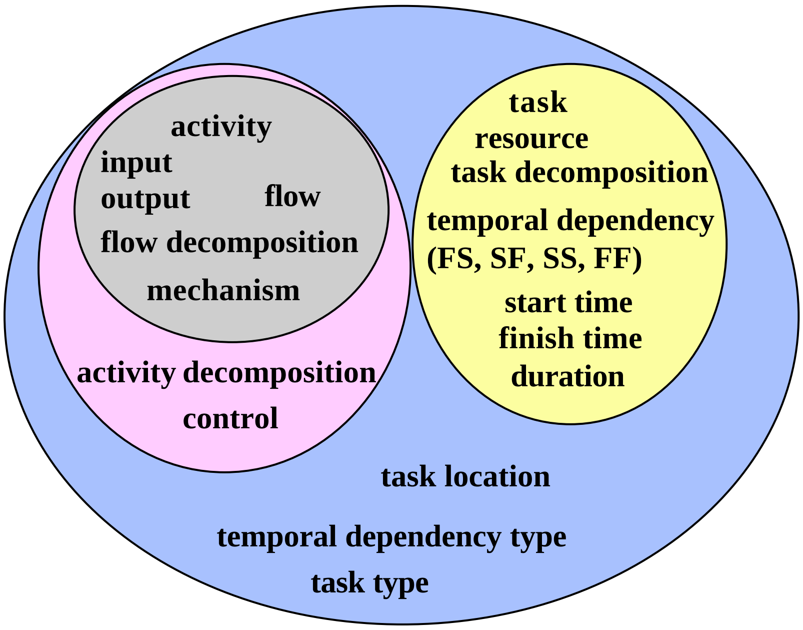
Vesa Karhu

A generic construction process modelling method

A model based approach for process description



KUNGL TEKNISKA HÖGSKOLAN
Royal Institute of Technology
Construction Management and Economics

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in the SAL L1 at the Royal Institute of Technology on the 9th of November 2001, at 1 pm.

Respondent:	Vesa Karhu, LicTech
Opponent:	Prof. Andrew Baldwin
Examination Committee:	Prof. Per Christiansson
	Prof. Rob Howard
	Prof. Jerker Lundequist
Supervisors:	Prof. Bo-Christer Björk and Prof. Örjan Wikforss

Cover page by Vesa Karhu.

ISBN 91-7283-162-6

Royal Institute of Technology, Industrial Economics and Management, Division of Construction Management and Economics
S-100 44 STOCKHOLM, Sweden

Abstract

A variety of modelling methods has been used to model construction processes and projects, either during normal project planning or for process re-engineering efforts or research. One common method, which is widely used by construction industry practitioners, is scheduling. In addition to schedules, some companies have used a simple box-and-arrow method, which graphically resembles schedules, for analysing their working processes. More formal methods such as IDEF0 have been used in re-engineering projects and by researchers. All these methods are limited in scope and cannot be used to model all the aspects of the processes that practitioners are interested in.

A new generic construction process modelling method, GEPM, was developed to overcome the deficiencies of the current methods. GEPM uses object-oriented principles, and has borrowed features, such as activity, task, and temporal dependency, from methods like IDEF0 and scheduling. GEPM is flexible in the sense that the conceptual model can be changed to achieve additional special features. This capability is also supported by the database implementation, which enables users to interact with the developed process models through views that represent partial models. The views support the IDEF0, scheduling, and simple flow methods. There are, though, rules for how to convert between the partial models through views.

The evaluation of GEPM showed that more modelling features, i.e. modelling power, are obtained in comparison with the earlier methods. One of the essential features of GEPM is the distinction between activities and tasks. Activities define how an action will be carried out, generally using predetermined inputs to achieve a predetermined output, whereas tasks are activities with additionally specified starting and finishing times, duration and location. Moreover, a task has a type-attribute that refers to an activity where its overall template is defined.

Before the actual evaluation, case material from a real project was preliminarily tested with GEPM along with the prototype application. It turned out that some additions were needed to the conceptual model of GEPM and to the prototype application.

GEPM can be used for process improvement, process management, and for enhancing communication in a construction process. One usage scenario for GEPM is to define quality systems and reference models, using the activity sub-model and storing the results in the GEPM database. A project-specific model can be derived from the reference model using conversion rules, and it eventually turns into a project specific-schedule with tasks.

Keywords: process, modelling, generic, method, model, database, view

ISBN 91-7283-162-6

Preface

The research presented in this thesis was carried out at VTT Building and Transport (VTT), previously VTT Building Technology, and the Royal Institute of Technology (KTH) between 1992-2001. Several projects have contributed to the results over the years. From the academic point of view, the first part of the research yielded a licentiate thesis in 1997. A natural continuation of the work gradually evolved into a doctoral thesis.

I would like to thank especially Professor Bo-Christer Björk for encouraging me to continue the work and for indispensable comments and suggestions during the research. After Professor Björk moved to a new position at the Swedish School of Economics and Business Administration in Helsinki, Professor Örjan Wikforss has succeeded him as my supervisor. I like to thank him as well as Professor Anders Ekholm, from Lund Institute of Technology, who has been my preliminary inspector and has suggested valuable improvements to the thesis. I would also like to thank Mr Kari Karstila for his expertise on EXPRESS models and modelling in general, Mr Matti Hannus for fruitful discussions, Dr. Kalle Kähkönen for valuable comments and material, and Dr. Pertti Lahdenperä, who was the co-author of the second journal article. My colleagues from KTH have also made many valuable comments during seminars over the years. I thank VTT, which has provided me with an opportunity to finalise the thesis and has been my employer throughout these years. For financing, I would like to thank The Technology Development Centre of Finland (Tekes), VTT, KTH, and the companies that were involved in the MoPo project, Rakennusliike Lipsanen Oy, Parma Betonila Oy, and YIT Corporation.

As I wrote in the preface of my licentiate thesis in 1997, an important factor during my years of research has been my kayaking hobby. A good one-hour kayaking training session is the best way to refresh the mind and soul, and besides, leads to a good physical condition.

Finally, special thanks go to my wife Katarina.

Espoo, September 2001

Vesa Karhu

List of papers

1. Karhu, V. 1997
   Product Model Based Design of Precast Facades
   Electronic Journal of Information Technology in Construction. Vol. 2. http://itcon.org/1997/1/

2. Karhu, V. and Lahdenperä, P. 1999
   A formalised process model of current Finnish design and construction practice
   International Journal of Construction Information Technology, Vol. 7 No 1, pp. 51-71.

3. Karhu, V. 2000
   Proposed new method for construction process modelling
   International Journal of Computer Integrated Design and Construction, Vol. 2 No 3, pp. 166-182.

4. Karhu, V. 2001
   A view-based approach for construction process modelling
   Journal of Computer-Aided Civil and Infrastructure Engineering. (accepted for publication in August, 2001)

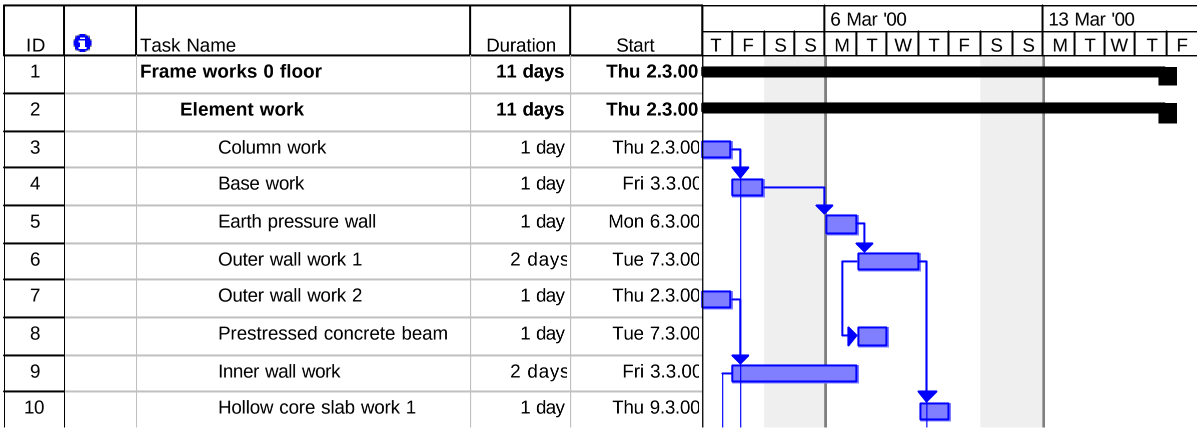
Contents

1 INTRODUCTION

1.1 Background

Construction projects are becoming more complex, involving a large number of companies, in teams varying from one project to another, that collaborate for a limited period of time to design and construct unique engineering artefacts, e.g. buildings. This complexity often leads to specialisation among the various disciplines and so to a possible breakdown in communication [Ahuja 1976]. Because of this complexity, there is a need to model construction processes from many points of view. The importance of visualising the operations involved in a construction project and of facilitating communication has been recognised [Ahuja 1976]. Moreover, Curtis et al. [1992] add that the purposes of modelling are also to facilitate human understanding, and to support process improvement and process management.

Several different variations of construction process modelling methods have been used for construction planning and management. Scheduling has been one of the purposes of these methods. A well-known method using Gantt bar charts has been used by project managers since the early 1900s [Melin and Whiteaker 1981]. Networking techniques have been available since the late 1950s [Fondahl 1980]. Examples of these methods are, for instance, the critical path method (CPM) [Kelley and Walker 1959] and the program evaluation and review technique (PERT). Both of these methods use graphical arrow diagrams and the projects are broken down into their component activities (or tasks). Moreover, another variation is a precedence diagramming technique, which emphasises clearly each activity's precedence in a graphical diagram. An example of a process schedule using a modern software tool is depicted in Figure 1.



ID		Task Name	Duration	Start
1		**Frame works 0 floor**	**11 days**	**Thu 2.3.00**
2		**Element work**	**11 days**	**Thu 2.3.00**
3		Column work	1 day	Thu 2.3.00
4		Base work	1 day	Fri 3.3.00
5		Earth pressure wall	1 day	Mon 6.3.00
6		Outer wall work 1	2 days	Tue 7.3.00
7		Outer wall work 2	1 day	Thu 2.3.00
8		Prestressed concrete beam	1 day	Tue 7.3.00
9		Inner wall work	2 days	Fri 3.3.00
10		Hollow core slab work 1	1 day	Thu 9.3.00

Figure 1. Example of part of a schedule.

Functional modelling methods, such as the IDEF0 [IDEF0 1993], have been used particularly in R&D projects, for instance by Sanvido et al. [1990], to model the process of facility provision, or to model current Finnish design and construction practice [Karhu and Lahdenperä 1999]. In addition, these methods have been used by industry practitioners to model specific problem areas, for instance to describe the process of a building audit [Karstila et al. 2000]. The IDEF0 method defines activities, and flows between the activities, using hierarchical diagrams. However, it leaves out the time concept; i.e. the activities that are represented in a diagram do not have a starting time, a finishing time, or duration, as in scheduling.

Another widely used method in the construction industry is a simple 'box-and-arrow' method, where boxes represent activities and arrows represent flows between the activities. Graphically, such a diagram resembles a schedule. Moreover, there have been attempts to improve project management, for instance in the UK construction industry where the work of the so-called Process Protocol [Kagioglou et al. 1998a, Kagioglou et al. 1998b] has resulted in a formal reference process model. Some authors have used data flow diagramming for modelling the processes of construction [Austin et al. 1996].

A difficulty in modelling the processes is how to model all the required aspects of the underlying reality in such a way that project participants may use them in an effective manner. Another difficulty is how to use reference models in determining project-specific models. Construction process reference models can be described using IDEF0 but specific models are described in schedules [Hannus and Pietiläinen 1995]. Moreover, concerning the process modelling methods themselves, the methods are often limited in scope to a particular viewpoint and thus cannot successfully be used for modelling processes from other points of view. In other words, each participant has a particular point of view from which he or she regards the total process. In addition, many recent R&D projects have demonstrated that process models developed using methods such as IDEF0, which researchers find useful, are often difficult for practitioners to understand [Berg von Linde 2000].

Because of this divergence of views and information needs, or put in another way, the lack of a unified approach with harmonised views, misunderstandings easily arise in interactions between different project participants as they communicate about the process. On the other hand, building-product modelling has tried to define standardised object-oriented descriptions of a building in order to enable seamless exchange of data between CAD tools and building analysis programs [Eastman 1999]. An example of this is the standardisation effort STEP, where application protocols are designed for data exchange of product models of buildings and their parts [ISO 1994a]. A more recent effort is the International Alliance for Interoperability, which is defining so-called Industry Foundation Classes (IFC) for the same purpose [IAIWEB 2000].

The object-oriented approach used in these product-modelling efforts can similarly be used to describe processes. This approach has been used in this thesis to describe a new type of construction process modelling method, where central object classes are activities, tasks, flows, and relations between these. The object-oriented approach also enables a computerised approach, using databases for storing the representations of construction process models.

1.2 Objectives, scope and phases of the research

The main objective of this thesis is:

- To propose a new generic construction process modelling method.

From the above, one may derive secondary objectives such as:

- To propose conversion rules between process modelling methods

- To contribute new knowledge to the process of using reference models and schedules in a combined approach.

The objective of contributing new knowledge to the conversion between reference process models and specific models can be formulated into a hypothesis that it is possible to convert a reference model, described as an activity model, into a schedule with tasks.

Figure 2 depicts the steps that have been taken during this research and that have led to the development of the generic process modelling method GEPM (note that the obvious mechanism, the researcher, has been left out of the figure). The requirements of industry have played a triggering role in the research. The need for a new type of a process modelling method became apparent through the development of a process (and product) model for the design and manufacturing of precast concrete facades, through the development of a reference model of traditional Finnish design and construction practice, and through the study of existing process modelling methods that have been used, or proposed, for modelling construction processes.

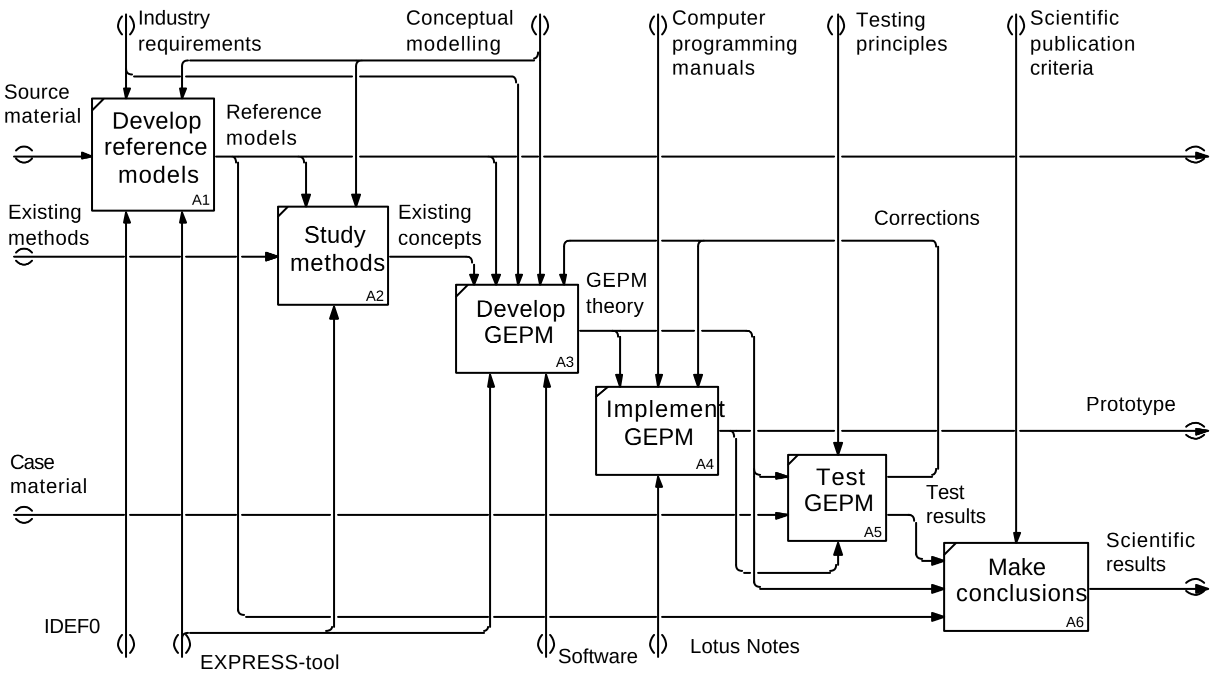


Figure 2. Steps for GEPM development.

The new generic construction process modelling method, called GEPM, is based on six earlier process-modelling methods. These are scheduling, simple 'box-and-arrow' (reference for instance [Hoffner 1997]), IDEF0 [IDEF0 1993], IDEF0v [Austin et al. 1998], IDEF3 [Mayer et al. 1995], and PetriNets [Anon. 1997] (these are described in more detail in Chapter 3). The new modelling method has also been influenced by the object-oriented principles used in product modelling research. The method distinguishes between activities and tasks and may be used to define a building construction process model.

The results have been published as scientific papers in refereed journals. The implementation of the GEPM theory has resulted in a prototype application.

The scope of this thesis is limited to building-construction processes, even though the principles could be applied to other industry fields as well. This limitation is because the test case material used in the various research projects is from building construction projects. A minor limitation is that, since the new generic process modelling method is based on other methods, this thesis can be considered as an attempt to combine these methods.

1.3 Structure of the thesis

This thesis consists of four separate papers that have been published in scientific journals, and a summary of the definition of the research problem, the research process, the results, and the conclusions.

Chapter 1 introduces the subject and discusses the current problems in construction process modelling in general. It presents the objectives, and clarifies the research steps undertaken during the project.

Chapter 2 presents the research methodology used in this thesis. It addresses conceptual modelling, the object-oriented approach, prototyping, and testing as the main approaches. Additionally, this chapter briefly discusses the issue of research and development and relates these to the projects carried out at VTT (the employer of the author).

Chapter 3 presents the state-of-the-art process modelling methods. It also describes how they have been used by researchers or practitioners for describing the building construction process, or parts of it. Furthermore, this chapter presents a number of construction information models. Additionally, a discussion of the requirements for a construction process modelling method and a process modelling software tool is presented. The state-of-the-art descriptions do not include directly the newly developed process modelling method called GEPM, or the associated papers, since these are presented in later chapters of this thesis.

Chapter 4 presents, briefly, the four papers. Three papers have been published in refereed scientific journals, and one paper has been accepted for publication. The presentation of the second paper [Karhu and Lahdenperä 1999] includes a discussion of the respective contributions of the two authors.

Chapter 5 presents the main results, the conceptual model of GEPM and its implementation on a Lotus Notes database platform. It also presents some further developed ideas and discusses the advantages and disadvantages of GEPM. A brief comparison with other existing models is given.

Chapter 6 presents a usage scenario of the method in practice, i.e. how it can be used in a construction company. This chapter also presents the final conclusions.

Appendix I is a re-print of the paper "Product Model Based Design of Precast Facades" from the Electronic Journal of Information Technology in Construction [Karhu 1997].

Appendix II is a re-print of the paper "A formalised process model of current Finnish design and construction practice" from the International Journal of Construction Information Technology [Karhu and Lahdenperä 1999].

Appendix III is a re-print of the paper "Proposed new method for construction process modelling" from the International Journal of Computer Integrated Design and Construction [Karhu 2000].

Appendix IV is the author's version of the paper "A view-based approach for construction process modelling" that was accepted in August, 2001 for publication in the Journal of Computer-Aided Civil and Infrastructure Engineering [Karhu 2001].

2 RESEARCH METHODOLOGY

2.1 Methods used

An important part of research involves the methodology of how to approach and solve a problem in a certain domain or application area, here the problem of developing a new construction process modelling method. One of the main targets in problem solving is to obtain a general view of the problem area, not simply one single part [Gustafsson et al. 1982].

Ackoff [1962] divides a procedure for scientific work into six phases as follows:

- Formulation of a problem
- Construction of a model
- Testing of the model
- Derivation of a solution
- Testing and controlling the solution
- Implementation of the solution

Ackoff [1962] continues by stating that these phases occur concurrently and that they are completed at the same time. Put in another way, this means that the above procedure is iterative in practice, i.e. after testing the model it is necessary to re-formulate the problem and the model. The above procedure has been applied in this thesis (see also Figure 2 illustrating the steps of this research).

Systems analysis provides methods and techniques for describing, analysing, and planning complex systems [Gustafsson et al. 1982]. An example of a system is a manufacturing process such as designing and constructing a building. According to Gustafsson et al. [1982] a systems analysis project is often carried out in stages as follows

- Problem perception
- Problem formulation
- Systems modelling
- Validation
- Problem solving
- Evaluation
- Presentation
- Implementation

Gustafsson et al. [1982] also state that the above stages are iterative by nature, as in scientific work.

In this thesis, the problem perception and formulation, i.e. the starting point, including the recognition of the problem area, was performed in several steps, in parallel with developing the reference models and studying the existing process modelling methods. Two basic methods that are often used in research projects, and also in this thesis, for problem formulation and for gathering background information are:

- Literature studies,
- Interviews with practitioners.

The literature studies were conducted by studying various process modelling methods in general, and process modelling methods used for construction process modelling in particular. Interviews with construction industry practitioners have played an important role during all the phases of this research, especially the interviews with companies participating in the MoPo project [Karstila et al. 2000]. The interviewing techniques included questionnaires, especially in the earlier stages of this research, and also more unstructured interviews, carried out during the workshop meetings with industry practitioners. The practitioners included architects, structural engineers, building services engineers, contractors, manufacturers, building developers, software developers, and authorities. In addition, many researchers from VTT, and other research institutes and associations were interviewed. In other words, practitioners representing a wide spectrum of the parties involved in the building industry were interviewed.

Process modelling is one of the methods used in this thesis for various purposes. First of all, a new process modelling method has been developed, but also one specific method, IDEF0 [IDEF0 1993], has been used to understand the problem area and to understand the procedures of scientific work (see Figure 2).

For the latter stages, i.e. construction of a model, the problem solving and the implementation, the following approaches were used:

- Conceptual modelling
- Object-oriented approach
- Information systems development
- Prototyping
- Evaluation.

Note that the terms tool, technique, and method are often used interchangeably in the research literature. Ackoff [1962] defines the terms as follows. A tool is a physical or conceptual instrument that is used in the analysis or construction of models, such as a computer program or a mathematical symbol. A technique is a way of accomplishing a scientific objective, i.e. a technique is a way of using tools. A method is a way in which the techniques to be used in a particular research project are selected. In the context of this thesis, the term 'method' is preferred, for instance to describe the different process modelling alternatives such as IDEF0, since the methods are often used in conjunction with a user's manual or other guidebook describing the usage area and background.

2.1.1 Conceptual modelling and choice of modelling language

Construction of an information system needs a clear understanding of that part of reality in which the system works [Boman et al. 1991]. This reality consists of things and phenomena, called objects or entities. Examples of the everyday objects and things are buildings and kayaks. The reality is also called a universe of discourse, abbreviated as UoD. An object system is a part of the reality. The term 'conceptual modelling' is an activity that constructs a model of an object system. The term 'model' can be defined as a simplified "picture" or an abstraction of the aspects of the problem area [Gustafsson et al. 1982]. In conceptual modelling, a model is called a conceptual model.

In conceptual modelling, the main principle is that all relevant aspects of the object system should be described. The term 'relevant' means that there should not be any extra information, other than the conceptual model. In other words, in practice there is no need to include aspects that are not of interest. The requirements for a model are that it must be detailed and unambiguous enough that subsequent steps can be taken in a systems development process [Boman et al. 1991].

Conceptual modelling needs a language to describe the objects. The language can be either a natural language such as Finnish or an artificial language such as EXPRESS [ISO 1994b]. The relationships between a term (symbol), a concept, and a reality are illustrated in the so-called Ogden's triangle, Figure 3 [Ogden and Richards 1972].

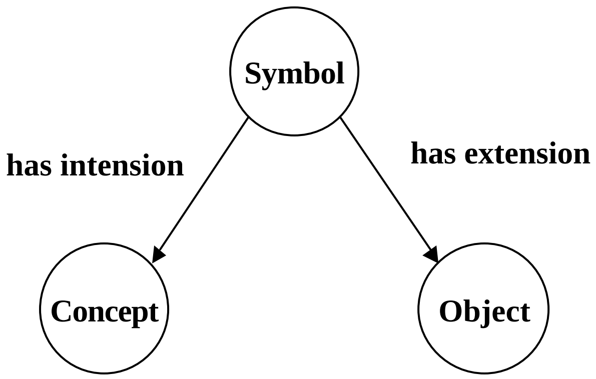


Figure 3. An illustration of Ogden's triangle.

The extension means the object or the set of objects to which a word or an expression refers. An example of the extension of the term 'task' is the set of all tasks in the world. The intension, on the other hand, means the sense of a word or an expression. Thus, the intension of the word 'task' could be an activity that has a starting and a finishing time and is performed on a construction site. Intention is related to the definition of a concept.

People tend to structure and organise what they perceive [Boman et al. 1991]. One grouping or organising mechanism is that the reality is divided into separate objects such as summer houses and kayaks. Another grouping mechanism arises from the fact that objects are parts of other objects, or, vice versa, objects are composed of other objects. Many objects possess similarities and thus the third grouping mechanism is called classification. The classes are often organised into hierarchies, i.e. a class is a subclass of another class. For the construction industry, the classification plays an important role, for instance in specifications and structuring of documents, and is a means to facilitate communication [Ekholm 1996]. Abstraction provides a means either to generalise or to specialise concepts and is used when certain properties of an object are not in focus.

The EXPRESS-language and its graphical counterpart EXPRESS-G [ISO 1994b] were used in this research for conceptual modelling. EXPRESS is a formal language and thus the models can be described as formal, according to Gustafsson et al. [1982]. There were several alternatives available with approximately the same functionality, for instance IDEF1X [Appleton 1985], NIAM [Nijssen and Halpin 1989], and UML [OMG 2000], but EXPRESS was chosen because it is the leading language in the product modelling domain. Consequently, researchers in the construction IT domain are usually familiar with it, which helps in communicating the results of this research to colleagues.

Note that there is a proposed process modelling extension available for the EXPRESS language, called EXPRESS-P [ISO 1995], which was not used in this research. The main reason was that the basic EXPRESS language was already sufficient for modelling the concepts. The second reason was that the extensions proposed, i.e. methods, interfaces, systems and subsystems, processes, and external entities, were not as such suitable for the conceptual model (GEPM).

2.1.2 Object-oriented approach

In object-oriented analysis and design, software systems are modelled as collections of objects, where objects are treated as instances of classes within a hierarchy of classes [Booch 1994]. In other words, the characteristics of abstract or real objects are modelled using objects and classes. Several object-oriented programming languages exist nowadays, e.g. Java [Gosling et al. 2000].

An object is some private data and a set of operations (methods) that can access these data [Cox 1986]. From the software, and thus programming, point of view an object is a software bundle of variables and related methods. The idea of the object-oriented approach is illustrated in Figure 4 where an object is represented by a circle that has methods around it, and variables in its centre.

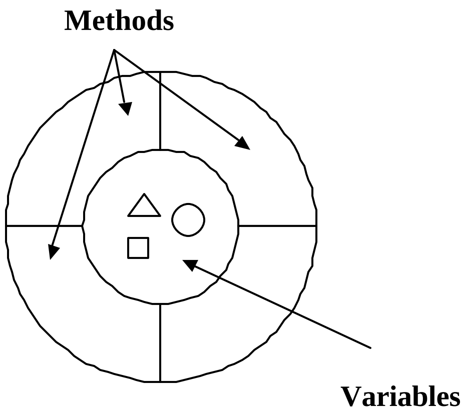


Figure 4. A common visualised representation of an object as a circle with methods and variables.

Encapsulation is the foundation for the object-oriented approach [Cox 1986]. Encapsulation means that the variables of an object cannot be accessed directly, but through the methods of that object. The methods are used to manipulate the variables, e.g. a task object in a project scheduling software application has methods for setting and retrieving the values of variables such as the starting and finishing times. The methods are invoked by sending messages to the object. This process can be compared to applying a function to data in conventional programming languages. From the programmer's point of view, the encapsulation provides a different approach in that importance is focused on what methods can be applied to data rather than how these methods are performed.

Many object-oriented programming languages use declarations that define the methods as public, and thus the variables can be accessed through the methods. Another central characteristic in object-oriented programming is that all objects have unique identifiers that unambiguously distinguish them from other objects. The unique identifier is used with messages.

The objects are defined in the programming languages using classes. In general terms, a class is a type or template that defines the variables and the methods that are common to all objects of a certain kind [Booch 1994]. A class can inherit variables and methods from another class and is then called a subclass. Inheritance is a powerful means for adding modularity to a software application as there is thus no need to define new variables or methods in a subclass, these are being inherited automatically from the superclass. In the object-oriented approach, the inheritance can be multiple, i.e. a class may inherit from more than one superclass. In many applications a superclass may also be an abstract class that has no instances.

An instance represents an object (in fact a class) where its variables are set to values. Figure 5 illustrates an instance of an example task class using the notation of Figure 4. The example task has a name 'Place reinforcement', a unique identifier (ID) with a value of '123', the starting date set to '23.1.2001', and the finishing date set to '25.2.2001'. It also illustrates some methods, e.g. 'Set starting date', that are used to set the values of these variables.

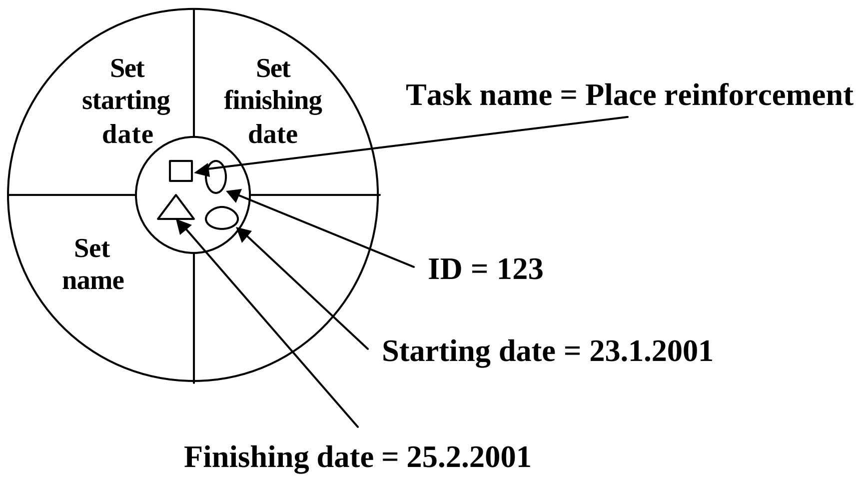


Figure 5. An example of an instance of a task class.

Variables can be separated into two categories: class variables and instance variables. The reason for this is that certain variables might be the same for all instances of the same class and so class variables are used to avoid redundant declarations. The same principle applies to methods where some methods might be the same for all instances of a class.

2.1.3 Information systems development

Organisations use information systems to assemble, store, process, and deliver relevant information. Information systems modelling addresses the problem of capturing fuzzy, ill-defined, informal real world and user requirements, and transforming these into formal, complete, and consistent specifications [Bubenko 1992]. An example of an information system is a project planning and control system [Avison and Fitzgerald 1995].

The conventional approach for information systems development [Avison and Fitzgerald 1995] can be divided into six different stages as follows:

- Feasibility study
- Systems investigation
- Systems analysis
- Systems design
- Implementation
- Review and maintenance.

The feasibility study is used to analyse current information systems, e.g. current process modelling tools. This is described in the third paper [Karhu 2000] by analysing process modelling methods and, at the same time, by analysing the corresponding software tools. During the feasibility study, a number of alternatives are suggested and finally, a recommended solution is proposed with an outline functional solution. Usually, the feasibility of a proposed system must consider legal, organisational, technical, and economical aspects.

The systems investigation is a detailed fact finding phase. The requirements of existing and new systems, possible constraints, exception conditions, and problems of present working methods are clarified. One uses several methods to obtain this information, such as interviews and questionnaires.

Systems analysis concentrates on the current system and clarifies why problems exist, whether there are any alternative methods, etc. This stage is used to understand all aspects of the current system. It should also indicate how the system could be improved.

Systems design analysis usually takes into account the results from the feasibility design phase and leads to the design of a new system. The documentation of this stage results in a description of input, output, processes, structure of the computer, security and back up, and testing and implementation plans. The specification of the new software application, GEPM browser, was developed at this stage.

Implementation of a new system involves practical software development including the programming of the necessary number of code lines, in practice often using more than one programming language. Considering the implementation of GEPM, this has meant programming mainly in LotusScript language [LotusScript 1995] and Lotus formula language [Lotus Notes 1995].

Review and maintenance is the final stage of systems development and can be undertaken when the system is operational. Often, errors are corrected and possibly some changes are made because of the technical advances. The changes may also be due to reorganisation.

2.1.4 Prototyping

Prototyping has been one important part of this thesis. Budde et al. [1992] define the term 'prototyping' as an approach to an evolutionary view of software development, having an impact on the whole development process. Prototyping is often part of an information systems development effort in an organisation. A (software) prototype is an approximation that demonstrates the essential features of the final (software) version [Avison and Fitzgerald 1995]. Prototypes serve many purposes, such as providing a basis for discussions between developers and users [Budde et al. 1992]. In other words, prototyping is important to give users something tangible and get preliminary user requirements. Lundequist [1995] divides the prototype development (or design) into three general stages: conceptual design, systems design, and detail design. Avison and Fitzgerald [1995] suggest the following stages for prototyping:

- Analysis phase to understand the present system and suggest functional requirements
- Prototyping phase to construct a prototype for users
- Evaluation and modification of the prototype
- Design and development of the final application, using the prototype as a specification.

An important aspect of prototype development is to determine the possible users of the prototype. In the context of this thesis, the prototype software application, the GEPM browser, has been developed and used only by the author. Using in this context means that the author has created the models with the browser. On the other hand, the prototype has been shown and used in the interviews with the industry practitioners as a demonstration tool. It has also been used for evaluation of the new modelling method. One can conclude that the GEPM browser can be used as a specification tool for further development. Moreover, some efforts result in more or less evolutionary prototypes, i.e. the prototypes are never intended to be final but can be developed further through iterative processes.

2.1.5 Validity, testing concepts and kinds of evidence

The purpose of the validation of a model is to check that it gives a sufficient description of the system so that the problems formulated can be solved [Gustafsson et al. 1982]. This means that it is essential to be assured that the model is adequate for its purpose, not necessarily that the model is true or false as such. Gustafsson et al. [1982] state that the validation process can be divided into two separate parts: system validation, and technical validation. System validation means validation of hypotheses. Technical validation concerns computer implementation. The evaluation has played an important role in this research and is reported specifically in the fourth paper [Karhu 2001] where the GEPM method with its prototype implementation was tested and evaluated against the chosen modelling criteria.

In testing, the ideas are evaluated against certain chosen criteria. Meredith [1993] has recognised that the primary difficulty in using models to analyse situations is obtaining sufficient simplification while still maintaining reasonable realism. The quality of a model can be considered from a syntactic, semantic, or pragmatic point of view [Lindblad et al. 1994]. From the pragmatic viewpoint, one may argue that the goal of conceptual modelling is not to make a model easy to understand but to ensure that the model is understood [Lindblad et al. 1994]. In the context of this thesis, some testing and evaluation criteria are discussed in the following sections.

Ackoff [1962] argues that comparisons of different methods can be made retrospectively or prospectively. By comparing the solutions, one may deduce which ways have been successful and satisfy the requirements. When applied to this thesis the retrospective comparison means that existing solutions with already known modelling methods have been compared with the new proposed modelling method. Prospective comparisons would mean a prediction of the future behaviour of the modelling methods under certain conditions. Since the lacks and deficiencies of the other modelling methods are already well documented and known, the retrospective comparison prevails in this research.

Amberg [1996] has used a pattern-oriented approach to a methodical evaluation of a business process modelling method and to a comparison with another modelling method. According to this approach, a specific set of patterns (principles) is defined by investigating the notations and the documentation of the methods to be compared. The set of principles was, in Amberg's case, the model and view principle, the as-is and to-be model principle, the model and requirement principle, task and resource principle, and the logical model and physical model principle. Amberg continued by determining an evaluation procedure, which includes steps that have to be taken to carry out the test. An approach similar to Amberg’s usability testing procedure was used in the third paper [Karhu 2000] for studying the characteristics of six existing process modelling methods, by defining the conceptual model of each method using the EXPRESS language and by trying to model one example case (a traditional Finnish summer house basement) using these methods.

Clayton et al. [1998] discuss the subject of how to compare the performance of an innovative CAD-tool with a conventional tool. Usually the validation of results falls into one of the following categories:

- A logical argument
- A worked example
- A demonstration
- A trial.

A logical argument is theoretical evidence and usually lacks practical evidence. To this category belongs also a so-called common-sense model that is based on general knowledge and information available to the modeller. Gustafsson et al. [1982] point out that a common-sense model can be used but the results of such a model should not be considered immediately self-evident. According to Clayton et al. [1998], a logical argument rarely convinces the more pragmatically oriented commercial community of the value of the research. A worked example may be tailored to suit the needs and purposes of the research [Ackoff 1962]. A worked example can also be used in the early stages of research and for reporting on the results. Thus, in the third paper [Karhu 2000] a worked example of the process of building a foundation for a traditional Finnish summer house is presented. A demonstration is given in the fourth paper [Karhu 2001], using real data from an office building project. The demonstration here is a retrospective comparison, using data already collected as input since the project data had already been documented in another report.

The trial usually requires a substantial effort to develop a software application that is robust and bug-free to a sufficient degree. In addition, many persons other than the researchers themselves need to be involved in testing, and thus, a trial was deemed to be beyond the scope of this project. This decision was also influenced by the fact that full trial testing would have required the installation of four software tools (Lotus Notes, MS Project, Visio, and BPwin) onto the computers of the testers. On the other hand, the aforementioned test case material has been derived from a well documented project report, and thus has served as a non-biased description of a building project as it really occurred.

The Charrette testing method, as described by Clayton et al. [1997], was also based on techniques of software usability testing and uses a scheme for carrying out the test itself. The main issues are the criteria for comparing and evaluating the procedures.

The development of GEPM included an important aspect concerning the learning of a new method. If the user is familiar with the constituent methods, i.e. scheduling, IDEF0, and simple flow, it can be argued that the user will be able to learn GEPM as well.

2.2 Choice of software tools for modelling

It is appropriate to mention the particular software tools that have been used in this research. The software tools for process modelling, using the IDEF0 method, have been Design/IDEF and, later, BPwin 2.0 [Logic Works 1997]. Process modelling in the form of tasks was carried out using MS Project, which is a general tool for scheduling [MS Project 1998]. Visio, a general purpose graphical tool that also has a programmable user interface [Visio 2000], was used to draw diagrams for the simple flow method. The conceptual models of process entities in EXPRESS were created with software tools for conceptual modelling in the latter phases of this research. Since then, because of this research, a new type of a software prototype tool, called a GEPM browser, has been developed and used for the modelling of a building construction process and for demonstration purposes. A more detailed description of the GEPM browser is found in the discussion part of this thesis and in papers [Karhu 2000, Karhu 2001].

2.3 Research or development

From a methodological viewpoint, one can ask whether the work presented in this thesis is research or development. Eriksson [1994] distinguishes between research and development as follows. Research has as an objective to produce theoretical knowledge, whereas the objective of development is to achieve new products, new processes, new systems, or essential enhancements to these. In the context of this thesis, theoretical knowledge means work that is accomplished by analysing, describing, clarifying, and possibly evaluating a description of a part of reality (e.g. a construction process). Other important aspects of research and development concern relevance and utility. Ackoff [1962] argues that pure research, in contrast to applied research, does not consider uses of its results outside the domain of science. Eriksson [1994] says that relevance concerns the usability of research results as an argument. Utility, on the other hand, may be related to research as well as to development. However, to evaluate the utility one needs to define the evaluation criteria.

A large part of the project work carried out by VTT (the employer of the author) belongs to the category of development. The work presented in this thesis is partly research and partly development. The development of the conceptual model of GEPM may be considered to be research. On the other hand, the prototype application can be categorised as development even if the distinction is not clear. Consequently, this leads to a question of whether this thesis could be a monograph instead of being composed of the refereed journal papers and a summary. Here, the pragmatic answer is that seldom does a single project at VTT (at least ones in which the author has been involved) lead to a doctoral thesis because of the project size, i.e. funding. Instead, many projects are carried out to elaborate the results. Hence, the four papers in this thesis reflect the process of developing ideas over the years. It can also be argued that the results presented in the papers are considered as intermediate results and thus differ from the results that are presented in the summary part of this thesis.

3 THE STATE OF THE ART OF PROCESS MODELLING USAGE AND METHODS

In order to understand the underlying reality of construction processes, it is important to examine process modelling methods, information models of construction and processes, and models that have been developed using these methods. It is also important to define what the term 'process' means here. A definition by Melan [1992] states that a process is a bounded group of interrelated work activities providing output of greater value than the inputs by one or more transformations. Halpin and Riggs [1992] define a construction process as a unique collection of work tasks, related to each other through a technologic structure and sequence. One more definition by Kartam et al. [1997] states that a process is a set of consecutive steps or activities with an end product or service being produced. The latter definitions prevail here since this thesis excludes the discussion of the value of inputs or outputs. The above definition logically leads to the definition of the term 'process modelling', which can be understood as a systematic means of modelling the activities and all relevant relationships of a process.

It is also of importance to recognise some fundamental characteristics of a construction process. These are as follows:

- Uniqueness of end products, i.e. buildings and bridges
- Iterative or repetitive activities
- Complex organisational schemes
- Order or dependency of activities
- Concurrency of activities
- Use of feedback
- Conditional branching.

The uniqueness of the end-products or construction projects, iterative or repetitive activities, and complex organisational schemes are often observed [Halpin and Riggs 1992]. The complexity often leads to specialisation of the various disciplines [Ahuja 1976]. The order of activities or sequencing is typical in many construction projects [Fondahl 1980] and is used in scheduling. The reasons for dependencies of activities vary from technical restrictions to a work gang's availability. The concurrency of activities is also well known, e.g. during the design phases of a building several participants work simultaneously. This is seen, for instance in the models developed of the current Finnish design and construction practice by Karhu and Lahdenperä [1999]. Moreover, Kagioglou et al. [1998a] suggest that concurrency should be promoted. Conditional branching is often involved in decision making and in alternative ways of carrying out tasks.

3.1 Process modelling methods

Many process modelling methods have been used or proposed for construction process modelling. These methods include:

- Scheduling

- The simple flow method and its variations
- Data flow diagramming
- IDEF0
- IDEF0v
- IDEF3
- Petri Nets

One method used for construction process simulation is called CYCLONE [Halpin and Riggs 1992]. This method was deliberately not studied in this thesis, because simulation is beyond its scope.

In all of the above process modelling methods, the graphical presentation of the resulting models is an integral part. The methods use different graphical presentations. Hence, in the following, the emphasis is on the core entities and these are presented using the EXPRESS-notation.

The term 'scheduling' is used here to denote the familiar general project planning or networking techniques (critical path, resource levelling, precedence method, PERT, etc.), which, along with software tools, have been under development for a longer period, as was noted in the introduction. Figure 6 illustrates one basic and simplified notation of scheduling, where, graphically, boxes denote activities or tasks, and arrows denote relationships between them. Some variations have used circles instead of boxes, or arrows have represented activities. The resulting schedules may be presented in variations such as Gantt charts (an example is seen Figure 1) or PERT charts.

Figure 6. An arrow between two tasks indicates a precedence relationship.

A conceptual interpretation of scheduling methods in general is illustrated in Figure 7 using the EXPRESS-G notation [ISO 1994b]. Activities are usually called tasks in scheduling. Tasks have attributes such as a starting and finishing time, duration, and resources. The precedence or other relationship between tasks can be expressed using a separate relationship entity (in Figure 6 the arrow represents a relationship). A separate entity also provides the possibility of including more information about the relationship than a simple precedence attribute, for instance. One more concept normally used in scheduling is the decomposition concept, which means that tasks can be decomposed into sub-tasks.

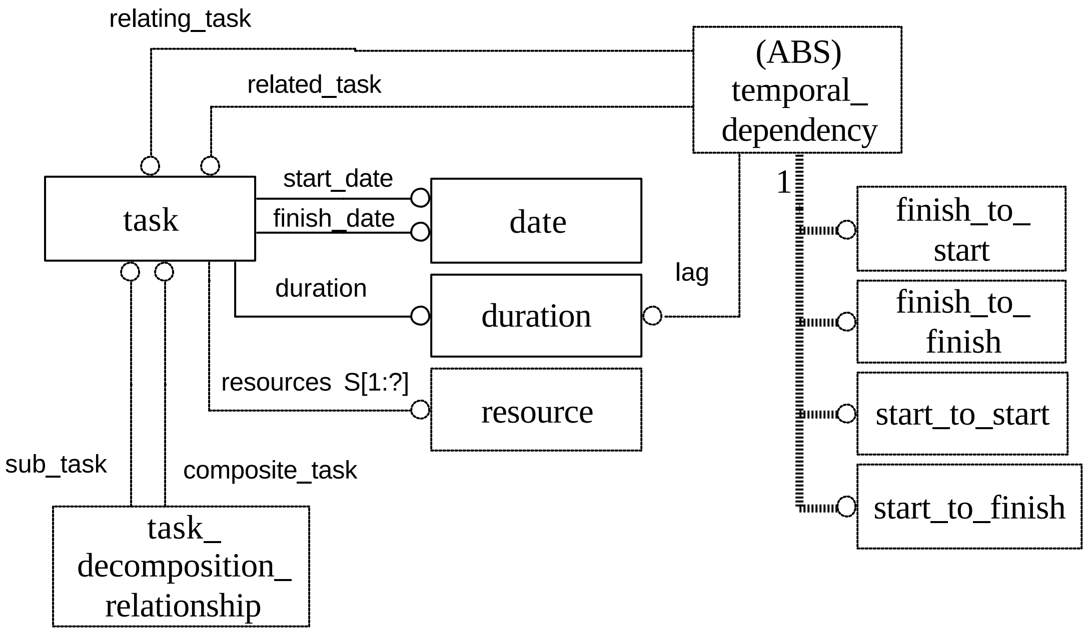


Figure 7. The conceptual model of the scheduling method.

One of the problems with scheduling methods is that they do not make a distinction between the different types of resources, nor do they distinguish controls for tasks. In other words, it is difficult to understand what a task does with all its resources. Another problem is that iterative tasks are difficult to model. On the other hand, the identification and sequence of tasks is modelled well.

The second method in the list, called here a 'simple flow method', has been used in many variations to describe construction projects. The method defines concepts that are activity, participant (mechanism), and a flow between the activities, and uses a simple 'box-and-arrow' notation. Figure 8 illustrates an example where two construction workers are responsible for two activities, respectively, displayed in rows, and there is a flow between these activities.

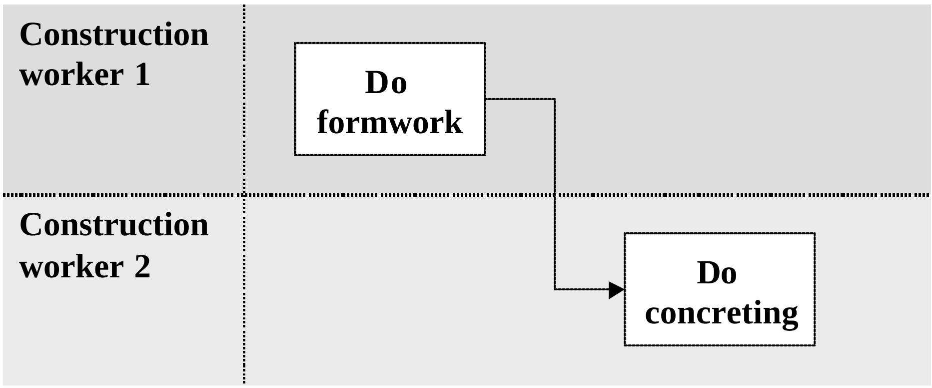


Figure 8. An example of a simple flow diagram.

One interpretation of the core entities of this method is illustrated in Figure 9 using the EXPRESS-G notation. For example, Hoffner [1997] uses the method to describe a general contractor's process, and interprets the flow as a precedence relationship. The method can be classified as a functional method, and a diagram may be called a crossfunctional flowchart. The term 'functional method' means that the method is used to model activities.

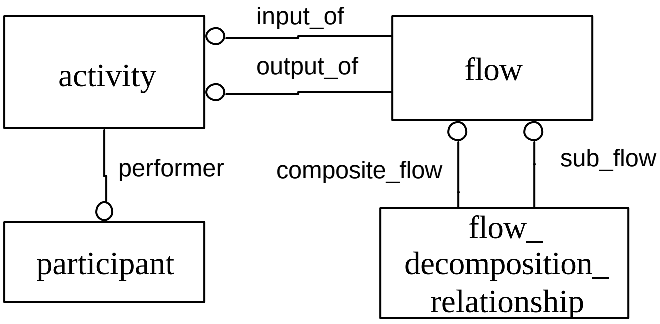


Figure 9. An interpretation of the core entities of the simple flow method.

Two of the main advantages of the method are that the diagrams can be produced using common general-purpose graphical drawing software tools and that they are generally easy to understand. One drawback is that there is no description of the kind of flow there is, i.e. if there is a predecessor relationship or another kind of flow such as input or output information. Another drawback is that the activities cannot be decomposed, though decomposition can be obtained at a software application level by exposing more activities in one row.

Data flow diagramming, as described by DeMarco [1979], has been used in software development in particular, but has also been used in modelling the construction processes, for instance to study communications between participants [Abou-Zeid and Russell 1993], or to model the building design process [Austin et al. 1996, Baldwin et al. 1999]. This method has four basic constructs: process (function), data or information flow, data store, and an external entity used as a source or sink of data flow. Austin et al. [1996] apply the constructs to a design process model and interpret processes as individual design tasks, flows as design information flows, data stores as drawings etc., and external entities as clients, local authorities, etc. An interpretation of the core entities of data flow diagramming is depicted in Figure 10 in an EXPRESS-G diagram. A process has one or more inputs and one or more outputs. The data or information flows can be external, i.e. they are produced or used outside the model. Additionally, the method has the capability of presenting the models in a hierarchical form through decomposition.

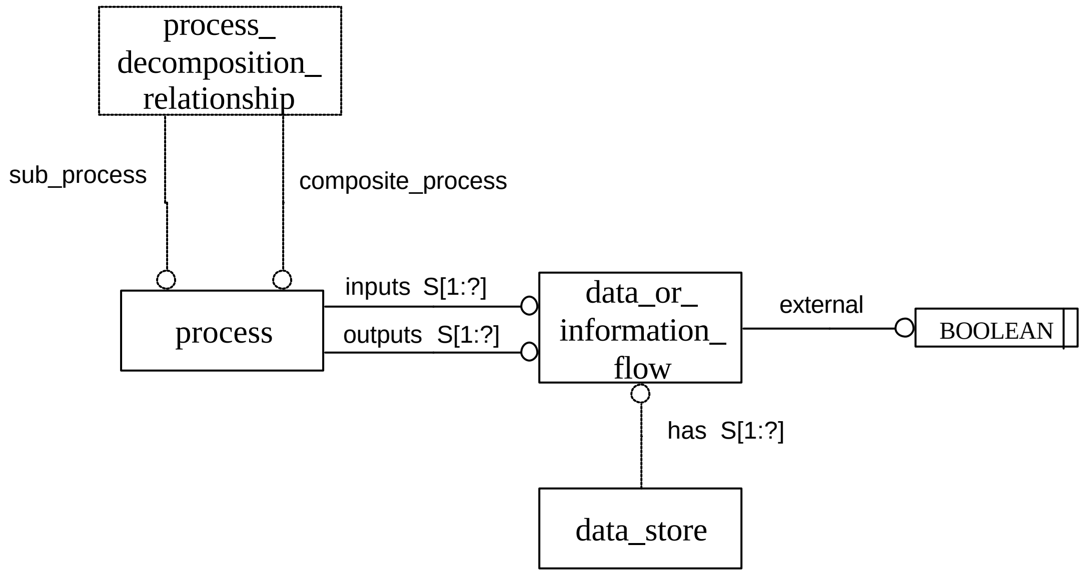


Figure 10. The core concepts of data flow diagramming.

Data flow diagrams (DFD) can be used effectively for modelling the functions and the information flows between them. They also support the iterative nature of many construction processes such as design work [Austin et al. 1996]. This is because the method is independent of time. On the other hand, the performer of a process is not modelled.

The IDEF0 method is a formal standardised modelling method [IDEF0 1993] for functional modelling. The main concepts are the activity and the flow (called ICOM, derived from the words input, control, output, and mechanism), in much the same way as in the simple flow method. On the other hand, the flow can be used as input, output, control, or mechanism, depending on where it (graphically) enters the box from (left part of Figure 11). Input enters the box from the left, control from the top, and mechanism from the bottom. Output leaves the box on the right. For instance, the same flow can be simultaneously an input of one activity and an output of another. In other words, the role of the flow determines whether it becomes an input, output, control, or mechanism of an activity.

The presentation of the IDEF0 diagrams is hierarchical, with diagrams at lower levels representing more detailed activities than those at the higher levels (right part of Figure 11). This hierarchical presentation implies a part-of decomposition of activities and says nothing about whether the activities represent specific activities or non-specific activities. A specific activity means here, for instance, an activity for installing a window with an identification number 123456. Put in a more formal way, an interpretation of the conceptual model of IDEF0 is illustrated is Figure 12 as an EXPRESS-G diagram.

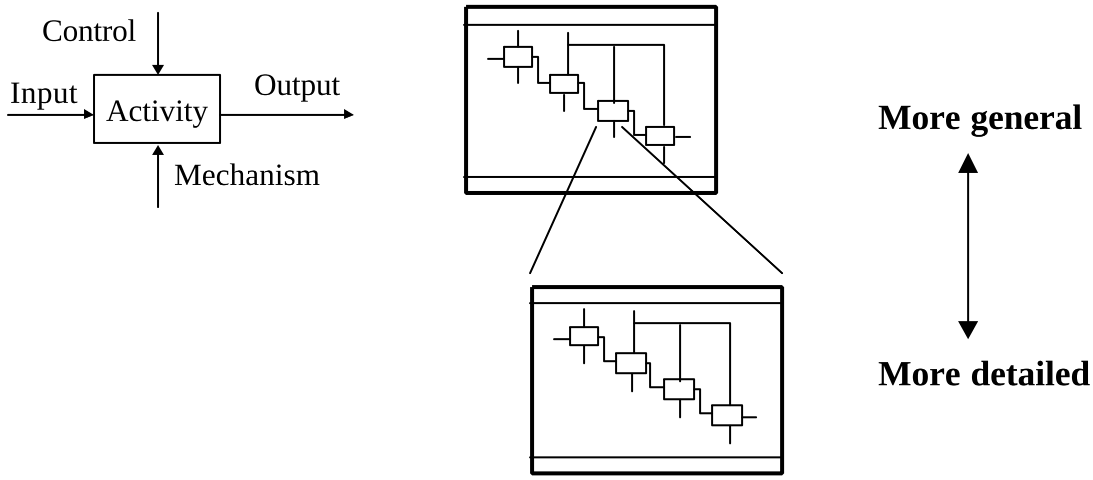


Figure 11. The basic concepts of the IDEF0 method, an activity box with its flows (on the left) and the hierarchy of the diagram (on the right).

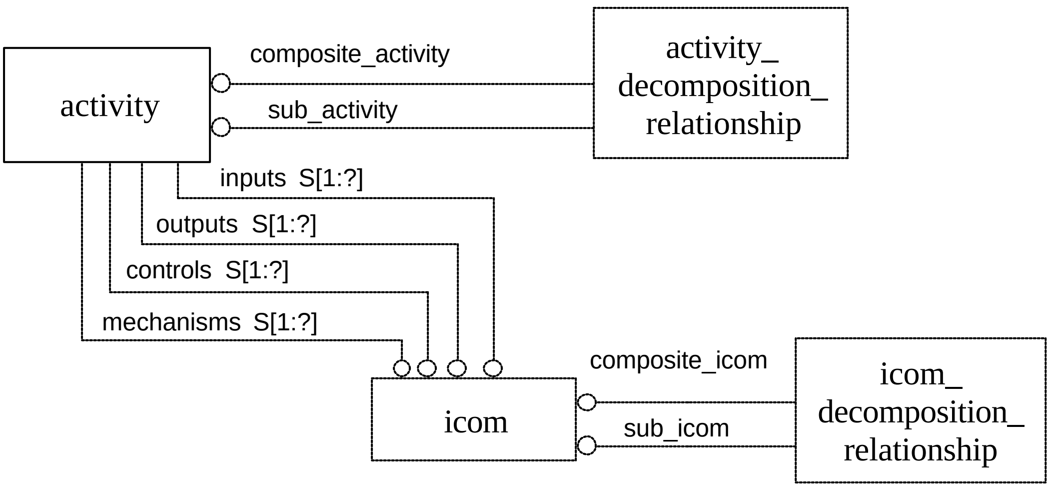


Figure 12. The conceptual model of core entities of IDEF0.

The major drawbacks of IDEF0 are that one cannot derive a logical sequence of activities, it does not apply the time concept, and that it cannot be used to model an 'or' branching explicitly. On the other hand, a branching of an arrow into two or more arrows does not necessarily mean a logical branching that is typical in workflow methods. In IDEF0, branching and its counterpart, joining, can mean a decomposition or merely that the same information flow is present in more than one direction, i.e. an arrow segment that is a part of an arrow. A practical problem is that IDEF0 models tend to be interpreted as sequences of activities instead of being general descriptions of activities that in the reality may occur simultaneously. Moreover, the developed models tend to become complex and difficult to understand. On the other hand, IDEF0 is very flexible in a sense that it can be used to model a large number of different processes at a detailed level or at a more general level. IDEF0 can be used to model concurrent activities but it needs human interpretation since no explicit concept for concurrency exists, i.e. models are independent of time. A concurrency is indicated in diagrams, for instance where an output arrow is split into two or more parts that enter other activities as inputs or controls.

The IDEF0v method was developed by Austin et al. [1998] for improving the detailed design process of buildings. The method has been developed from IDEF0 by dividing inputs into three subcategories; intra-disciplinary input, cross-disciplinary input and external input. It retains most of the characteristics of the IDEF0 method, for instance diagram hierarchy, but excludes the control and mechanism concepts. The exclusion of control and mechanism is a disadvantage of this method. On the other hand, it has been successfully used in a particular process model [Austin et al. 1998].

The IDEF3 method is a scenario-driven process-flow modelling method, the purpose of which is to capture descriptions of the precedence and causality relations between situations and events [Mayer et al. 1995]. A scenario can be a recurring situation, a set of situations that describe a typical class of problems addressed by an organisation or system, or the setting within which a process occurs. The main concept of IDEF3 is the so-called unit of behaviour (UOB). A unit of behaviour covers a wide range of objects, for instance it can be an activity or a process. The unit of behaviour is graphically represented by a box. The IDEF3 method can be used in two modes: to model the process flow or to model the object-state transition. The method has similarities with other methods, especially with workflow techniques. From the viewpoint of this thesis, an important concept is a junction. A junction can be used to model clearly a branching of activities or a joining of a number of results of various activities. In comparison with IDEF0, an 'or' junction is an effective means to model alternative sub-processes. A disadvantage of IDEF3 is that it does not distinguish between different types of input, output, control, or mechanism, as in IDEF0.

Petri Nets can be used, among other methods, for modelling business and software processes, and for descriptions of existing systems prior to re-engineering. It has been proposed as a standard [Anon. 1997]. Some authors have used Petri Nets for construction process modelling. The constructs for construction processes, i.e. applicable entities for construction, may be identified as sequential execution, conflict, concurrency, synchronisation, merging, and confusion [Viswanadham and Narahari 1992]. Volkhard [1989] uses Petri Nets to model the activities on a construction site. Wakefield and Damrianant [1999] use it for construction process re-engineering. In its basic form, a Petri Net (PN) defines transitions that are interpreted as activities, and tokens that are interpreted as resources. The idea is that resources (tokens) are used by activities (transitions), and then moved from one place to another. The concept of time can also be associated with a transition. Moreover, pre- and post-conditions can be added to a Petri Net and additional attributes (colours) can be added to tokens. Petri Nets can be used effectively for the simulation of construction processes. They are also used in a modified version by Augenbroe and Amor [1997] to describe project windows that represent parts of the total construction process. On the other hand, the decomposition or hierarchy cannot be modelled using the basic Petri Nets approach. Petri Nets are aimed more at simulation, and thus are beyond the of scope of this thesis.

3.2 Process models

Many construction process models have been developed in recent years. Perhaps the most influential model, based on later research, was Sanvido et al.'s [1990] model for supporting the provision of a facility. The aim was to support computer-integrated construction and to define the critical success factors for construction projects. The modelling method used was the IDEF0 method. The model itself was divided into five main activities: manage facility, plan facility, design facility, construct facility, and operate facility. Another model, defined using the IDEF0 method, was developed by Zhong et al. [1994]. It covers the overall building process by dividing a building construction project into four main phases: the initial phase, the design phase, the tendering phase, and the construction phase. Bakkeren [1995] and Nederveen [1995] use IDEF0 to model specific view type models. Merendonk and Dissel [1989] model a construction process. The current Finnish design and construction practice has been described by Karhu and Lahdenperä [1999] as six different process models corresponding to building procurement, architectural design, structural design, building services design, geotechnical design and constructing. The models were created using IDEF0.

The process protocol work is an attempt to improve project management in the UK construction industry [Kagioglou et al. 1998a, Kagioglou et al. 1998b]. One of the incentives for the process protocol has been the manufacturing industry. The process protocol work defines a process map, a kind of reference model, for a construction project. The process itself is divided into ten phases describing the participants, their major activities, and deliverables over the period of a project. From the point of view of decision making during the process, there are so-called soft gates and hard gates. Soft gates imply that decisions may be approved conditionally afterwards. On the other hand, hard gates indicate final and firm decision points whether or not to proceed to the next phase of the process. This is because late design changes often cause substantial additional costs in construction projects. One more aspect of the process protocol is that it suggests the use of legacy archives that can be used as feedback for subsequent and future projects.

3.3 Conceptual models of construction information

In addition to the modelling methods and process models, many conceptual models of construction information have been proposed and discussed in the research literature. A difference between these information models in comparison to modelling methods is that they merely define information entities in the construction process.

The unified approach model by Björk [1992] suggests the use of a single conceptual modelling language for modelling all kinds of construction process information, including activities, building parts, documents, etc. Concerning processes, the unified approach model suggests three basic categories: activities, results, and resources. The results are further divided into sub-categories such as document, physical object, and service. The common upper class for all these entities is called a construction process entity, which among others has an attribute that defines a unique identifier. The conceptual model itself relates the above entities as follows: an activity produces a result and is performed by an agent that in turn can be an organisation, person, machine or facility. The concept of contract is also introduced in the model. A contract concerns results.

The general construction project model (GenCOM) was developed in order to improve the integration of project management tools by using object-oriented models of construction projects [Froese 1992]. The high-level object classes, or entities, are activity, resource, component, method, action, and project participant. Activity performs actions and uses methods. An activity is the responsibility of a project participant. The relationship to the physical components of a facility is achieved through a relationship that an activity operates on a physical component. Moreover, the activities are parts of a construction plan. The GenCOM model defines an entity 'resource use'. The relationship with an activity is such that an activity uses the entity 'resource use'. An entity 'resource' is used by 'resources use'.

Luiten et al. [1993] define a conceptual project model called IRMA (information reference model for AEC). It was developed by combining features of the unified approach model, GenCOM, and models developed at TNO in Holland. The aim of IRMA is to clarify terminology, since similar concepts are often used in a slightly different manner. Figure 13 illustrates the relationships between the entities. The central concept is project object that has four subclasses: product, activity, resource, and contract.

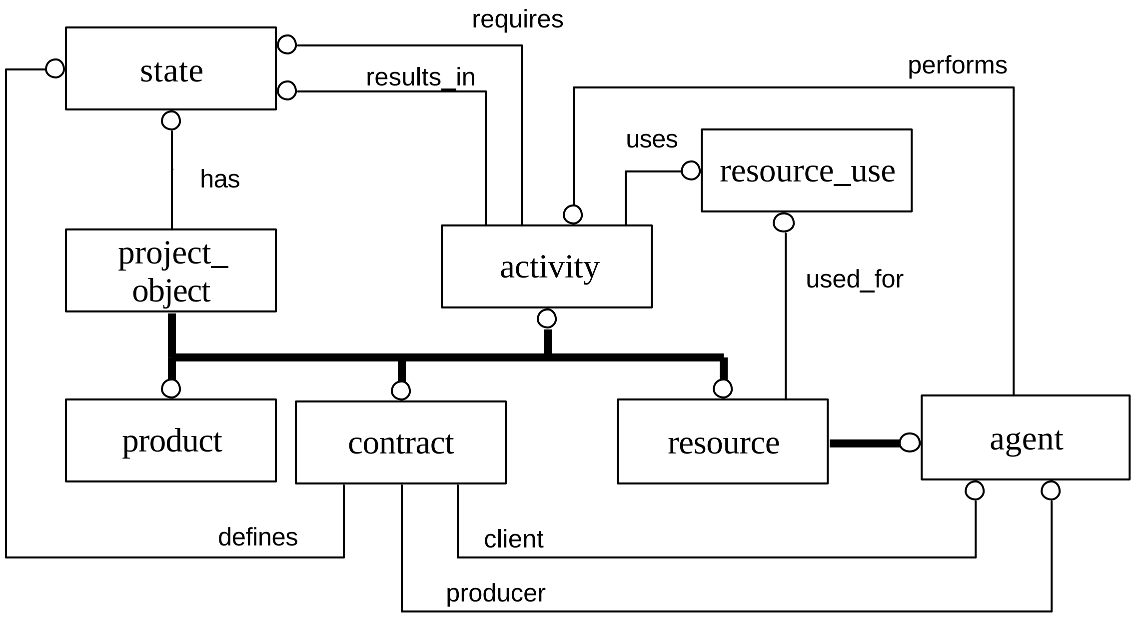


Figure 13. The main relationships of IRMA.

The 'state' concept plays an important role in IRMA. The state concept can be used to describe that, for instance a concrete mould on the building site may be in an 'as-required' or 'as-built' state. In IRMA, an activity results in a state of a project object. According to the principles described by Luiten et al. [1993], an activity requires that an object (or many objects) be in an initial state. The resources are used by activities, or in more precise terms, the concept 'resource use' defines the usage of resources. Luiten et al. [1993] further demonstrate the concepts of IRMA with a simple IDEF0 model where arrows were named in such a way that they can be interpreted as states of project objects.

Though the model did not lead to practical solutions or applications, it has served as an incentive for other modelling attempts.

The ICON model [Aouad et al. 1994] defines a planning activity that is a part of a construction plan and corresponds to a building element. An activity has a dependency that is divided into the subclasses start-to-finish, finish-to-start, start-to-start, and finish-to-finish dependencies. Moreover, a task dependency has subclasses such as physical dependency, trade dependency, resource dependency, and safety dependency. An activity has a 'resource use' attribute that uses a resource. A resource is further divided into subclasses, for instance into material and labour.

Another conceptual model is STARGEN [Hannus and Pietiläinen 1995, Pietiläinen and Heinonen 1995]. It defines the concepts of project, contract, and project object. The project object is divided into two subclasses: information object and resource. The project objects have a state, which is either an input or output of an activity. The activities are controlled by information objects. An activity may be decomposed. Moreover, activities may have time and location attributes as well as dependencies though these are not clarified. Activities have subclasses that are 'inspecting', 'waiting', 'moving', and 'processing', and these describe the flow. A STARGEN model can be viewed from different points of view, which are a contract view, a resource view, an activity view, and a flow view.

Some information models have dealt with product and process entities and have defined relationships between these. In the context of this thesis, the process-related entities are of importance. The building project model (BPM) by Luiten [1994] applies the concepts of a functional unit and a technical solution, proposed by Gielingh [1988] to define activities and states of products. It distinguishes three stages, defines the concepts such as required activity, proposed activity, and realised activity, and applies the three stages to cover the products and the states of products.

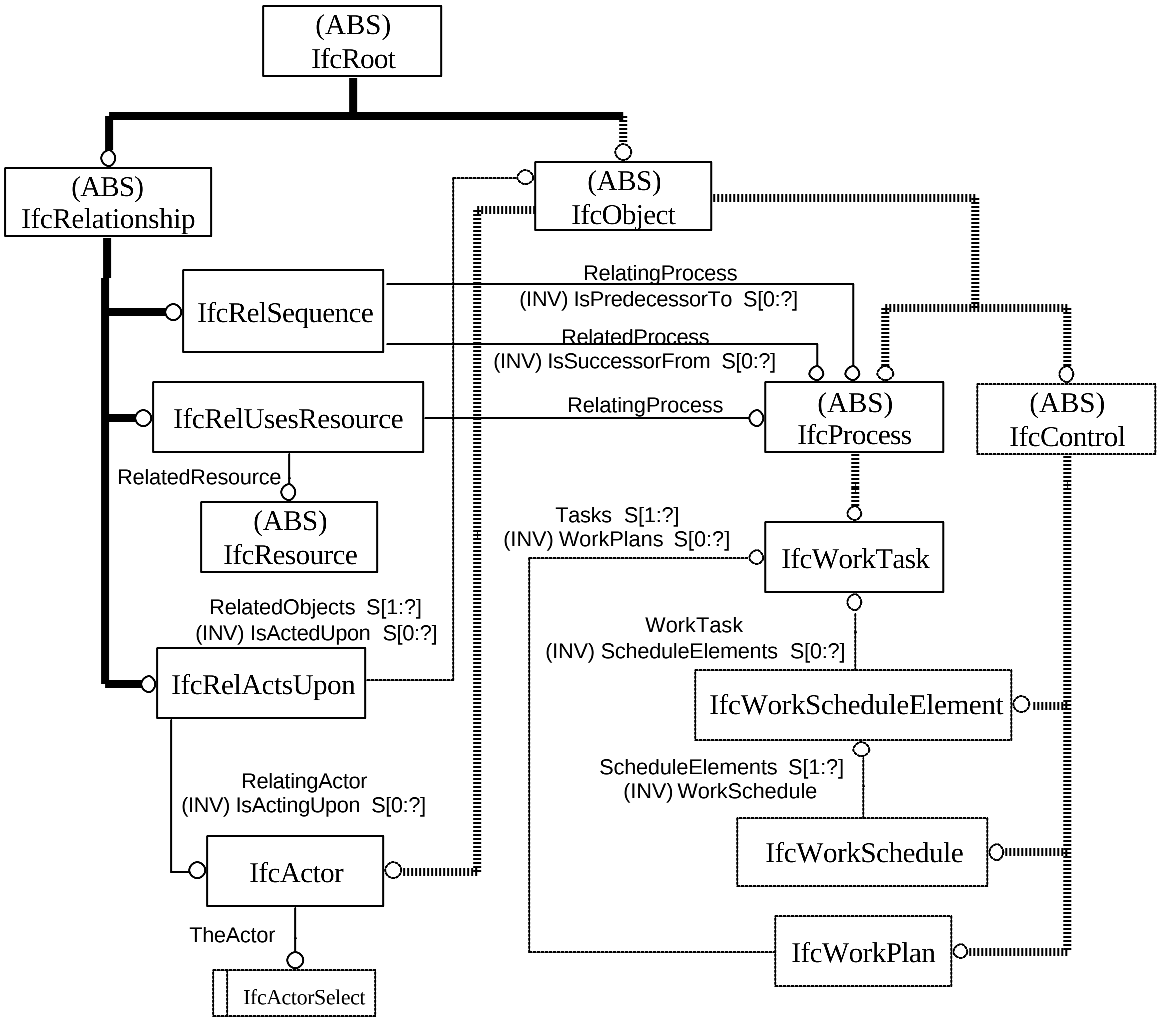


Figure 14. A part of the process extension of the conceptual IFC model.

The International Alliance for Interoperability has defined a process extension (draft) for the IFC model [IAIIFC 2000]. A simplified EXPRESS-G diagram is presented in Figure 14. The central concept here is a process defined as an abstract entity 'IfcProcess'. The task 'IfcWorkTask' is a sub-type of 'IfcProcess'. The 'IfcWorkTask' has attributes for an ID and name (though not shown in the figure). The process may have predecessors and successors, which are defined using a relation object 'IfcRelSequence'. The relation object is in turn a sub-type of an abstract entity 'IfcRelationship'. The 'IfcRelSequence' has an attribute to indicate the time lag. The relation object may have types such as finish-to-start, start-to-finish, etc. The 'IfcResource' entity is further divided into sub-types such as labour resource, product resource, etc.

3.4 Requirements for a modelling method

What should be required from a construction process modelling method? The fundamental characteristics of construction processes and projects have been discussed at the beginning of this chapter, and thus it is logical that a modelling method should cover those aspects. Wakefield and Damrianant [1999] identify some disadvantages of present modelling tools as a lack of user friendliness, a lack of flexibility, and a complexity of models. Chung [1989] recognises a few basic requirements for a model such as it should accurately represent the construction industry and it should be easy to understand and use. Human communication is essential [Ahuja 1976]. Kochikar and Narendran [1994] use the concept of modelling power to evaluate a model of flexible manufacturing systems. The modelling power is provided by the universe of system features that a method can represent, in other words the modelling power is measured by the number of features. Specifically, for a general model of construction, Koskela [1995] adds to the above the criteria for a construction focus, breadth and depth. The breadth means a comprehensive model that covers widely differing processes in any construction process. By depth is meant that the model covers all the important basic constructs. In addition to modelling power, a modelling method should be verifiable and should provide the opportunity for system evolution. A verifiable method is defined as one that makes it possible to see if a resulting model is correct or not, and often a graphical representation is a particular aid. System evolution means that a system evolves in time.

From the conceptual models, one can derive some essential entities that are used or are present in a construction process. In addition to the entities themselves, the relationships between them are of importance. Froese [1996] summarises the core process relationships between activities as input, output, control, precedence, actor, method, cost, and hierarchy.

A traditional approach used to describe processes is the so-called input-process-output (IPO) paradigm, Figure 15, which emphasises the conversion nature of the process. The process may be broken down into sub-processes. This corresponds, in fact, to the definition of a process as a set of activities or as a set of steps, and leads logically to a need for the concept of hierarchy. Moreover, Chung [1989] suggests that a construction process should be broken into smaller units and a model should be modular. The hierarchy and abstraction are also suggested by Kochikar and Narendran [1994] as necessary aspects. In addition to the conversion view of the construction process, construction can be viewed as composed of flow processes [Koskela 1992].

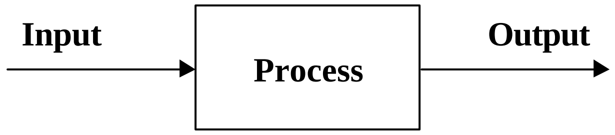


Figure 15. Input-process-output paradigm.

A central question here is whether an input is consumed or not by an activity. The IDEF0 method [IDEF0 1993] distinguishes the control from input as follows: input is transformed by an activity into output; control is a condition required by an activity to produce correct output. According to the standard [IDEF0 1993], control may be transformed by an activity, creating output. On the other hand, a practical convention for how to use input and control is that control is not transformed by the activity. This convention is used, for instance in Sanvido et al.'s [1990] model.

The IDEF0 definition [IDEF0 1993] discusses the different inceptions of activities under different conditions. Thus, in general terms, the activities need something, i.e. prerequisites need be present in one form or another. The concept of precedence may be interpreted sometimes as a prerequisite for an activity. The performer, actor, or mechanism is also necessary for an activity, and denotes who is doing the work. In scheduling, precedence occurs, for instance, when the same work gang has the responsibility for two or more tasks and thus can only do one task at a time.

Since the primary purpose of the prototype application in this thesis is to demonstrate the new process modelling method, many refinements of the user interface have been left out of the prototype. Moreover, the end-user has been the author. On the other hand, a few basic requirements for the prototype application are as follows:

- Handling of many models in one database
- Error checking routines
- Consistency checking and repair routines for models
- Modularity

For process modelling tools in particular, Chung [1989] suggests that the chosen modelling tool should be available to all participants of a process modelling effort, and that the tools should be compatible with other tools, if possible, and also with tools of other industry areas.

4 SUMMARY OF PUBLISHED PAPERS

4.1 Product Model Based Design of Precast Facades

The paper presents the development of a product data model of precast concrete facades and of a process model for the corresponding design process [Karhu 1997]. The main objectives were to:

- Define a basic activity model for the design process of precast concrete facades, emphasising the architectural design phase
- Analyse the problems occurring in a typical design process using the above model as a framework
- Define a product data model of a precast concrete facade
- Define prototype check-lists for the data input and output requirements of different design activities
- Develop prototype software utilising the product model and test it in a real project
- Decide if the product model supported architectural design process enhances the whole building design process.

The results relating to the objectives above are explained briefly in the following.

The traditional building process was used as a framework for studying the data exchange problems during the process phases. The design process was divided into six sub-phases:

- Briefing
- Programming
- Conceptual design
- Detailed design
- Design during construction
- Design during maintenance and usage.

The problems encountered are that the data describing the facades, which are produced during the different phases of the architectural design process, do not meet the information content and requirements of the other parties, and that there is insufficient feedback of requirements, experience data etc. from the latter phases to the architectural design. One solution to this problem is to develop a product data model of the precast concrete facade. The facade consists of separate elements, which in turn consist of structural layers.

A part of the activity model for designing a facade is shown in Figure 16 [Karhu 1997]. The information flows between the activities are data included in the product data model of the facade. The product data model was combined with the activity process model by defining checklists. The checklists include information about the process phases and the corresponding product data used as input and output for the phases. One reason for using the checklists was that industry practitioners are more familiar with such lists than, for instance, with IDEF0 models. Moreover, the checklists are easy to use and can be used as additional material for many purposes, such as for defining subcontract boundaries.

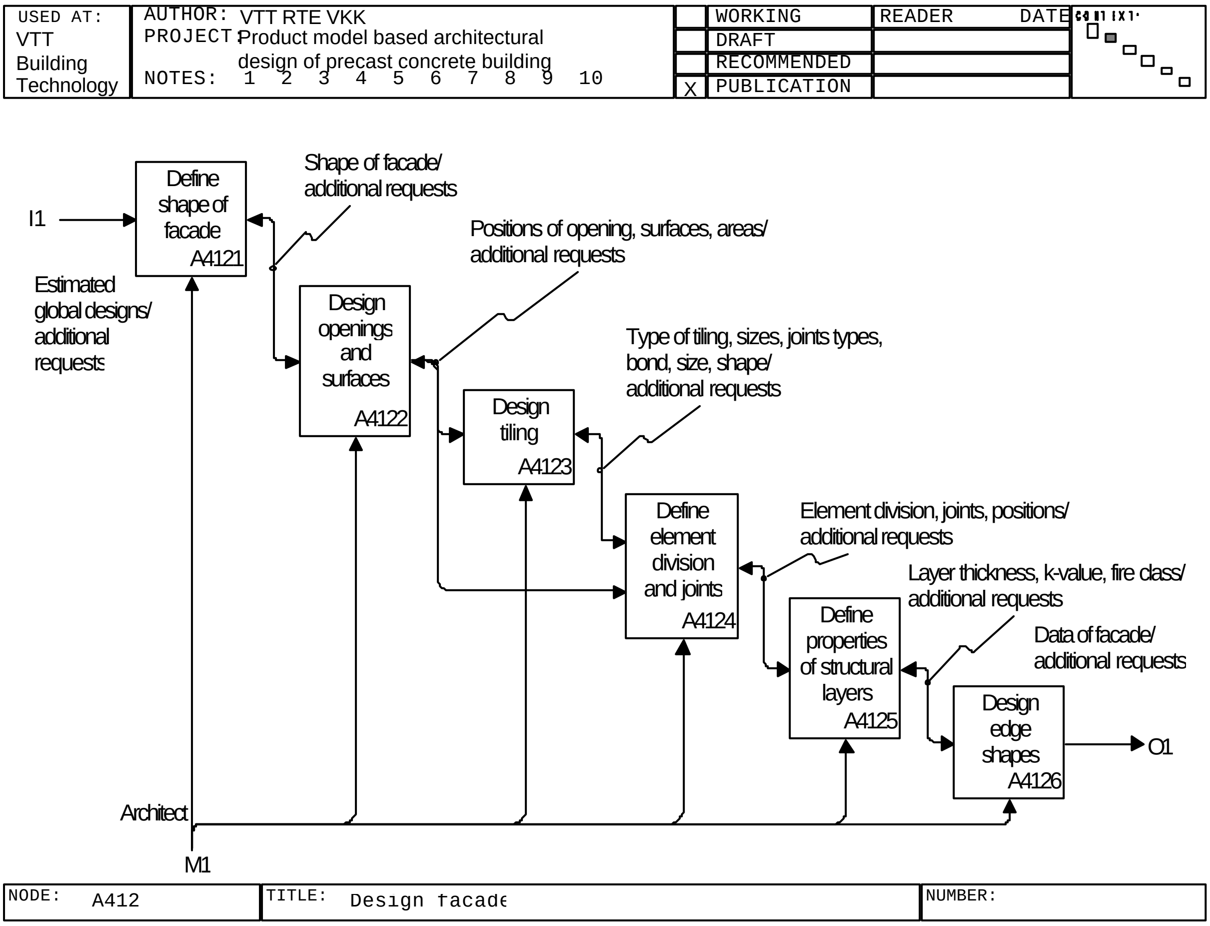


Figure 16. A part of the process model for designing a facade.

The conclusions from the prototype work are that it is possible to develop software tools that support the architectural design of facades. It is essential that the data exchange should be standardised among the participants. For instance, the architect designs surfaces, openings and edge shapes, and this information needs to be conveyed further downstream. Later, the structural engineer may divide the facade into a number of elements and will need information about, for instance, the surfaces. The question of who divides the facade into elements depends on the agreement between the participants.

It was shown that there are usable data already defined in architectural designs but that these data are difficult to pass on to the other participants in a structured format. It also became clear that the integration of the process model with the process models describing the work of the other participants is of importance.

4.2 A formalised process model of current Finnish design and construction practice

The objectives of the paper [Karhu and Lahdenperä 1999] were to:

- Compile a reference description of the overall construction process, covering the design and construction of a building, from the identification of the client's needs and briefing to completion for hand over

- Develop a systematic and generic model of current industrial practice in Finland
- Test the usefulness of formal process modelling methods, such as IDEF0, for the definition of construction processes and for communication with practitioners.

The paper presents traditional Finnish design and construction practice using, the IDEF0 modelling method. Since a building project involves many participants, interaction and communication between them is essential during the project.

The process sub-models that were studied included the following:

- Building procurement
- Architectural design
- Structural design
- Building services design
- Geotechnical design
- Construction.

The term 'building procurement' was used in this paper instead of the term 'client's work' used in the source documents, i.e. the tasks lists published by the Building Information Institute [RT 1995], which are more or less a de facto standard in Finland. The models contain more than 300 activities altogether, and the number of hierarchy levels is three.

The results of this stage of the research may be summarised as a more thorough study of reference models, identification of the problems encountered in presenting construction process models if many participants are involved, and guidelines on how to overcome the integration problems of various design disciplines. It was also concluded that the efforts for the development of processes should be focused more on the entire building process instead of the sub-optimisation of the particular processes of different participants.

It also became clear that the IDEF0 method alone is not enough to describe a complete building construction process from all the necessary points of view. Features such as conditional branching or temporal dependency between activities (finish-to-start, etc.) cannot be modelled with IDEF0. Another practical difficulty concerns the description of iteration in the model. In practice, iteration is related to the concept of time, which is not part of the IDEF0 method. On the other hand, the method as such allows the use of feedback loops and iteration (cf. Figure 2, the loop from activity A5 to A3) and does not indicate the order of activities. If iteration is present, it means that the same activities are carried out several times. On the other hand, IDEF0 models are independent of time. One more idea that became apparent is the use of a modular approach, where activity modules, or some kind of activity templates, are stored in a database and are used to build up a process model. This approach would support the use of reference models.

One additional ad hoc feature was developed during the definition of the reference models. Because the sub-models were, in practice, stored in separate files (using the Design/IDEF software), a need arose to add more information about the origins of certain flows coming from other sub-models. Had the whole process model been stored as one file this need would not have arisen. The flows that came from outside the sub-model in question were marked with an abbreviation in parenthesis to indicate the origin of the flow. For instance, the flows coming to the structural design from the architectural design were labelled '(ARCH) Basic solution'. This feature proved to be very useful for the communication and understanding of the models.

The above results and discussion lead to a conclusion that there is a need for a new construction process modelling method that overcomes the deficiencies of the IDEF0 method.

From an academic point of view, a brief discussion about the contribution of the authors, Karhu and Lahdenperä, is presented here. Karhu has developed the IDEF0 models, has suggested the use of an ad hoc solution to enhance the reading of the models (since they were physically in separate files), and has reported his experience of applying the chosen modelling method. Karhu has also suggested the use of a modular approach, which, in fact, follows the object-oriented approach that was already used in another paper [Karhu 1997]. Karhu has proposed a re-formulation of the task list of the architectural design. In other words, Karhu's main contribution from the point of view of this thesis, is the use and experience of modelling, of reference models, and of the IDEF0 method in general. Lahdenperä has contributed to the description of processes from the viewpoint of how to organise the management of the construction projects. Lahdenperä's contribution reflects his PhD thesis [Lahdenperä 1995].

4.3 Proposed new method for construction process modelling

This paper presents a new generic construction process modelling method called GEPM [Karhu 2000]. The objectives of this phase of the research were:

- To study and compare a number of existing formal process modelling methods in a systematic way, using a formal information modelling language
- To propose a new method as a synthesis of existing methods.

The new method is a synthesis of the features of other existing process modelling methods. These were scheduling, the simple flow method, IDEF0, IDEF0v, IDEF3 and Petri Nets. These other methods have all been used or proposed for the description of building construction processes. These methods were originally developed for specific purposes and it is thus often difficult to use them for describing processes from viewpoints other than those originally envisaged.

One of the main purposes of GEPM is to distinguish between an activity and a task, and how these are combined. Figure 17 depicts the principal relationships using the EXPRESS-G notation. An activity denotes the doing of something in general. Specific activities are called tasks. The tasks are of a certain type, indicated with an attribute 'task_type'. This type refers to an activity. The figure also shows a few other attributes that are borrowed from IDEF0 and scheduling. The relationship between an activity and a task can be illustrated by an example of installing a window. A process activity model would model an activity 'Install window' but the schedule of a project for installing windows would assign tasks 'Install window #123456', 'Install window #654321' etc., indicating the specific identification of the window. All the tasks would have a type that also indicated the inputs, outputs, controls, and mechanisms of the activity.

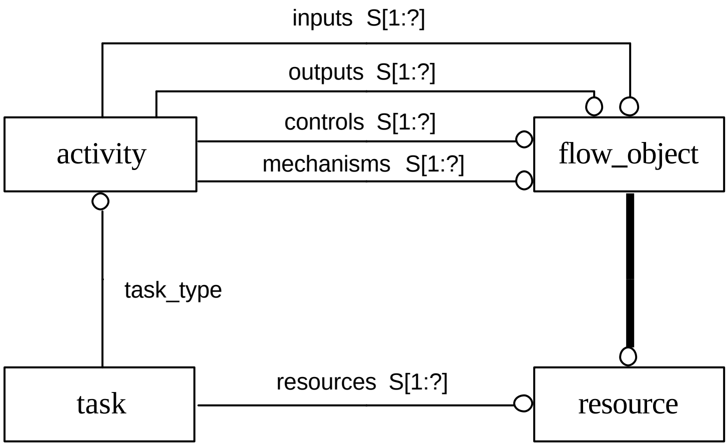


Figure 17. The relationship between an activity and a task.

A prototype application called the GEPM browser was also developed to test the basic principles of the method itself. The main idea behind the browser was to use the database approach and to study how different user interfaces could be used to interact with the same process model.

The features in the first prototype are

- Import of IDEF0 models using the IDL format, which is a de facto standard for data transfer between IDEF0 tools. The IDL defines all graphical information in the form of arrow segments and box data, their interconnections, diagram data, and possibly a glossary
- Support for the creation of tasks, temporal dependencies, and resources
- Export and import of tasks and resources to and from a scheduling tool.

In addition to the above-mentioned features, the prototype application functions as an object browser. Figure 18 shows four small screen-dumps of flow objects, one task with its predecessors, one unit of behaviour (UOB, from the IDEF3 method) with predecessors and corresponding junction type, and one resource with its state and where it is used.

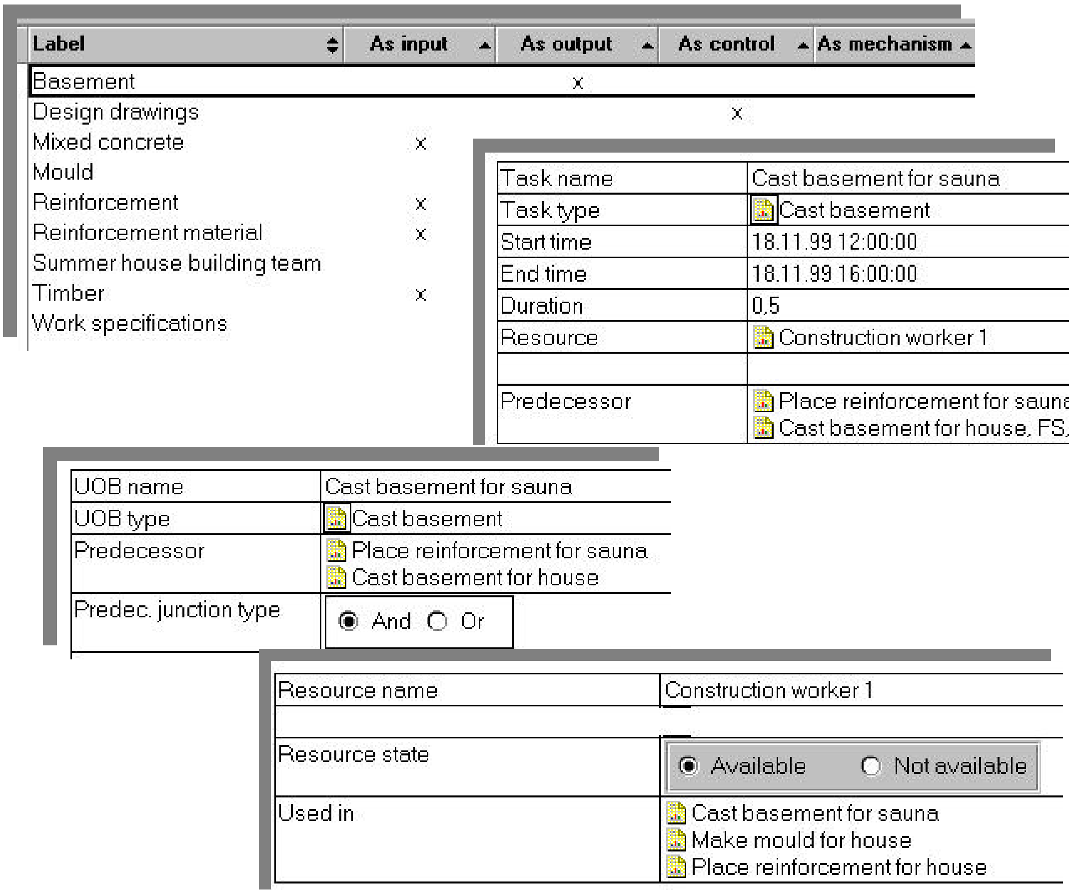


Figure 18. Examples of flow objects, a task, a unit of behaviour, and a resource.

One more result is a comparison of the six earlier process modelling methods with GEPM at the entity level. An activity in GEPM corresponds to an activity in the simple flow method, IDEF0, and IDEF0v, to a unit of behaviour in IDEF3, and to a transition of Petri Nets. A task in GEPM corresponds to a task in scheduling. A complete table of the corresponding entities between the methods can be found in the paper [Karhu 2000].

4.4 A view-based approach for construction process modelling

This paper presents the test results and a further developed generic process modelling method GEPM [Karhu 2001]. The objectives were:

- To evaluate a view-based approach to modelling methods for construction process modelling
- To evaluate the generic process modelling method GEPM, using case material from a specific building construction project
- To compare GEPM with other methods through conversions from one method to another.

The term 'view' means here a partial model, i.e. a partial GEPM model that includes features of other methods. Central indicates that the GEPM model is stored in a database.

The test case material was taken from an office building project, which had been thoroughly documented in an earlier research project at VTT. The documentation describes more than 1000 tasks, including more than 850 dependencies and more than 450 flows [Tanhuanpää and Lahdenperä 1996].

The prototype application, the browser, was at this stage developed further because of findings related to the test case material. The browser is based on a database solution using Lotus Notes, version 4.6. Figure 19 illustrates the different views of a GEPM database and the specific software applications that can be applied to these views. The views are:

- Schedule
- Dependency table
- Simple flow diagram
- IDEF0 diagram.

The reason why these were chosen is that the interviews with the companies participating in the MoPo-project showed that these particular views are important and useful. These views have also, in different combinations and variations, been used in many process modelling efforts and studies in construction companies. Figure 19 shows, additionally, the GEPM browser interfaces with the other software tools (Visio, MS Project, BPwin). The dependency table is generated from the data in the GEPM database and exists in conjunction with tasks.

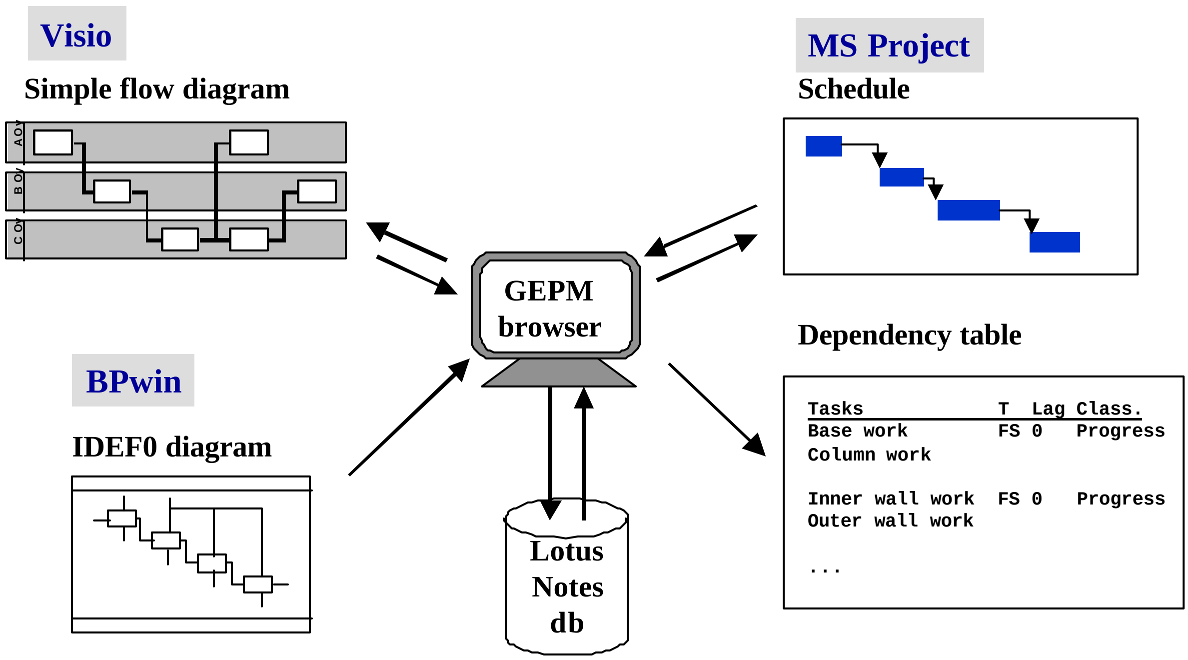


Figure 19. GEPM browser interfaces.

GEPM adds more modelling power to process modelling than the other earlier existing modelling methods that were used as a starting point for the development of the method. This is because GEPM has more concepts (i.e. task type and location) than the other methods, even when they are combined.

A complete model may be created using the GEPM browser but, in practice, it is often useful to use the separate software tools for the different views. These tools can be used to work on partial models. In this way, the conversions between the chosen views were tested.

One conclusion is that an IDEF0 model may be converted to a simple flow model relatively directly, but information is lost in this conversion. A simple flow model does not distinguish between an input and a control, and thus a practical choice of how to handle a control is a software implementation matter. A user selection for this has been implemented in the GEPM browser. Another observation concerns how to handle the levels of the hierarchy of an IDEF0 model. In addition, in this case, the developed browser has an option for selecting the activities to be converted from a particular level. One may ask whether information is lost in a conversion or not. If a model is stored in the database before converting it to another view, information is not lost but is not seen in the resulting view. A conversion from a simple flow model to an IDEF0 model requires additional information since a simple flow model does not have all the features included in IDEF0.

On the other hand, the conversion of these activity models into a schedule is not possible directly at all. This is because the scheduling method differs conceptually from the other methods, e.g. activities in IDEF0 have no starting time, finishing time or duration. The solution to this to use activities such as in IDEF0 as templates for tasks, i.e. a task has a type defined as an IDEF0 activity.

Conversion from a full GEPM model to the other methods is possible directly since the GEPM method includes all the concepts involved in the other methods. Here, the conversion covers the core entities, and the graphical presentation is considered to be an implementation.

This paper also briefly initialises the discussion of how to use reference models and how to proceed to project-specific-model, with the help of GEPM.

5 DISCUSSION

The main results presented in this thesis can be summarised as follows:

- A new construction process modelling method with more modelling power has been developed
- A new process modelling software application has been developed
- The distinction between an activity and a task, and the relationship between these, has been defined in the modelling method.

These items are discussed in more detail in the following sections.

5.1 Conceptual model of GEPM

The conceptual model of GEPM is shown in Figure 22 as an EXPRESS-G diagram. The figure presents the conceptual model at an entity level only. This version differs from the earlier versions that were presented in the papers [Karhu 2000] and [Karhu 2001]. The method has borrowed many concepts from the other methods, for instance activity, task, temporal dependency, flow object (very similar to the ICOM concept of IDEF0), resource, and junction. A relationship between an activity and a task is essential. A task has a type attribute, i.e. task type. Activities are defined as general descriptions of how to carry out some action whereas tasks are specific descriptions with starting time, finishing time, duration, and location. The dependencies between tasks are modelled using two enumeration types instead of using classes and subclasses. These two separate types define the type of dependency and the cause of dependency. The textual version of these (since EXPRESS-G does not show these) is shown in Figure 20.

```
TYPE task_dependency_type = ENUMERATION OF
 (FINISH_TO_START, FINISH_TO_FINISH, START_TO_FINISH, START_TO_START);
END_TYPE;
TYPE task_dependency_cause = ENUMERATION OF
 (TECHNICAL, HIERARCHY, LOCATION, RESOURCE);
END_TYPE;
```

Figure 20. Dependency type and cause of a task in textual format.

The dependencies between activities are also modelled using an enumeration type as shown in Figure 21.

```
TYPE activity_dependency_cause = ENUMERATION OF
 (TECHNICAL);
END_TYPE;
```

Figure 21. Activity dependency cause.

The causes of dependencies in both cases (between tasks, or between activities) have one common reason: a technical dependency. Hence, if two activities have a technical dependency between them, the resulting tasks have a technical dependency relationship as well. A question arises here as to whether an enumeration is needed if there is a single choice.

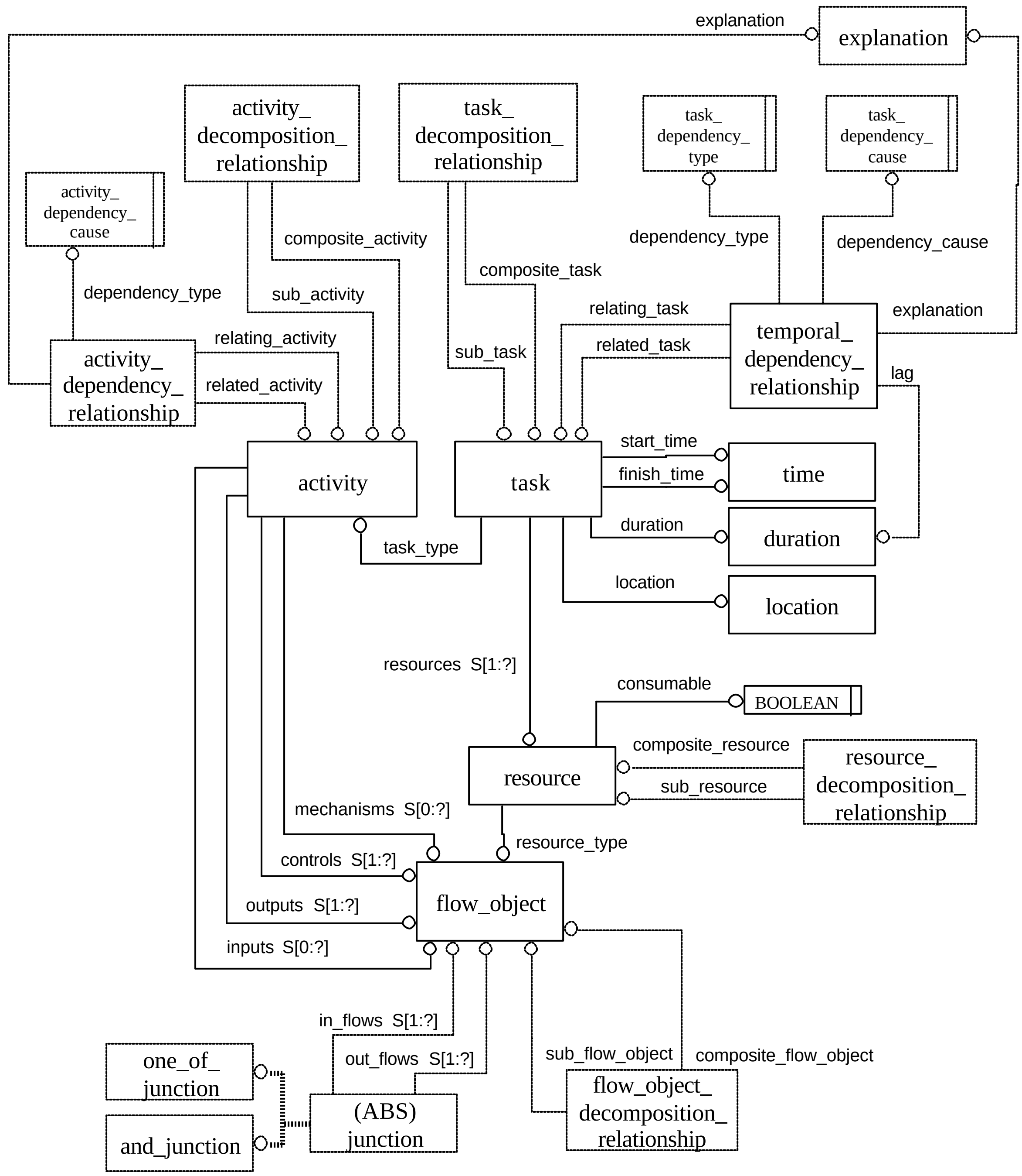


Figure 22. Conceptual model of GEPM shown at the entity level.

The decompositions of activities and tasks are separated in the version presented here, in contrast to the previous versions that were presented in the papers [Karhu 2000, Karhu 2001]. This means that an activity may be decomposed into several activities, not into, for instance, activities and tasks. Another modelling choice is that the activity and task have one common superclass for the decomposition relationship but there is a rule to exclude a combination of decompositions.

A practical modelling choice is that the attributes start time, finish time, and duration are related to each other, e.g. duration can be derived from start and finish times, or finish time can be derived from start time and duration. The choice between these modelling options has been considered to be an implementation issue, and thus the conceptual model in Figure 22 does not show any of these as attributes as derived.

GEPM adds more modelling power to construction process modelling [Karhu 2001]. Testing GEPM demonstrated that is has more modelling power than the other methods studied. The concepts of the other three methods that have been compared with GEPM are illustrated in Figure 23, in which the principal concepts of each method are listed inside circles. GEPM (largest circle) covers all concepts. Moreover, GEPM has a specific relationship between the activity and the task.

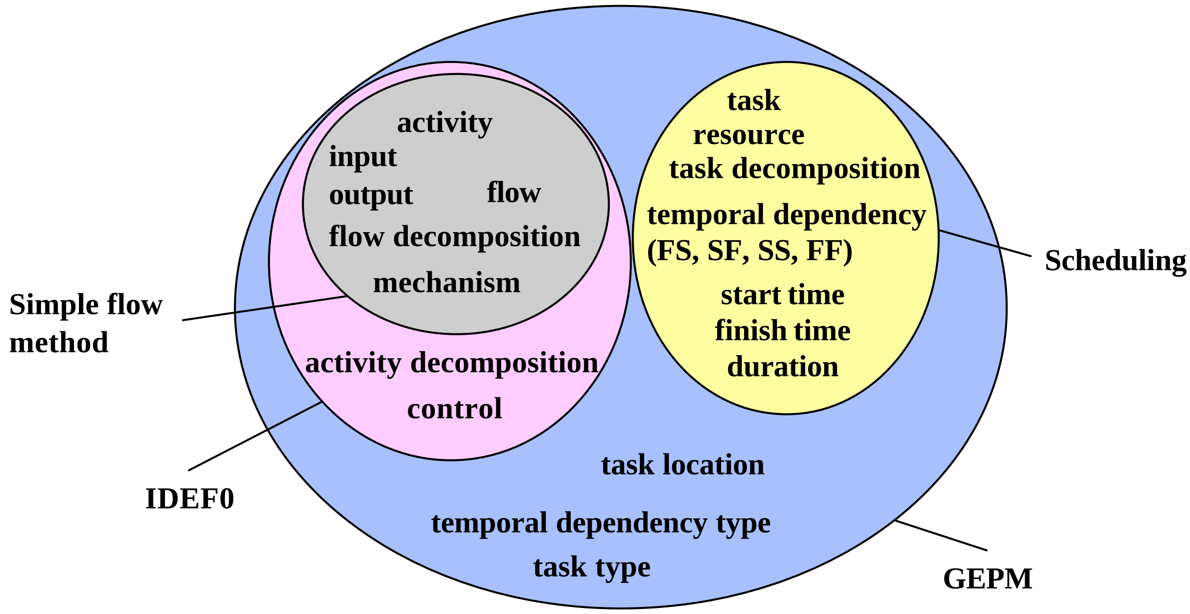


Figure 23. A schematic illustration of the modelling power of the different methods.

The conceptual model in Figure 22 does not show all the concepts that are needed for a prototype application. The implementation of GEPM is discussed in the next section.

5.2 Prototype implementation of GEPM

The conceptual model has been implemented by the author of this thesis on a Lotus Notes, version 4.6, platform. The resulting software application is called a GEPM browser. Lotus Notes is a groupware tool that uses, in Lotus Notes' terms, a document-object database where users interact with objects, and the data of these objects are held in Lotus Notes documents [Lotus Notes 1999]. For readers not familiar with Lotus Notes, it should be noted that Lotus Notes is not a relational database. The documents may include free text, pictures, attachments, and fields. The fields are of the following types: text, time, number, keywords, rich text, authors, names, and readers [Lotus Notes 1995]. The three latter field types have not been used in the GEPM browser because their primary purpose is to manage user access rights. The templates for document-objects are called forms. An overall and simplified interpretation of the internal structure of Lotus Notes documents is illustrated in Figure 24. Using the object-oriented terminology, one can say that the forms represent object classes.

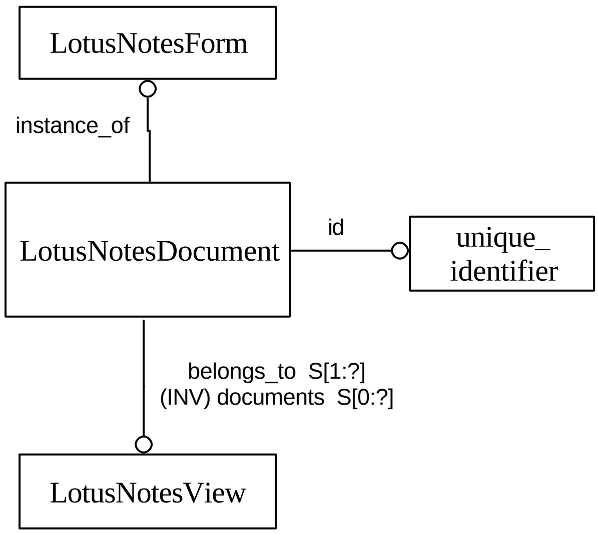


Figure 24. A Lotus Notes document, form, unique identifier and view, and their internal relationships.

A Lotus Notes document, called here 'LotusNotesDocument', is an instance of a Lotus Notes form, called here 'LotusNotesForm'. When compared to the terminology of relational databases, one may say that the forms define the fields for tables of a database. A Lotus Notes document automatically receives an internal unique identifier that is a 32-character combination of letters and numbers. The unique identifier also appears in Figure 24. It was a logical choice to use this unique identifier directly for all documents in the database (it is not shown in the schema of GEPM in Figure 22). The values of many documents can be observed in different Lotus Notes views, in addition to the documents themselves. The term view as defined by Lotus Notes is not the same as the definition of a view in GEPM. Thus, a Lotus Notes view contains a set of documents. It is noted that one document may belong to more than one view. The programmer may select which fields or field values are to be displayed in the views.

The forms are also used to define methods for manipulating the data. Most of the routines to handle a model in a GEPM database have been developed using the LotusScript programming language. LotusScript is a special programming language for the Lotus Notes environment and supports the object-oriented approach [LotusScript 1995]. The object-oriented features of LotusScript were not needed eventually even though they were tested at the beginning of the prototype development.

The views are also used for building queries for user interfaces. A query may be dynamic, e.g. based on a value of another field, or static, that is used normally for querying all values of a view. Thus, by using queries to create selections of other documents, one can handle instances of entities of a conceptual model, such as GEPM, in an effective manner because one can, in practice, build all kinds of relationships between the documents. Figure 25 shows an example of a user's dialog box for selecting tasks. In this example, the tasks that have been selected are shown with a little 'hook' at the front of the task name.

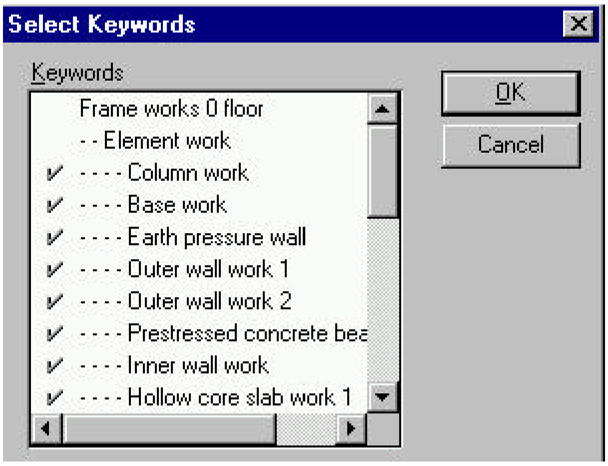


Figure 25. An example user interface dialog box for selecting tasks.

In Lotus Notes, the hyper-links between documents should be placed and shown in the rich text fields. It means, from the programmer's point of view, that the unique identifier must be stored in one text field and the hyper-link is then automatically generated on the fly into the rich text field. In fact, a programme routine was developed for this purpose. The text fields may contain many values and rich text fields may contain many hyper-links. For readers not familiar with Lotus Notes, this programming choice was provided because the value, i.e. the unique identifier, cannot be retrieved from the hyper-link itself and thus there is a need to store the unique identifier in a separate field.

The practical implementation of the conceptual model of GEPM has been developed so that each major entity has its own Lotus Notes form object that defines the required fields. These forms correspond to the entities as follows:

- Activity
- Flow object
- Junction
- Resource
- Task
- Temporal dependency relationship.

An implementation matter concerning whether to use a separate entity for a relationship or not, depends on whether there is a need to describe the relationship in more detail by assigning more attributes to it. In the GEPM browser, the decomposition relationship was implemented using a simple relationship through an attribute and thus differs slightly from the conceptual model of GEPM. An example is illustrated in Figure 26 (using EXPRESS-G notation) where the activity decomposition is shown.

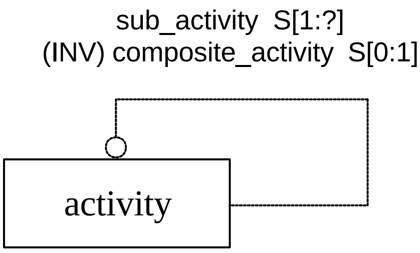


Figure 26. Implementation of the activity decomposition in the GEPM browser.

On the other hand, the entity for a temporal dependency of tasks, as defined in the conceptual schema of GEPM, is implemented in a separate (Lotus Notes) form that has the necessary database fields for its attributes 'relating_activity' and 'related_activity'. In addition, it has fields for its other attributes such as the type of dependency, its cause, lag, and a field for a free textual explanation. Using the example of temporal dependency, a set of two tasks and one temporal dependency between them results in three separate documents in the database. It should be noted that this latter implementation requires more programming efforts than the simpler version.

The GEPM browser has been linked for data exchange with BPwin, MS Project and Visio. The data exchange links with MS Project and Visio were programmed in LotusScript and Visual Basic, using OLE automation. On the other hand, the data exchange from BPwin was carried out using files in the IDL-format [IDL 1994]. In practice, this choice was influenced by the fact that BPwin 2.0 does not support OLE-automation. However, the IDL-format is a de facto standard data exchange format between IDEF0-tools, and thus any IDEF0 tool that can generate a file in the IDL format can be used. BPwin 2.0 generates some additional information (e.g. colour codes for lines, etc.) in the IDL file but this information can be easily removed during the import. An extract from an IDL file is presented in Figure 27, showing some diagram and box information.

```
DIAGRAM GRAPHIC A0 ;
 CREATION DATE 2/3/2000 ;
 REVISION DATE 29/3/2000 ;
 TITLE 'Procurement' ;
 STATUS WORKING ;
 BOX 1 ;
  NAME '{LWI I 4 255 255 255}Plan material<CR>procurement' ;
  BOX COORDINATES (0,17583;0,28455) (0,30207;0,12195) ;
 ENDBOX ;
 BOX 2 ;
  NAME '{LWI I 4 255 255 255}Make<CR>invitation to<CR>tender' ;
  BOX COORDINATES (0,33814;0,43089) (0,44635;0,28455) ;
 ENDBOX ;
...
ENDDIAGRAM ;
```

Figure 27. An extract from an IDL file showing some diagram and box information.

The data exchange is always initialised from the GEPM browser. The unique identifier set by Lotus Notes is used in Visio and MS Project to retain the consistency of the model. A user-defined template was required in Visio to store the GEPM entities and to help in managing the software-specific features. For instance, during the import of new entities from a simple flow model, the unique identifier is set to the custom properties of the activity boxes of Visio. Thus, the graphical model can easily be updated and managed consistently later because the same unique identifier is used in both software applications. The same principle is used with MS Project, which can store additional information about tasks. No special template was required for MS Project, because MS Project has a number of free text fields for users' special purposes and needs, and these have been used to store the unique identifier. In practice, the user may make changes to the model in the GEPM browser or make changes to the partial models by using Visio and MS Project. On the other hand, the IDL file from BPwin 2.0 is used when importing a new model into the GEPM browser. The implementation of an IDEF0 template for Visio would have been one alternative but was deemed beyond the scope of this project because it would not have brought any new knowledge to the prototype itself or its testing.

Considering the data exchange from a practical point of view, the prototype application stores more information about an IDEF0 model than is seen in the conceptual model (Figure 22). This information is the node numbering of activity boxes and the diagram identification (such as A1, A2, etc.) information. Additionally, the names of files are stored in the GEPM database, as the models are interpreted as partial models.

5.3 Graphical constructs of GEPM

This section describes briefly the ideas and suggestions for the graphical constructs of a GEPM model. The graphical constructs can be borrowed from the other process modelling methods. Figure 28 illustrates the basic graphical constructs of GEPM for an activity model. The IDEF0 method provides good basic choices. Hence, a graphical box denotes an activity and an arrow denotes a flow object. A junction, denoted here as an octagon, can also be added to a model. The octagon has either an 'O' inside, denoting an or-junction, or an '&' (ampersand) inside, denoting an and-junction. A technical dependency is denoted by an arrow with a round ending, with the ending pointed towards the related activity. One can add a label to the activity box, to the flow object arrow, and to the relationship arrow.

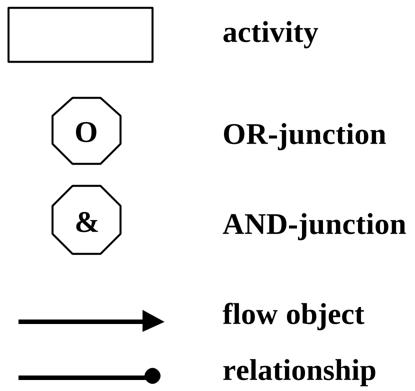
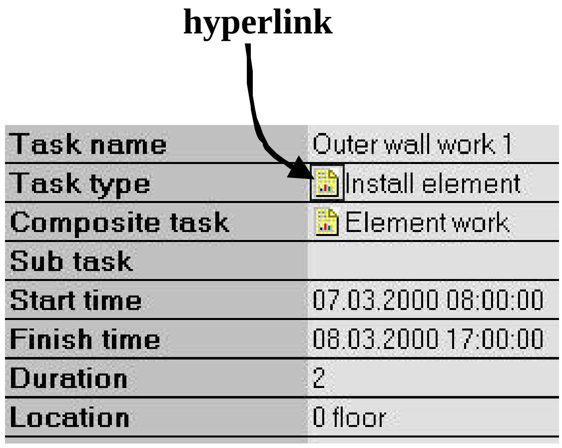


Figure 28. Graphical constructs of a GEPM model.

The graphical constructs of tasks, i.e. schedules follow the traditional Gantt chart or PERT chart notation and thus need no additional features (see Figure 1 for an example). A graphical notation for denoting the task type in diagrams has not been defined in this context, and this is left for further discussion in the future. In the present version, the GEPM browser uses a hyperlink between a task and an activity, Figure 29. The hyperlink 'spot' is indicated by an arrow in the figure, and by clicking the hyperlink, the corresponding activity type is displayed.



Task name	Outer wall work 1
Task type	Install element
Composite task	Element work
Sub task	
Start time	07.03.2000 08:00:00
Finish time	08.03.2000 17:00:00
Duration	2
Location	0 floor

Figure 29. A task type is indicated by a hyperlink in the GEPM browser.

5.4 Views and conversion of partial models

Converting a complete GEPM model into the partial models is always possible since a GEPM model includes all the core concepts of the other methods. On the other hand, conversions between the different views are of interest and constitute the principles of how to model using GEPM. Thus, a conversion of this type is a specific view of the information of the model. On the other hand, a conversion from one partial model into another partial model is of interest. The conversion here is a mapping from one conceptual model onto another. The mapping is enabled because of the way the corresponding concepts have been interpreted [cf. Karhu 2000].

Activities may be modelled using either the simple flow view or the IDEF0 view. For a simplified IDEF0 example (Figure 30), converting from IDEF0 to simple flow means a reduction of the information contained in the model because the control concept is not converted. This is because control is not part of the simple flow method. On the other hand, if the feedback 'Reinforcement' was modelled as an input to the activity 'Place formwork', it would become a part of the model in the simple flow view (note: here formwork refers to the wooden part of the mould). Thus, the conversions depend on the modelling choices, i.e. the way in which the modelling work is carried out. Another difference concerns the hierarchy of an IDEF0 model. Hierarchy is not part of the simple flow method. In practice, in a conversion from IDEF0 to simple flow, the user selects the level of activities, which are to be converted. The developed prototype application supports this kind of procedure. The resulting simple flow views may thus represent the different levels of hierarchy of the IDEF0 sub-model - but one level at a time. The logical consequences of the above conversion procedures are that modelling rules and guidelines are needed, and that experience in modelling is essential.

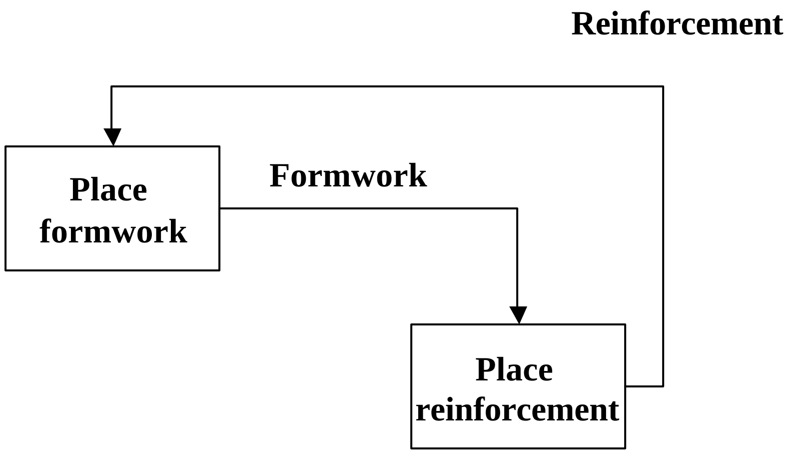


Figure 30. The modelling of feedback or iteration in IDEF0.

This raises the question of whether information is lost or not during a conversion from IDEF0 to simple flow. One might say that information is not lost because the GEPM database has all this information but it does not appear in the view. One may, in practice, using the GEPM browser, convert a control into an input. On the other hand, if GEPM is used simply as a converter from one software application to another (or from one corresponding model to another), it means that information is lost (or must be added) because the different methods do not all have the same concepts.

The above conversions between activities of the different methods are relatively straightforward. The more complex case concerns the conversion of an activity model into a schedule. GEPM has two essential types of entities: activities and tasks. These are orthogonal in the sense that they do not have common attributes (see also Figure 23) and thus it means that an activity model cannot be converted directly into a schedule but a rule must be applied.

As has been stated, the type attribute of a task refers to an activity. The idea comes partly from the fact that it is not necessary to model all specific activities, for instance activities to install specific pre-cast concrete elements on a site, but instead the activities are modelled as sort of templates. The specific tasks may then all have the same type. Figure 31 illustrates an activity 'Install element' in a reference model and three specific tasks 'Install element 1', 'Install element 2', and 'Install element 3' in a more specific model. The arrows in this figure simply illustrate a type relationship. These different models belong to a reference model layer and a specific model layer, respectively. This approach thus clearly supports the distinction of a reference model and a specific model.

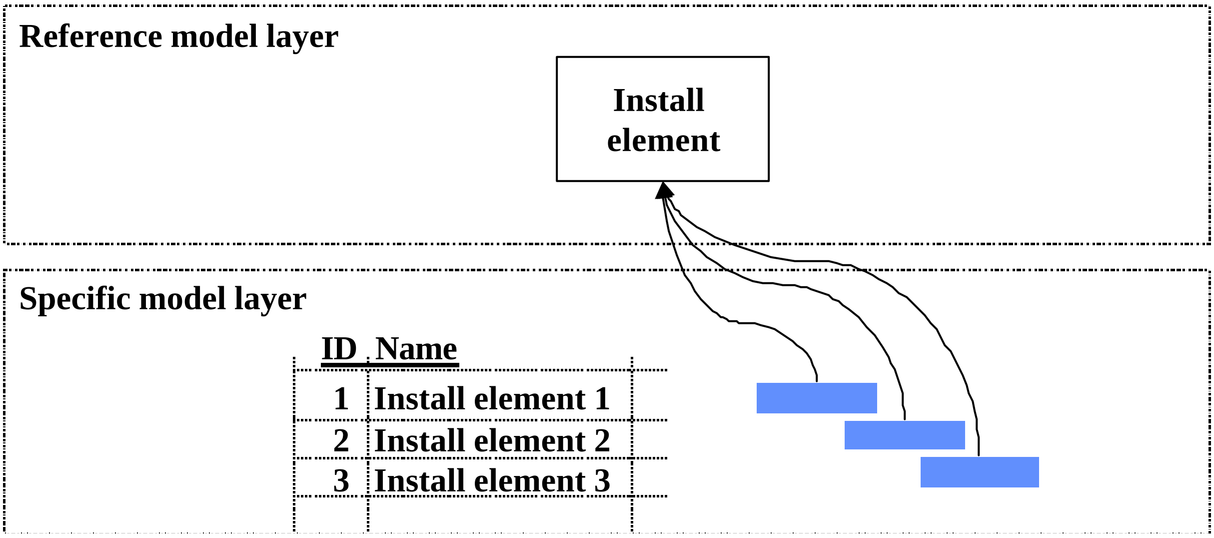


Figure 31. Reference model layer vs. specific model layer.

The distinction between activities and tasks, as defined here, exposes the critical question of whether it is possible to convert activities into tasks directly. This depends on how the constituent methods are used. The IDEF0 method (a sub-part of GEPM in this context) is flexible and does not as such restrict the activities to be modelled at a detailed or at a general level. Sometimes, when activities are modelled at a very detailed level, the resulting schedule looks almost the same. On the other hand, if feedback or any other iteration is present, the conversion is not straightforward. The problem can also be illustrated in Figure 30, where the two activities 'Place formwork' and 'Place reinforcement' form a sequence with a feedback 'Reinforcement' from the latter. The sequence here means parallel activities and is typical for this kind of process, where many iterations are often needed to get a satisfying solution, i.e. a formwork that can hold the reinforcement on its place. The arrows are used as constraints in IDEF0 [IDEF0 1993]. Thus, the direct instantiation of these two activities, i.e. conversion to the tasks of a schedule, would lead to an impossible loop because the instantiation of 'Place formwork' would need the presence of the feedback 'Reinforcement' but which is the result of 'Place reinforcement'. Another solution to this problem might be to distinguish more precisely the type of arrows (flows), for instance by separating constraining and optional arrows.

5.5 Comparison with other information models

One can compare GEPM with the other information models. The models are the unified approach model [Björk 1992], the general construction project model GenCOM [Froese 1992], the information reference model IRMA [Luiten et al. 1993], the ICON model [Aouad et al. 1994], the STARGEN model [Hannus & Pietiläinen 1995, Pietiläinen and

Heinonen 1995], the core process model suggested [Froese 1996], and the draft IFC model [IAIIFC 2000].

All the above models have defined a concept 'activity', which corresponds to either an activity or a task in GEPM. The performer of an activity, i.e. a mechanism in GEPM, can be compared with an agent in the unified approach model and IRMA, and with an actor in STARGEN and the core process model.

GEPM defines output, which is also found in STARGEN and the core process model. The term 'result' was used instead of 'output' in the unified approach model. The corresponding concept for input is more complex. In GEPM, input is a flow object that becomes input when it enters the (graphical) box from the left. This idea has been borrowed from IDEF0. STARGEN defines input as a state of a project object, which can be, for instance, information. IRMA defines input in a similar manner. The state concept is also used for output in IRMA and STARGEN. The state concept is related to the transition of an object over time, e.g. an object may be in an 'as-required' state or in an 'as-designed' state. In the conceptual model of GEPM presented in this thesis, the state concept has been omitted. It was, however, presented in the papers [Karhu 2000] and [Karhu 2001], having a simple meaning of whether a resource is available or not.

The concept 'control' in GEPM is recognised in the core process model. The control concept is borrowed from IDEF0 and thus, the meaning of it is the same, i.e. a control is used as a constraint for an activity and does not become part of an output.

GEPM defines precedence through temporal dependencies (finish-to-start, etc.). The idea of precedence is also suggested in the ICON model and in the core process model. GEPM defines, additionally, the reason behind the dependency, e.g. a technical dependency, which is also present in ICON. GEPM adds the dependency cause. Moreover, the technical dependencies may also be present between activities.

The IFC model relates some entities in a very similar way to GEPM. The use of a separate relationship object instead of using an attribute such as 'predecessor', which refers to another task, is very similar to the situation in GEPM, where the temporal dependencies (and thus predecessors) are modelled using a separate object 'temporal dependency relationship'. The resource for a task in the IFC model is defined by 'resource use' instead of resource directly.

On comparing solutions, a question arises as to whether a task can be modelled as a subclass of an activity, Figure 32. This is another modelling choice. It would mean that the task would inherit inputs, outputs, controls, and mechanisms from the activity (see Figure 22). On the other hand, the modelling choice used in GEPM corresponds more to the traditional scheduling approach.

Figure 32. A task as a subclass of an activity.

5.6 A general remark on a database as a platform

The idea of using a database as a platform is not new but it brings some advantages, cf. Figure 33. The process views, which are the primary way in which the end users interact with the process models, can be obtained through queries from a database in a similar way to the extraction of design drawings from CAD tools. Advanced CAD systems store the models in databases and users interact with the designs through views, often to certain scales.

Modelling work is often carried out in an iterative way, sometimes involving several cycles. The examples presented in the fourth paper [Karhu 2001] were also created in an iterative way. This was partly because there was a need to test the prototype application, and partly because the creation of tasks and other entities was performed in several steps. This procedure was used particularly for schedules, where many task dependencies were set up. The dependencies were checked and corrected several times before the final schedules were completed. The project planning software was used for the creation of tasks, and the GEPM browser was then used to import the tasks into the database. Thus, the iteration proved useful, not only for the test material but also for the development of the GEPM method along with the prototype application, as can be seen in Figure 2 depicting the steps that were taken to develop GEPM.

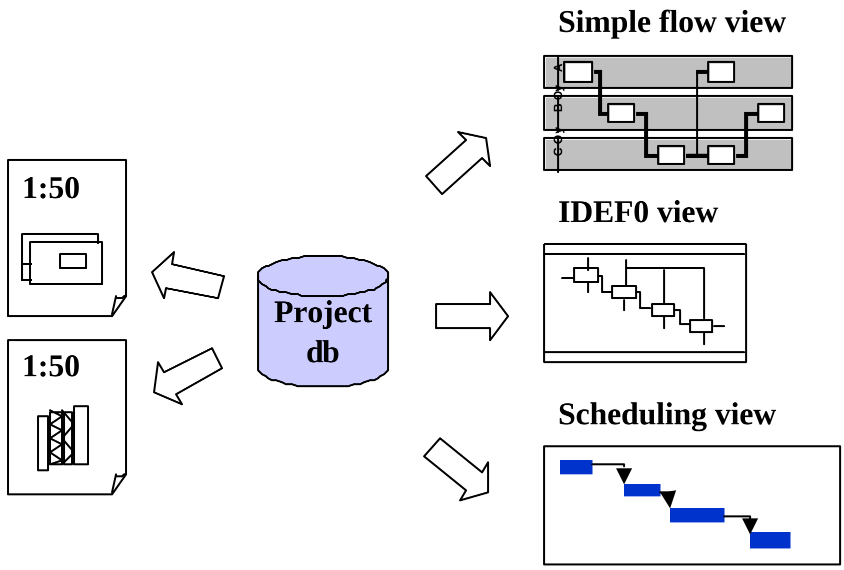


Figure 33. There is a similarity between how drawings are generated from a project database, on the left, and how process views are generated from database information, on the right.

The above description of the use of iteration in the definition of tasks raises the question as to whether the task type is eventually needed - if tasks are created before determining their type. The answer is that the tasks can be created first using the GEPM browser and then exported to the project planning application for setting the basic temporal dependencies. It should also be noted that the activities can be created easily before any schedules are determined.

5.7 How does GEPM meet its requirements?

Does GEPM meet the requirements? The characteristics of a construction process and its requirements have been presented in Chapter 3 on the state-of-the-art. The answer to the above question is obviously positive because of the developmental history of GEPM (i.e. borrowing the best features from other methods). Thus, the question is reduced, in fact, to the modelling rules and guidelines, i.e. how to model using GEPM.

Input, output, control, and mechanism or actor can all be modelled with GEPM. Concurrency of activities can be modelled using the junction. Concurrent tasks can be modelled as well. The order and dependency of activities mostly concern tasks but they may also exist as technical dependencies between activities. These can be modelled using GEPM. Iterative activities and use of feedback can be modelled using activities. Iterative activities are common in building construction as has been shown in the case of formwork. An example of feedback or iteration was illustrated in Figure 30. There can be iterative tasks, but from a scheduling point of view, these would be separate tasks, each having their own starting and finishing times. GEPM does not restrict in any way the modelling of such tasks. A more common interpretation is a repetitive task in this case.

An example of conditional branching, derived from the Finnish task lists [RT 1995] for how to select building designers, is illustrated in Figure 34. The branching of the output of 'Decide on selection method' is depicted as an octagon with 'O' inside meaning 'or'. The branching is relevant to activities since, as in Figure 34, one of the activities after the branching will be activated, i.e. in GEPM one task with the corresponding type activity ('Select directly', 'Select based on tenders', 'Negotiate', or 'Organise competition') will be created. Moreover, branching requires rules to be associated with the junction but these have neither been included in GEPM nor in the prototype application.

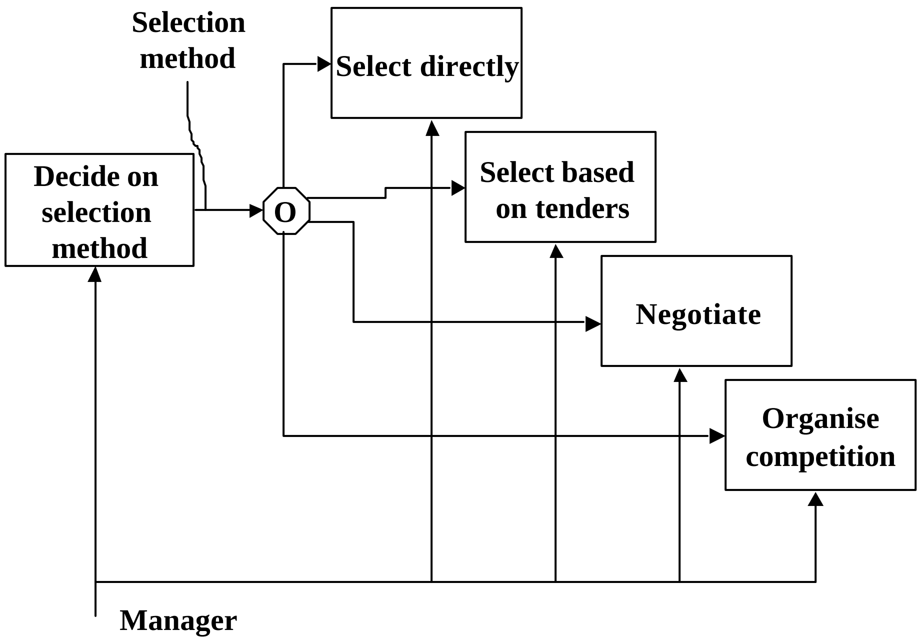


Figure 34. An example of conditional branching.

The use of hierarchy has many interpretations. The decomposition of activities, tasks, and flow objects are examples. The decomposition in GEPM is a 'part-of' decomposition relationship. The level of detail depends on the hierarchy and does not indicate whether activities are specific or not.

Modelling of organisations can be carried out to a certain extent. GEPM does not define the entity 'organisation' as such. On the other hand, the concepts of resource and mechanism can be used for modelling organisations. The flow object and resource of GEPM can be decomposed and thus an organisation hierarchy can be achieved.

In addition to the specific conceptual requirements, there are many general requirements that GEPM should fulfil. Thus, how does GEPM satisfy the requirements posed by the purpose and usage of process models? The previously mentioned requirements, i.e. the facilitation of human understanding and communication, the support of process improvement, and the support of process management, will be discussed below. 'Human understanding and communication' means that a group (here the construction project participants) is able to share a common representational format [Curtis et al. 1992]. Applying this to GEPM, the common format is the views that represent the partial models GEPM. Communication is further enhanced because GEPM includes many concepts from other existing, and well-known methods and can be run partially using views. Moreover, GEPM brings more modelling power (i.e. more modelling features) and thus models in GEPM contain enough information. Model verifiability is best achieved through graphical representations and common sense [Kochikar and Narendran 1994]. The fourth paper [Karhu 2001] tested the verifiability, and thus GEPM satisfies this requirement.

The support for process improvement requires a basis for defining and analysing processes. The different process components can be identified using GEPM. One more feature supporting process improvement is the re-use of well-defined process descriptions. Well-defined means here processes that have proved useful in construction projects. This leads logically to modularity. The database implementation of GEPM supports this approach. On the other hand, since GEPM is not a simulation tool, one cannot argue that GEPM can be used for estimation of the potential impact of changes in the process. The entities of cost and method of tasks are not included in the conceptual model of GEPM but could be added to the model.

The process management issues mean, from the point of view of GEPM, that project-specific processes and project plans can be developed. GEPM can be used to store existing process alternatives in a modular way, and thus can enhance the management of the process. Process management involves monitoring the project. This means, in practice, that GEPM can be used to model the issues that are to be monitored. As described earlier, GEPM is not a simulation method, which means that it is not possible to determine whether GEPM can be used to forecast the progress of the project.

6 CONCLUSION

A new generic construction process modelling method has been developed. The new method overcomes the deficiencies of the earlier methods and thus provides a solid foundation for further research and development projects. Research as such has an objective to produce new theoretical knowledge. An important aspect of research at VTT is that most projects are funded partly by the industry, and thus one cannot neglect the applicability of the results to practice. Hence, the next sections discuss a practical usage scenario for the new method. Furthermore, some further research topics are suggested.

6.1 A practical usage scenario of GEPM for industry

The fourth paper [Karhu 2001] presented the principles of how to use GEPM. The principles were described through examples of how to define activity models, how to convert between the different views, and how to proceed from activity models into schedules. This section discusses a scenario for using GEPM in practice and how to re-use existing process models in a construction company. The incentives for this scenario have come from the construction industry, especially from the companies involved in the MoPo project (during 1999-2001).

Figure 35 depicts a usage scenario, consisting of three steps. Quality systems, in Step 1, have been developed and are in use in many companies, not only in construction companies but also in manufacturing companies, design and engineering consultancy companies etc. Quality systems often contain manuals, checklists, and instructions on how to perform certain operations. They also contain information to help in deciding responsibility limits etc. In other words, a quality system describes the 'best-practice' inside a company. The IDEF0 method can be used to describe parts or certain aspects of the quality system in a formal way. A model is thus a quality system reference model. There can be many different alternative models with related documents and instructions for the various purposes in the quality system. The GEPM browser can be used to store this information as a partial GEPM model, and in a database format. Then the users, e.g. design engineers, interact with the quality system through views such as the IDEF0 view.

In Step 2, a project-specific model can be created from the reference model in the quality system, using the simple flow method to emphasise participants, possible sequences of activities etc. It is also possible to extract information for various purposes and determine what levels and details should be included in the more specific model. Here, the GEPM browser works as a converter of the general process model into a project-specific model. All necessary information is also stored in the GEPM database.

Furthermore, in Step 3, the specific project model is then scheduled for the actual work. In a similar way to the procedure in the earlier steps, the schedule is stored in the GEPM database. At this stage, the process starting from the quality system is documented in one database.

As has been described in the previous chapters, GEPM and the prototype browser can be used in an iterative manner. Thus, the user may change the models as necessary.

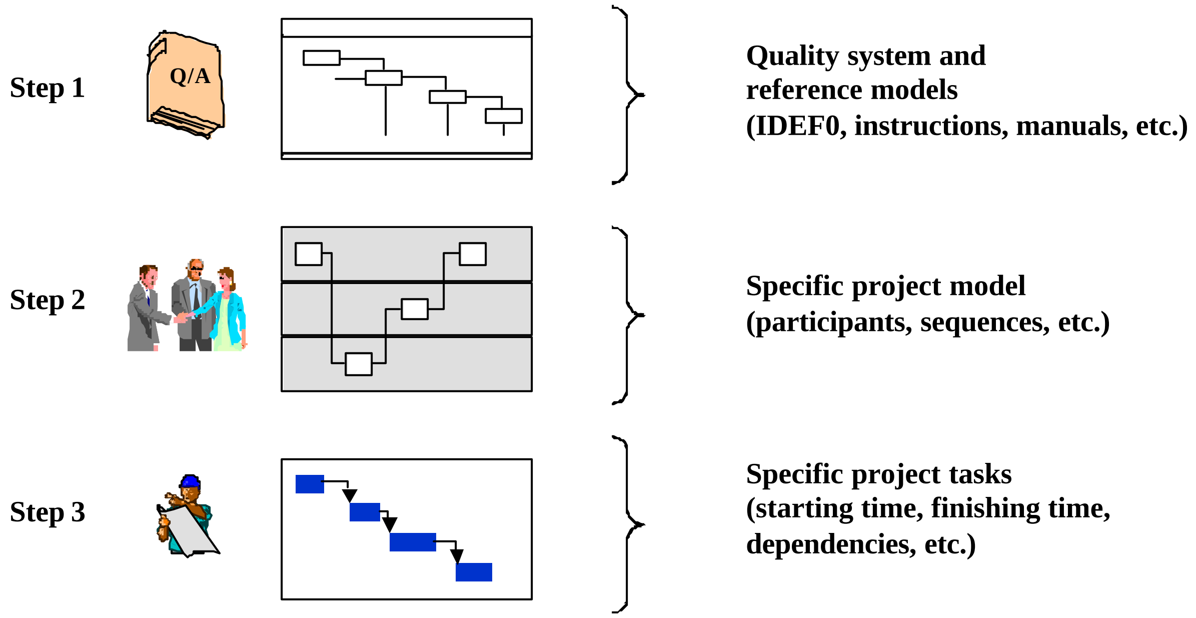


Figure 35. Usage scenario for GEPM.

6.2 Further research topics

The reference models may serve as a basis for project-specific models. Often, the reference models contain alternatives or more activities than are needed for a specific project. An interesting further research topic would be to clarify what the relevant model activities for a chosen activity are, i.e. certain activities 'bring' other needed activities into the model. This can be illustrated with the example of selecting designers for a construction project (see also Figure 34). The selection can be made by direct selection, through negotiation, based on tenders from the designers, or by organising a design competition. These activities in turn have some other specific activities that are required. Another typical example concerns the selection of the frame structure of a building, whether the frame is made of steel or concrete, for instance. The concrete alternative can further be divided into two more alternatives: cast-in-situ frame, and prefabricated element frame. Similar procedures also exist in other parts of the construction process.

This leads to the question of alternative solutions and to the question of how to perform a comparison of the solutions with respect to cost, time, and efficiency. Moreover, a possible further research topic would be, if it were possible, to create a kind of 'user manual' for a project by selecting the necessary activities and tasks from the quality system. This 'users manual' would serve specifically determined projects.

Concerning the prototype application, another possibility for the further development of GEPM is the implementation of support for more methods (beyond the currently supported IDEF0, simple flow and scheduling) and the determination of the conversion rules between them. Here, Petri Nets, IDEF3, and simulation methods are of particular interest.

6.3 Final conclusions

Process modelling in the construction industry is still lagging behind the overall development of IT-based tools. The object-oriented principles for process modelling combined with databases, graphical user interfaces, and internet technology may offer substantial advantages to the modelling efforts. This approach is also supported by the requirement for and use of quality and document management systems. I hope that the research presented in this thesis will make a valuable contribution to the development and use of a new generation of process modelling methods and tools.

REFERENCES

Abou-Zeid, A. and Russell, J. S. (1993)
Using data flow diagrams to study communication process between construction project participants.
In: Proceedings of the 5th International Conference on Computing in Civil and Building Engineering, Anaheim, USA

Ackoff, R. L. (1962)
Scientific method. Optimizing applied research decisions.
John Wiley & Sons, Inc.

Ahuja, H., R. L. (1976)
Construction performance control by networks.
John Wiley & Sons, Inc., New York.

Amberg, M. (1996)
A Pattern Oriented Approach to a Methodical Evaluation of Modeling Methods.
In: AJIS Australian Journal on Information Systems, Vol. 4, No.1, September 1996, pp. 3-10

Anon. (1997)
High-level Petri Nets - Concepts, definitions and graphical notation.
Committee Draft ISO/IEC 15909, 2 October, Version 3.4

Aouad, G., Brandon, P., Brown, F., Cooper, G., Ford, S., Kirkham, J., Oxman, R., Sarshar, M. and Young, B. (1994)
Integration of Construction Information (ICON), Final Report
University of Salford, UK.

Appleton. (1985)
Information modelling language IDEF1X.
Appleton Company Inc., US

Augenbroe, G. & Amor, R. (1997)
Project Control in an Integrated Building Design System.
CIB W78 Workshop on Information Technology Support for Construction Process Re-Engineering, IT-CPR-97, Cairns, Australia, 9-11 July, pp 53-66.

Austin S, Baldwin A, and Newton A. (1996)
A data flow model to plan and manage the building design process.
Journal of Engineering Design, Vol. 7, No. 1, pp. 3-25

Austin S, Baldwin A, Li B and Waskett P. (1998)
Development of the ADePT methodology.
Department of Civil and Building Engineering, Loughborough University, Loughborough, UK. (ISBN 1 897911 06 8), (Interim report, unpublished)

Avison, D.E. and Fitzgerald, G. (1995)
Information Systems Development: Methodologies, Techniques and Tools.
McGraw-Hill International Limited, UK (2nd Edition)

Bakkeren, W. (1995)
View type Model for the Structural Engineer for RC Detailing.
TNO Building and Construction Research. (unpublished), Delft, NL.

Baldwin, A. N., Austin, S. A., Hassan, T. M. and Thorpe, A. (1999)
Modelling information flow during the conceptual and schematic stages of building design.
Journal of Construction Management and Economics, Vol 17, pp. 155-167.

Berg von Linde, R. (2000)
A usability study of a construction related IDEF0 model.
In Proceedings of the Third European Conference on Product and Process Modeling in the building and related industries (ECPPM2000), Lisbon, Portugal, September 25-27, 2000.

Björk, B.C (1992)
A unified approach for modelling construction information.
Building and Environment, Vol. 27, No. 2, pp. 173-194

Boman, M., Bubenko, J. and Johannesson, J. (1991)
Conceptual modelling.
Department of computer and systems sciences, Stockholm University. Second (revised) edition.

Booch, G. (1994)
Object-oriented analysis and design with applications. 2nd ed.
Addison Wesley Longman, Inc.

Bubenko, J. Jr (1992)
On the evolution of information systems modelling - a scandinavian perspective.
In: Lyytinen, K. and Puuronen, S. (eds). Computing in the Past, Present and Future. Department of computer science and information systems, Jyväskylä, Finland, pp. 31-66

Budde, R., Kautz, K., Kuhlenkamp, K. and Züllighoven, H. (1992)
Prototyping. An evolutionary approach to system development.
Springer-Verlag, Berlin, Germany.

Chung, E. K. (1989)
A survey of process modeling tools. Technical report No. 7.
The Pennsylvania State University, University Park, PA, US

Clayton, M. J., Fischer, M. A. and Kunz, J.C. (1998)
CAD Prototype Testing: Worked Examples, Demonstrations, Trials, and Charrettes.
In Computing in Civil Engineering, Proceedings of International Computing Congress. ed. K. C. P. Wang. American Society of Civil Engineers, pp. 106-116.

Cox, B. J. (1986)
Object-oriented programming. An evolutionary approach.
Addison-Wesley Publishing Company.

Curtis, B., Kellner, M. & Over, J. (1992)
Process modeling.
Communications of the ACM, 35(9), pp. 75 - 90.

DeMarco, T. (1979)
Structured Analysis and System Specification.
Yourdon Inc., New York.

Eastman, C. M. (1999)

Building product models: Computer environments supporting design and construction.
CRC Press, Fl, US.

Ekholm, A. (1996)
A conceptual framework for classification of construction works.
Electronic Journal of Information Technology in Construction. Vol. 1. http://itcon.org/1996/

Eriksson, J. (1994)
Forskning, utveckling, nytta - Om nytto- och relevansbedömningar av BFR-stödd forsknings och utvecklingsverksamhet. [Research, development, utility - On the assessment of utility and relevance in research & development supported by the Swedish council for building research]
Byggforskningsrådet, NOGA:s skriftserie 1994:12

Fondahl, J. (1980)
Networking techniques for project planning, scheduling, and control.
In Frein J.P (editor): Handbook of construction management and organization.
Van Nostrand Reinhold Company, New York.

Froese, T. (1992)
Integrated computer-aided project management through standard object-oriented models.
Stanford University, PhD thesis

Froese, T. (1996)
Models of Construction Process Information.
Journal of Computing in Civil Engineering, Special section on Data, Product, and Process Modeling. ASCE, Vol. 10, No. 3, July 1996, pp. 183-193.

Gielingh, W. (1988)
General AEC reference model.
ISO TC 184/SC4/WG1 doc. 3.2.2.1, TNO building and construction research, BI-88-150. Delft.

Gosling, J., Joy, B., Steele, G. and Bracha, G. (2000)
The Java language specification. Second edition. Draft.
Addison-Wesley (An imprint of Addison-Wesley Longman, Inc.)

Gustafsson, L., Lanshammar, H. and Sandblad, B. (1982)
System och modell. En introduktion till systemanalysen. (System and model: An introduction to systems analysis)
Studentlitteratur, Lund, Sweden

Halpin, D. W and Riggs, L. S. (1992)
Planning and analysis of construction operations.
Wiley & Sons, Inc. New York

Hannus, M. and Pietiläinen, K. (1995)
Implementation concerns of process modelling tools.
In: Fischer, M.A., Law, K.H. & Luiten, B. (eds.) Modeling of Buildings Through Their Life Cycle. CIB W78 & TG10 Workshop on Computers and Information in Construction, Stanford 21-23 August 1995. Stanford, CA: Stanford University & CIB. P. 449 - 458. (CIB Proceedings, Publication 180).

Hoffner, L. (1997)
Projekteringsprocessen vid egen regiprojekt hos JM Byggnads AB. En modellering och analys av processen och dess ingående dokument. (The design stage of the building process at

JM Byggnads AB. A modelling and analysis of the process and the documents involved).
Master's thesis no. 318. (In Swedish), Department of Construction Management and Economics, Royal Institute of Technology, Stockholm, Sweden.

IAIIFC. (2000)
IFC 2.X Beta Implementers' release. IFC2X_Beta (15-Feb-2000).
International Alliance for Interoperability, Virginia, US.

IAIWEB. (2000)
Home page of International Alliance for Interoperability. (http://iaiweb.lbl.gov/)
International Alliance for Interoperability, Virginia, US.

IDEF0. (1993)
Integration definition for function modeling (IDEF0).
Federal Information Processing Standards publication 183.

IDL. (1994)
Design/IDEF. Interface Languages Manual. Version 3.0.
Meta Software Corporation. Cambridge, MA.

ISO. (1994a)
10303-1 Industrial automation systems and integration - Product data representation and exchange. Part 1: Overview and fundamental principles.
International Organisation for Standardisation, Geneva

ISO. (1994b)
10303-11 Industrial automation systems and integration - Product data representation and exchange. Part 11: Description methods: The EXPRESS language reference manual.
International Organisation for Standardisation, Geneva

ISO. (1995)
The process modelling language EXPRESS-P.
ISO TC184/SC4/WG5 N230. Working draft.

Kagioglou, M., Cooper, R., Aouad, G., Hinks, J., Sexton, M. and Sheath, D. (1998a)
Final report. Generic design and construction process protocol.
University of Salford, UK.

Kagioglou, M., Cooper, R., Aouad, G., Hinks, J., Sexton, M. and Sheath, D. (1998b)
A generic guide to the design and construction process protocol.
University of Salford, UK.

Karhu, V. (1997).
Product Model Based Design of Precast Facades.
Electronic Journal of Information Technology in Construction. Vol. 2. http://itcon.org/1997/1/

Karhu, V. (2000)
Proposed new method for construction process modelling.
International Journal of Computer Integrated Design and Construction. Vol. 2 No 3, pp. 166-182.

Karhu, V. (2001)
A view based approach for construction process modelling.
Journal of Computer-Aided Civil and Infrastructure Engineering. (accepted for publication in August 2001)

Karhu, V. and Lahdenperä, P. (1999)
A formalised process model of current Finnish design and construction practice.
International Journal of Construction Information Technology, Vol. 7 No 1, 1999, pp. 51-71

Karstila, K., Karhu, V. and Björk, B-C. (2000)
Development of process modelling tools for the construction industry.
In: Proceedings of the third European conference on product and process modelling in the building and related industries, Lisbon, Portugal, 25-27 September.

Kartam, S., Ballard, G. and Ibbs, C.W. (1997)
Introducing a new concept and approach to modeling construction.
Journal of Construction Engineering and Management, Vol. 123 No 1, pp. 89-97.

Kelley, J. E. and Walker, M. R. (1959)
Critical-Path Planning and Scheduling.
In: Proceedings of the Eastern Joint Computer Conference, pp. 160-173.

Kochikar, V.P. and Narendran, T.T. (1994)
On using abstract models for analysis of flexible manufacturing systems.
International Journal of Production Research, Vol. 32 No 10, pp. 2303-2322.

Koskela, L. (1992)
Application of the new production philosophy to construction.
Technical report Number 72, Stanford University, US.

Koskela, L. (1995)
On Foundations of Construction Process Modeling.
Modeling of Buildings through Their Life-Cycle. CIB Workshop on Computers and Information in Construction, August 1995, Stanford University, Stanford. CIB proceedings, Publication 180, pp 503 - 514.

Lahdenperä, P. (1995)
Reorganizing the building process. The holistic approach.
Espoo, Finland. VTT Publications 258.

Lindblad, O.I, Sindre, G. and Sølvberg, A. (1994)
Understanding quality in conceptual modelling.
IEEE, March 1994, pp. 42-49

Logic Works (1997)
BPwin v. 2.01. User's Manual and software.
Logic Works, Inc.

Lotus Notes (1995)
Lotus Notes. Programmer's guide. Part 1 and part 2. Release 4.
Lotus Development Corporation.

Lotus Notes (1999)
The Architecture of Lotus Notes.
Technical paper. Number 114654. Lotus Development European Corporation.

LotusScript (1995)

Language reference. LotusScript. Cross-Product BASIC Scripting Language. Release 3.
Lotus Development Corporation.

Luiten, G. (1994)
Computer Aided Design for Construction in the Building Industry.
Ph.D. thesis, Delft University of Technology.

Luiten, G., Froese, T., Björk, B.-C., Cooper, G., Junge, R., Karstila, K., and Oxman, R. (1993)
An Information Reference Model for Architecture, Engineering, and Construction.
Proceedings of the First International Conference on the Management of Information Technology for Construction, World Scientific, Singapore, 1993, pp 391-406.

Lundequist, J. (1995)
Design och produktutveckling. Metoder och begrepp. [Design and product development. Methods and concepts.]
Studentlitteratur, Lund, Sweden (in Swedish).

Mayer, R. J., Menzel, C. P., Painter, M. K., deWitte, P. S., Blinn, T. and Perakath, B. (1995)
Information integration for concurrent engineering (IICE): IDEF3 process description capture method report.
Knowledge based systems, Inc. Interim Report, College station, TX, US.

Melan, E.H. (1982)
Process management: methods for improving products and service.
McGraw-Hill, Inc.

Melin, J.W and Whiteaker, B. (1981)
Fencing a bar chart.
Journal of the Construction Division, ASCE, Vol. 107, pp. 497-507.

Meredith, J. (1993)
Theory building through conceptual models.
International Journal of Operations & Management, Vol. 13, No 5, pp. 3-11

Merendonk, P. van and Dissel, D. van. (eds.). (1989)
Bouw informatie model; versie 3.1. IOP-Bouw.
App., Rotterdam, Netherlands.

MS Project. (1998)
Microsoft Project 98. User's manual and software.
Microsoft Corporation.

Nederveen van, S. (1995)
View type model for the architect in the global design stage.
TNO Building and Construction Research, Delft, Netherlands (unpublished)

Nijssen, G. and Halpin, T. (1989)
Conceptual schema and relational database design, a fact oriented approach.
Prentice Hall, London, UK.

Ogden, C. K. and Richards, I. A. (1972)
The Meaning of Meaning.
Kegan Paul, London.

OMG (2000)
OMG Unified Modelling Language Specification. Version 1.3.
Object Management Group, Inc., US.

Pietiläinen, K. and Heinonen, R. (1995)
STARGEN – Generic modelling of construction processes.
Working paper, 1995 (unpublished manuscript)

RT. (1995)
Task lists. RT 10-10575. 1995. Rakennuttamisen tehtäväluettelo [Scope of client's work in building construction], RT 10-10576. 1995. Arkkitehtisuunnittelun tehtäväluettelo [Scope of work in architectural design], RT 10-10577. 1995. Rakennesuunnittelun tehtäväluettelo [Scope of work in structural design], RT 10-10579. 1995. Talotekniikan suunnittelun tehtäväluettelo [Scope of work in building services design], RT 10-10580. 1995. Geosuunnittelun tehtäväluettelo [Scope of work in geotechnical design].
Rakennustieto Oy, Helsinki, Finland. (in Finnish).

Sanvido, V., Khayyal, S., Guvenis, M., Norton, K., Hetrick, M., Al-Muallem, M., Chung, E., Medeiros, D., Kumara, S. and Ham, I. (1990)
An integrated building process model. Technical Report no 1. CIC Research Program.
The Pennsylvania State University. University Park, PA, US.

Tanhuanpää, V-P. and Lahdenperä, P. (1996)
Rakentamisprosessin malli. Toteutumatiedot rakennushankkeesta [Construction process model. Actualized data of an office project]. Espoo. Technical Research Centre of Finland, VTT Tiedotteita – Meddelanden – Research Notes 1768.(in Finnish)

Wakefield, R. and Damrianant, J. (1999)
Petri Net Modelling. A tool for construction process reengineering.
International Journal of Computer Integrated Design and Construction. Vol 1, No 2, pp 20-34.

Visio (2000)
Visio 2000 - International version.
Visio Corporation. http://www.visio.com/.

Viswanadham, N. and Narahari, N. (1992)
Performance Modelling of Automated Manufacturing Systems.
Prentice-Hall, Englewood Cliffs, New Jersey, USA.

Volkhard, F. (1989)
Planung und Steurung komplexer Bauprozesse durch Simulation mit modifizierten höheren Petri-Netzen.
Kassel, Germany. PhD thesis (In German)

Zhong, Q., Mathur, K. and Tham, K. (1994)
A construction process model by using Design/Idef.
CIB W78 workshop on computer integrated construction, Helsinki, Finland, August 22 - 24, 1994. Helsinki: VTT Building Technology & International Council for Building Research Studies and Documentation CIB.

LIST OF FIGURES