		no-update

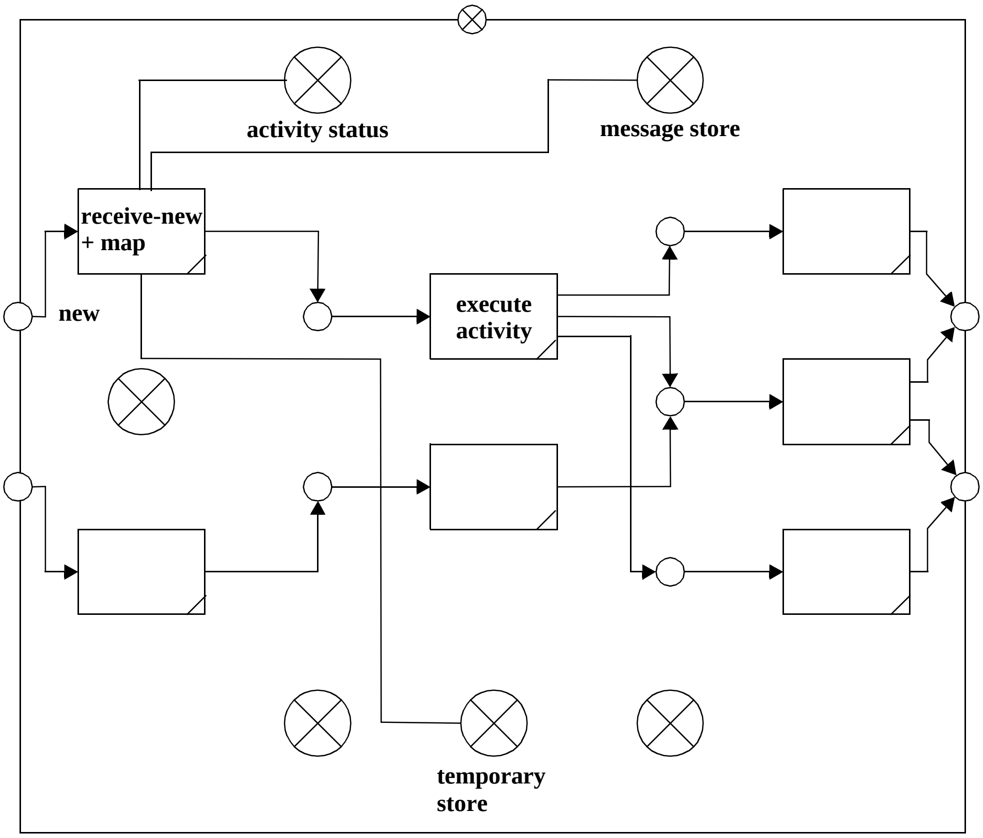
TABLE 3. Instance of the message example: relationships

relationship	object 1	nr	object 2	nr	aspect
decomposition	building	1	staircase	1	location
decomposition	building	1	elevator	1	location
decomposition	building	1	store	1	cost
decomposition	store	1	room	1	location
decomposition	store	1	room	2	location
decomposition	room	1	floor	1	shape
decomposition	room	1	wall	1	shape
decomposition	room	1	wall	2	shape

In the table representation an extra field “nr” is introduced to relate individual objects to their aspects [Table 2] and to mutually relate objects [Table 3]. The “nr” field is necessary because the object name is not a unique identification of an object (see Figure 32 on page 62). Aspect type *Q* stands for a rational number and DWG stands for a geometrical description format [Autodesk 1992]. In the case of a shape description, the aspect value contains all geometrical data representing that specific object. Null values as aspect values indicate that the aspect is present, but not yet known. The “reply” message that is sent as a consequence of the “new” message described here, will contain only the object/aspect combinations that were marked with an “update” operation. The relationship of type “decomposition” between two objects references an aspect. A “realization” relationship is not part of the example. Both tables show just part of the complete set of all objects that are present in the architect’s design.

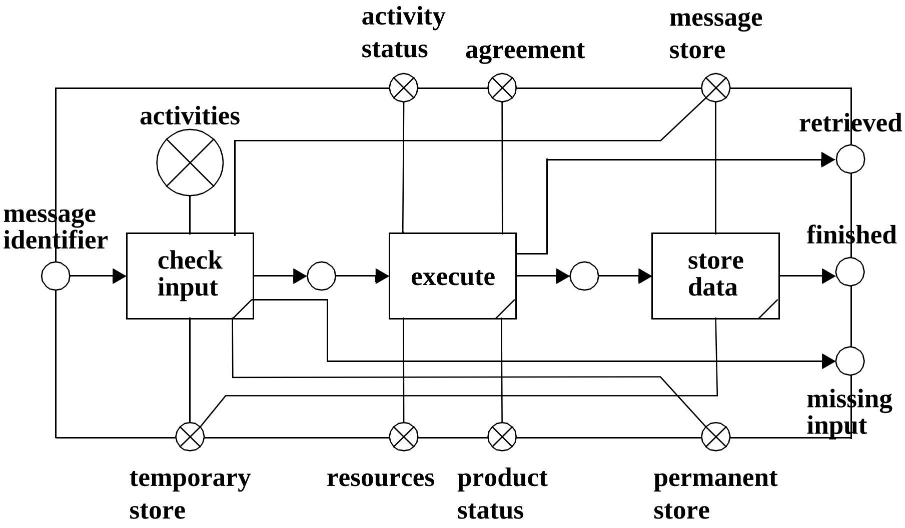
The participant’s system of the project process model [Figure 34 on page 65] is decomposed into seven subsystems and five internal stores. The transition system diagram will be explained in Figure 37 to Figure 41. Sending and receiving messages is preceded respectively followed by mapping the data structure of the temporary store and the message store to the message data structure [Section 4.3 on page 34]. Whether mapping is necessary at all and what kind of mapping is determined along with the definition of the agreement. A mapping example will be shown at the end of this section.

FIGURE 37. Participant process model: receipt of a new message



Every “new” message that arrives contains information that needs to be stored. If the message contains objects that have more recent aspects than the equivalent objects that are available in the “temporary store”, then the “temporary store” is updated with these data. New objects are added to the “temporary store”. The “message store” gets an exact copy of the message that is received. When a “new” message arrives, the “activity status” store gets an entry indicating that a “task” or “request” is yet in state “not completed”. After storing a “new” message, a token containing a message identification is forwarded to the “execute activity” process. The “execute activity” system is decomposed into the transition system of Figure 38.

FIGURE 38. Execute activity process model



In case of a “task” a check is made whether all required input information is available. The activity name and type are retrieved from the “message store” by the “message identifier”. The “temporary store” and the “permanent store” are checked for available input data. The definition of activities in the “activity” store of a participant can be the same as the activity definitions that are part of the “agreement”, but this need not be so. The “activities” store contains the core business definitions that are project independent. The “agreement” contains the activity definitions that are utilized for the specific project. Only activity definitions of type “task” can be dissimilar. Differences can appear on the input side and on the output side.

TABLE 4. Example core business and agreement definition (1)

core business activity				agreement activity			
in		out		in		out	
object	aspect	object	aspect	object	aspect	object	aspect
labor	cost	floor	cost	labor	cost	floor	cost
concrete	cost			floor	geometry		
floor	geometry						

Table 4 shows an example of an agreement definition of an activity with less input than the core business definition. In this case the missing input data have to be acquired by sending a “request” to another participant that has the missing data in store (a concrete factory is asked for its current price).

TABLE 5. Example core business and agreement definition (2)

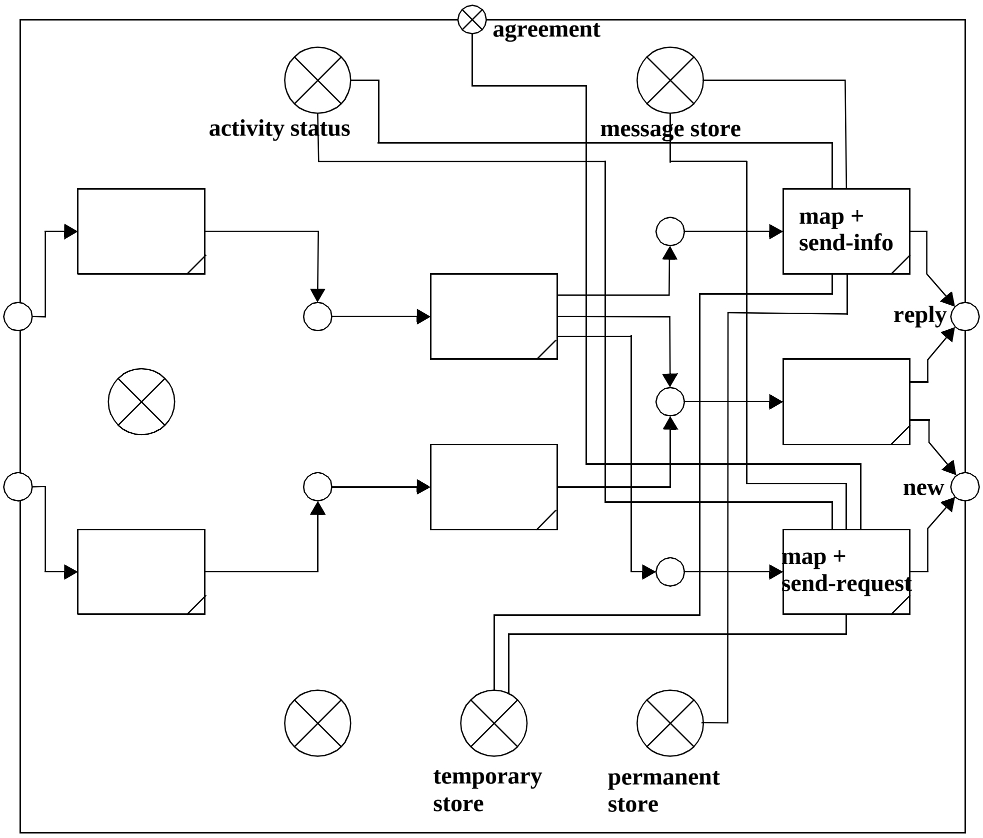
core business activity				agreement activity			
in		out		in		out	
object	aspect	object	aspect	object	aspect	object	aspect
labor	cost	floor	cost	labor	cost	floor	cost
concrete	cost			concrete	cost	transport	cost
floor	geometry			floor	geometry		

Table 5 shows an example of an agreement definition of an activity with more output than the core business definition. In this case a “task” is sent to another participant that is capable of producing the missing data (a transport company is asked to calculate the transport cost).

To continue with the execute activity process model of Figure 38, if not all required input is available, then a token is forwarded to the “map + send-request” process via the “missing input” channel. If the input check is passed then the activity is executed and the “activity status” is changed to “executing”. Execution takes as much time as is specified in the “agreement”. During the execution resources are allocated from the “resources” store. Within the activity network of a participant extra preconditions can be defined for starting an activity by using the “product status” store. The “product status” store is checked at the start of an activity and it may be updated at the end of the activity. The newly produced information is stored in the “temporary store”. In the “message store” the “task” is updated with the newly generated objects and aspects. After finishing the activity, a token is forwarded to the “map + send-task, send-result” process via the “finished” channel.

In case the received message has started an activity of type “request”, then the input check is passed, the request is executed and a token is forwarded to the “map + send-info” process via the “retrieved” channel. Since there is no new information generated, there is no “temporary, permanent or message store” updated.

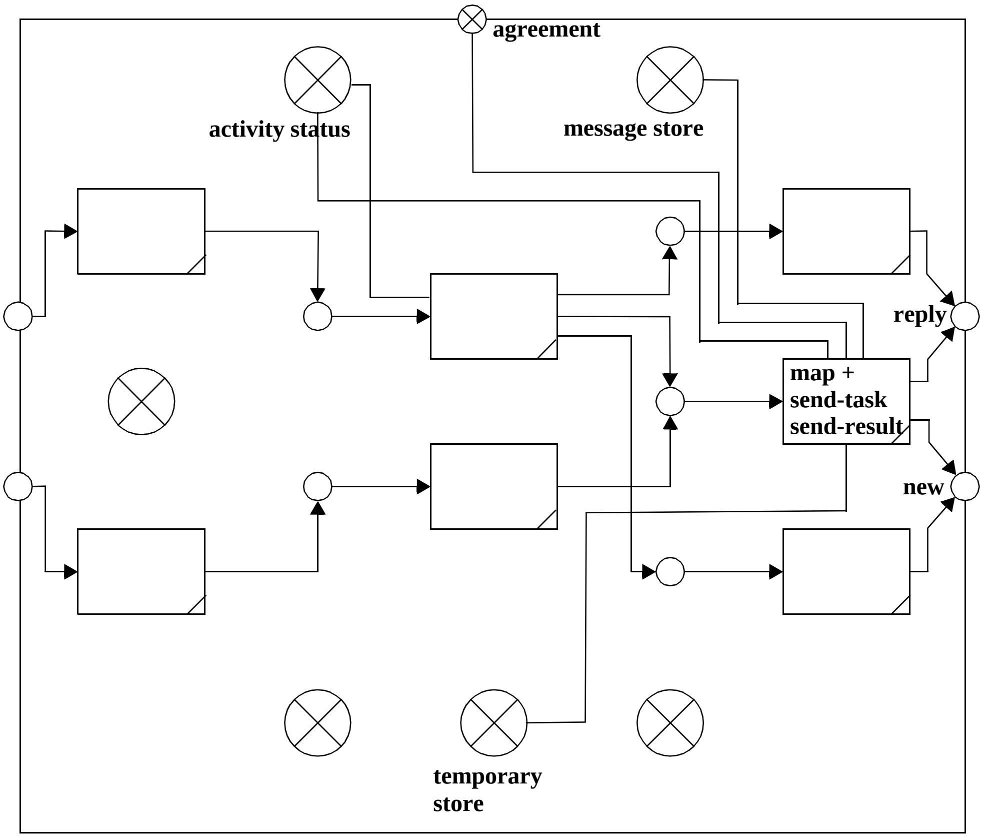
FIGURE 39. Participant process model: information request or reply



In the “map + send-info process” a message is created that contains the requested information. The requested information is retrieved from the “temporary” store or the “permanent” store. The “activity status” of the message that started off the request is changed to “completed”.

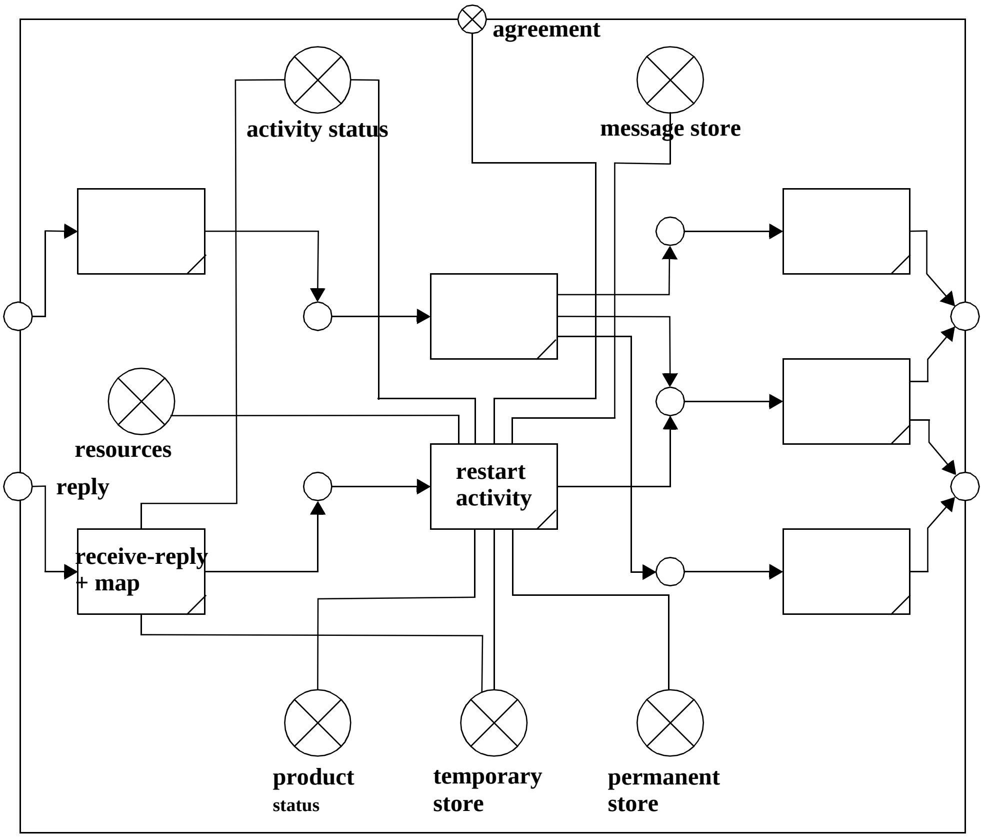
In the “map + send-request” process an activity is selected from the “agreement” of type “request”, that can deliver the missing input. Of course the required input for the selected activity must be available at this point in the “temporary store”. If no activity is found, the process halts. This situation does not occur if the agreement is defined correctly. To send a “request” a message is created containing the required input and the object/aspect combinations with an “update” operation specifying the requested output. The “activity status” of the message is set to “not-completed”.

FIGURE 40. Participant process model: task request or result

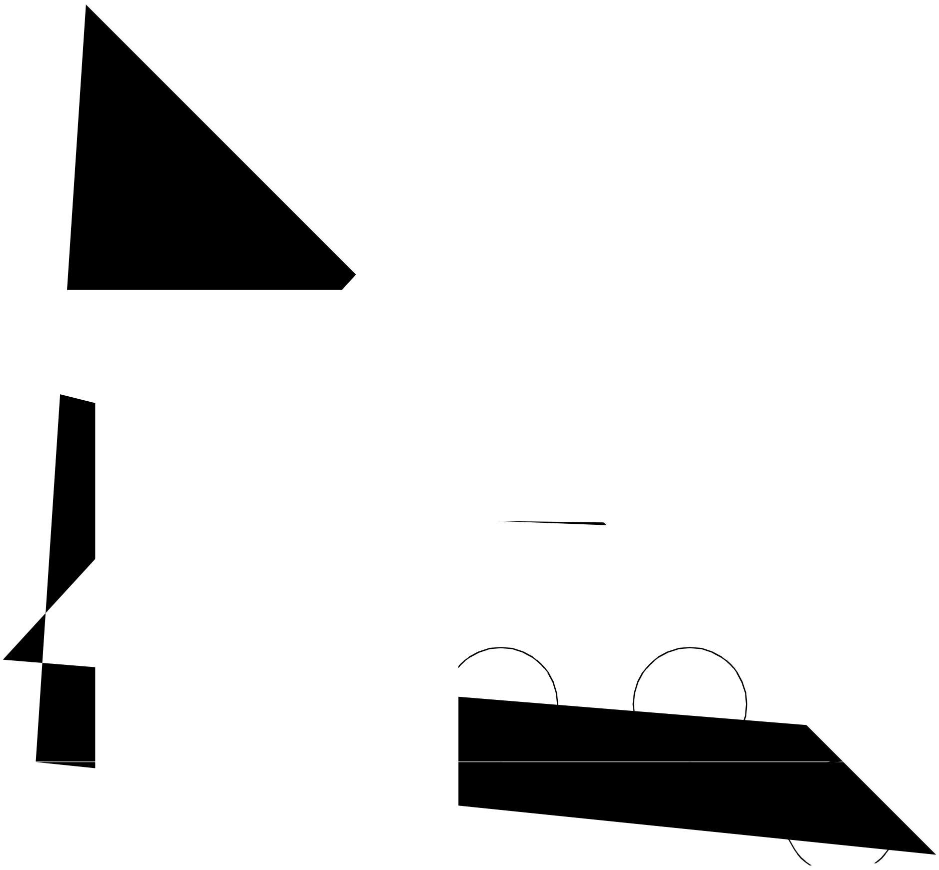


In the “map + send-task, send-result” process, a check is made whether the “task” that just has been finished, has been finished completely. A “task” that is in state “not-completed” has been updated partially. In that case there is still missing output. The agreement is searched for activities that can produce the missing output. A message of type “task” is sent containing the required input from the “temporary store” and the object/aspect combinations with an “update” operation specifying the requested output. If no activity is found, the process halts. This situation does not occur if the agreement is defined correctly. If a “task” is fulfilled completely, the state in the “activity status” store is changed to “completed”. A “reply” message containing the result of all activities retrieved from the “temporary store” is sent to the participant that started off the single activity or the sequence of activities.

FIGURE 41. Participant process model: receipt of a reply message



A “reply” message contains new information that must be stored in the “temporary store”. At the same time the “activity status” of the message that preceded the received message is changed from “not completed” to “completed”. In the case that the preceding message was a “task”, the message copy of the preceding message is updated with the newly received objects. Reception of a reply will produce a token that is forwarded to the “restart activity” process. This process will check the “activity status” store for all messages that have been received from other participants containing a “task”, that are still in the “not completed state”. With the newly received information available maybe one or more of them can be executed. If so, then the process that follows is analogous to the “execute activity process”. Nevertheless, the only possible outcome of a restart is a token in the “finished” channel (see



ırizes in which states the messa a
l. It also shows that the state tra ′
ined in the message (“task” or “
ved (“new” or “reply”). The act
cursive procedure that will con
und that can produce the missin
. The selection priority is a foll

one activity is available to pro
contributes the most data to the

than one activity is available tc ı
with the shortest execution time

than one activity is available, tl

Only the “execute task” and “execute request” states are time consuming. All other states will be passed instantly.

After the project has been completed a selection is made from the information in the temporary store that will be moved to the permanent store. Which information is to be stored permanently depends on the use of filed information by applications in new projects.

To conclude the process and data integration section, an example will be shown and explained in more detail of the mapping of data structures. (For an overview of software tools that support data structure mapping, see [Verhoef 1995]). As stated before the mapping situation that exists is determined along with the agreement definition. As an example we will again look at the data structures of the architects preliminary design [Figure 19 on page 38] and the data structure of the calculation of the quantity surveyor [Figure 20 on page 39]. Solutions to overcome the differences are:

1. Define a mapping algorithm in two directions.

   For one direction the example is worked out in Section 4.3.

2. Adapt a ‘neutral’ data structure for both the architect and the quantity surveyor as a message data structure.

   The architect and the quantity surveyor have to define mapping algorithms to the ‘neutral’ data structure.

3. Add more information to both data structures, so that matching of objects is possible.

   Following the guidelines for message data structures as described in Section 5, the data structure of the architect and the data structure of the quantity surveyor are changed accordingly. This solution is worked out in more detail below.

FIGURE 43. Example data model store architect

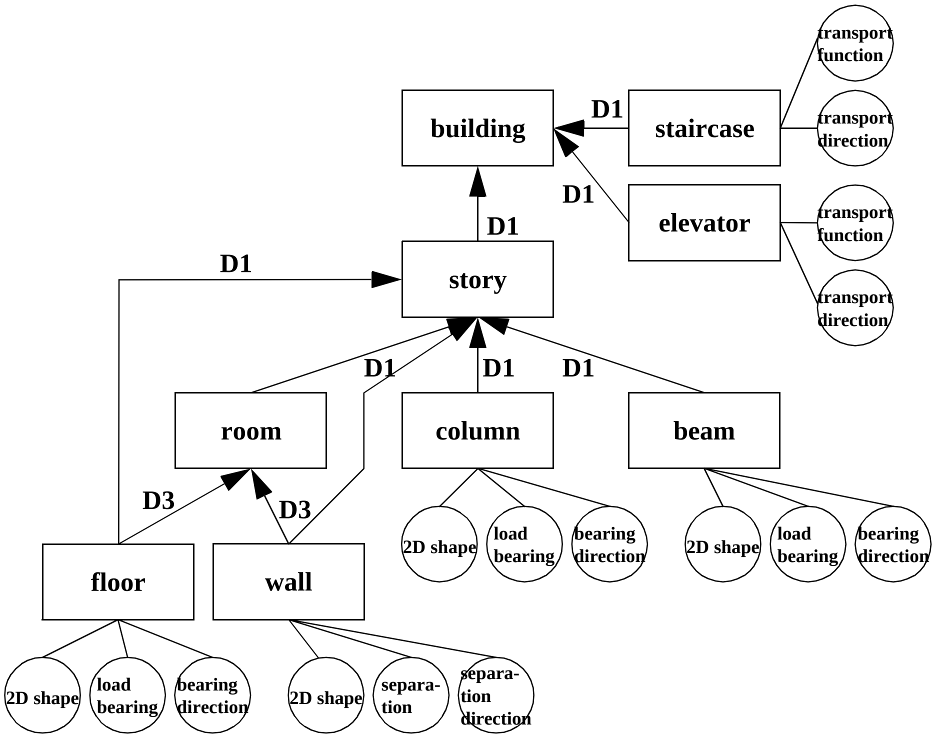


FIGURE 44. Example data model store quantity surveyor

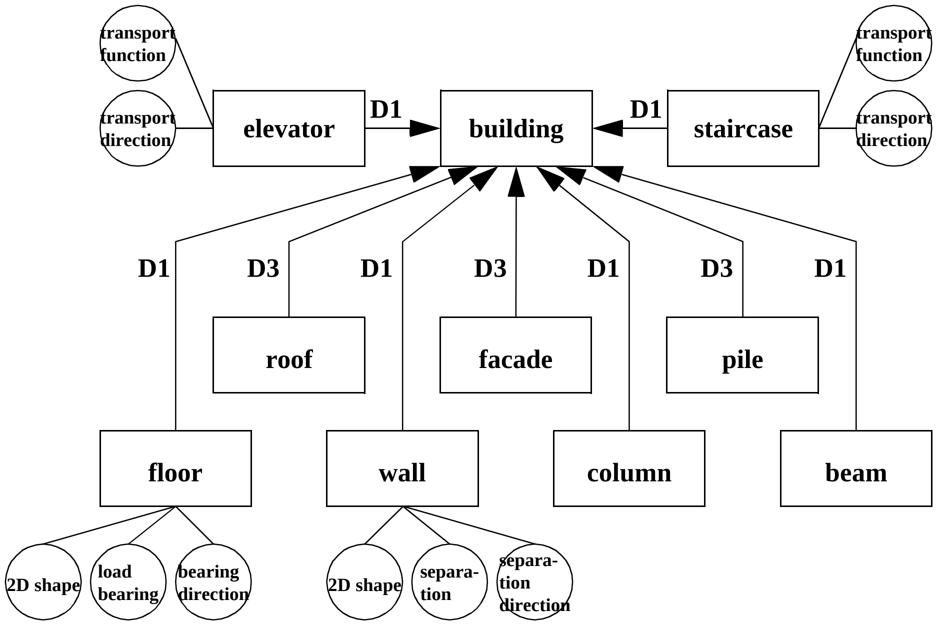


Figure 43 and Figure 44 show the adjusted data structures that respectively the architect and the quantity surveyor should adapt, to allow for data structure mapping in both directions. In comparison to the original data structure, the entity types “surface element” and “line element” in the quantity surveyors data structure are removed because they are not “physical” nor “spatial” object types. Instead of the “level” entity in the original data structure of the quantity surveyor that was used to determine the differences in for instance a floor and a roof, the actual objects are added: roof, facade and pile. Functional aspects are added in both data structures to be able to split up objects and to combine objects. For instance the “floor” object of the architect is split up into another “floor” object and a “roof” object at the quantity surveyor. This can be done only by looking at differences in the aspect values of the “load bearing” aspect and “bearing direction” aspect. In this way the quantity surveyor himself defines exactly what he ‘understands’ to be a roof and what is a floor. Relations are used to determine if an object is inside another object or not. For instance, an architect’s “column” with a “decomposition” relation with “location” reference (D1) to a “story” is also a quantity surveyor’s “column”. An architect’s “column” with a “decomposition” with “2D shape” reference (D3) to the “building” is a quantity surveyor’s “pile’.

6.2 Disturbance of the process

The message exchange model defined in this section describes the message exchange process as a continuous process without interruption. The building project is carried out as it was defined in the agreement before the project started. In practice, however, external and internal factors will disturb the ideal agreed process. External disturbances can appear at the following system levels:

1. The environment of the project.

   For instance, the weather is a very unpredictable factor that can have great influence on the construction process.

2. The project.

   For instance, the principal changes the start state and/or the end state of the building product.

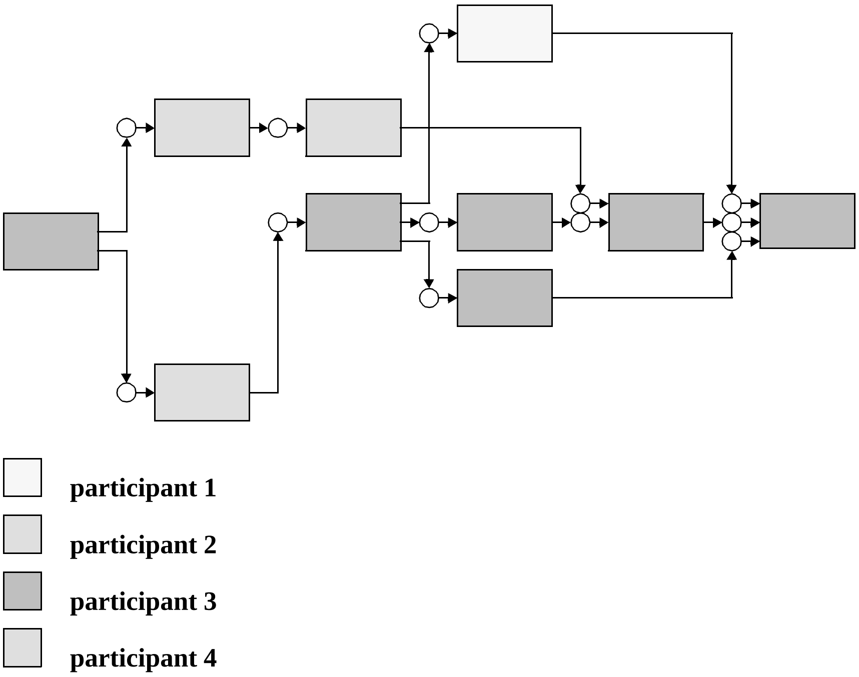
3. A participant.

   For instance, a participant suffers from a lack of resources so that the agreed execution time cannot be reached.

External disturbances result in a new agreement. A new agreement may contain a new start state of the building product. In the start state product information that has already been generated can be included. When the new agreement is accepted by the (new) set of participants, then all former project activities are aborted and a completely new project is started.

Internal disturbances do not effect the agreement, but hinder the normal execution of activities. This happens when messages are received that override previously received data. In such cases the activity network that contains all activities from all participants must be traced to find activities of which the input data are directly or indirectly dependent on the changed data. An activity network like the one shown in Figure 45 can be derived from the project agreement. To find the part of the network that must be re-executed, all channels that connect the activities are inspected after the first activity. The activity network is modelled as a transition system of which the channels contain the required input data to start an activity. The earliest existence in the network of the changed data determine which part(s) must be re-executed. All participants that have dependent activities will be sent a message to change the states of these activities to “abort” [Figure 26 on page 48]. After that, the process is resumed with the earliest activity or activities that have new input data. Participants to whom these activities belong will be sent new “tasks” or “requests” containing the new input data.

FIGURE 45. Example of an activity network for a specific project



6.3 Classification, methods and media

To implement electronic exchange of messages a transfer protocol and a transfer medium must be selected. Part of the transfer protocol is the structuring method, i.e. the method used for ordering data in a meaningful way. Structuring methods can be considered separately from transfer media, although in practice they are often strongly related. For instance, EDIFACT messages can only be exchanged using electronic networks. The message contents can initially be derived from the message exchange process definition described in this section. Secondly, parameters are required by which the process can be measured to help to decide which available structuring method and transfer medium are most suitable at each moment. These parameters are:

- The equivalence between the data structures of the messages that are sent and received and the data structure of the (temporary and permanent) stores of a participant.
- The frequency of messages.
- The amount of data.

Equivalence of messages is measured by the existence or absence in the message data structure compared with the store data structure of:

- Meta object types.

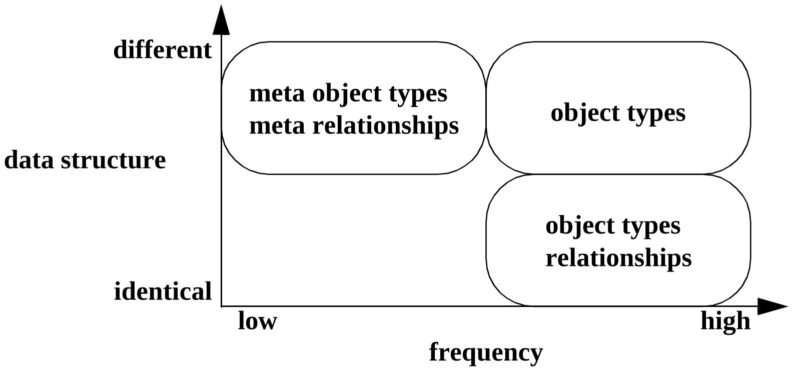
  A meta object type defines the data structure of an object type.
- Object types.

  An object type defines a class of objects with an equivalent set of aspects.
- Meta relationships.

  A meta relationship defines a set of relationships between object types.
- Relationships.

  A relationship defines an association between two object types.

FIGURE 46. Equivalence-frequency diagram



A diagram with three areas can be constructed with the message equivalence along the vertical axis and the message frequency along the horizontal axis. It should become clear from the message exchange process whether object types and relationships are similar to the stores' object types and relationships or not. If not, then the abstraction level must be lifted to the meta level to find similarity. Assumable similarity is related to frequency. If information is exchanged more frequently, the store data structures of participants are probably more similar. There is a stronger motive to adapt to the data structure of another participant if mappings have to be made frequently. Frequently exchanged messages with an incompatible data structure will soon lead to an attempt of the participants to reduce the differences. Usually the first step is the creation of a common set of objects types that will be used by the participants. However, in this thesis the standardization of aspect types not object types is argued. Having a set of normalized aspect types available, each participant can create his own object types. Matching the different objects is achieved by comparing their aspect types and aspect values.

The equivalence criteria are not only used to classify messages but also to classify message data structuring methods. The data models that were explained in the research inventory [Section 3.2 on page 16] can be considered as methods for structuring information. These methods also try the cover the problem of differences in data structures. The level of abstraction that is used varies. In the list of structuring methods EDIFACT is split up into EDIFACT message and EDIFACT syntax, and the Message Exchange Model is added. Moreover, the criteria list is extended with the existence or absence of "constraints". After classifying the messages that are exchanged, Table 6 can be used to choose a structuring method.

TABLE 6. Structuring methods

Information structuring method	Meta object type	Object-type	Meta relation-ship	Relation-ship	Con-straints
BPM	•			•	
BTMS	•			•	
Classification and coding	•		•		
EDIFACT message		•		•	
EDIFACT syntax	•		•		
EDM	•			•	•
STEP AP203		•		•	
STEP AP225		•		•	
MEM	•	(•)	•		•

Classifying information transfer media is done using frequency and the amount of data as parameters. Frequency is determined by the number of messages of a specific type that are sent or retrieved during one year. The amount of data is determined by the number of instances of all objects of a specific message type. The amount of data (I) that is sent and received is normally at its peak ($f'(I)_{max}$) in the middle of the project as shown in Figure 47.

FIGURE 47. Amount of data and the production of data during project time.

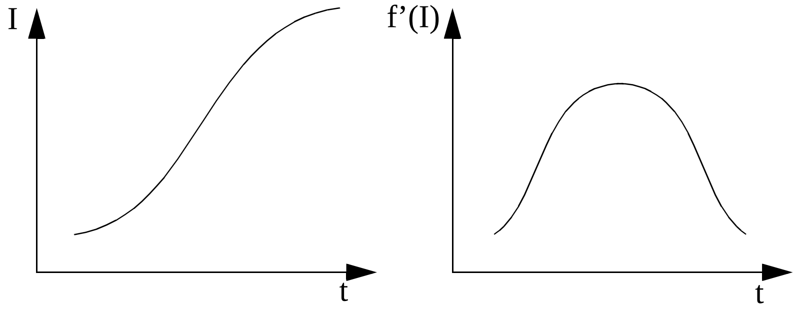


Table 7 shows the normative classification results that hold for the moment the thesis was written. The "-" symbol stands for a low number and the "+" symbol stands for a high number. It is evident that the list of transfer media and the results change when new technical solutions for electronic transfer become available.

TABLE 7. Transfer media in 1995

medium	frequency	amount
manually produced document	-	+
automatically produced document	+	+
telefax	+	-
electronic mail	+	-
diskette, tape, CD	-	+
electronic network (LAN)	+	+
electronic network (WAN)	+	-

6.4 Message development and media selection

To enable electronic information exchange between participants in building projects, a strategy is described using the formerly defined formal description of a building project and the classification methods.

The message development and media selection strategy includes the following steps:

1. Select the participants that are involved in the project.

   When trying to implement electronic information exchange, one needs a group of participants that are willing to consider it. Usually this group already has communicated in former projects and its participants are capable of producing and consuming information electronically. They have a feeling that there might be a profit in introducing or extending electronic information exchange.

2. Select a reference project.

   To create a formal description of the current process a reference project is selected. In principle all types of projects should be analyzed to get the complete overview of all types of messages and their frequency. A more pragmatic approach is to select a project that is close to the average type of project of the involved participants. Afterwards a check is made whether the contents of the messages will vary when the building products varies or when the set of participants vary. If necessary more reference projects must be analyzed.

3. Define start and end state of the building product.

   Within the selected project a building product is created by the participants. The data structure of the start state and the end state of the product is defined as an instance of the meta models for objects [Figure 27 on page 54], relationships [Figure 28 on page 57] and constraints [Figure 29 on page 58].

4. Gather all documents that were produced.

   The tangible output of activities are an important source for the information analysis and for the information flow analysis. It requires much expertise of the design and construction process to retrieve input and output data from these documents. To support the analysis additional interviews can be done with the persons that created the documents.

5. Define the core business activities for each participant.
   a. Select or define the aspect library.
   b. Define object types.
   c. Define relationships.
   d. Define constraints.

   An activity is defined as the smallest unit of work that can be advertised by a participant [page 43]. Advertisable activities are atomic and can be integrated in the project activity network. Aspect libraries described in the message exchange model are not available nowadays. Some aspect types are formally defined in standards and regulations. Many aspect types are terms that are also used in natural language and do not need additional definition, other than the dictionary description. New aspect types are only introduced when a definition can be found that differs from already existing aspect types. Working this way will result in the creation of lists of synonyms. All physical and spatial entities that can be recognized in the documents and drawings are defined as object types. Object names are chosen as close as possible to the terms that are used by the persons that have to deal with the information, avoiding adjectives that may overlap with aspect types. Relationships are added if the list of objects with their aspects is not sufficient to reconstruct the message data structure. Whether this is necessary or not depends on the mapping situation that exists in the project. A “decomposition” relation requires that both related objects share a common aspect. Most objects of type “labor” have a “realization” relation. Constraints are added as sets of rules, that reduce the domain of possible instances. Usually these are rules of thumb that can be defined using some specification language.

6. Create a first-order agreement from the activity definitions.

   A first-order agreement is created by simply copying all activity definitions, only adding the activity type (“task” or “request”) and the execution time. The execution time will depend on the estimated work load that a participant expects at the time the activity will actually be executed.

7. Check the information exchange process for completeness and consistency.

   At this point all information is available to check whether the end state of the building product can be reached. In other words, if the requested input is available to execute all activities. Incompleteness can be solved by changing the core business definitions of the participants. This is not a very appropriate solution, however, because one should expect the core business definitions to be well defined. Properly speaking another set of participants with other activities must be selected. Secondly, a check is made whether some participant does work with outdated information while executing an activity. This problem can be solved by introducing “request” messages and their accompanying activities. Before a new activity is started the participant sends a request message to acquire the actual information, preventing him from using outdated information. The request will start an activity at the recipient, which will retrieve the actual information from his temporary store.

8. Create a project process model from the agreement.

   To make activity dependencies visible, an activity network is constructed from the agreement.

9. Create a project time schedule from the agreement.

   To detect time-consuming activities, a time schedule is reconstructed with time along the x-axis and the activities along the y-axis.

10. Look for opportunities to re-engineer the project process in order to:
    a. Exclude ‘unnecessary’ activities.
    b. Detect ‘critical’ activities.
    c. Define project types.

    Activities that do not contribute to getting to the end state of the building product will not be part of the project. Candidates for this kind of activity are all activities that are concerned with verification. Verification activities only change the status of objects and do not add new information. If activities are not part of the activity network, one should seriously look at the objectives of such activities. Reconsideration might lead to redefinition of activities.

    Activities that produce information that other activities depend upon, and that are very time-consuming have great influence on the time schedule of the whole project (critical path). A closer look at such an activity might show the possibility to split it up into independent smaller ones. If the independent activities can continue without waiting for the critical one, then the overall project time can be shortened. Otherwise, adding more resources to the activity will generally shorten the project execution time anyhow.

Studying the complete project process should make clear whether the reference project is an example of a project type. A *project type* is a set of projects that is independent of the set of participants for a specific start and end state of a building product. For each project type a reference project must be selected and analyzed according to the described strategy.

11. Create a second-order agreement by:
    a. Grouping sequences of activities into one new agreement activity definition.
    b. Removing output from the agreement activity definition of type “task” that will be produced by subcontracting.
    c. Removing input from the agreement activity definition of type “task” that will be retrieved as readily available information.
    d. Adding activities to the agreement of type “request” to provide actual information.

    The second order agreement is created from the first order agreement to adapt to the organizational structure of the building project. In the first order agreement all activities are initiated by the “starter” of the project model. Depending on the type of organization, some participants will take the responsibility for parts of the project. These participants themselves can start activities at other participants. Project dependent or project independent information can be deliberately retrieved to be sure that the most up-to-date information is used.

12. Create a message exchange diagram from the second order agreement.

    From the second order agreement a message exchange diagram can be constructed showing the sequence of the messages and their source and destination. Since every activity is started by a message, participants also send messages to themselves. These internal message are useful for product status control, but will be left out of the message exchange diagram because it is focused on external exchange.

13. Create a message scenario from the message exchange diagram,

    by adding:
    a. “Confirm” messages.

b. “Planning” messages.

by deleting:
c. “Reply” messages.

In the message exchange model all activities will execute as soon as possible. In practice this will not always be the case. Therefore “confirm” messages can be introduced, to confirm the receipt of a message to the sender and the delay time until the activity will be executed. Internal and external “reply” messages sometimes contain information that is useful, like planning information for activities that will be executed later on. In that case a “planning” message is introduced containing information that can be used for resource planning by other participants.

“Replies” contain information about a building product in the “virtual” and/ or in the “physical” state. When a “reply” message merely consists of a product in the “physical” state, the message can be dropped because the received physical object contains all information in itself.

14. Compare the contents of the stores with the already existing stores and adjust if possible.

    ‘Execution’ of the message model will result in an ‘ideal’ data structure of the temporary and permanent store. Many participants in the building industry have storage facilities such as relational databases, etc. These stores must also be analyzed and compared with the ‘ideal’ ones. Careful judgement incorporating interoperability, investment, flexibility, etc. should result in adjustments of the existing data structure or perhaps in a new data structure.

15. Determine equivalence-frequency domain for each message.

    Messages are classified using data structure equivalence and frequency as parameters. The message data structure is compared to the existing store data structure. Frequency is determined by the times that a message with the same object/aspects combinations and relations appears within the reference project. Each message will be put into one of the three domains of the equivalence-frequency diagram [Figure 46 on page 81].

16. Select a structuring method and transfer medium for each message.

    In the first place a structuring method is selected from Table 6 for each message using the parameters from the equivalence table. Selecting a structuring method that uses predefined object types means that the objects types are adapted from the standard. This is of course only possible if all object types that are part of the message also exist in the standard or have a synonym. The same counts for relationships. When adaptation of object types and relationships to a standard is not possible, then a meta structuring method needs to be selected. The message data structure is designed in accordance with the structuring method. Next, a transfer medium is chosen from Table 7 with frequency and amount of data as parameters. (Evidently the time schedule must be adjusted to include the time spent on sending a message.) Message frequency is magnified by the number of projects within one year of the project type. The amount of data will heavily depend on the building product complexity. The reference project must be based on a building product with an average complexity to get an indication of the amount of data.

17. Decompose business activities into logical activities and non-logical activities.

    Business activities were defined as non-logical activities in Section 4.1. Non-logical activities can sometimes be decomposed into other non-logical activities and logical activities. Non-logical activities are the 'creative' activities that require human capacities, whereas the logical ones can be automated. Computer applications transform input data into output data without human interference. Decomposition of non-logical activities results in an activity network with two types of processors: human beings and computers.

18. Evaluate investment and profit of:
    a. Process re-engineering.
    b. New communication media.

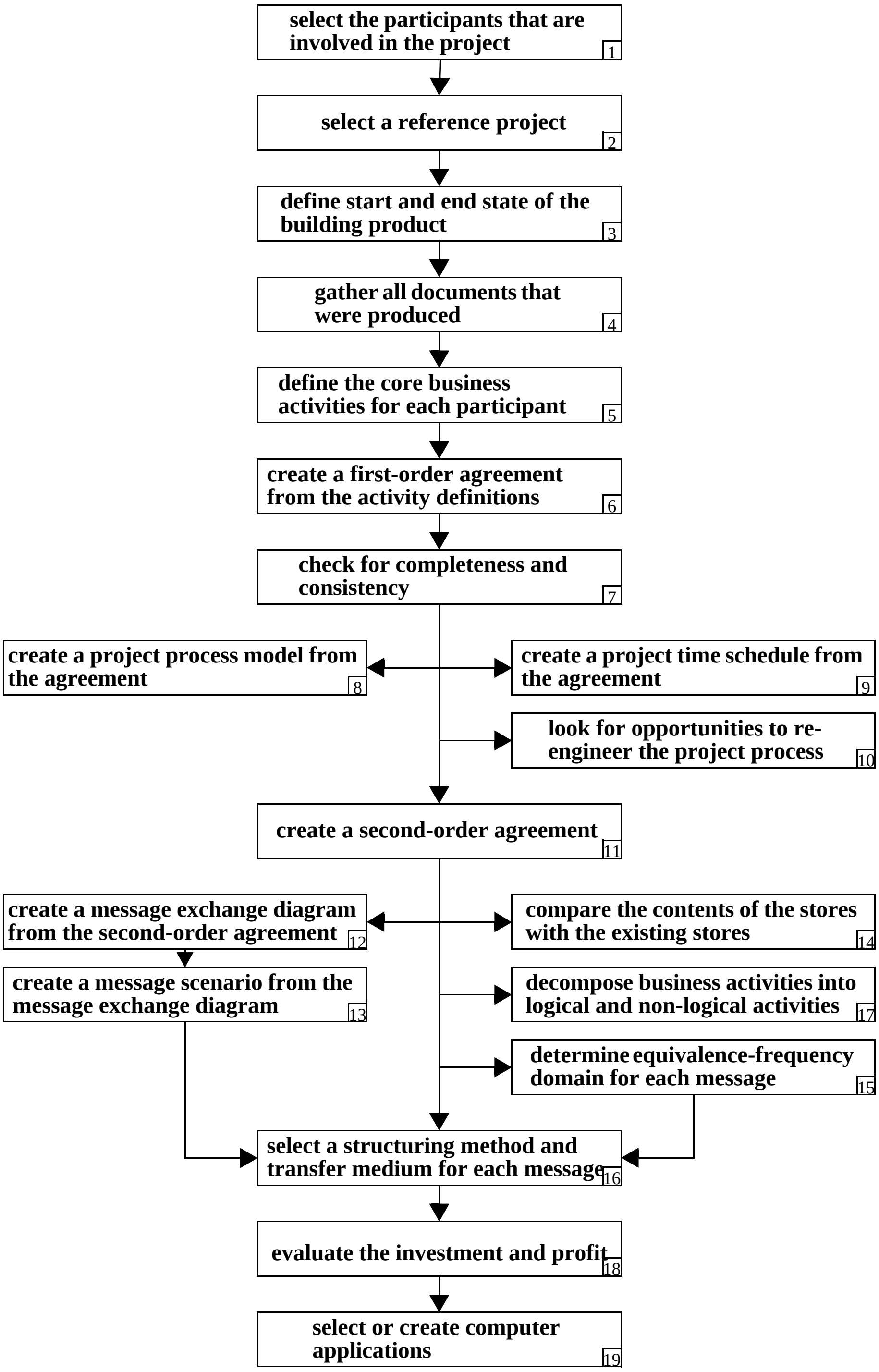
    Of course, before starting to implement new activities, data structures, electronic data interchange facilities, etc. a careful judgement has to be made about the pay off. New communication media do not necessarily mean electronic communication. In fact, the selected transfer media need to be reconsidered from time to time. Criteria for this judgement are beyond the scope of this thesis.

19. Select or create computer applications that support the storage and retrieval of information from the stores and that support message exchange by the selected medium.

    In the end computer hardware and software has to be bought or developed. Some participants are capable of managing it themselves, others will hire external consultants for this job. As with the evaluation phase, the implementation phase is beyond the scope of this thesis.

The message development and media selection strategy is presented in a graphical schema in Figure 48. Application of the strategy is demonstrated in a case study in Section 8.

FIGURE 48. Message development and media selection schema


select the participants that are involved in the project
1
select a reference project
2
define start and end state of the building product
3
gather all documents that were produced
4
define the core business activities for each participant
5
create a first-order agreement from the activity definitions
6
check for completeness and consistency
7
create a project process model from the agreement
8
create a project time schedule from the agreement
9
look for opportunities to re-engineer the project process
10
create a second-order agreement
11
create a message exchange diagram from the second-order agreement
12
create a message scenario from the message exchange diagram
13
compare the contents of the stores with the existing stores
14
decompose business activities into logical and non-logical activities
17
determine equivalence-frequency domain for each message
15
select a structuring method and transfer medium for each message
16
evaluate the investment and profit
18
select or create computer applications
19

7 Simulation model

To be able to effectively use the message exchange model in the actual practice of introducing new communication media, a specific tool was developed. This tool, a simulation model, was implemented using another tool, namely a process modelling program called ExSpect. In the ExSpect environment the information exchange process of a specific building project can be simulated. The information flow can be traced and the contents of the messages can be inspected that are exchanged between the participants of the building project.

7.1 Prerequisites

In Section 6.4 the steps that have to be taken for message development and media selection are explained. The first steps have to do with analyzing the reference project. In the middle the process part and the data structure part of the message exchange model are defined. From the project agreement that has been created, several schemata are drawn that give insight into the project process. The final part considers the selection of a structuring method and of the transfer medium for each message.

After defining the message exchange process described in Section 6, it is possible to deduce the schemata from the definition. More precisely such a deduction should give the following results:

- A list of inconsistencies.
- A project process model.
- A project time schedule.
- A message exchange diagram.
- A list of messages and their contents.

Definition of the activities by their input and output will create a set of data structures. Checking the activity definitions for completeness and consistency, and deduction of schemata from the agreement is hardly possible manually. To cope with this problem the message exchange model is implemented in a computer program using a process modelling tool. The resulting model is called a *simulation model* because if all definitions are filled in correctly, the behavior of the process model should come close to the project behavior in reality. *Behavior* in this context is defined as the flow of information between participants in a specific building project. It is important to keep in mind that aspect values are not part of the simulation model. Only object types and aspect types are used in the data structures. As a consequence, dependency of activity output on aspect values input cannot be taken into account (e.g. design or decision criteria). Instances that enclose aspect values are used in the actual message exchange process. Within the context of the simulation model, meta information is exchanged.

7.2 Limitations of the simulation model

The simulation model is a limited implementation of the message exchange model. The limitations are not imposed by the process modelling environment, but follow from practical reasons to keep the use of the simulation model simple. All limitations are argued below and how to work around them.

Since the scope of the simulation is a building project, there is no history available at the start of the project. This means that the permanent stores of the information systems of all participants will be empty. Almost always a participant has information in the permanent store that will be used in a new project (e.g. reference projects and products). To solve this problem there are three solutions:

1. The permanent stores are loaded before the project starts.

   The problem then is how to acquire all the information that every participant has in store.

2. An actor is introduced that can deliver all information that should be available. This actor can be a participant in the project or an information system in the environment.

   This is a very suitable solution for suppliers of building products. For other information that is assumed to be available in a participant’s store, a non-existent actor is introduced that supplies the information.

3. Permanent information is considered to be available.

   This means that the permanent store is neglected. The activities assume that the required information is available at the time that they are executed. The permanent information is implicitly defined in the activity definition.

The product status store is neglected because its purpose is the status control within a participant, whereas the research is focussed on the communication between participants. Product status control will of course influence the information quality, but not the information flow between participants, as far as meta information is considered.

To simplify the project specifications which are the input for the simulation model, relationships are not considered. Creation of the input and output activity definitions from the documents will start with object type identification. The (building) object types are retrieved from the documents and drawings. Then, if more semantics are necessary to map the set of object types to a standardized message data structure or the data structure of existing stores, relationships are needed. To reduce complexity of the simulation model and to concentrate on creating a common set of object types in the first place, relationships are omitted. Thus the simulation model will merely use sets of objects with their corresponding aspects to define activities, agreements, stores and messages[1]. While finally defining a message data structure in the implementation stage, the structuring method [see Section 6.3 on page 80] determines which relationships or meta-relationships are applicable.

In contrast to the message exchange model, aspect types cannot be decomposed into other aspects types. This facility is needed for complex aspects types. Most aspects types appear to be basic aspect types in practice, so omitting the aspect type decomposition is not a severe drawback and simplifies the simulation model a great deal. Object type definitions merely consist of a set of basic aspect types. It just means that the complex aspect cannot be described by their basic types. When finally defining a message data structure, aspect type decomposition has to be added and checked ‘manually’. The references of an aspect type to a typed set, a value and the unit are omitted since aspect values are not part of the simulation model as stated before. Again, these aspect attributes have to be added and checked afterwards.

Expression of object and relationship constraints is not included. The information exchange process is primarily determined by the availability of objects and their aspects. The availability of objects and their aspects can be checked simply by checking their existence in store. Checking the availability of a constraint on an object or relationship requires a uniform language for constraint expression. Evidently the parse procedures for constraint checking are far more complex than the existence checking of objects and aspects. Development of parse procedures for a certain kind of constraint expression language is excluded from the research and thus constraints are not part of the simulation model.

The resource store is skipped from the simulation model, because the availability of resources can only be described on an instantiation level. Object types do not give any information about the number of instances.

[1] Since the simulation model transfers meta information, objects and aspects used in this context refer to object types and aspect types.

The activity state “abort” is not allowed in the simulation model [Figure 26 on page 48]. The simulation model should support the development of an information flow that prevents re-execution of activities. Instead of restarting an activity, the project continues while a special store *obsolete* is filled with outdated object/aspect combinations.

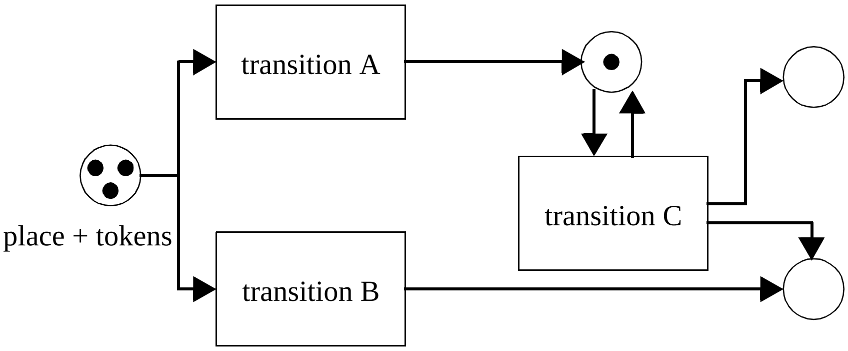
The simulation model does not support the “no-update” operation. During the execution of the simulation, messages will be generated containing exactly the required input with operation “none” and the requested output with operation “update”. Nevertheless, the recipient of a message may update object/aspect combination with operation “none” and may add unrequested new objects and/or aspects. Unwanted updating or addition of object/aspect combinations that are part of the input of an activity which is not in the “completed” state, will result in an entry in the store “obsolete”.

The state attribute (“virtual” or “physical”) of an object is omitted. Instead object names are extended if necessary to express the difference between a virtual or physical object. This adjustment keeps the input and output definitions of the activities that have to be created ‘manually’ as limited as possible.

7.3 ExSpect

ExSpect stands for Executable Specification tool [Hee 1989]. The tool has two main parts: the design editor and the simulation tool. In the design editor, systems and processors are created graphically and interconnected by channels. At the lowest level of decomposition mathematical functions describe how input data are transformed to output data in a processor. The theoretical background is based on timed hierarchical colored Petri nets. (For colored Petri nets, see: [Jensen 1991].) ExSpect has been developed by the Department of Computing Science of Eindhoven University of Technology.

FIGURE 49. Petri Net



A Petri Net has three basic entities: *transitions, tokens* and *places* [Figure 49]. Transitions are connected by directed arcs to places. Every place can contain one or more tokens. The type definition of a place specifies the token type (*color*) it can contain. A transition fires if every input place of a transition contains one or more tokens. The transition then produces zero or more tokens in each of its output places. Decomposition of a transition into one or more other transitions creates a *hierarchy* in the process model. In a *timed* Petri net tokens carry a time stamp. A transition can only consume tokens with a time stamp that is equal to or less than the actual time. In a transition a delay time is specified for each token that is produced and put into a place. The time stamp of the newly produced token is the actual time plus the delay time.

In the representation of the process model in ExSpect five basic entities are used:

1. System

   A system contains one or more processors, channels, stores and subsystems. At the lowest level of decomposition a system only contains processors and may contain stores and channels.

2. Processor (= transition)

   A processor has one or more input pins, output pins and store pins. Input pin, output pin and store pin each have a type definition. The type definition describes the data structure of tokens. Tokens are consumed by the processor from an input channel or store and they are produced in an output channel or store. Mathematical functions define how consumed input and produced output are related.

3. Channel (= place)

   Systems and processors are interconnected by channels. A channel has a type definition and can contain one or more tokens of that type. The type definition of a pin of a system or processor and the type definition of the channel connected to the pin, must be the same.

4. Store

   A store has a type definition and always contains exactly one token. A store is connected to a system or processor by a store pin with the same type definition. A store can be considered a special type of channel.

5. Pin

   A system or a processor has one or more input pins, output pins and store pins. A copy of a system or processor can be used at several places in the process model. Therefore they are ‘installed’ by connecting input and output pins with the channels of the super system and connecting store pins with stores.

In the project process model [Figure 34 on page 65] the agreement has been modelled as a store that is shared by all participants. This is done because in the pre-project phase, participants negotiate about the agreement and store it in the common agreement store. In the simulation model the pre-project phase does not exist. The agreement definition and the activity definitions for each participant are input for the simulation. Therefore they are modelled as parameters of a participant’s system. The top level system consists of all participants, installed with their specific parameter values. The parameter values define the behavior of each participant and of the project as a whole according to the information flow.

Time plays an important role in the message exchange process. In the simulation model the current time is retrieved from a special *time* store. The project process starts at time zero. After the execution of an activity the time store will contain the new time which is the old time plus the execution time as stated in the agreement.

In the message exchange model messages are uniquely identified by a time stamp. The assumption here is that “new” messages and “reply” messages are accepted one at a time and that acceptance of message is a process that takes more than zero time. In the simulation model only activities are time-consuming, all other processors ‘fire’ in zero time. For this reason an extra store *count* is introduced that is shared by all participants, which keeps a counter for message identification (messid). A situation can occur in which two different participants retrieve a number from the “count” store at exactly the same time. This means that the store is accessed twice for a new message number without increasing the message counter in between. This synchronization problem is solved by adding a *flag* store. Whenever a new number is retrieved from the “count” store, the “flag” store gets a token. The existence of this token prevents other participants from retrieving a number from the “count” store before it is increased. After the store is increased the token from the “flag” store is removed.

The parameters that must be given a value at installation of a participant are:

- Activityin

  A set of building object types and their normalized aspect types defining the required input for each activity.

- Activityout

  A set of building object types and their normalized aspect types defining the produced output for each activity.

- Agreemin

  A set of building object types and their normalized aspect types defining the required input for each activity as agreed upon for the project.

- Agreemout

  A set of building object types and their normalized aspect types defining the produced output for each activity as agreed upon for the project.

- Agreements

  The name of the participant, the activity type (task or request), the execution time for each activity as agreed upon for a specific project.

- Partname

  The name of the participant as one of the actors in the message exchange process.

The pins that must be connected upon installation of a participant are:

- Newmess_in and newmess_out

  The receipt and the sending of a message containing a task or request.

- Replymess_in and replymess_out

  The receipt and the sending of a message containing a reply to a task or request.

- Time

  The project time.

- Count

  The counter for adding unique numbers to messages within the project.

- Flag

  The flag that is needed for handling concurrence by getting message numbers from the counter.

The data types of the participant system parameters and of the system pins are specified using the ExSpect language [ExSpect 1993] as follows:

```
activity := str
activities := activity -> $[object:str, aspect:str]
activityout := activities
activityin := activities
agreementout := activities
agreemntin := activities
agreements := activity -> [type:str, participant:str, exectime:real]
partname := str

newmessenvelope := [recipient:str, sender:str, activity:str, messid: int]
newmessdata := $[object:str, aspect:str, modtime:real, operation:str]
newmess := newmessenvelope -> newmessdata
replymessenvelope := [recipient:str, messid: int]
replymessdata := $[object:str, aspect:str, modtime:real]
replymess := replymessenvelope -> replymessdata

count := int
flag := int

-- Comments:
-- type = task or request
-- state = virtual or physical
```

The graphical representations of the data types of the first and the second part of the ExSpect specification can be found in Figure 35 on page 66 and Figure 32 on page 62 respectively. Note that only part of the complete message exchange model is implemented in the simulation model (see Section 7.2). A distinct difference between the data models and the ExSpect specifications is the absence of the one to many relation between object types and aspect types. In the simulation model object/aspect combinations are implemented as one tuple. Inspection of object/aspect combinations happens many times during the execution of the model and is very much simplified if implemented as one tuple.

7.4 Participant system in ExSpect

Taking the limitations and adjustments into account as explained in the previous section, the participant system in the simulation model is modelled using the ExSpect design tool. The ExSpect process schema of Figure 50 can be compared to the transition schemata of a participant that can be found in Figure 37, Figure 39, Figure 40, Figure 41, starting on page 69.

FIGURE 50. ExSpect schema of the participant process model

The type definitions of the channels and stores that are used are:

```
message_store :=
    messid -> $[object:str, aspect:str, modtime:real, operation:str, operstatus:str]
temporary_store :=
    [object:str, aspect:str] -> modtime:real
activity_status :=
    mess_id -> [recipient:str, sender:str, activity:str, actstatus: str]
accnew,accreply,reqcompl,taskcompl,missinp := messid

-- Comments:
-- operstatus = ok or not_ok
-- actstatus = completed or not_completed or missinput or missoutput or executing
```

In ExSpect as in hierarchical Petri nets the systems are decomposed into subsystems. At the lowest level of decomposition a system is called a processor and denoted as a triangle in the schema. A processor encompasses a transition definition. A transition definition consists of a set expressions that define how the data from the input channel(s) and store(s) are transformed to the output channel(s) and store(s) that are connected to the processor.

As an example the decomposition of the “send_taskorresult” system is shown in Figure 52. The “send_taskorresult” system performs the following function.

When an activity has been completed, then the input channel “taskcompl” gets a token containing the message identifier of the message that started the activity. The “checkstatus” processor consumes the token and checks whether all data that were requested in the original message have been updated. If this is the case then the message identifier token is put into the “ok” channel. Next, the “defineresult” processor fires, which creates a reply message containing all results and puts the message in the output channel “outresult”. If not all the requested data were produced then the message identifier token is put into the “missoutput” channel. Following that, the “newtask” system fires which will search the agreement to see if there are any activities that can contribute to the missing data, having the data available in the “database” as their input. Note that the agreement is passed to the system as a parameter value. If an activity is found that meets the conditions, then a new message is created and put into the “outtask” channel. During the message creation process the “count” and “flag” stores are accessed to retrieve a new message identifier.

FIGURE 51. “Send_taskorresult” system schema

As an example of a processor specification, the processor heading and body of the “checkstatus” processor is listed below:

```
name:
   checkstatus
in:
   taskcompl  :  messid
out:
   missoutput  :  messid
   ok  :  messid
store:
   activity_status  :  activity_status
   message_store  :  message_store
```

```
body:
if all[x:message_store.taskcompl | x@operstatus=’ok’]
then message <-
upd(activity_status,taskcompl,upd(activity_status.taskcompl,[actstatus:’completed’])),
    ok <- taskcompl
else message <-
upd(activity_status,taskcompl,upd(activity_status.taskcompl,[actstatus:’missoutput’])),
    missoutput <- taskcompl
fi
```

The processor body shows that the “message_store” is checked to see whether all object/aspect combinations have an operation status “ok”. If this is the case then the message entry in the “activity_status” store will get operation status “completed”. Otherwise the operation status will become “missoutput” as its new value.

The ExSpect implementation of the participant system at all levels of decomposition totals to:

- 20 systems
- 51 processors
- 10 stores

A formal representation of the participant schema can be found in Appendix A.3.

7.5 Using the simulation model

In its uninstalled form the simulation model only consists of the participant definition. For each participant the parameters have to be given a value during installation. The values can be assigned in the ExSpect design environment. The activity definitions and agreement definitions, however, tend to become long lists that need another tool like a database or word processor to be typed in. For that purpose a simple C program was written to convert ASCII text tables to ExSpect language. After converting the activity and agreement file, it is added to the ExSpect file containing the participant installations.

All participants are connected to the “new” channel and the “reply” channel. An extra processor called “starter” is created that will send the first message. The processor definition only contains the definition of this message. The message defines the start state and the end state of the building product by defining the available input data and the required output data. The prescribed end state is extended with the output of all activities in the project. The extension is needed because activities are selected during the simulation process by their contribution to the end state. To make selection of all activities possible intermediate results must be added to the final result. The start message is sent to the participant that is considered to be the initiator of the project.

Now the system is ready to execute the simulation. In the simulation environment of ExSpect, each channel can be inspected to see the contents of each token that has passed the channel. The main purpose of the first order agreement is the creation of a consistent project definition. When the process stops before the reply containing the requested end state is returned to the starter, then an inconsistency exists. By inspecting the channels of the simulation model it is easy to find out what causes the inconsistency. When inconsistency problems are solved then the agreement is adjusted to fit to the organizational project management as described in Section 6.

7.6 Results of simulating a project

Executing a simulation of a building project will fill all channels and stores with a list of tokens that have occupied the channel or store. Each token carries a time stamp of the time that has passed the channel or store. Channels and stores that are of special interest are:

- Channel “new” and channel “reply”

  To create a project time schedule, a message exchange diagram and a list of messages and their contents, these channels contain sufficient data. These data are the message sequence number, the activity name in case of a “new” message, the time the message was sent, the name of the sender in case of a “new” message and the name of the recipient.

- The participants’ stores obsolete

  Within the “receive” system of each participant the store “obsolete” resides where all object/aspect combinations are stored that have been received and that have a time stamp that is more recent than the time stamp of identical object/aspect combinations that are part of the input of “tasks” or “request” that were sent to other participants and that are not yet completed. The message number of the message that contained these data is stored along with the object/aspect combination. If store “obsolete” does not remain empty then at least one participant is working with outdated data in a certain activity. The message number is used to trace back the participant’s name and activity.

- The participants’ stores *actseq*

  The project process model cannot be deduced from the project time schedule because activities can end simultaneously. Then it is not clear what dependencies exist with the following activities. For this purpose a store “actseq” can be accessed from the “send task” system and from the “send request” system. A new “task” or “request” is always fired by a previous one with exception of the first task that is sent by the “starter”. To reconstruct a process model the “actseq” store is filled with a list of prior and next messages numbers. These numbers are used to trace back the activities that were started by the messages.

7.7 Other usage of the simulation model

Though the simulation model primarily has been developed to analyze information exchange in a building project, it offers more possibilities. A few possibilities in the building industry one can think of are:

- Planning complex processes

  Logistically the most complex part of a building project is the construction part. Many activities take place concurrently and lots of different kinds of resources are used. To support construction management planning programs have been developed. The simulation model can be considered as a sophisticated planning program. The result of a project simulation can be translated to a PERT chart, which is rather common in project management or into another presentation form. Because the simulation also includes the duration of activities, the project can be optimized for the total project time. Looking at the time schedule that is the result of a certain agreement, it is rather easy to find out what the ‘critical’ activities are. A major limitation in this respect is the absence of resource stores. Thus, used as a planning program the simulation model does not count for lack of resources.

- Analyzing the consequences of changes

  Updating input data of activities while they are in the “executing” state is not allowed in the context of the simulation model, but happens in practice. Often it is very hard to oversee all the consequences of these ‘forbidden’ changes and in many cases they are the cause of errors. These errors cannot always be detected in an early stage. Most errors do not appear until the construction stage of the building project. The simulation model is a tool to instantly grasp the consequence of a ‘forbidden’ change. With this tool the project management can decide which activities have to be re-executed and which activities can continue.

- Creating documented company procedures

  The basic step in implementing the IS0 9000 standard is the creation of a documented management system (also indicated as a quality system) and retaining records of the operation that makes the complete management process ‘visible’ [Peach 1994]. The formal process description based on the Message Exchange Model and the results from the simulation model are suitable as part of the company procedure documentation. Clauses and subclauses from the ISO 9001 standard that are covered are: design and development planning, design input, design review, design output, design verification, design changes, document and data control, and corrective and preventive action.

8 Case study

A case study was carried out at a company that produces concrete floor slabs. From the company’s point of view the aim of the exercise was to get more insight about the applicability of communication standards. From the research point of view the case was used to test the concepts of this thesis. To achieve both goals, a specific building project was analyzed and an executable model was created in the simulation environment. The results of the simulation are interpreted into several schemata referencing the ‘ideal’ project process. The same schemata were created from the actual building project as executed in real practice. Comparing the two sets of schemata leads to conclusions about the business activities, project organization, the information flow and the contents of the exchanged information. Finally, a global indication is given on how to obtain the message parameters for message classification in order to support the selection of a specific communication method.

8.1 Aim of the case study

A case study is presented to show an example of the application of the message development and media selection strategy [Section 6.4], using the executable simulation model [Section 7.5]. The case study is not a proof of correctness but a proof of applicability of the concepts described in the previous sections. Especially the preparation, the use and the results of the simulation tool are focussed on, because at this point bottom-up system analyses and abstract theoretical concepts meet. The case study is not a fully worked out example of the development and implementation of an electronic data interchange system. Despite that, it demonstrates the main purpose of the strategy, namely to provide the knowledge to determine the potencies of electronic communication. The decision making on investment and implementation is very time dependent and beyond the scope of this thesis.

8.2 New or completed case

The Message Exchange Model (MEM) is applicable to both new and completed cases. For new cases though, the prerequisite is that activity definitions are available for all activities from all participants that are involved in the project. If one or more case studies have been worked out for several reference projects, this can be the case. Thus, a new project can be analyzed with specific product start and end state definitions and a specific set of participants. As long as there are undefined activities in the project, they must be defined first by analyzing completed projects. Since the MEM was applied the first time, a completed case was used.

8.3 Description of the case

A company of prefabricated concrete products named ‘BetonSon’ was willing to contribute by providing all information they had about a specific building project. The prefab concrete company produces a great variety of concrete products such as piles, parts of bridges, floor slabs and facade elements. The company has three locations in the Netherlands. About 750 people are employed in the company and the sales totalled 200 million Dutch guilders in 1994. The organization of the company is divided into four divisions: foundation, building elements, floor slabs and pipes. After a short orientation it appeared that the floor slabs division had the conditions that were favorable for the case study. These conditions are: enthusiasm for new technologies and much communication with other companies.

The prefab concrete company is involved in and shows interest in several IT projects. At the moment that the case study was started the company was working on a long-term project for a highly automated floor production process. As a prerequisite, process schemata were produced to get a better view of the current situation. The process schemata proved to be very useful as a start for the analysis phase of the case study. Severe doubts existed in the company about the return on investments of the introduction of electronic data interchange. On the other hand the company was aware of standards such as EDIFACT and STEP. The interest of the company in the case study was that it could give more insight into the applicability of new communication standards. More precisely the aim of the case study was formulated as:

> *Investigate the applicability of new communication standards and indicate the consequences for the internal information flow in the company.*

8.4 Selection of the building project

As a reference project, the design, production and deliverance of a specific type of floor slabs was selected which generates much communication. A reference project is an instance of a specific project type (see page 86). A project type is strongly related to a product type. From the list of products, one of the three types of prefab floor slabs was selected. This specific type of floor slab is not a product one can pick from a catalogue, but it is tailor made to a high degree. Especially the position of the openings in the slab requires interaction between the floor supplier and the main contractor. The openings also are the cause that every floor is redesigned according to the amount and the position of the reinforcement. As a result, much communication takes place inside and outside the company to produce a prefab floor slab.

An ‘average’ project was selected that was in the production stage at the time of the case study. The project is average with regards to the project size, the product complexity and the role of the other participants. The order consisted of the design, production and deliverance of 30 floor slabs. The number of floor slabs has no impact on the result of the simulation, because in the simulation model object types are transferred and not their instances. The variety of the floor dimensions and the position of the openings may influence the level of detail of the information that is necessary to describe every floor slab. There is little variety in the role of the participants in the case of the floor slab product type. The main roles are played by the contractor as the principal, and the prefab concrete company as the supplier. Subcontractors are involved to do parts of the job such as transport, reinforcement production etc.

8.5 Analyses of the building project

First of all the boundaries of the case study have to be determined. Therefore participants, activities and the product’s start and end state have to be identified. Identification precedes definition. Identification determines which participants are involved, which activities are considered ‘atomic’, which abstract and physical objects are available at the start and which objects have been added at the end.

8.5.1 Identification of the participants

In this case study an emphasis is of course laid on the part of the prefab concrete company in the project, from now on indicated as the floor company. To create a view of the information flow inside the company, four departments of the company are recognized as participants in the simulation model. Outside the floor company only the companies that had a direct contact are considered as participant in the project. This is an arbitrary but logical choice. On the one hand participants that are contacted indirectly (via a ‘direct’ participant) often influence the information exchange process and the contents of the information. On the other hand, formally these influences should be shielded from the direct contacts, because they should take full responsibility and should have full knowledge of the information they pass through. These assumptions lead to the following list of participants:

- The sales department of the floor company.
- The design department of the floor company.

- The production department of the floor company.
- The production preparation department of the floor company.
- The main contracting company.
- The structural design consultant.
- The transport company.
- The reinforcement production company.

8.5.2 Identification of the activities

All information that was produced during the project was analyzed to identify business activities. About 25 documents (letters, drawings, schemata, tables) were collected. The process schemata that were available served as a blue print for the activity identification. Main guidance at this point is the definition of an activity in the Message Exchange Model, namely the smallest unit of work that can be advertised by the participant (in principle). Activities that return repeatedly during the process have to be entered separately (e.g. one-day floor production 1 to 5; see below). This is necessary because in the model there is no information about object quantities. Nor are resource stores present to determine the object quantity that can be processed at one time by an activity.

The following activities were distinguished for each of the participants:

The sales department of the floor company:

1. order acquisition

The design department of the floor company:

2. floor dimension design
3. strength analysis
4. reinforcement design
5. complete floor design

The production department of the floor company:

6. one-day floor production 1
7. one-day floor production 2
8. one-day floor production 3
9. one-day floor production 4
10. one-day floor production 5

The production preparation department of the floor company:

11. production preparation

The main contracting company:

12. order definition
13. floor opening indication
14. floor sequence indication
15. floor slab placement 1
16. floor slab placement 2
17. floor slab placement 3
18. floor slab placement 4

The structural design consultant:

19. structural design verification

The transport company:

20. floor stack transport 1
21. floor stack transport 2
22. floor stack transport 3
23. floor stack transport 4

The reinforcement production company:

24. reinforcement production
25. reinforcement transport

8.5.3 Identification of the product start and end state

The project starts with a request from the main contractor to the floor company to calculate the cost of the design, production and deliverance of the floor slabs. The project ends when all floor slabs are in place in the building site. The set of floor slabs are considered as one product in the simulation model. From the viewpoint of the floor company this is a complete project. Before the project starts, some information is available about the floor slabs. The architect has indicated the dimensions of the floor in his design. The main contractor encloses this information in his cost calculation request. Because the architect is excluded from the simulation, the artificial participant called ‘starter’ will send the main contractor the information as a prerequisite.

8.6 Analyses of the available information

After identification of the participants, the activities and the product start and end state, they need to be defined. Definition basically includes determining objects and their aspects while analyzing the available documents. Object and aspect definitions should comply to the rules of the Message Exchange Model [Section 6].

8.6.1 Definition of the activities

Central to the simulation model are the activity definitions. For each activity input and output are defined in terms of lists of objects and for each object a list of aspects. Activities usually require one or more input documents and produce zero or more output documents. The existing documents in this case are of course an important source, but a major problem is that almost all documents contain more information than is strictly required for their purpose. In the simulation model only the required input and the produced output are added to the activity definition list. To create the correct list, knowledge of the actual work that is done by the employees of the company is indispensable. Object names are chosen that are identical to the common terminology, with respect to the object type definition in Section 5.1.1. Following the aspect definition in Section 5.1.1, aspect names should be retrieved from national or international building standards. Much depends on the knowledge of the person who creates the object/aspect definitions to which extent this goal is reached. For the sake of proving the usefulness of the simulation tool this is not an important issue. But to provide the opportunity of building regulation conformance checking an easy match with these standards is a precondition.

8.6.2 Definition of the participants

A participant’s definition consists of the set of activity definitions that the participant can perform during the project. Participant definitions are core business definitions and thus should be project independent. In the simulation model a participant can bring in the complete set of activities that he is capable of. In the case study only the activities actually executed are analyzed and listed.

8.6.3 Definition of the product start and end state

At a minimum the first activity and maybe some other activities require information that is available at the start of the project. All required input that is not produced as output by any of the activities from the participants belongs to the *product start state definition*. The information contained in the product start state is far less than the information the architect has forwarded to the floor company. The cause of the superfluous information is that the drawings that are sent are ‘general purpose’ drawings, which are used not only by the floor company, but also by the main contractor, the local authorities, etc. In the message that is sent by the “starter” and that contains the product start state, all object/aspect combinations have an “operation” attribute “none”, i.e. all data

8.7.1 First order agreement

The first order agreement is defined simply by copying all activity definitions from the participants. Additionally, the activity type is defined for each activity and the execution time is set to one (unit) for all activities. More formally: an instance is made of the business activities, adding a value for the activity type and the execution time. While executing the simulation model, a check is made against:

- Completeness

  Incompleteness will result in quitting the simulation before the product end state has been reached. Because creation of the activity definition is done ‘manually’, input and output definitions of successive activities will probably not completely ‘fit’. When not all the required information is available to execute any activity, the simulation process will stop. Evidently one or more activity definitions need adjustment to continue the process. If the end state is reached then the information required to produce the product is complete.

- Consistency

  Inconsistency is checked by inspecting the store “obsolete” at each of the participants. In the store “obsolete” all object/aspect combinations, indicated as outdated data, are stored that have a time stamp that is more recent than the time stamp of identical object/aspect combinations that are part of the input of a “task” or “request” sent to another participant and that are not yet completed. Executing activities with outdated data is not allowed to happen. Inconsistency can be resolved by adding a new activity of type “request” with the participant that owns the most recent information. To prevent “obsolete” information the participant then can send a request to retrieve the actual information before starting the activity.

8.7.2 Second order agreement

A second order agreement is created manually from the first order agreement in accordance with the project organization. The organizational structure of the project controls the information flow between the participants. Hence, the estimated or measured execution times of the activities are entered. The first order agreement is adjusted by:

1. Grouping sequences of activities into one new agreement activity definition, to exclude messages that a participant sends to itself.

   A new activity “design and analyze floor” at the floor production company consists of: “floor dimension design” + “floor opening indication” + “strength analyses” + “reinforcement design” + “structural design verification”.

   A new activity “produce floor slabs part 1 and 2” at the floor production company consists of: “complete floor design” + “production preparation” + “one-day floor production 1” + “one-day floor production 2”.

   A new activity “produce floor slabs part 3, 4 and 5” at the floor production company consists of: “one-day floor production 3” + “one-day floor production 4” + “one-day floor production 5”.

   A new activity “produce and deliver reinforcement” at the reinforcement production company consists of: “reinforcement production” + “reinforcement transport”.

   A new activity “deliver floor stacks 1 and 2” at the transport company consists of: “floor stack transport 1” + “floor stack transport 2”.

   A new activity “deliver floor stacks 3 and 4” at the transport company consists of: “floor stack transport 3” + “floor stack transport 4”.

2. Removing output from the agreement activity definition that will be produced by subcontracting.

   From the new activity “design and analyze floor” that is sent to the floor company, the sub-activities “floor opening indication” and “structural design verification” are performed by subcontracting the main contracting company and the structural design consultant respectively.

3. Removing input from the agreement activity definition that will be retrieved as already available information.

   This feature was not used in the case study.

4. Adjusting the execution time of all activities to the number of working days spent.

   At the moment the case study was done, the planned working days of all activities were known and are presented in Table 8.

TABLE 8. Planned execution time per activity

activity	working days
order acquirement	1
floor dimension design	7
strength analysis	5

TABLE 8. Planned execution time per activity

activity	working days
reinforcement design	3
complete floor design	3
one day floor production	1
production preparation	1
order definition	1
floor opening indication	7
floor sequence indication	1
floor slab placement	5
structural design verification	1
floor stack transport	1
reinforcement production	6
reinforcement transport	1

8.8 Results of the simulation

The simulation of the building project will result in a list of messages that are exchanged between the participants and stores filled with project dependent information. Each message contains: the sender name, the recipient name, the activity name, the time the message was sent, the objects/aspect combinations with a time stamp and the message sequence number. The temporary store that resides at each participant is filled with that latest object/aspect combinations that were received from other participants or that were produced by the participant itself. The data that follow from the messages and store contents are presented in the following schemata:

1. Activity network.
2. Time schedule.
3. Message scenario.
4. Message contents.

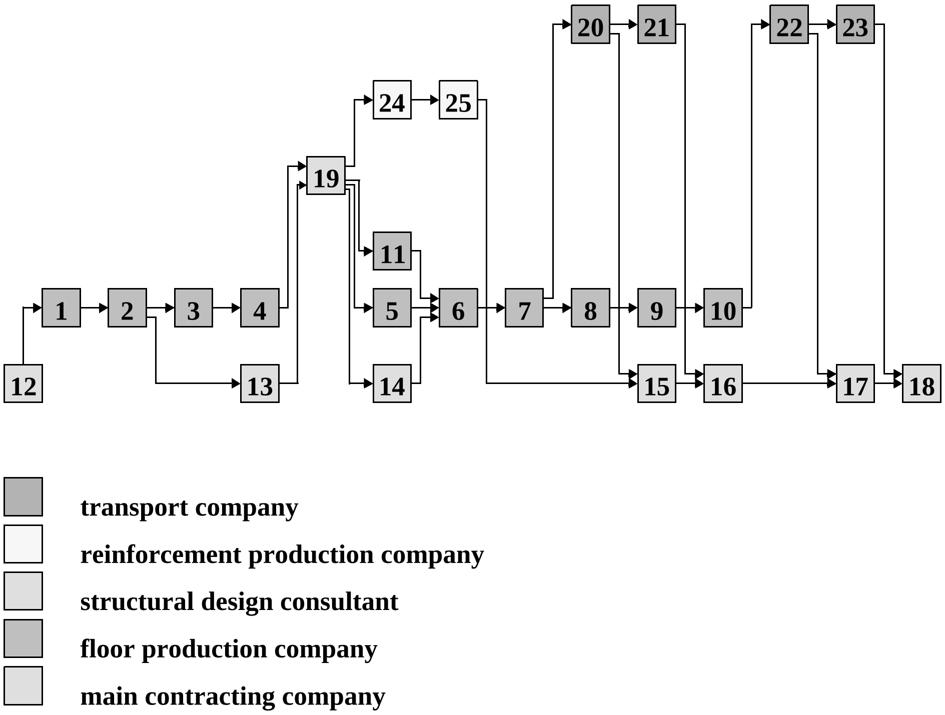
In the next sections it is explained how the schemata are derived from the available data and for what purpose.

8.8.1 Activity network

An activity network shows the information flow between activities. The activity network is manually deduced from the list of messages that is generated from the first order agreement. All activities are set to take equal time to execute and each activity will be started by the participant that initiated the project. The most important property of the network is that an activity can only start executing if all input channels contain the required information. The squares in Figure 52 symbolize the activities. Their sequence numbers can be found in Section 8.5.2. The arrows symbolize the channels through which the information flows from one activity to another. Since the case study is focused on information exchange between participants, all departments of the floor production company are concentrated in one company. The channels that have our special interest are the channels that connect activities between two different participants. From the activity network it becomes clear:

- Which activities are interdependent.
- Which activities can take place concurrently.
- Which bottlenecks exist in the information flow.

FIGURE 52. Activity network: ideal information flow

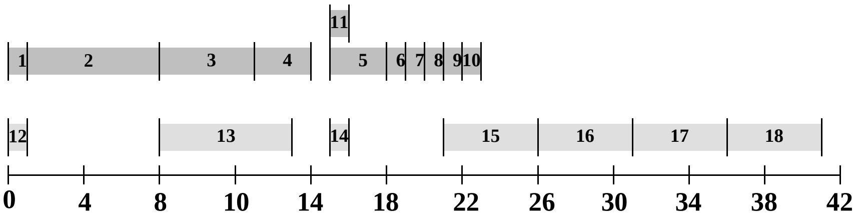


8.8.2 Time schedule

The time schedule shows the number of working days and the starting time for each activity. The time schedule is manually deduced from the list of messages that is generated from the second order agreement. An important property of the simulation model is that it will execute an activity as soon as the required information is available. The activity numbers can be found in Section 8.5.2. From the time schedule it becomes clear:

- At which time an activity can possibly start.
- Which activities form the ‘critical path’ (in combination with the activity network of Figure 52).

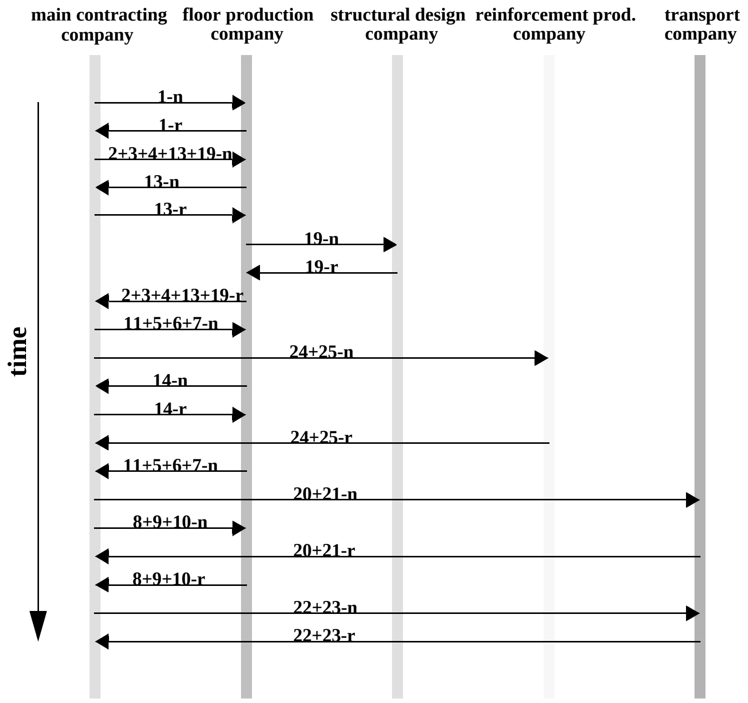
FIGURE 53. Time schedule: ideal situation



8.8.3 Message scenario

A message scenario shows the sequence of messages between participants. The message scenario is manually created from the second order agreement. This procedure is explained in more detail in Section 8.7.2 on page 112. Messages carry the activity number in Figure 54 of the activity that is started by the message, with a ‘-n’ (new) extension, or the activity number of the activity that produces the result, with a ‘-r’ (reply) extension. Activity sequences are grouped into one message if no intermediate information exchange with other participants occur. Grouping of activities is indicated with a ‘+’ sign. Subcontracting is introduced to assure that the floor production company is responsible for getting the floor opening indications and for verification of the floor design by the structural design consultant. In the message scenario the subcontracting activities are added (with the ‘+’ sign) to the actual activities. The message scenario starts at the top. Some messages that are sent concurrently (e.g. 11+5+6+7-n and 24+25-n) are presented one after another, otherwise the schema would have been unreadable.

FIGURE 54. Message scenario: ideal situation



8.8.4 Message contents

As an example the contents of the two messages that were sent and their replies are listed in Table 9 to Table 12. Apart from the object names and their aspects, “new” messages contain an “operation” attribute. To present the message contents a table format is used. The implementation format may differ from the table format.

TABLE 9. Example message-new: order acquisition

object	aspect	operation
reinforcement	surface distance	none
floor	height	none
floor	geometry	none
floor	environment	none
floor	cost	update
roof	geometry	none
roof	material	none
staircase opening	geometry	none
bearing	geometry	none
bearing	material	none
roof surface	material	none
slab	span	none
ordering	juridical condition	none
payment	juridical condition	none
construction	juridical condition	none

TABLE 10. Example message-reply: order acquisition

object	aspect
floor	cost
slab	height
slab	width
slab	delivery date
reinforcement	amount
reinforcement	function
reinforcement	distance
reinforcement	load type
layer	load
layer	concrete quality
ordering	juridical condition
payment	juridical condition

TABLE 11. Example message-new: floor opening indication

object	aspect	operation
opening	size	update
opening	location	update
wall	location	none
wall	geometry	none
wall	material	none
stair	geometry	none
stair	location	none
slab	geometry	none
slab	location	none
outlet	location	update
outlet	type	update

TABLE 12. Example message-reply: floor opening indication

object	aspect
opening	size
opening	location
outlet	location
outlet	type

If the activity network of Figure 52 is combined with the message contents for each channel, then the activities can be deduced that have to be re-executed if already accepted information is changed afterwards, as happens in real practice. Therefore all messages must be scanned for the earliest existence of the changed objects. The section of the activity network that uses the changed objects as part of their input must be re-executed with the new data.

8.8.5 Temporary store contents

As an example the temporary store of the design department of the floor company after the project is completed is listed in Table 13.

TABLE 13. Temporary store design department of the floor company

object	aspect
layer	load
layer	location
layer	concrete quality
floor	load
floor	load area
floor	load location

TABLE 13. Temporary store design department of the floor company

object	aspect	object	aspect
slab	geometry	floor	shear force
slab	location	floor	bending moment
slab	delivery date	floor	measuring point
reinforcement	surface distance	floor	environment
reinforcement	diameter	floor	cost
reinforcement	geometry	wall	geometry
reinforcement	location	wall	location
reinforcement	load type	wall	material
bar	diameter	net	size
bar	geometry	net	location
bar	location	net	geometry
bar	steel quality	net	steel quality
bar	function	outlet	location
bar	cross section area	opening	location

The data store of the design department shows part of the total set of object/ aspect definitions that were defined during the analysis phase. Looking at the definitions the following observations can be made:

1. Some object definitions contain redundant aspect definitions. For example the bar has the aspects geometry and diameter, whereas the diameter could be part of a 3D geometry description. As a consequence, activities that produce this information should take care of consistency within the exchanged information.
2. Some aspects only have semantics if they are available in conjunction. For example the bending moment, shear force, measuring point, load, load area and load location are related to each other. To maintain consistency the same condition holds as indicated above.
3. An object name may enclose redundant information that is also specified as an aspect of that object. To allow the use of commonly used terminology on one side and the use of standardized aspect names on the other side, there can be an overlap (e.g., object: bar, aspect: geometry). The object names however are not used as selection criteria in the Message Exchange Model. The aspect values of object instances must contain enough detailed information to identify a specific building object. Preferably, the same aspect names are also used in building regulations or standards (e.g. concrete quality, load type).

4. A geometry description can contain 2D, 3D graphical or even textual data. The recipient must be able to read the format and interpret it for his own usage.
5. The object’s state is not a separate attribute in the simulation model as it is in the Message Exchange model (see “Limitations of the simulation model” on page 92). If ambiguous situations are possible then the object name is extended with ‘_physical’ or ‘_virtual’.

8.9 Comparing the results to the current process

In contradiction to what one might expect, the simulated process and the actual process are often not identical. The ‘ideal’ process is generated from the second order agreement. The second order agreement is created from the core business definitions and the project organization. The information flow is fully determined by the agreement and is not controlled or hindered some other way. In practice there usually are factors that influence the information flow. The most important factor is the human interference in information transfer. Examples of interfering attitudes are:

- Waiting to send information until one is asked for, although it is available at an earlier time.
- Not being aware of information already available.
- Considering information to be irrelevant.
- Passing on information without contributing new information.

To be able to compare the results of the simulation with real practice, the same schemata were reconstructed from the actual process. The ‘actual’ schemata were created with the help of the floor company’s employees. While comparing, an explanation of the differences with the ‘ideal’ situation is given.

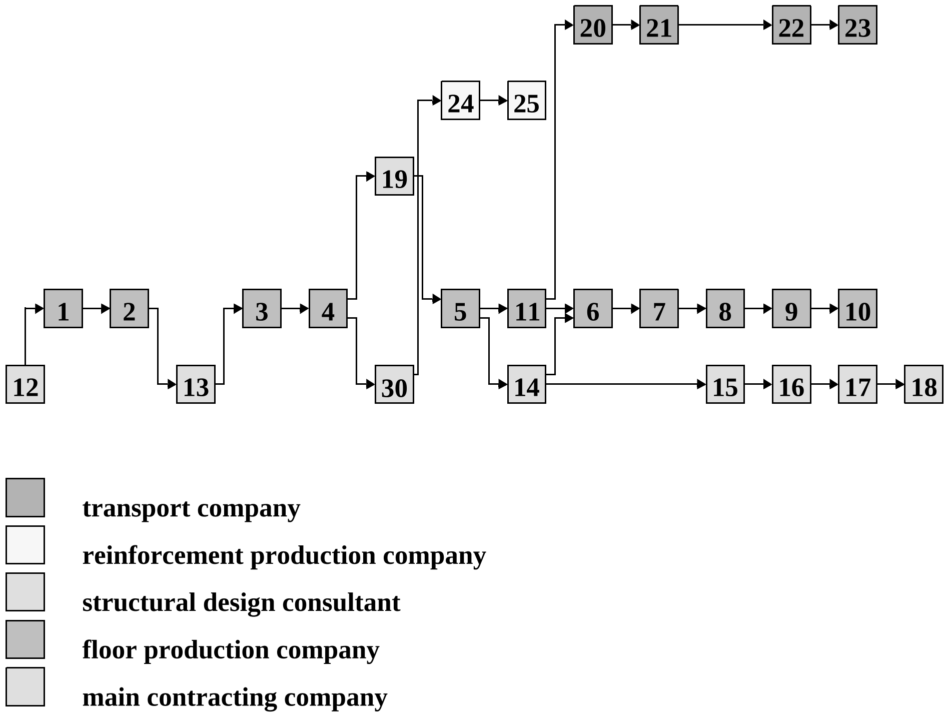
8.9.1 Activity network

Channels in the simulation model transfer information about abstract and physical objects. If information merely consists of physical objects, then instead of transferring the information, the physical objects can be transferred. This situation occurs if a stack of floor slabs is transported from the floor production company to the building site. Instead of sending a message containing new information about the location of the floor slabs, the main contractor receives the floor slabs at the building site. The output channels that connect the transport activities (20, 21, 22, 23) to the floor placement activities (15,16,17,18) are absent in actual practice.

Two activities are executed sequentially instead of concurrently with one or more other activities, namely “floor opening indication” (13) and “production preparation” (11).

An extra activity “floor design verification” (30) is carried out by the main contractor. The main contractor is formally responsible for the correctness of the structural design. In practice he will leave the verification task up to the structural design consultant. While performing his task, the structural design consultant asks the main contractor to send the actual information due to construction on the building site. To save time, the floor company sends the complete floor design to both the main contractor and the structural design consultant. The reply is only received from the structural design consultant.

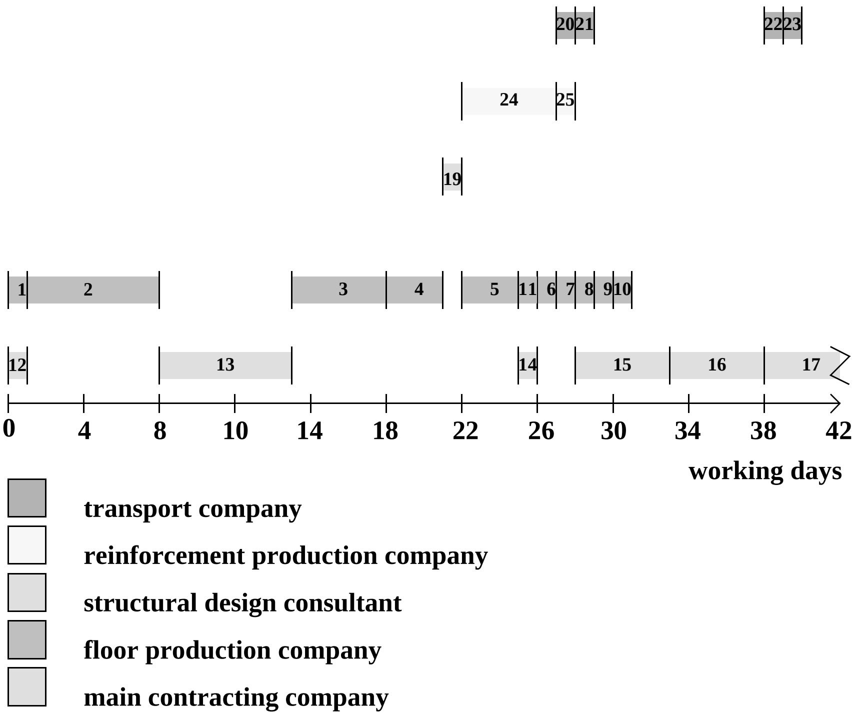
FIGURE 55. Activity network: actual information flow



8.9.2 Time schedule

Concurrent execution of activities that are performed by different participants exists in the current situation. On the other hand, concurrences within a participant were not found. In comparison with the ‘ideal’ situation the role of the “floor opening indication” activity (13) is most striking. Whereas the simulation continues with the strength analyses at the moment that the floor design is sent to the main contraction company, in practice the design department waits for the reply before it will continue.

FIGURE 56. Time schedule: actual situation



The percentage of time that is spent on input consumption and output production is perhaps even more important than the total execution time of an activity. Activities that have a high percentage of time spent on input and output gain much benefit from electronic communication. The percentages are roughly (0, 25, 50,75 percent) in estimate [Table 8]. In general, input consumption is more time-consuming in the beginning of the project because text and drawings are created from scratch. Afterwards information is extended and added. ‘Critical’ activities that could gain much profit from electronic input are “strength analysis” and “structural design verification”. A ‘non-critical’ activity such as “floor opening indication” can save time if information can be transferred electronically instead of redrawing and retyping information.

TABLE 14. Time spent in input/output

activity	working days	input/output part
order acquirement	1	50%
floor dimension design	7	25%
strength analysis	5	50%
reinforcement design	3	25%
complete floor design	3	25%
one day floor production	1	0%
production preparation	1	25%
order definition	1	50%
floor opening indication	7	50%
floor sequence indication	1	0%
floor slab placement	5	0%
structural design verification	1	75%
floor stack transport	1	0%
reinforcement production	6	25%
reinforcement transport	1	0%

8.9.3 Message scenario

The message scenario resembles the ‘ideal’ scenario with a few differences. Evidently at those places that the channels are absent in the ‘actual’ information flow (see Figure 55 on page 122), there will be no message exchange. In the simulation model every activity is immediately followed by another activity. In reality there will be delay times, because for instance there are not enough resources available or because of disturbances [Section 6.2]. For this reason “confirm” messages are inserted to confirm the receipt of a “new” or “reply” message and to convey the delay time. The essential difference between a reply message and a confirm message is that a reply message is sent after finishing a task, whereas a confirm message is sent immediately after reception of the task. Obviously the delay time has to be added to the total execution time of the project.

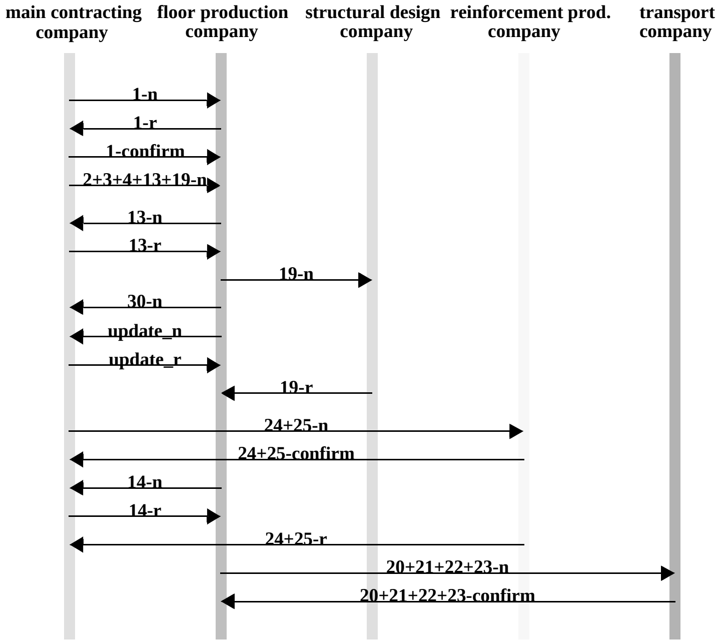


FIGURE 57. Message scenario: actual situation

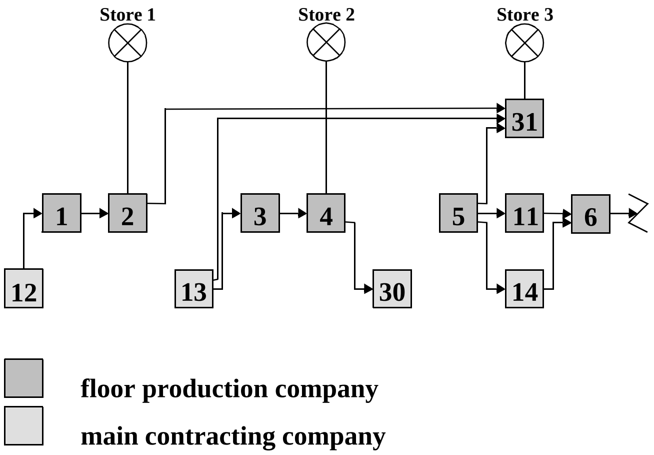
8.9.4 Message contents

In Section 8.8.4 the message contents of two messages have been presented: “order acquisition” and “floor opening indication”. The order acquisition is sent as a letter with a request for cost calculation and the enclosed drawings from the architect. The juridical conditions of the order are added as separate pages. The reply is a letter recapitulating the most important information from the drawings and of course the price for the design, production and deliverance of the floor slabs. Also, additional reinforcement that should be constructed by the main contractor is described. To enable the main contractor to indicate the floor openings (activity nr. 13), a special drawing is created for this purpose by the floor company. The main contractor will pass the drawing to the mechanical engineer and to the electrical engineer in most cases, to draw the size and locations of the openings. When the engineers have finished their task they will send back the result to the main contractor, who will in return send it to the floor company. An employee in the floor company will read the drawing and enter the opening’s data in their floor design system.

8.9.5 Store contents

In the Message Exchange Model (MEM) each participant has one temporary store and one permanent store. To create a view of the internal storage of information in the floor production company, the activity network of the actual information flow [Figure 55 on page 122] is reduced to floor production company and the main contractor and it is extended with the three stores that are being used. Moreover, the process is limited to the first part, until the start of the floor slabs production. An extra activity “status logging” (31) is added to the schema since in the floor company it is a separate activity.[1] Store 1 contains the floor design information including the openings. Store 2 contains the information about the geometry and location of the reinforcement. The dimensions of the floor are passed from Store 1 to Store 2 through the channels. For production planning and management Store 3 is filled on the basis of instructions that are sent or received after an activity has been finished.

FIGURE 58. Activity network: current information storage



All stores are dedicated to the activities they are connected with. The data structures of the stores are unknown. To create a view about the contents of Store 1, the floor design system that is used in activity 2 was studied in more detail. General remarks about the store contents are:

- Data consistency is maintained on entry level. The constraints that are evaluated on data entry are very hard to reproduce from the data storage.

[1] In the MEM only activities that produce new information are part of the activity network. Product status information is accessed by every activity and resides in the “product status” store (see Section 6).

- Much additional information is stored about the representation on paper (plotting instructions).
- Data value sets and data value ranges are ‘hard-coded’, thus it is nearly impossible to adapt to new building standards or new product specifications.
- No common geometrical format is used for the storage of shape and location data.

8.10 Opportunities

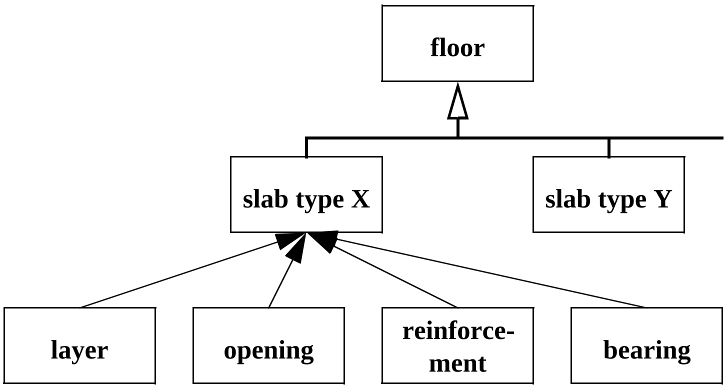
Apart from determining the differences between the ‘ideal’ and the actual situation, interpretation of the differences also indicate opportunities for business process rearrangement and enhancement. The following opportunities were detected:

- Status logging is performed as a separate activity in the current situation. Instead, status information can be checked at the start of each activity and be updated at the end. In such a situation each employee will ‘automatically’ maintain the product status store. Moreover, the execution of activities is not only dependent on the availability of the required input data, but also on the approval of the use of the information by some control mechanism (e.g. a logistics support application).
- In both the ‘ideal’ and in the current situation the structural design verification activity (19) constitutes a bottleneck in the information flow through the activity network. The complete floor design and strength analysis is verified by an independent specialist. Formally, the structural engineer should merely perform the verification, but in practice it turns out that he also updates the input data with the actual information from the main contractor. The most drastic and effective way to remove the bottleneck is the elimination of the verification activity. The main contractor and the floor company could agree upon a quality assurance for the complete floor design. Instead of the structural design company, the floor company takes full responsibility.
- The floor company arranges the transport of the stacks of floor slabs. The ‘ideal’ activity network (Figure 52 on page 115) shows that instead, the management of the transport should reside at the main contracting company. There is a direct information flow between the floor stack transport activities and the floor slab placement activities. In the ‘ideal’ message scenario the transport is arranged by the main contracting company on-call, increasing his flexibility in the construction management. As a consequence this may lead to more storage room demands by the floor production company.

- The main contractor passes on information to and from the floor company, the mechanical engineer and the electrical engineer. Instead, the floor company could consider direct communication with the mechanical engineer and the electrical engineer. In that case the control activity of overlapping openings (now a subprocess of the “openings indication” activity nr. 13) is moved from the main contractor to the floor company.
- Concurrent execution of the floor openings indication and the structural design is possible if the following conditions are taken into account:

  A clear difference must be made between large and small openings. The large openings are part of the floor design (e.g. staircase opening, shaft, etc.). The small openings (e.g. water pipes and gas pipes) can be located anywhere on the floor without influencing the reinforcement, if a few constraints are validated. These constraints could be incorporated in an application program that allows the mechanical and electrical engineer only to make approved openings. Such a facility in combination with the elimination of the “structural design verification” activity would reduce the “complete floor design” activity to the control of opening’s overlap by the mechanical and electrical engineer.
- Many of the computer applications that are currently used, consume data and produce data which are interpretations of the complete set of ‘raw’ data. To enable communication with the company data model, applications should be selected that allow access to the raw input and output data. Because the interpretation process is often very time-consuming (Table 8 on page 113), the total execution time of such activities can be reduced drastically.
- Instead of using dedicated data storage, a database management system can be introduced in the company for central management and maintenance of data. In the Message Exchange Model this storage function is performed by the “temporary” store. A more pragmatic approach is the replacement of current dedicated stores [see Figure 58 on page 126] by Database Management Systems. The main goal that should be aimed at is the development of a company-wide product information model. Starting point for the development of a such a model are the object/aspect combinations of a participant’s store that are generated by the simulation model [Table 13 on page 119]. A product information model (see Section 5.1 on page 49) is the source for creating an implementation model for a DBMS. During the development of the implementation model specific requirements of the DBMS are taken into account. Data structures of all stores in a company should comply to the company product information model. As an example of the result of the modelling technique, the top level of the implementation model of Store 1 is presented in Figure 59.

FIGURE 59. Implementation model for a DBMS



8.11 Classifying messages

Classification of messages enables the selection of a structuring method and of a transfer medium (Table 6 on page 82 and Table 7 on page 83). The parameters that must be determined are:

- Message frequency.

  In the case study the frequency of the messages is the same as the project frequency per year.

- Amount of information.

  The amount of data is determined by counting the occurrences of all instances of all object types in a message. For this purpose the original drawings and documents must be analyzed.

- Structure equivalence.

  Since the messages in the simulation model only consist of lists of objects, the structure equivalence with existing data stores does not include relationships. Determining structure equivalence requires the availability of a data model (or at least a record structure) of the actual stores. In current practice much information is filed on paper and the computer applications that are used do not allow access to the data structure from outside the application. Before implementing electronic information exchange (and thus classifying the messages that will be used), a data model of the actual stores or even better, a company data model should be developed. A first step in this process can be found in Figure 59 of Section 8.10. The entire development of the data model(s) is beyond the scope of this thesis.

The case study is based on one reference project. More reference projects, including the other prefab concrete products that are produced by the company, are required to obtain enough information for classification of the complete range of messages that can be used.

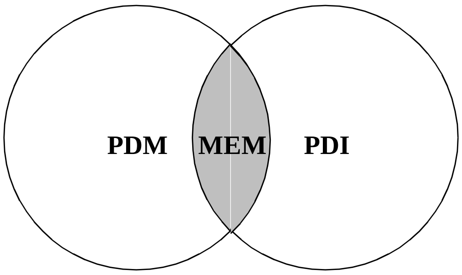
9 Discussion and conclusion

The results presented so far are reviewed in the context of other adjacent research developments. Furthermore, the applicability of the simulation model in information management practice and the validity of the case study is discussed. A proposal is made for the use of the concepts of the Message Exchange Model in future research. Finally, the main conclusions are summarized.

9.1 Message Exchange Model

The Message Exchange Model (MEM) should be applied in combination with other information management strategies. The MEM is focused on information exchange between building companies, considering the activities that are executed within each of the companies. The application area of the MEM is part of two other areas of information management, namely internal communication and external communication. The first area is usually indicated as Product Data Management (PDM), the second area as Product Data Interchange (PDI).

FIGURE 60. PDM, PDI and MEM



In the next two sections the functionality of the MEM is compared to PDM and PDI respectively.

9.1.1 Product Data Management

In industry the awareness of the need for management of all data that are stored in the company is growing rapidly. Within almost every company data storage takes place on a variety of media such as paper archives, computer disks, etc. The objective of data storage is to retrieve part of the data at some time in the future. To support data retrieval, data storage must be managed. Management of documents has a long tradition. Implementations of document management systems are currently available. Industry-specific demands concerning products are more complex, because reuse of data in the design and production process requires storage on a product data level. A fairly new group of applications have come to the market to manage product data storage and retrieval. PDM applications are often built on top of Data Base Management Systems. The applications claim to provide the following services [CIMdata 1994]: controlled data storage, finding by name, etc., part number generation, check-in/check-out, change control & tracking, version control, master database, product structure (bill of materials) data, and product documentation.

PDM applications are primarily developed for manufacturing industries. A major difference between manufacturing industries and the building industry is that in the manufacturing industries new products generally are assemblies of other existing products. In building industry a product is generally completely newly designed. PDM applications are focused on the management of product assemblies. Product changes imply replacement of a product part with another one with different characteristics. Changes in the product assembly leads to a new product version number in PDM.

PDM is closely related to two other types of management, namely Work Flow Management (WFM) and Project Management (PM). WFM supports the control of data flow in the business process [Joosten 1995]. PM supports the control of the execution of activities in a business project [Spinner 1982]. Both WFM and PM require accurate data to produce management information for decision support.

Features that are part of PDM, WFM, PM, and can be found in the MEM as well, being:

- Version control and change tracking.

  In the MEM, version control and change control are established by attaching a time stamp to every aspect value of each instance of an object type. If the received or produced information is more recent than the information that is in store, this automatically results in a new product definition. Product changes are registered by storing all messages that were sent and received. Information flow inside the company can be rerouted to outside by ‘forcing’ the company to send a message to itself. In that way internal information will be filed like external information.

- Change control.

  In the “product status” store the status is kept as an attribute of each object/aspect combination. An activity can only start if the status of all object/aspect combinations of the input meet a certain precondition that is defined as part of the activity definition. Monitoring the status of a product requires a monitoring application that is connected to the “product status” store and the “temporary” store.

- Check-in/check-out.

  Inconsistency because of concurrent use of data can be signalled from the moment there is a project description using the MEM. The problem is solved by adjusting the agreement and/or by creating a message scenario that includes “request” messages to provide the actual data. Thus consistency maintenance is implicitly part of the message development strategy.

- Master database and product structure.

  The “temporary store” contains all information that has been received or has been produced during the building project. The database structure is developed from a set of objects that a company has in common with its fellow companies. The database structure is extended with company specific information and with extra information that is necessary to enable mapping to non-identical message data structures.

- Action messages.

  Messages are the only instrument that is available in the MEM to trigger new activities. Messages contain the required input information and information about the tasks to be fulfilled.

- Business process engineering.

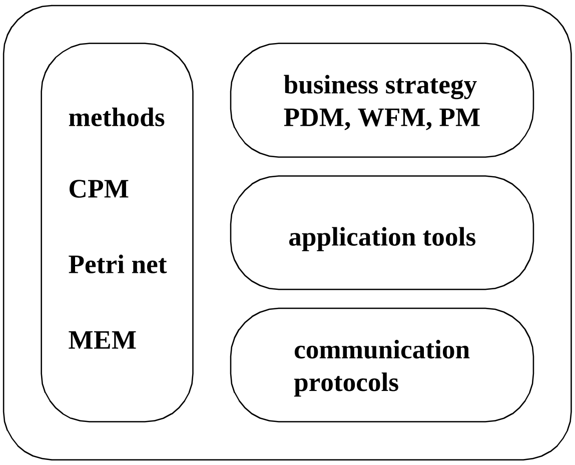
  Activities are defined as ‘atomic’ information processors. This characteristic allows re-arrangement of activities to create new business processes. Because the activities are formally defined, lack of information can be deduced from the process description.

- Project time management.

  As a result of the activity definition, an ‘ideal’ time schedule of a project can be reconstructed. The time schedule can be optimized by enlarging the resources of the ‘critical’ activities.

The objectives and applications of the MEM and of PDM, WFM and PM are different. The MEM is a model to formalize information exchange inside and outside the company. PDM, WFM and PM are management strategies that are implemented in computer programs.

FIGURE 61. MEM, PDM, WFM, PM



Project organizations can use the MEM as a theoretical basis for describing their information exchange needs. From that, PDM, WFM and PM applications can be selected that support company management, but also the use of new communication media.

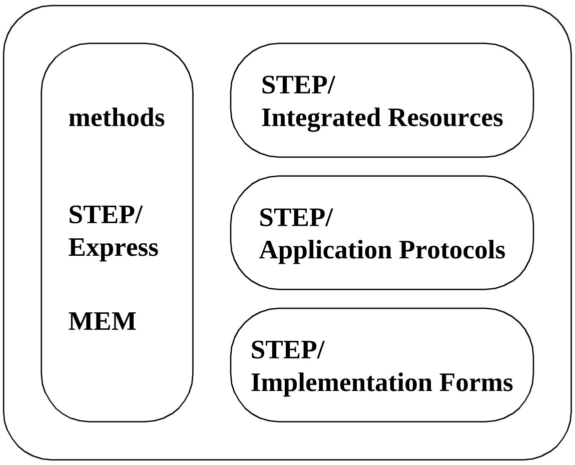
9.1.2 Product Data Interchange

Electronic data interchange as a technique has initiated special purpose developments such as Product Data Interchange (PDI). PDI has almost become synonymous with STEP (STandard for the Exchange of Product model data), at least in the international research institutes. Two STEP Application Protocols are briefly explained in Section 3.2.6 on page 22 of this thesis. An Application Protocol (AP) specifies the portions of STEP required for a specific communication, how they should be used, and in which context. To maintain consistency between application protocols, common data models (e.g. Geometry) are defined, called Integrated Resources (IR), that are referenced by the AP’s. The objectives of STEP are to enable the following services [Gielingh 1994]:

- Inter-enterprise communication.
- Inter-discipline communication.
- Inter-life cycle communication.
- Inter-application communication.
- Communication between computer applications from different vendors.

The MEM does not provide a data model like STEP does, but it provides a method to determine the contents of the actually required information that is to be exchanged to make a specific building project proceed.

FIGURE 62. STEP, MEM (adapted from [Gielingh 1994])


methods
STEP/ Express
MEM
STEP/ Integrated Resources
STEP/ Application Protocols
STEP/ Implementation Forms

The contents of the information exchange is presented as a message containing a set of objects and their aspects. The set of objects can be compared to the set of objects that are part of an Application Protocol. After an AP has been found that encloses all information that is required, the mapping situation (see Section 4.4 on page 35) must be determined.

A fundamental question about the STEP approach is, whether it is possible to describe a product life cycle at a generic level. During the design, development and construction stage a product definition will change. A comprehensive product model should enclose all information in all stages. This implies that such a model also relates the data structures of the different stages. In the MEM the process description between the start and end state is described as a sequence of states. The activities that move the product from one state to the next are merely defined by input and output. Decomposition of activities will result in logical and non-logical activities (see Section 4.1 on page 27). Non-logical activities are the cause of the fact that it is not possible to relate the data structures of all the successive states. Up to now STEP AP’s were primarily focused on the end state of the final design of a product. Thus, it is not clear at this point how the problem of the non-logical activities will be coped with.

STEP has reached the status of an ISO standard, but implementation in building applications is still very rare. From the viewpoint of the MEM, STEP Application Protocols are an optional data structure for the implementation of a message. A message usually consists of a small subset of an AP. In the most ideal situation all software vendors and building companies will adapt their company specific data structure to STEP AP's. In that case all information can be exchanged without object mapping. As with all emerging standards there are considerations and circumstances that hinder the achievement of the ideal situation, namely:

- Products are not static.

  Standard product definitions run the risk of becoming outdated. Formal standardization procedures take time before a new product specification can come into effect.

- Companies have a company specific data structure.

  Before STEP IR's and AP's have emerged, companies already have developed a data structure that satisfies their specific needs. It is not likely that this data structure will match the STEP data structures. Data structure mapping seems unavoidable.

- Computer programs have an application specific data structure.

  Data structures of computer programs are often fully integrated in the software and not recognized as such. Computer program data structures are optimized for the purpose of the computer application. Today, most applications have a facility to communicate with external database systems. The data structure of the database system reflects the internal data structure of the computer program. It is not likely that this data structure will match the STEP data structures. Data structure mapping seems unavoidable.

- Uncertainty about the adoption of the standard.

  Standards do not come into effect until they become part of government regulations or become part of the guidelines of leading industries. Good standards from a technical point of view may fail if there is no major driving force for adoption of the standard. Even then, competitive standards often co-exist, like for instance the UNIX variants of the UNIX operating system.

9.2 Simulation model

In Section 7.2 the following limitations of the simulation model in comparison to the Message Exchange Model are put forward, namely the absence of:

1. A permanent store.
2. A product status store.
3. Relationships.
4. Aspect decomposition.

5. Constraints.
6. Resource stores.

Filling in the limitations is possible, but it doubtful whether it leads to really new applications. Introduction of the stores will make the simulation model applicable for other purposes such as: project planning, work-flow planning and resource planning. However, for these kind of applications, management systems are on the market. The major advantage of the extended simulation model is that it covers an integrated approach of all management areas.

The simulation model can be extended with aspect decomposition and relationships. Instead of the simple lists of object/aspect combinations, complete data structures can then be transferred between the participants. The purpose of the simulation model will shift in that case from a development tool to an implementation tool. It could prove to be very useful to test whether a choice for a specific structuring method will satisfy all needs. The simulation environment allows experimenting with data structures without interfering with the actual information flow and information storage.

In the MEM object constraints and relationship constraints are evaluated with aspect values as parameters. Object instances and thus aspect values are not part of the simulation model. Constraint checking requires a uniform constraint expression language. In today’s information exchange, there is little transfer of explicitly expressed constraints. Usually it is regarded as part of the task to be executed to perform constraint checking. Evidently additional constraints enhance the semantics of a message a great deal. Value ranges in particular (e.g. a color palette) are rather easy to implement. Adding complex constraints (e.g. formulae) will shift a message to an expert application. It is questionable whether companies will ‘envelope’ their expertise in a message. A more pragmatic approach is the implementation of ‘rules of thumb’ and code lists to define aspect dependencies and define object libraries respectively.

The simulation model is restricted to the execution of one project. Obviously, building companies are usually involved in more than one project at a certain moment. The role of a building company as a participant in a specific project may vary. To extend the functionality of the simulation model to more than one project, multiple agreement definitions should be allowed and all messages that are sent should have a project agreement identification. Concurrent project execution in combination with the existence of resource stores can be very useful for resource requirement planning.

Project time schedules tend to be adjusted during the project execution because of external disturbances of the process (see Section 6.2). This effect can be simulated by introducing a statistical function for activity execution time extension (or maybe shortening). The statistical function is related to project experiences in the past.

ExSpect has proved to be a very powerful modelling tool for processes. Execution of the process model implicitly proves the correctness of the designed model. The behavior of the simulation model is an abstraction from actual practice and allows for experimenting. ExSpect is a process modelling and simulation environment. If a designed model has proved to be of use in, for instance, the development of electronic communication, then the requested functionality of this model can be implemented as an application program. The application program can be designed with a user interface specifically for the simulation of information exchange processes. Input of the participant and agreement definitions and output of the schemata can be refined for ease of use.

9.3 Case study

The case study is discussed using the four organizational influence factors: business goals, business process, IT strategy and IT relations as a guidance (see Section 1.1). For each of the influence factors a statement is made, fed by the experiences that were gained during the case study with the Message Exchange Model, the simulation model and the current use of information systems in the company.

9.3.1 Business goals

- A company should not only produce complete products (or services) but also should advertize ‘atomic’ activities containing special knowledge.

  Specialization of design and construction activities have created (new) companies that can do part of the job more efficiently than those companies that offer the integrated processes. Activities that merely have information as input and output can easily be relocated if information transfer time is reduced to virtually zero. Relocation can be effected inside and outside the company. A company that has built up valuable knowledge on a specific subject for its own usage may very well exploit it outside the company. Usually, these specific activities will be closely related with the primary process. Conversely, activities that have been added as support activities may better be executed by a specialized company.

9.3.2 Business process

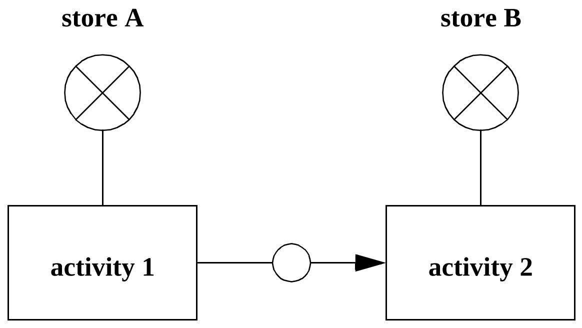
- Project activity networks describe an instance of the business process (i.e. a building project) explicitly, which is required for process evaluation.

  Many (mainly graphical) representation techniques have been developed for describing business processes (e.g., ISAC [Lundeberg 1981], IDEF0 [IOP 1989], DFD [Aerts 1991]). Activity networks used in the Message Exchange Model which are based on Petri nets, are described using a formal specification language. Formal specifications can be evaluated to check for completeness, consistency, performance etc. Business processes modelled using the MEM help information managers to create a clear and concise view of the information exchange process. Discussion with company managers about the activity network and about the schemata that result from the formal description analyses (i.e. the project simulation) help the company to re-arrange or perhaps re-engineer the business process if necessary. One case study of a specific project will not directly result in a drastic re-arrangement of the business process. However, it appeared that presumable enhancements were confirmed by the outcome of the simulation experiment. A major improvement as aimed for by Business Process Re-engineering [Hammer] requires a more comprehensive analysis and of course the full support of the management of the company.

- Information exchange in practice also includes information management of distributed information.

  Though the message exchange process in the case behaved very much the same in the model as in real practice, it appeared that the simulation model is not able to represent the existing data storage circumstances accurately. The major difference between the ‘ideal’ situation and the ‘real’ situation is that there is more than one information store. For pragmatic reasons such as performance, location distances, maintenance complexity, etc., the central stores are split into several application stores. However, the contents of the store of the simulation should still be considered as a reference. In the Message Exchange Model information transfer between activities always induces information storage. Volatile information is kept within an activity. In practice the difference between volatile (not stored) and temporary information (stored during project lifetime) is often not clear. Much information is transferred between activities that has not been stored (e.g. a preliminary design which is input for the next design phase). In the model only message identifications are transferred through the channels inside an organization. If activities can be connected to different stores, the impact on the model is that the channels which connect the activities will contain data with complex data structures.

FIGURE 63. Multiple application stores.



Instead of copying data, references to these data can be included in the message. The distributed DBMS should take care of linking all referenced data at the time of data retrieval and of maintaining data consistency.

Summarizing, in practice a channel can contain a combination of the following information (see Figure 63):

a. Message identification and project identification.
b. Volatile information.
c. A subset of store A that is copied to store B.
d. Reference to a subset of store A that is copied to store B.

- Application of the Message Exchange Model helps to detect and control ‘critical’ activities.

  In business processes time often is a critical factor. Activities are scheduled to reduce the overall production time. Activities that act as bottle necks in the information flow are often critical activities too. The process is difficult to control if a critical activity has to be executed by another company. Integrating remote activities in the process planning and process control can be improved using IT. Project agreements and electronic information exchange avoid the lack of information and reduce the time spent on input/ output and transfer.

- IT enables process control without creating business activity overhead.

  Product status acts as a precondition (apart from the required input) for activity execution. Obtaining a specific status is often related to the availability of certain information about the product. Product status depends on product information but can be managed independently. Determining the product status does not require any special activity, since the actual product data reside in the project store. Product status management for logistics control for example is a project independent activity, not part of the business process.

9.3.3 IT Strategy

- Informal information exchange should be made explicit if new information is added.

  The Message Exchange Model does not include informal information exchange. During the execution of an activity information exchange can take place (e.g. a telephone call) that does not add new information, but serves as a clarification or negotiation (e.g. delivery data). A distinction should be made between the information exchange that needs to take place (formal) and information exchange that may take place (informal). If really new information is added, then a separate activity must be defined for this purpose. Evidently, control of informal information exchange is impossible and thus should be avoided.

- Replace information verification by product quality assurance.

  The Message Exchange Model pointedly leaves out activities that do not contribute to the product end state. This characteristic is the cause of the fact that verification activities (which do not produce new information) are skipped by the simulation program. To ‘force’ the execution of a verification activity, an aspect must be added to the output of the verification activity, for instance “approval”. Nevertheless it seems a good habit to minimize verification activities and replace control of the *correctness* information by a quality assurance at the root of the information production. Control of the *availability* of information at the right time is inherent to the MEM.

- ‘Atomic’ core business activities can be executed independently of other ‘atomic’ activities in the company.

  During the activity definition phase of the case study, it proved to be a very fruitful exercise to disentangle work tasks and to locate ‘unnecessary’ activities. Because several activities are often performed by one person, it seems that they depend on each other, which need not be true. Careful analysis of the required input and produced output will relate only really dependent activities. The less dependencies exist, the more flexibly an activity network can be (re)arranged.

- The Message Exchange Model supports consistent data model development.

  Application of the product information part of the MEM (see Section 5.1) should be part of the company data model development. The product information model serves as a development framework, that keeps modular development of parts of the company product data model consistent. Implementation models will not contain all features (e.g. object and relationship constraints) of the product information model normally. A data model development framework is necessary because the Object Oriented paradigm is a very powerful concept, but allows for many good and yet very different solutions to the same problem. The product information model is based on basic design and communication principles and it structures the Universe of Discourse of the building industry in a consistent way. By structuring building information consistently, software maintenance is more easily managed.

- Normalization of aspects is a major challenge inside and outside the company.

  An important aspect of the product information part of the Message Exchange Model (see Section 5.1) is the existence of normalized aspects. Normalized aspects should reference building regulations. Evidently, in the current situation it will not always be possible to find an adequate reference. Many aspects (e.g. length) are so obvious that formal definition seems overdone. Nevertheless, a concise and univocal aspect definition that is valid independent of its context, can prevent much misunderstanding.

9.3.4 IT relations

- Backward information flow requires special activities to induce the upstream information transfer.

  Interaction between design and construction goes forwards and backwards. Improving the integration of design and construction requires both flows of information [Luiten 1994]. In the MEM the information flow is determined by the information requirement of the activities. This implies that the backward information flow must be modelled explicitly by introducing ‘consultation’ activities (e.g. check realizability).

- Relationships between companies based on project agreements will improve process integration.

  Project agreements determine the contribution of each participant to the product to be created, by defining the required input and the produced output for all activities. Project agreements are the formalizations of the project activity networks. Activity networks can be optimized towards total production time. Parameters in the optimization are activity execution time, information transfer time and the possibility of concurrent execution. Participants that integrate their activities and offer fast information transfer will of course gain better process performance. Agreements between participants which exceed the project lifetime can improve activity communication and integration.

- Tasks described using an operation attribute for each object/aspect combination will decrease misunderstanding and disappointment.

  Tasks nowadays are often described on the product level (e.g., structural analyses). Creating task descriptions by adding an operation “update” or “none” to each object/aspect combination appeared to be a very simple concept that forces the creator of a message to be very explicit about his expectations. On the other hand, the freedom the recipient of the message has to search for appropriate solutions can be clearly bounded by a “no-update” operation. Evidently, the activity result can never be fully determined, otherwise the sender of the message would take care of the job himself.

9.4 Future research

The Message Exchange Model is a new approach to modelling information and the information flow for a very restricted purpose, namely the information exchange process between participants in a building project. The MEM could be extended to meet the specifications of a message exchange handling system. A *message exchange handling system* (MEHS) is a computer application capable of identifying and processing incoming and outgoing electronic messages of a participant. Identification is possible by checking the agreement that two (or more) participants have settled. After identifying a message, the message data structure and the mapping rules are determined. Importing the received information and storage in the company database can then be performed automatically. Exporting information takes the same steps in reverse order. A MEHS takes away the trouble of searching for the most efficient information exchange method for each information exchange event. It ensures that all information that should be stored is actually stored in the ‘right’ data structure. It allows flexibility in choosing a message data structuring method and message transfer medium. It prevents the start of the execution of an activity before all required information is available.

The fundamental difference between the MEM and the MEHS is that the MEM is a model of the information exchange process, whereas the MEHS is a system to support information management. A model of the MEHS could very well be implemented in ExSpect. Everyone in a company that communicates with other companies should have access to the MEHS. The company management is of course responsible for making agreements about communication with other companies. In contrast with the MEM, the MEHS agreement exceeds the project lifetime. Companies that frequently communicate about complex products over a long period of time (e.g. five years), are good candidates. ‘Agreed communication’ and ‘agreementless communication’ can coexist within a company. Agreed communication will improve information transfer efficiency and should result in cost reduction.

9.5 Conclusions

Companies in the building industry hesitate to adapt to emerging standards for product models and/or communication protocols. The main emphasis up to now is laid on developing comprehensive product models as enabling technology for electronic communication. Though there is awareness about new information technologies, it is not clear to companies which technology fits their business process. Moreover, many technologies are not yet stable [Janssen 1993].

The best way to stimulate the introduction of electronic communication is to create a clear view of the current information exchange process inside and outside the company. From that, the potencies of new technologies can be explored. The consequences of the adoption of communication standards can be justified at any moment in time. A company’s management can decide on whether to and when to implement new communication media.

Creating a view of an information exchange process requires a formal description of the information flow. The Message Exchange Model is developed to serve this purpose, specifically for a building project. Part of the Message Exchange Model is implemented in a simulation model. The output of the simulation in comparison with the current situation shows the implications and possibilities of new communication media.

Implementation of electronic communication will probably lead to long-term agreements among companies that exceed the project lifetime. Even the use of a specific standard needs to be agreed upon before a project starts. Mapping of a message data structure to a company specific data structure is unavoidable in most cases. Companies that make investments on mapping software will require enough lifetime to make it profitable. Implementation of electronic communication will involve small groups of companies that communicate frequently. Development of electronic communication standards will involve representatives from a certain discipline. It appears that implementation and development stem from a different background.

Many problems that occur during the information exchange process could be avoided if a formal description of the business activities of every company was available. This conclusion is probably one of the main reasons for the existence of the ISO 9000 standard, which aims at making the management process visible. The unexpected absence of output and/or input could be determined beforehand. Formal activity descriptions should be added to the glossy brochures that are used nowadays to present product specifications. Products in this context are physical and abstract products. The product specifications are the output of the activity description. If product descriptions are replaced by formal activity descriptions, checking of the information flow can be automated.

Normalized aspect libraries are essential in performing object mapping of message data structures onto company data structures. Presuming that object mapping is unavoidable, aspects are required to identify ‘identical’ objects. Whether two objects are considered to be identical depends on the context of the application. Univocal aspect definitions enable object matching based on identical object characteristics. Aspect definitions should reference libraries that fulfil the same function as language dictionaries. Aspect libraries with typical building terminology should be contained in building regulations.

Companies demand flexibility in using application specific data structures on the one hand and adaptation to new standards for product or message data structures on the other. To serve this flexibility an enterprise integrated data model should be described. Message data structures and application data structures can be mapped to this model. The enterprise data model need not be implemented, but serves as a stable reference model. It should be defined carefully and completely, independent of existing data structures and organization structures, using the Message Exchange Model for instance.

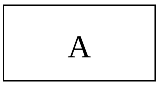
Appendix

A.1 Object modelling methods

The graphical representation techniques used to present functional object models is explained briefly. A comparison is made between functional object modelling and other commonly used modelling methods.

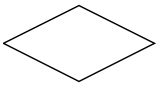
A.1.1 Functional Object Modelling

An object type is a class of entities in the Universe of Discourse and is uniquely determined by its relationships to other objects.



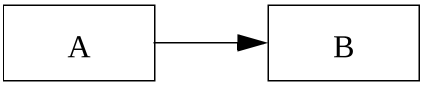
object type A

An association is an object type that is not part of the Universe of Discourse. It is typically used for modelling many to many relations.



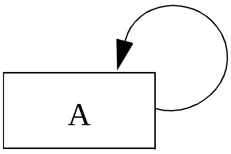
association

In functional data modelling relations between objects are depicted as mathematical functions. If two object types A and B are related by a function *f* then for every element of the domain *f* there is exactly one element in the range of *f*. Domain and range of *f* are instances of object type A and object type B respectively. The graphical representation of a functional relationship between two object types is:



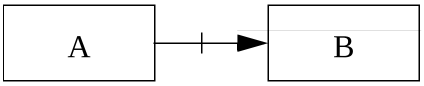
functional relationship

An object can reference itself by a functional relationship.



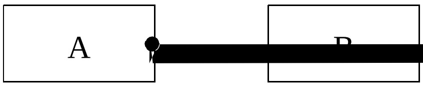
functional relationship

A function is said to be injective if every element of the range of *f* is associated to maximally one element of the domain of *f*.



injective relationship

If every instance of A is an element of the domain of *f* then the function is called total.



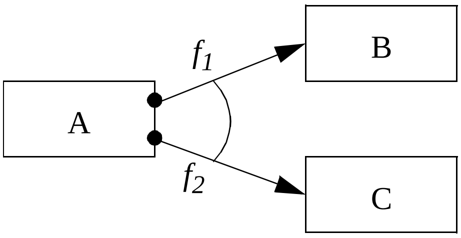
[redacted]

The inheritance relationship is defined as a total and injective function from object type A to object type B

A relationship constraint is called exclusion if the domain of function f_{AB} and the domain of f_{AC} do not intersect.

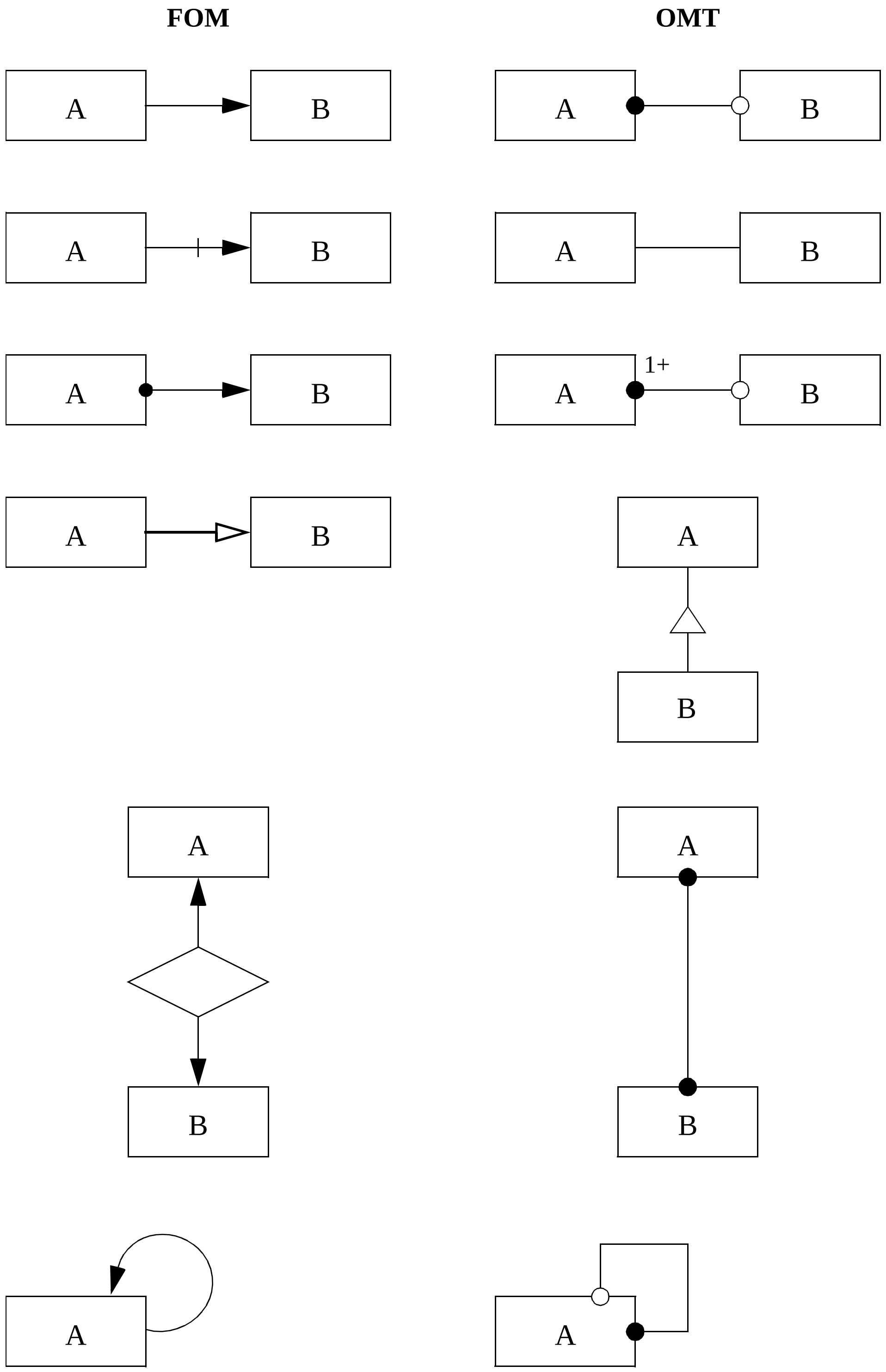
[redacted]

A relationship constraint is called a key constraint if an instance of object type A is uniquely identified by an element of the domain of a total function f_1 and an element of the domain of a total function f_2.



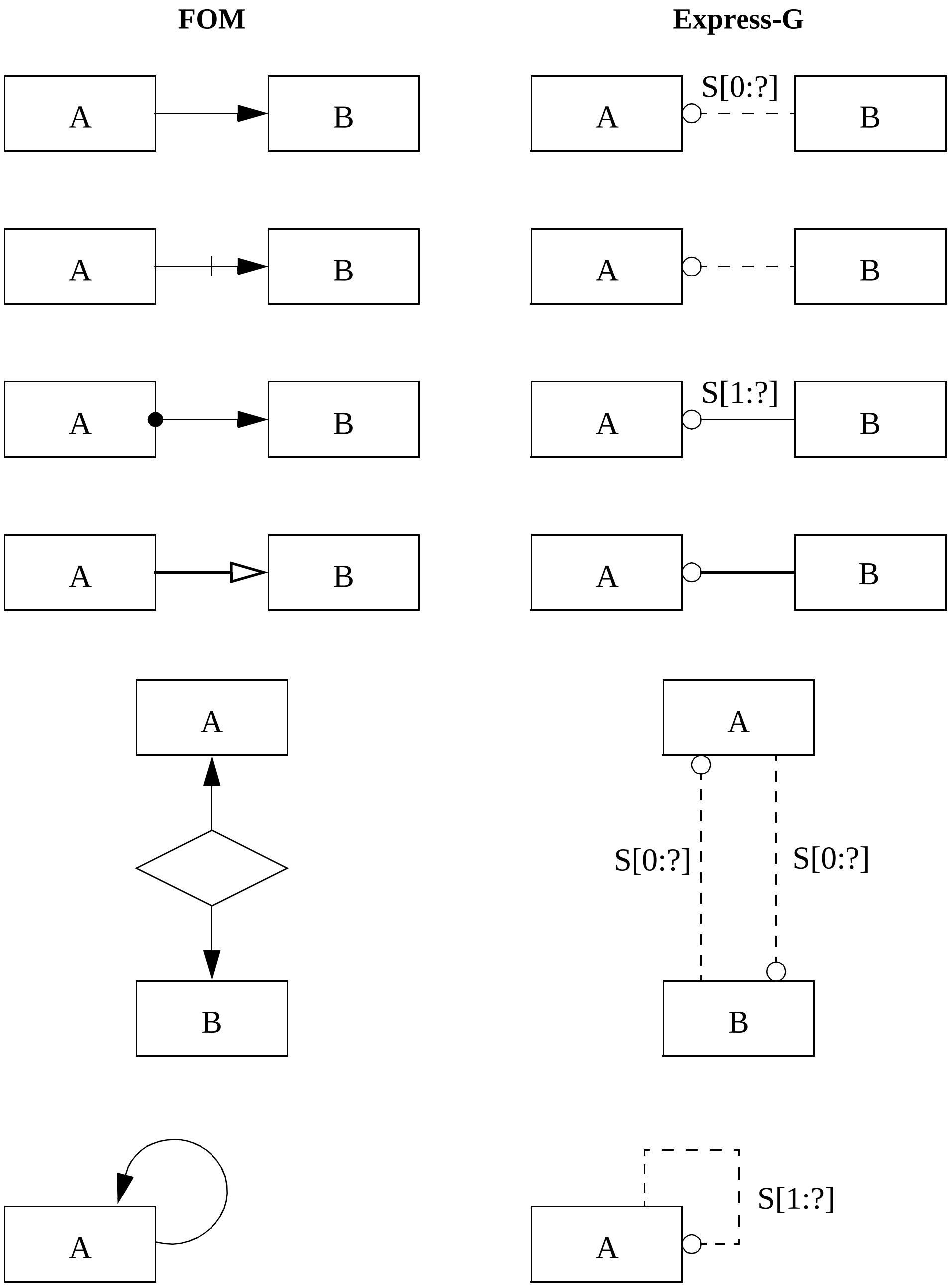
A.1.2 Object Modelling Technique

For those who are familiar with the OMT notation [Rumbaugh 1991], a comparison is made with Functional Object Modelling. Since each notation has its own way to express objects, relationships and constraints it is not always possible to present the alternative graphical notation.


FOM
OMT
A
B
A
B
A
B
A
B
A
B
1+
A
B
A
B
A
B
A
B
A
B
A
A

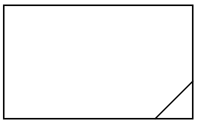
A.1.3 Express-G

Express-G is a graphical subset of Express [ISO/TC184 1991]. A comparison is made with Functional Object Modelling as far as possible. Sets in Express-G are denoted by the letter *S* with a lower and upper bound and specification. When a ‘?’ is given instead of a fixed number, the number of elements is not bounded from above.

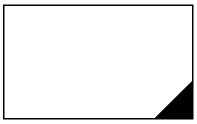


A.2 Transition system

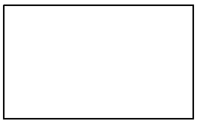
Transition systems are modeled in this thesis as hierarchical colored Petri nets (see Section 7.3 on page 94). In hierarchical Petri nets complex transitions can be decomposed into elementary transitions. Complex transitions are also referred to as systems (as in ExSpect and in system theory [Leeuw 1972]), elementary transitions are also referred to as processors (as in ExSpect). At every level of decomposition transitions can be connected to places. A special type of place is a store, which can contain exactly one token. Places are connected to transitions by directed arcs. Stores always have a bidirectional connection to an elementary transition. The following graphical notations are used to define a Petri net:



complex processor

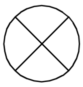


elementary processor



complex or elementary processor

channel



store

A.3 System definitions

The Message Exchange Model is described using a combination of graphical representations for object modelling and process modelling. The top level system in the MEM is the project system. The project system is composed of a set of interconnected participant systems. A formal definition of the project schema and the participant schema is given using a mathematical specification language [Hee 1994].

Definition A.4 A participant schema consist of a tuple *(partname, activityin, activityout),* where

i. partname = string

ii. activityin ∈ activities

iii. activityout $\in$ activities

where

activities = activity $\rightarrow$ $\boldsymbol{P}$(object $\times$ aspect)[1]
activity = string
object = string
aspect = string

Definition A.5 A <u>project</u> schema consists of a tuple *(partactin, partactout, agreemin, agreemout)*, where

i. partactin $\in$ partname $\rightarrow$ activityin[2]

ii. partactout $\in$ partname $\rightarrow$ activityout

iii. agreemin $\in$ activities

iv. agreemout $\in$ activities

v. agreements $\in$ activity $\rightarrow$ (acttype $\times$ partname $\times$ exectime)

where

acttype = {‘task’, ‘request’}
exectime = real

As an example of schema formalization, the core business definition of the floor_openings_indication activity of the contracting company and the first order agreement of this activity in the case project is presented. Note that in a first order agreement the activity definitions are a exact copy of the core business definitions. To understand the relation between activity definitions and the messages that are sent, compare the schemata presented with Table 11 and Table 12 on page 119.

partactin.contracting_company.floor_openings_indication[3] =
{(wall, location),
(wall, geometry),
(wall, material),
(stair, geometry),
(stair, location),
(slab, geometry),
(slab, location)}

partactout.contracting_company.floor_openings_indication =

[1] $\boldsymbol{P}$ denotes a powerset

[2] $\rightarrow$ denotes a function

[3] Function application is denoted as f.x instead of f(x)

{(opening, size),
(opening, location),
(outlet, location),
(outlet, type)}

agreemin.floor_openings_indication =
{(wall, location),
(wall, geometry),
(wall, material),
(stair, geometry),
(stair, location),
(slab, geometry),
(slab, location)}

agreemout.floor_openings_indication =
{(opening, size),
(opening, location),
(outlet, location),
(outlet, type)}

agreements.floor_openings_indication =
(‘task’, contracting_company, 5.0)

References

[Aerts 1991]
Aerts, A.T.M., Alblas, G., Hee, K.M., *Conceptueel Modelleren van Informatiesystemen: een leidraad voor systeemontwikkeling*, Schoonhoven: Academic Service, 1991, (in Dutch).

[Alberts 1993]
Alberts, L.K., “YMIR: an ontology for engineering design,” Ph.D. thesis, Enschede: University of Twente, 1993.

[Autodesk 1992]
Autodesk, *AutoCAD Reference Manual,* Neuchatel: Autodesk bv, 1992.

[Boekholt 1984]
Boekholt, J.T., “Bouwkundig ontwerpen: een beschrijving van de structuur van bouwkundige ontwerpprocessen,” Ph.D. thesis, Eindhoven: Eindhoven University of Technology, 1984.

[Booch 1994]
Booch, G, *Object-oriented analysis and design: with applications*, Amsterdam: Benjamin/Cummings, 1994.

[Bouman 1973]
Bouman A.M., Siewertsen M., Zee A. Th., *Bouwclassificaties: grondslagen, onderzoek*, Stichting Bouwresearch-rapport nr. 40, Alphen aan den Rijn: Samsom, 1973, (in Dutch).

[Carnap 1939]
Carnap, R., “Foundations of Logic and Mathematics,” *International Encyclopedia of United Science Series*, Vol. I, No.3, Chicago: University of Chicago Press, 1939.

[Cherry 1966]
Cherry, C., *On human communication: a review, a survey and a criticism,* Cambridge: MIT Press, 1966.

[CIB 1973]
CIB, “The SfB system; authorized building classification system for the use in project information and related information,” CIB-rapport no. 22, Rotterdam: CIB, 1973.

[CIMdata 1994]
CIMdata, Inc., “State of PDM Industry,” Proc. PDM Europe 1994, Amsterdam: CIMdata, 1994.

[Eastman 1991a]
Eastman, C.M., Bond, A.H., Chase, S.C., “A data model for design databases,” Los Angeles: University of California, 1991.

[Eastman 1991b]
Eastman, C.M., Kutay, A., “Transaction management in design databases,” in: Duvvuru Sriram, Robert Logchner and Shuichi Fukuda (eds.), Proc. Computer aided cooperative product development, MIT-JSME workshop, November 20-21, 1989, Cambridge, Berlin: Springer, 1991.

[EDI 1994]
EDI forum, “The journal of electronic data interchange,” vol.7(1994), No.4, Oak Park: EDI Group, 1994.

[ExSpect 1993]
ExSpect team, “ExSpect User Manual,” Eindhoven: Eindhoven University of Technology, 1993.

[Gielingh 1994]
Gielingh, W.F., “European R&D related to Product Data Technology,” in: A.G. Stuut, L. Zuijdgeest (eds.), PDI/CALS Jaarboek 1994, Zoetermeer: PDI/CALS centrum, 1994, (in Dutch).

[Hammer 1994]
Hammer, M., Champy, J., *Reengineering the corporation: a manifesto for business revolution*, London: Nicholas Brealy Publishing Limited, 1994.

[Hee 1989]
Hee, K.M., van, Somers, L.J., Voorhoeve, M., “Executable specifications for distributed informations systems,” in: E.D. Falkenberg, P. Lindgreen (eds.), Information system concepts: an in-depth analyses, Amsterdam: North Holland, 1989.

[Hee 1994]
Hee, K.M., van, *Information systems engineering: a formal approach*, Cambridge: Cambridge University Press, 1994.

[Hofman 1994]
Hofman, W.J., *A conceptual model of a Business Transaction Management System*, Ph.D. thesis, Den Bosch: Tutein Nolthenius, 1994.

[IOP 1989]
IOP, “BIM: Bouw Informatie Model: versie 3.1,” ed. by P.H. van Merendonk en D.J. van Dissel, Rotterdam: IOP-Bouw, 1989.

[ISO 1984]
ISO, “Information Processing systems - Open Systems Interconnection - Basic Reference Model,” ISO 7498, Geneva: ISO, 1984.

[ISO 1985]
ISO, “Concepts and Terminology for the Conceptual Schema and the Information Base,” Geneva: ISO, 1985.

[ISO 1988]
ISO, “Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) - Application Level Syntax Rules,” ISO 9735, Geneva: ISO, 1988.

[ISO/TC59 1993]
ISO/TC59, “Classification of Information in the Construction Industry,” in: ISO Technical Report, ISO Technical Committee 59, Sc 13, WG2, Geneva: ISO, 1993.

[ISO/TC184 1991]
ISO/TC184, “Part 11: Descriptive Methods: The Express Language Reference Manual,” in: Industrial automation systems and integration - Product data representation and exchange, Draft International Standard, ISO DIS 10303-11, Geneva: ISO, 1991.

[ISO/TC184 1993a]
ISO/TC184, “Part 203: Configuration controlled design,” in: Industrial automation systems and integration - Product data representation and exchange, Draft International Standard, ISO DIS 10303-203, Geneva: ISO, 1993.

[ISO/TC184 1993b]
ISO/TC184, “Part 1: Overview and fundamental principles,” in: Industrial automation systems and integration - Product data representation and exchange, Draft International Standard, ISO DIS 10303-1, Geneva: ISO, 1993.

[ISO/TC184 1993c]
ISO/TC184, “Part 42: Integrated resources: Geometric and topological representation,” in: Industrial automation systems and integration - Product data representation and exchange, Draft International Standard, ISO DIS 10303-42, Geneva: ISO, 1993.

[ISO/TC184 1994]
ISO/TC184, “Part 225: Building Elements using Explicit Shape Representation,” in: Industrial automation systems and integration - Product data representation and exchange, Draft International Standard, ISO DIS 10303-225, Geneva: ISO, 1993.

[ISO/TC184 1995]
ISO/TC184, “The Process Modeling Language EXPRESS-P,” in: Industrial automation systems and integration - Product data representation and exchange, Working Draft, Geneva: ISO, 1995.

[Janssen 1993]
Janssen, A.P.J., “Perspectieven voor EDI en PDI bij de Hollandsche Beton Groep nv: Strategisch Technologiebeleid,” Eindhoven: Eindhoven University of Technology, 1993, (in Dutch).

[Jensen 1991]
Jensen, K. “Colored Petri Nets: A high level language for system design and analyses,” in: G. Rozenberg (ed), Advances in Petri Nets 1990, Lecture Notes in Computer Science 483, Berlin: Springer Verlag, 1991.

[Joosten 1995]
Joosten, S., “Conceptual Framework for Workflow Management Support Systems,” Enschede: Centre of Telematics and Information Technology, University of Twente, 1995.

[Leeuw 1972]
Leeuw, de, A.C.J., “Models, information and control,” Eindhoven: Eindhoven University of Technology, 1972.

[Lucardie 1994]
Lucardie, G.L., “Functional object-types as a foundation of complex knowledge-based systems,” Ph.D. thesis, Eindhoven: Eindhoven University of Technology, 1994.

[Luiten 1994]
Luiten, G.T., “Computer Aided Design for Construction in the Building Industry,” Ph.D. thesis, Delft: Delft University of Technology, 1994.

[Lundeberg 1981]
Lundeberg, M., Goldkuhl, G., Nilson, A., *Information systems development: A systematic approach*, Englewood Cliffs: Prentice Hall Inc., 1981.

[Peach 1994]
Peach, R.W., *The ISO 9000 handbook 2nd edition*, Fairfax Virginia: CEEM Information Services, 1994.

[Rumbaugh 1991]
Rumbaugh, J., et al., *Object-oriented modeling and design*, Englewood Cliffs: Prentice-Hall International, 1991.

[SBR 1990]
SBR, “Informatie-overdracht via tekeningen,” rapporteur J. Vermaas, co-rapporteurs D.B. Joseph, R.P. Geraedts, Stichting Bouwresearch-rapport nr. 211, Rotterdam: Stichting Bouwresearch, 1990, (in Dutch).

[SBR 1992]
SBR, “Bouworganisatievormen in Nederland,” Rotterdam: Stichting Bouw Research, 1992, (in Dutch).

[Sowa 1984]
Sowa, J.F., *Conceptual Structures: Information Processing in Mind and Machine*, Amsterdam: Addison-Wesley, 1984.

[Spinner 1982]
Spinner, M., *Elements of Project Management*, Englewood Cliffs: Prentice-Hall International, 1982.

[STABU 1990]
STABU, “STABU standaard 1990: bestekssystematiek voor de woning- en utiliteitsbouw,” publicatie van de Stichting Standaardbestek Burger- en Utiliteitsbouw, Ede: STABU, 1990, (in Dutch).

[Toffler 1990]
Toffler, A., *Powershift: knowledge, wealth, and violence at the edge of the 21st century*, London: Bantam Press, 1990.

[Tolman, et al. 1989]
Tolman, F.P., et al., “Four years of product modelling: collected papers,” BI-89-40, Rijswijk: TNO Building and Construction Research, 1989.

[Verhoef 1995]
Verhoef, M., Liebich, T., Amor, R., “A multi-paradigm mapping method survey, Proc. CIB W78 Seminar on Modeling of buildings through their life-cycle,” CIB-publication 180, Stanford: Stanford University, 1995.

[Waard 1992]
Waard, M., de, “Computer aided conformance checking: checking residential building designs against building regulations with the aid of computers,” Ph.D. thesis, Delft: Delft University of Technology, 1992.

[Wagter 1995]
Wagter, H., “Synchronizing the company: Freedom and Discipline,” in: The Works, vol.2(1995), No.3, Eindhoven: Origin Technology in Business, 1995.

Bibliography

Vries, B. de, “The minimal approach, Proc. CIB W78 Seminar, The computer integrated future,” Eindhoven: Eindhoven University of Technology, 1991.

Vries, B. de, “EDI en PDI, Platformbijeenkomst IGBI,” Utrecht: IGBI, 1992 (in Dutch).

Vries, B. de., “Normalized Message Exchange in Building Industry,” in: K.S. Mathur, M.P.Betts, K.W. Tham (eds.), Management of information technology for construction, Singapore: World Scientific & Global Publishing Services, 1993.

Deiman, E.P., Hubers, J.C., Leeuwen, J.P., Smeltzer, G.T.A., Tacke, H.C.A., Vries, B. de, Zuthpen, R.H.M., “Het Intechno Bouwkundig Model,” BIT Notes 94/5, Eindhoven: Eindhoven University of Technology, 1994, (in Dutch).

Vries, B.de, Somers, L.J., “Simulation of the communication process,” in: R.J. Scherer (ed.), European Conference on Product and Process Modelling in the Building Industry, Rotterdam: A.A. Balkema, 1994.

Vries, B. de, Somers, L.J., “Message exchange in the building industry, Automation in Construction,” Vol. 4(1995), No. 2, pp. 91-100.

Vries, B. de, Wagter, H., “Message development in the building process, Proc. CIB W78 Seminar on Modeling of buildings through their life-cycle,” CIB-publication 180, Stanford: Stanford University, 1995.

Bauke de Vries

Molvense Erven 35
5672 HH Nuenen
040 2472388 (work) 040 2834330 (home)
B.d.Vries@tue.nl

Education:

1957 **Born in Velp, The Netherlands**

1969 to 1977 **Mavo, Havo, Atheneum-B at Kohnstam-Mavo in Velp and Rhedens Lyceum in Rozendaal**

1977 to 1982 **Department of Architecture at the Delft University of Technology.**

1985 **Master's of Science degree at the Department of Architecture and Building at the Eindhoven University of Technology.** The study was concluded with a (structural) design of an accountant's office and a proposal for a simplified method for non-linear calculation of structural building elements. The building design and the numerical research was performed under supervision of Prof. J.W. Kamerling.

1996 **Ph.D. at the Eindhoven University of technology.** Thesis: Communication in the Building Industry: A strategy for implementing electronic information exchange. The research was performed under supervision of Prof.ir. H. Wagter (Department of Building and Architecture) and Prof.dr. K.M. van Hee (Department of Computing Science)

1999 **Sabbattical at the Eidgenossische Technische Hochschule Zürich, Department of Architecture, CAAD Group.** Research subject: Manipulation of Objects in a 3D environment using Gestures and Voice. Supervision: Prof. G. Schmitt and Prof. M. Engeli.

Employment:

1987 to present **Associate Professor in the Design Systems group of the department of Building and Architecture at the Eindhoven University of Technology.**

Research expertise:

Computer Aided Architectural Design

Product and Process Modelling

Design information storage and retrieval

VR technology and Interfaces

CAD tools and Sketch tools

Geometric modelling in Architectural Design

Knowledge Based Systems

Educational activities

CAAD and Design Methodology course for first years students

Capita Selecta course for graduate students

ICT module in the Post Doctoral programme on Architectural Design Management Systems

Management activities

Chair ad interim of the Design Systems group

Management of the research on the VR-DIS platform

1987 to 1996 **I.T. Manager of the Department of Building and Architecture**

Advising the management of the department about local hardware and software investments.

Negotiating with the Computing centre of the university about central hardware and software investments.

Writing I.T. strategy reports for the department.

Defining functional specifications for software applications and management of the implementation.

Developing client applications and server applications using a Oracle DBMS.

Using software engineering tools such as SDW.

1985 to 1987 **Software engineer at W&B Software.**

Analyses of design and calculation methods.

Software development of building applications.

Dissertation

Vries, B. de (1996) *Communication in the Building Industry: A strategy for implementing electronic information exchange*, Ph.D. Thesis, Department of Architecture, Building and Planning, Eindhoven University of Technology.

Scientific Publications:

Vries, B. de, A.J. Jessurun and M. Engeli (2000) Development of an intuitive 3D sketching tool, In: *Proceedings of the 5-th International Conference on Design and Decision Support Systems in Architecture and Urban Plannin*g, Nijkerk, The Netherlands, August 22-25, 2000, (forthcoming).

Leeuwen, J.P. van and B. de Vries (2000) Modeling with Features and the Formalization of Early Design Knowlwedge, In:*Proceedings of the 5-th International Conference on Design and Decision Support Systems in Architecture and Urban Plannin*g, Nijkerk, The Netherlands, August 22-25, 2000, (forthcoming).

Dijkstra, J., H.J.P. Timmermans and B. de Vries (2000) A Multi Agent Model for Visualisating Simulated User Behaviour within the Building Environment, In: *Proceedings of the 5-th International Conference on Design and Decision Support Systems in Architecture and Urban Plannin*g, Nijkerk, The Netherlands, August 22-25, 2000, (forthcoming).

Segers, N., B. de Vries and H.H. Achten (2000) Architectural Design Tools, In: *Proceedings of the 5-th International Conference on Design and Decision Support Systems in Architecture and Urban Plannin*g, Nijkerk, The Netherlands, August 22-25, 2000, (forthcoming).

Vries, B. de, S.M. Mallory-Hill, E.H. de Groot and P.G.S. Rutten (2000) VR-DIS: Een Interactieve Integrale Ontwerpomgeving, TVVL Magazine, Mei, 2000, Jaargang 29, (forthcomming).

Leeuwen, J.P. van and B. de Vries (2000) A Decision Support System for Building Refurbishment Design, In: Proceedings of The International Conference on Construction Information Technology, Reykjavik, Iceland, June 28-30, 2000, (forthcoming)

Achten, H.H. and B. de Vries (2000) DDDoolz: A Virtual Reality Sketch Tool for Early Design, In: *Proceedings of The Fifth International Conference on Computer Aided Architectural Design Research in Asia, Singapore*, May 18-19, 2000, (forthcoming).

Tan, A., H.J.P. Timmermans and B. de Vries (2000) Investigation of Human Behavior in a Virtual Environment, Position paper on The Virtual Worlds and Simulation Conference, San Diego, California, January 23-27, 2000.

Vries, B. de, A.J. Jessurun and R.H.M.C. Kelleners (2000) Using 3D Geometric Constraints in Architectural Design Support Systems. In:*Proceedings of the 8-th*

International Conference in Central Europa on Computer Graphics, Visualization and Interactive Digital Media 2000, University of West Bohemia, Czech Replublic, Februari 7-10, 2000, pp.76-83

Wijk, J.,J. and B. De Vries (1999) Towards an Understanding 3D VR Architectural Design System, In: *Proceedings of the 15-th Twente workshop on Language Technology, "Interactions in Virtual Worlds"*, Enschede, The Netherlands, May 19-21, 1999, pp. 219-224.

Mallory-Hill, S.M. and B. de Vries (1999) Leren van verleden bij ontwerpen binnenruimten, De Bouwadviseur, vol. 41 afl. 6, pp 30-33.

Groot, de, E.H., S.M. Mallory-Hill, R.H.M. van Zutphen and B. de Vries (1999) An experimental design system for the very early design stage. In: *Proceedings of the 8th International Conference on Durability of Building Materials and Components*, Vancouver, Canada, May 30 – June 3, 1999.

Achten, H.H., S.M. Mallory-Hill and B. de Vries (1998) Workshop Case Based Reasoning -–Featuring Professor Mary Lou Maher, Designs Systems Report, Eindhoven University of Technology.

Vries, B.de and H.H. Achten (1998) What offers Virtual Reality to the Designer. Paper presented at the workshop *The Art of Design at the Third Biennual World Conference on Integrated Design & Process Technology*, Berlin, Germany, July 6-9, 1998.

Vries, B. de and A.J. Jessurun (1998) An experimental Design System for the Very Early Design Stage, In: *Proceeding of the 4-th International Conference on Design and Decision Support Systems in Architecture and Urban Planning*, Maastricht, The Netherlands, 26-29 July, 1998.

Achten, H.H., A.J. Jessurun, J.P. van Leeuwen and B. de Vries (1998) *Virtual Reality Design Information System: Distributed Interactive Simulations*, Designs Systems Report, Eindhoven University of Technology.

Vries, B. de and A.J. Jessurun (1998) Features and Constraints in Architectural Design. In: *Proceedings of DETC'98, 1998 ASME Design Engineering Technical Conference* (CD-ROM), September 13-16, 1998, Atlanta, Georgia.

Vries, B, de., J.P. van Leeuwen, H. H. Achten, Design Studio of the Future (1997) In Drogemuller, R. (ed.): Proceedings of the CIB W78 workshop, Information Technology Support for Construction Process Reengineering, Cairns, Queensland, Australia, July 9-7, 1997, pp. 139-146.

Vries, B. de (1997) Ontwerpen van ruimte in de ruimte, De Bouwadviseur, juli/augustus 1997, p. 16.

Somers, L.J. and B. de Vries (1997) Simulation of Project Information Exchange, In: Proceedings of the ESS 97 conference "Simulation in Industry, Passau,Germany, October 19-22, 1997.

Vries, B. de (1997) Message Development in the Building Process, In:*Proceedings of the CIB W96 conference Managing Requirements & Design Information*, The Hague, The Netherlands, pp.39-51.

Vries, B. de (1995) Message Exchange in the Building Industry,*Automation in Construction*, Vol. 4(2), Elsevier, Amsterdam, pp. 91-100.

Vries, B. de (1995) Message Development in the Building Process, Proceedings of the CIB W78 workshop, Modeling of buildings through their lifecycle, Stanford, 1995.

Vries, B. de (1994) Simulation of the Communication Process, In R.J. Scherer (eds): *Proceedings of European Conference on Product and Process Modelling in the Building Industry*, A.A. Balkema, Dresden, pp. 233-236.

Vries, B. de (1993) Normalized Message Exchange in the Building Industry, K.S. Mathur, M.P. Betts, K.W. Tham (eds): *Managament of information technology for construction,* World Scientific & Global Publishing Services, Singapore, pp. 125-133.

Vries, B. de (1991) The minimal approach, In:*Proceedings of the CIB W78 workshop, The computer integrated future,* Eindhoven, The Netherlands, September 16-17, 1991.

Presentations

DDSS programme, presentation on DDSS Research meeting at the Department of Building and Architecture, Eindhoven University of Technology, 2000.

VRDIS: Een Interactive Intrgrale Ontwerpomgeving, presentation on TVVL seminar in Zeist, 2000.

ETH Sabbattical, presentation at the DDSS Research meeting at the Department of Building and Architecture, Eindhoven University of Technology, 1999.

Manipulation of Objects in a 3D environment using Gestures and Voice, presentation on Sabbattical at the CAAD group of the ETH i

Communication in the Building Industry, lecture for Department of Architecture, Delft University of Technology, 1996.

Proces and Datamodellen voor PDM, presentation on PDM Platform meeting, Eindhoven University of Technology, 1996.

EDI en PDI, presentation on IGBI seminar, Jaarbeurs Utrecht, 1995.

IT Beleid, presentation on Computing Center seminar, Eindhoven University of Technology, 1995.

Ph.D. Committees

G. Wets (1998)
M.A.D. Wang (1998)
J.P. van Leeuwen (1999)
A. Hendricx (2000)
S. M. Mallory-Hill
J. Dijkstra
M.K.D. Coomans
N.M. Segers

Sofware Tools

MEM
Communication process simulation system demonstrating the Message Exchange Model, implemented using ExSpect from the Department of Computing Science, based on Petri Nets and the Ph.D. research of B. de Vries

Constraint Tester
A prototype demonstrating the use of geometric constraints, implemented in C, using OpenGL and based on the Ph.D. research of R. Kelleners from the Department of Computing Science.

DDDoolz
A desktop-VR three-dimensional sketch tool for the support of mass and space studies in Architectural design, implemented using WorldUp and based on the Face Orientation Method developed by B. de Vries.

Review Committees

- American Society of Mechanical Engineering Conference
- Computer Aided Architectural Design Futures Conference
- Education in Computer Aided Architectural Design Conference
- Evaluation of Research Grant of the Katholieke Universiteit Leuven.

Organisation of Conferences

- Design and Decision Support Systems in Architecture and Urban Planning Conference,1998, 2000.
- Case-Based Reasoning Workshop, 1998.
- Design Research in the Netherlands Symposium, 2000.
- Computer Aided Architectural Design Futures Conference, 2001.

International Projects

- Brite Euram Project on the development of a Socio- technological- Commercial Framework and Decision Support System for Building Refurbishment.
- International Collaborative Design project for graduate students hosted by the PBL Laboratory of Stanford University and in cooperation with building enterprises. Funded by the Eindhoven University and The Hollandse BetonGroep (in progress).

National Projects

- Building Management Simulation Center for training construction managers under controlled but 'realistic' circumstances using VR technology. Funded by the European Community.
- Interactief Ruimtelijk Programma van Eisen prototype system for the support of interactive definition of the spatial layout of a building. Funded by the Architectuur Stimuleringsfonds(in progress).

Consulting

- Advice on use of electronic message exchange in the construction industry on request of the CUR committee.
- Advice on strategic ICT management on the University level on request of the University Board.
- Interim manager Notebook Service Centrum during half a year on request of the University Board

Memberships

- Vice Chair of the CAAD Futures Foundation.
- Auditing committee on the scientific research of the Department of Building and Architecture (VCW).

Courses

- Software Design and Development
- Oracle courses on Data Base Management and Data Base Application Development
- Object Oriented Data Bases
- Personnel Management

- Project Management

Languages

Native language is Dutch. Good command in writing and speaking Dutch and English. Good speaking in German.

Samenvatting

Informatie Technologie in het algemeen en electronische communicatie in het bijzonder zijn van invloed op organisatiestructuren. Nieuwe communicatiemedia zullen bestaande communicatie processen en -patronen wijzigen en zij kunnen de huidige activiteiten van bedrijven beïnvloeden. Om de invloed van communicatiemedia te kunnen analyseren is het noodzakelijk een duidelijk beeld te hebben van de richting van de informatiestroom en van de inhoud van de informatiestroom gedurende een willekeurig bouwproject. Gegeven dit beeld kan worden vastgesteld of de huidige organisatie en de bedrijfsactiviteiten op de feitelijke informatiebehoefte zijn afgestemd. Tevens kan aan de hand van de frequentie en de inhoud van de informatiestromen worden nagegaan welke opslagstructuur en welk transportmedium per uitgewisseld bericht het meest adequaat zijn. Doel van het onderzoek is een beschrijving van het informatie-uitwisselingsproces voor een bouwproject te definiëren die het hiervoor beschreven beeld als resultaat oplevert, waarbij volledigheid en consistentie van gegevens gedurende het project impliciet is gewaarborgd.

Communicatie media

De vakdisciplines die deel uitmaken van de bouw hebben een lange traditie in het uitdrukken van hun gedachten in de vorm van tekst en tekeningen. Een logische eerste stap om meer inzicht te verwerven in het communicatieproces, is het analyseren van de huidige communicatiemethoden. Ofschoon er vandaag de dag een grote diversiteit aan communicatiemedia beschikbaar is zoals video, maquette, etc., concentreert dit onderzoek zich op documenten bestaande uit tekst en/of tekeningen. Reden hiervoor is dat deze documenten vaak contractuele waarde hebben en dus aan een zeker kwaliteitsniveau dienen te voldoen. De aandacht is met name gericht op de wijze waarop de informatie is gerepresenteerd en gestructureerd. De basisconcepten die ten grondslag liggen aan de informatieuitwisseling op papier moeten terug zijn te vinden in de formele beschrijving van het proces.

Aangrenzend onderzoek

Communicatie tussen bedrijven in de bouw is onderwerp van onderzoek bij vele onderzoeksinstellingen en binnen diverse internationale commissies. Ieder onderzoek kiest daarbij zijn eigen invalshoek ten aanzien van het communicatie-probleem. Om de resultaten van de diverse onderzoeken te kunnen vergelijken wordt eerst een onderzoeksraamwerk gedefinieerd. Een beperkte selectie van lopende en reeds afgeronde onderzoeken wordt binnen dit raamwerk geëvalueerd. Met name is gekeken naar de wijze waarop het bouwproces en het bouwprodukt zijn beschreven en hoe deze twee beschrijvingen aan elkaar zijn gerelateerd.

Proces analyse

De algemene karakteristieken van het bouwproces zijn inmiddels genoegzaam bekend, te weten: unieke produkten, relatief kleine bedrijven en een steeds wisselende samenstelling van deelnemende bedrijven in een projekt. Ten aanzien van de informatieverwerking kenmerkt het bouwproces zich door formaliseerbare aktiviteiten zoals berekeningen en niet-formaliseerbare aktiviteiten, waarbij de creativiteit van de ontwerper een belangrijke rol speelt. De informatiepatronen binnen een bouwprojekt zijn nauw gerelateerd aan de vorm van de projektorganisatie. De projektorganisatie is weer van invloed op de wijze waarop de afbeelding van informatie plaats moet vinden tussen de zendende en de ontvangende partij. Traditioneel wordt het bouwproces veelal voorgesteld als een opeenvolging van bouwfasen. In dit onderzoek wordt een willekeurig bouwprojekt als uitgangspunt genomen en beschreven als een netwerk van aktiviteiten. Tussen de deelnemende participanten die elk één of meer activiteiten uitvoeren wordt informatie uitgewisseld.

Informatie analyse

Een bericht dat wordt uitgewisseld bestaat uit drie hoofdonderdelen: (1) produkt informatie, (2) de opdrachtomschrijving en (3) de enveloppe van het bericht. De informatiebeschrijving is ontleend aan de huidige situatie. De concepten die hieraan ten grondslag liggen gelden als basis voor het definiëren van berichten die wellicht worden uitgewisseld middels andere communicatiemedia dan de huidige. De informatiebeschrijving resulteert in een generieke berichtstructuur waarbinnen de informatie van elk van de drie onderdelen kan worden vastgelegd voor een specifiek bericht. Uitgangspunt daarbij is dat informatieuitwisseling mogelijk wordt gemaakt zonder menselijke tussenkomst en dat bewaking van de consistentie van de gegevens binnen een projekt mogelijk is.

Message Exchange Model

De concepten die zijn geïntroduceerd voor de beschrijving van het informatie-uitwisselingsproces en van de informatie zelf, worden gecombineerd tot één model, te weten het Message Exchange Model (MEM). Met dit model is het mogelijk om de informatie die wordt gebruikt op enig moment gedurende een bouwproject expliciet te beschrijven. De procesbeschrijving en de informatiebeschrijving zijn generiek omdat ze van toepassing zijn op elk willekeurig bouwprojekt. Toepassing van het model voor een concreet bouwprojekt betekent het afbeelden van de randvoorwaarden die gelden binnen het bouwprojekt op het MEM. Om een ‘afgewogen’ keuze te kunnen maken uit de beschikbare communicatiemethoden, wordt een classificatie voor berichten en transport media voorgesteld. Ten slotte wordt er een ontwikkelingsstrategie geformuleerd uitgaande van het MEM voor de introductie van electronische informatieuitwisseling tussen bedrijven.

Simulatiemodel

Om de ontwikkelingsstrategie te ondersteunen is een hulpgereedschap ontwikkeld, te weten een simulatiemodel voor bouwprojekten. Het simulatiemodel is uiteraard gebaseerd op het MEM, maar kent een aantal beperkingen. Het simulatiemodel is ontwikkeld in ExSpect. ExSpect is een ontwikkelomgeving voor het ontwerpen en simuleren van logistieke processen. In het simulatiemodel kan de richting waarin de informatie zich beweegt worden getraceerd, en de inhoud van de berichten die worden uitgewisseld kan worden geïnspecteerd. Tijdens het simulatieproces wordt gecontroleerd op volledigheid en consistentie. Na afloop van de simulatie is tevens van elke participant bekend met welke deelverzameling van de totale produktinformatie hij te maken heeft.

Case study

Een case study is uitgevoerd bij een bedrijf dat betonvloeren produceert. Vanuit het gezichtspunt van het bedrijf was de doelstelling meer inzicht te krijgen in de toepasbaarheid van de beschikbare ‘communicatiestandaarden’. Vanuit onderzoeksgezichtspunt was de doelstelling de concepten die voortkomen uit het onderzoek te testen. Om aan beide doelstellingen tegemoet te komen is een concreet bouwprojekt geanalyseerd en ingevoerd in het simulatiemodel. De uitkomsten van de simulatie zijn verwerkt tot schema’s die het ‘ideale’ project weergeven. Dezelfde schema’s zijn gemaakt van het bouwprojekt zoals dat in werkelijkheid heeft plaatsgevonden. Vergelijking van schema’s leidt tot conclusies over: de kernaktiviteiten van het bedrijf, de projectorganisatie, de informatierichting en de inhoud van de uitgewisselde informatie. Tenslotte is voor de berichten een aanzet gegeven van de classificatie die dient als hulp bij de selectie van een communicatiemedium.

Discussie en conclusie

Het resultaat van dit onderzoek bevindt zich op het grensvlak van twee andere onderzoeksgebieden, namelijk Product Data Management (PDM) en Product Data Interchange (PDI). Het Message Exchange Model (MEM) is geëvalueerd in de context van PDM en PDI. Hieruit blijkt dat PDM ondersteuning biedt bij informatiemanagement en dat PDI zich toelegt op de informatiestructuur-definitie. Het MEM is geen gereedschap maar een methode om proces en informatie binnen één model te beschrijven. In het model is zowel sturingsinformatie als inhoudelijke informatie aanwezig, waarmee de relatie met PDM en PDI is gelegd.

Algemene conclusies met betrekking tot de invoering van electronische communicatie zijn:

- Bedrijven zijn ongewis over de invloed van electronische communicatie op hun organisatie en nemen een afwachtende houding aan.
- Voordat implementatie van electronische informatie-uitwisseling op gang zal komen moet eerst een duidelijk beeld worden geschapen van de feitelijke informatiebehoefte.
- Om een duidelijke beeld van het informatie-uitwisselingsproces te verkrijgen is een formele beschrijving van het proces zelf noodzakelijk zoals bijvoorbeeld in het Message Exchange Model.
- Afbeelding en conversie van gegevensstructuren is onvermijdelijk omdat het bedrijfsspecifieke model, het applicatiemodel en het berichtmodel in de meeste situaties onafhankelijk van elkaar zijn ontwikkeld.
- Ontwikkeling van electronische berichten vindt plaats binnen de vakdisciplines, terwijl implementatie plaats moet vinden door participanten die met elkaar samenwerken in een bouwprojekt.
- In plaats van produktbeschrijvingen zouden bedrijven zich moeten toeleggen op activiteitsbeschrijvingen.
- Normen in de bouw zouden moeten worden opgesteld uitgaande van een eenduidig gedefinieerde verzameling eigenschappen, onafhankelijk van de objecten waarop ze betrekking hebben.
- Een bedrijfsdatamodel dient als referentiemodel bij de keuze van applicaties met hun eigen specifieke datastructuur en bij het maken van afbeeldingen tussen bedrijfsinformatie en produkt- of berichtstructuren uit (nieuwe) communicatiestandaarden.