CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

Automating Communication in Civil Engineering

by

Gijsbertus T. (Bart) Luiten
and Frits P. Tolman

CIFE Working Paper
Number 38

May, 1995

Stanford University

If you would like to contact the authors please write to:

c/o CIFE, Civil Engineering,
Stanford University,
Terman Engineering Center
Mail Code: 4020
Stanford, CA 94305-4020

Automating Communication in Civil Engineering

By Gijsbertus T. (Bart) Luiten[1] and Frits P. Tolman[2]

ABSTRACT: Automating communication is a powerful tool for realizing organizational changes currently pursued by building companies. Until now however, the building industry did not go further than automating the traditional means of communication. This paper presents a new approach to open, high-level, computer-interpretable communication between project participants. "Open" means independent of the participants and their computer systems, "high-level" that no human interpretation is required for handling information and knowledge, and "computer-interpretable" that no human interaction is needed for data transfer. The STEP product-modeling approach provides a basis for this kind of communication, but it needs an extension to project-modeling and more scope layers. The authors have developed such an extended approach for integrating design and construction in building projects and present here the need for and the background of this approach, a way to ensure (computer) integration in ever-changing project teams, and a test case.

INTRODUCTION

In the last decades, the building industry (like all industries) has faced an increase in demands from society, which demands higher quality, more variation, higher complexity, shorter lead times, lower costs, a lesser burden on the environment, and better working conditions. This increase in demands is only likely to continue (ARTB 1993).

One of the building industry's answers to these demands might be to make better use of information and knowledge that is available. The knowledge becomes available from prior experience, feedback from realized products, new research, new materials, new technologies, etc. In addition, newly introduced computer applications generate large amounts of project, company, and general building information. Unfortunately, because communication (i.e., knowledge and information transfer) between project participants did not evolve at an equal pace, islands of knowledge and information have resulted.

A good example of problematic communication is the interaction of design and construction. Most currently used means for communication do not support an improvement in this interaction. First, the relation between design and construction information is still not made explicit; the relation still exclusively exists in the heads of the professionals involved. Second, communication during design and construction management are just partly supported because only the results of these processes are communicated. Interaction between design and construction is thus hindered by not communicating the reasons behind and the relations between design and construction information. The research field of Design for Construction (DfC) tries to improve this interaction and is thus equivalent to Design for Manufacturing (Boothroyd and Dewhurst 1987) in the mechanical industry.

1 Postdoctoral Fellow, Dept. of Civ. Engrg., Stanford University, Stanford, CA 94305-4020, and Dept. of Civ. Engrg., Delft University of Technology, Delft, The Netherlands; Ph.: 415 723 9340; Fax: 415 725 8662; E-mail: Luiten@Cive.Stanford.EDU

2 Prof., Dept. of Civ. Engrg., Delft University of Technology, and TNO-Building and Construction Research, Computer Integrated Construction, PO Box 49, 2600 AA Delft, The Netherlands; E-mail: Frits.Tolman@Bouw.TNO.NL

Since design and construction are increasingly, and successfully, supported by computers, it only seems logical to support their communication with computers as well. Although Computer-Integrated Construction (CIC) has been studying this "integration"[1] for quite awhile (Fischer 1989), CIC (and thus computer-aided DfC) is still a technology in its infancy and by far the building industry has not mastered it. In this paper we outline an approach that will help computer-aided DfC to mature. This approach was developed during Ph.D. research at the Delft University of Technology (Luiten 1994).

In the paper we first describe some organizational changes that currently take place in the building industry and require improved communication. This results in requirements on computer-aided communication. The STEP product-modeling approach seems the most promising. We extended this *product*-modeling approach to *project*-modeling to support the interaction between design and construction and applied the new approach to precast concrete structures. The paper ends with conclusions about the future of automation and integration in the building industry.

ORGANIZATIONAL CHANGES AND COMMUNICATION

New forms of organization have appeared as the building industry has made efforts to cope with the growing demands of society, for example, the growing influence of suppliers, design-and-build contracts, automation of construction on site, Total Quality Management, and integration of life-cycle stages. New forms from other industries that the building industry has gradually taken over are Design for Manufacturing, lean production, concurrent engineering, and business re-engineering.

No form of organization suits every business goal, however; for example, there is a growing influence of suppliers on the design that results in designs that better suit their construction resources. Often, suppliers even design their part of the building completely, of course, within the conditions set by the client and the contractor. Lean production (Womack et al. 1990) is another example; it concentrates on the value-adding activities in the two main transformation activities: (1) design and production management (the main *information* transformation activities) and (2) production (the main *material* transformation activity)and aims at eliminating other activities and optimizing the main activities on the project level (not on the level of individual companies). In other words, it aims at eliminating "waste".

Fig. 1 presents a simplified IDEF0 diagram (SofTech 1981) that shows the two main transformation activities and some information and knowledge flows in building projects. The message conveyed by the figure is that feedback of constructability and performance knowledge are, in general, not supported electronically. And since organizational changes, by definition, relate to transformation activities and/or flows, improved communication between the activities results in improved business performance.

[1] Fischer (1989) defined integration as "the continuous and interdisciplinary sharing of goals, knowledge, and information among all project participants."

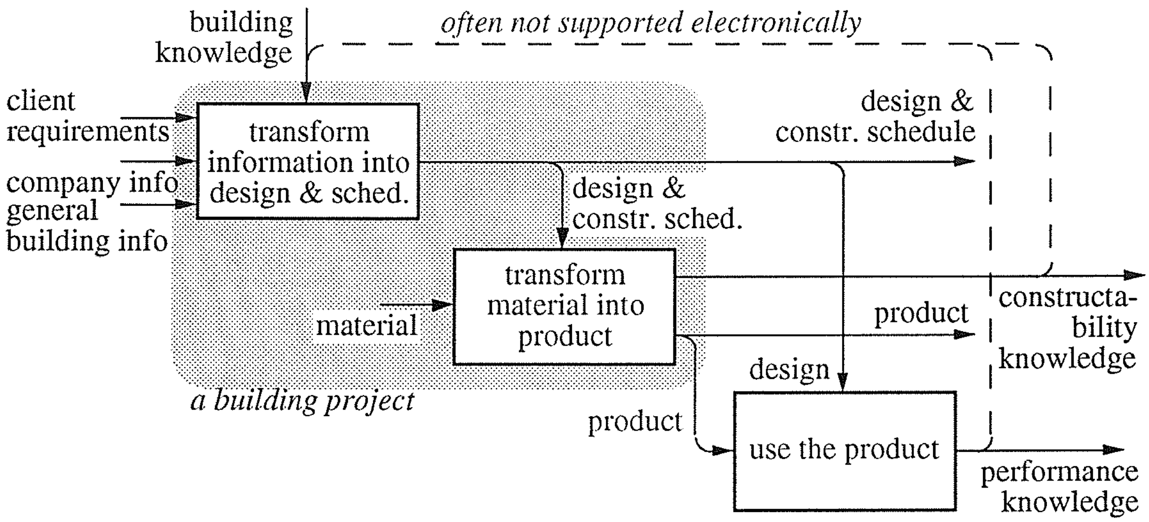

Fig. 1. The two main transformation activities in a building project.

COMPUTER-AIDED COMMUNICATION

Since improving communication seems essential to organizational changes, the building industry is already pursuing computer-aided communication. This section argues that the most commonly-used means for sharing knowledge and information are not suitable for computer support and that open, high-level, computer-interpretable communication is required. The STEP product-modeling approach provides such communication.

Open, high-level, computer-interpretable communication

Since the introduction of computers into the building industry, design and construction processes have been supported with numerous design, scheduling, and cost-estimating applications. However, integration of design and construction has not really changed: project partners share goals, knowledge, and information in the same way as before. Computer applications still produce traditional documents, such as specification documents, detail drawings and schedules. Humans have to interpret these documents, whether paper or electronic, before the next application can use them (see Fig. 2), but human interpretation is slow, costly, and prone to errors. Moreover, knowledge mainly resides in the heads of experienced professionals, which hinders its transfer and full use.

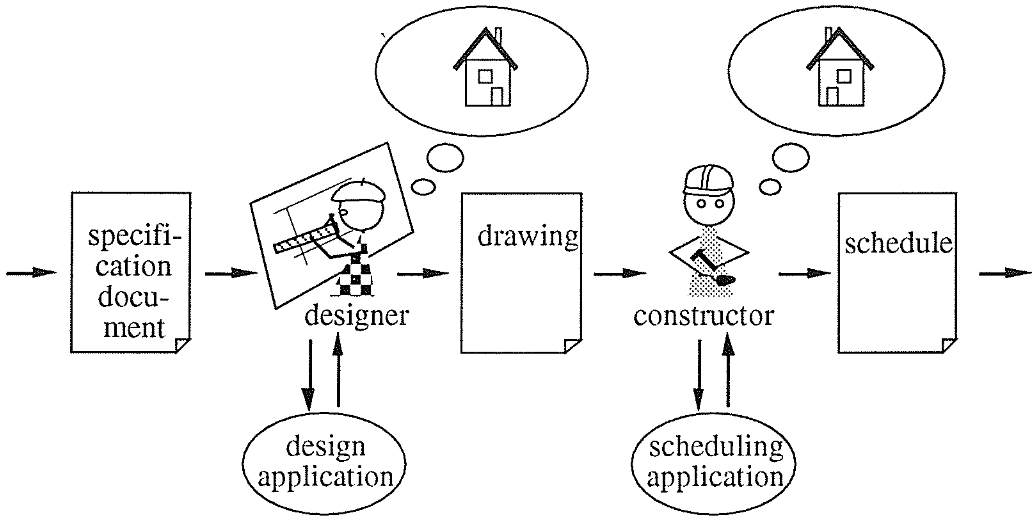

Fig. 2. Communication with documents as often seen in current practice.

Computer-aided DfC aims at improving the integration of design and construction by automating communication between applications used for these processes. Most computer applications, however, cannot interpret information from electronic documents as humans can. Therefore, the relations between design and construction information and the reasoning behind design and construction choices have to be made explicit in communication. To realize this, applications need information with more explicitly defined meaning, that is, information at a higher semantic level. High-level communication in a computer-interpretable format minimizes the need for human interpretation and thus enables integration of applications (see Fig. 3).

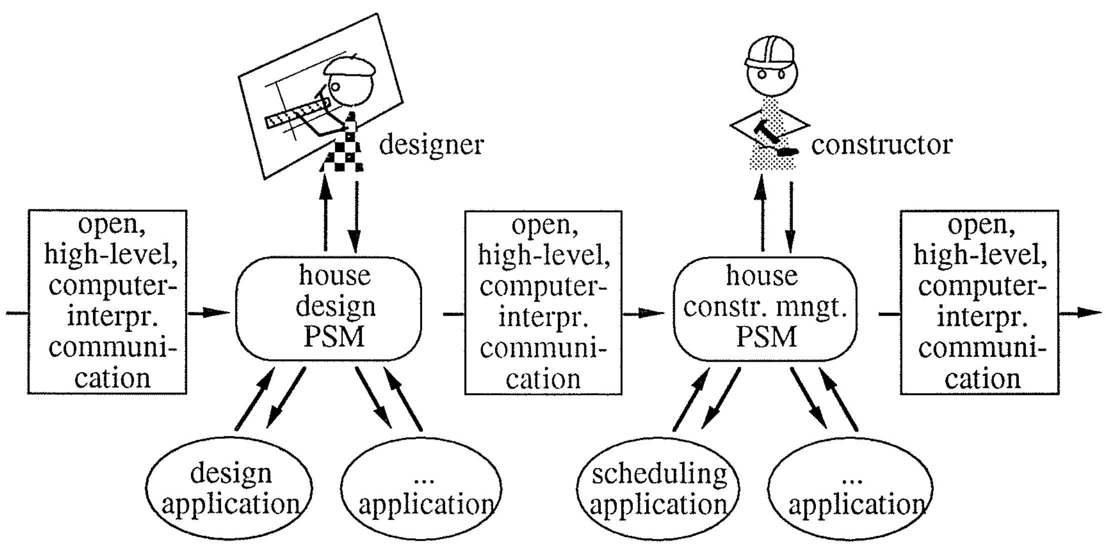

Fig. 3. Open, high-level, computer-interpretable communication between Product Specific Modelers (PSM).

To manage information and knowledge transfer, companies should have so-called Product Specific Modelers (PSMs). A PSM manages information and knowledge transfer between internal computer applications and uses transfer mechanisms for outside communication.

Therefore, a PSM should "know" the meaning of the information it handles. For example, to generate the input for a structural analysis application, structural engineers currently interpret the design and filter out the load bearing objects manually. If engineers are to automate this process, they have to be able to define their knowledge rules with the terminology and semantic level they use in practice. The rules should be expressed in terms such as *beam, load bearing, is supported by*, etc. Only with this high semantic level is a PSM "aware" of the function of the objects in a design and can it filter the information. In addition, the applications that are connected to the PSM should use the same semantic level: computers are only able to derive low-level information from high-level information, not the other way around. For example, it is possible to derive 2-D drawings from a high-level product-model of a beam, but it is impossible (or at least very difficult) to derive such a product-model from 2-D drawings. If two-way communication is to exist between applications and PSMs, both should use the semantic level of professionals in practice.

Besides technical feasibility of computer-interpretable communication, its economic feasibility is also a key issue. Such feasibility requires an open environment, independent of the computer applications used at the companies involved. The standardization of communication is one way of realizing this. Unfortunately, the building industry is too fragmented to standardize on the level of building projects; the configuration of building teams changes too often. Therefore, industry standards are needed, supported by all participants, with all their applications, and preferably on an international scale. However, because of the fragmented and project-oriented nature of the building industry, internationally accepted, industry-wide standards are very hard to realize.

Product-modeling

In the seventies and eighties, CAD systems were the central points for integration and thus for standardization. Models for the exchange of shape information grew more and more sophisticated, from wire frames and surface models, via solid models, to relational reference models. Nevertheless, it turned out that topology and/or geometry could not be used as the central point of integration because:

- the shape of a product is not stable during the design process,
- there is often already information before a shape is chosen, and
- different participants in the project often use different shape representations.

Therefore, geometric modeling evolved to *product-modeling*, in which the building model is described with the semantics used in practice. A product-model contains product information such as the properties of its parts. This includes geometry, topology, decomposition, material, behavior, etc. The product-model contains information for all life cycle stages and for all participants in the building process. Geometric information is not the core anymore, but one of the properties. Traditional documents, such as drawings, can ultimately be derived from the product-model.

Product-modeling builds on the strengths of other integration concepts. Classification (and coding) of building products, activities, and construction resources has proven its value in practice; its division into categories and its terminology can be a basis for product-models (Björk et al. 1989). Product-modeling is also the product of an evolution of feature modeling for mechanical parts to the more complex nature and larger scale of building products. Moreover, it offers the high-level semantic required to define knowledge rules. With its high-level, computer-interpretable communication, product-modeling forms a solid basis for computer-aided DfC.

ISO-STEP product-modeling approach

The major standardization effort in product-modeling today is ISO-STEP, the international STandard for the Exchange of Product-model data (ISO/TC184 1993). The purpose of this international standard is to "specify a format for the unambiguous definition and exchange of computer-interpretable product information throughout the life of a product." STEP's development started in 1984 and builds on the Product Data Exchange Specification (PDES) (Smith 1986).

STEP developed view and application independent languages for specifying information structures and exchanging information. The value of these languages can be measured by their widespread use, even outside the STEP community. STEP developed the lexical modeling language EXPRESS for specifying information structures. A subset of EXPRESS has a graphical presentation: EXPRESS-G. Other graphical presentations that may be used in STEP are NIAM (Nijssen and Halpin 1989) and IDEF1X (Appleton Company 1985). Because EXPRESS schemata are independent of computer-systems and have a standardized format, they are easily exchangeable. An EXPRESS schema defines a *conceptual model* with object-types and attributes, which denote sets of objects with common properties. A database can be structured according to a conceptual model. In this database, information about one particular product (e.g., the White House) of that type of product (e.g., governmental buildings) can be stored. Information in the database, i.e., the *product-model*, can be exchanged with a file in a standardized format or via standardized access to the database.

STEP standardizes conceptual models with different scopes, using EXPRESS. There are two types of conceptual models:

- integrated resources[1], and
- application protocols (APs).

Integrated resources define related parts of conceptual models that can be used in other conceptual models. There are two types of integrated resources: generic resources and application resources. Generic resources are application-type independent. Examples are geometry and topology representations, product structure and configuration management, dimensions and tolerances, and form features. Application resources are applicable to specified types of applications, such as drafting, finite element analysis, and presentation. An Application Protocol (AP) defines the information structure for one or more applications. In this context the definition of "application" is rather broad: "a group of one or more processes creating or using product data." As a result, there are application protocols for general processes, such as 2-D-drafting, but also for specific processes, such as ship or road design.

Despite some unresolved issues, many integrated resources and APs are already under development. The development of APs is somewhat bottom-up, driven by the needs of computer applications in question. In the near future dozens of new STEP resources and APs will be submitted for evaluation and integration.

BUILDING PROJECT-MODEL FOR DFC

As shown above, the STEP product-modeling approach provides the open, high-level, computer-interpretable communication needed for industry-wide *product* information and knowledge sharing. In this section we introduce the Building Project-Model (BPM), which

---

[1] STEP uses the term *resource* to denote a conceptual model that can be used in other conceptual models. These STEP resources should not be mixed up with models of the *construction resources* that are used at the building site. To clarify the reading, we will use *construction resources* when we talk about the building site resources and only *resources* when we talk about conceptual models or their implementation.

extends the product modeling approach to project modeling for DfC. We discuss the two main abstraction mechanisms of the BPM--product-activity and concretization--that make relations between design and construction management information and the reasoning behind decisions explicit. The section ends with a prototype implementation.

Models with different scopes

Gielingh proposed the General AEC Reference Model (GARM) (1988), which was one of the first efforts made to develop a comprehensive model for the building industry. The GARM can be used to model a product and its characteristics in different life cycle stages. It also contains information structures that support decomposition, evaluation and selection of alternative solutions, connectivity, shape representation, and classification of building aspects. Originally, models like the GARM were intended to be instantiated directly as product-models that represent one particular product. However, soon the need for more *product-type* specific conceptual models merged.

Tolman (1991) proposed the introduction of *product-type models* structured according to a common reference model. Product-type models contain information structures valid for one family of products only, for example, for a type of precast concrete element. The terminology used to define a product-type model closely follows the terminology commonly used for product descriptions in practice. For example, a product-type model for precast concrete elements follows the terminology of engineers at a precast concrete element supplier. Tolman uses reference models to structure these product-type models uniformly, which results in three types of conceptual models, each with its own scope:

- general resources,
- reference models, and
- product-type models.

General resources correspond to STEP integrated resources and are applicable to several types of industry. A reference model contains information structures applicable to one type of industry. For example, the GARM can serve as a reference model for the building industry. A product-type model, which corresponds to a STEP Application Protocol, is a conceptual model for a specific type of product.

After this interpretation, the GARM has successfully been used as a reference model for product-type models, proving the value of this architecture for conceptual models. A reference model ensures interoperability between product-type models, because product-type models based on the same reference model can now be used in one product-model. In addition, it enables the development of Project Specific Modelers (PSMs) that are easy to integrate, because they are based on similar information structures. Another big advantage is reusability of software: software based on one reference model can be used in many product-type models.

We applied this interpretation of layered models with different scopes to our research. We focused on a reference model that improves communication between design and construction and called it the Building Project-Model (BPM). The BPM not only models product information, but also other project information, such as activity and resource information. The BPM refers to existing general resource models for the definition of shape and material we applied it to a project-type model for precast concrete structures in the test case (see next section).

Abstraction mechanisms

A comprehensive reference model such as the BPM should emphasize essentials and suppress irrelevant details. In conceptual modeling research this is called *abstraction* (Smith and Smith 1977). Two abstraction mechanisms are relevant for DfC: (1) product-activity and (2) concretization. The product-activity mechanism emphasizes the relations between product, activity, and construction resource information in a building project. The concretization mechanism emphasizes the information needs during problem-solving processes such as design and construction management. The following two subsections describe the two mechanisms, their models, and their use.

Abstraction mechanism 1: Product-Activity

If we are to integrate design and construction, we have to relate product, activity, and construction resource information ànalogously to the way professionals talk and think about it. In current practice, these types of information can be found in design drawings, activity schedules, and construction resource plans. The relations between these three types of information often exist only in the minds of the professionals involved; in other words, the relations exist only *implicitly*. Normally different persons perform design and construction management, and they may interpret the design, the schedules, and their relations differently. As long as the relations remain implicit, such differences in interpretation may exist.

Until recently, product, activity, and construction resource information were modeled separately, e.g., by using the STEP *product*-modeling approach and by using the IDEF0 *activity* analysis method. With the BPM we want to relate the three types of information in one conceptual model. Fig. 4 models products, activities, and construction resources as subtypes of project objects. The figure contains a short legend of NIAM (Nijssen and Halpin 1989), the graphical information modeling technique used in this paper. The figure shows that a project consists of project objects, which are either products, activities, or construction resources.

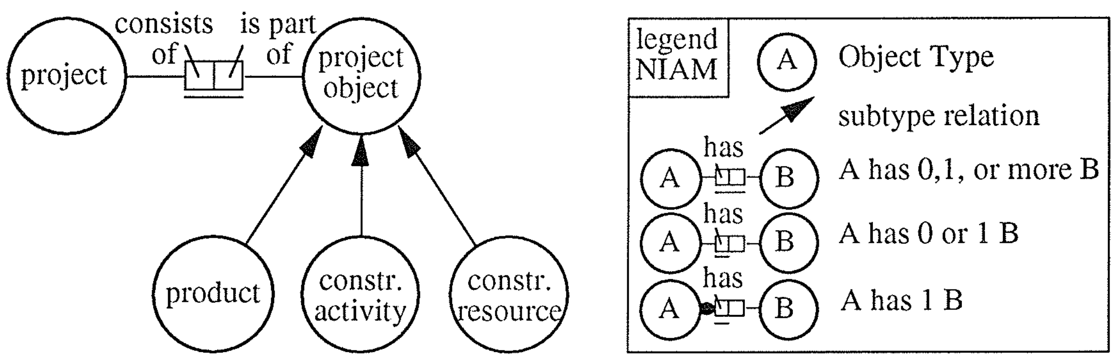

Fig. 4. Starting point of the product-activity model in the BPM: a project consists of products, activities, and construction resources.

Fig. 5 models the object-types and relations by combining observations in practice, Von Wright's logic of change (1963), the ideas behind activity analysis methods such as IDEF0 (SofTech 1981), and other reference models (Björk 1991; Froese 1992; Gielingh and Suhm 1993; Luiten et al. 1993). The figure models the relations between the product, activity, and construction resource information explicitly. It shows that both products and construction resources have construction states, which are realized by construction activities. Construction states succeed each other. A construction activity is performed by a constructor (a general contractor, sub-contractor, or supplier) who should be able to perform that activity. An activity uses construction resources, which should be at the disposal of the constructor.

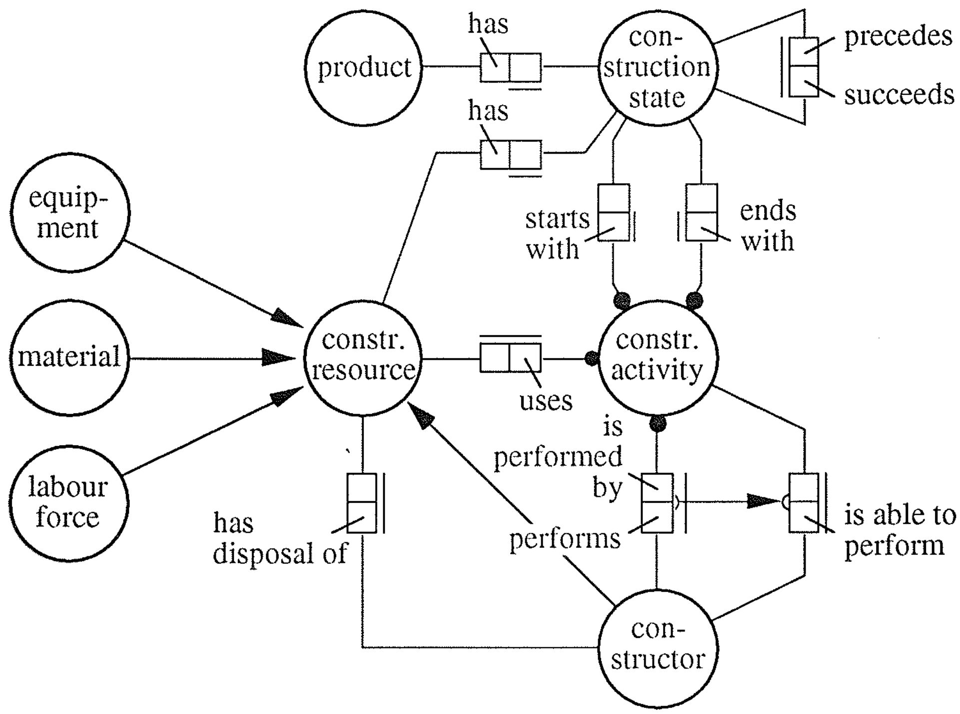

Fig. 5. The product-activity model in the BPM: relating products, activities, and construction resources.

The three main types of information in building projects have characteristic properties. For products, the characteristic properties are shape and *material*, which we can define using the STEP resource models. For activities, the characteristic property is *time*, such as early start, late finish, duration, and float. For construction resources, the characteristic property is *cost*. Subtypes of construction resources are labor forces, materials, and equipment, or compositions of these three basic construction resources. Examples of these compositions are work crews, or even the constructor companies, such as (sub-)contractors and suppliers.

Modeling product, activity, and construction resource information explicitly in the product-activity model contributes to communication between design and construction in several ways. The product-activity model might support the forward exchange of information from design to construction using one, gradually growing, shared project-model. First, the designer proposes a product and stores it in the project-model. This designed product is the starting point for the planner, who then defines the successive states of the product and the construction resources during construction and adds them to the project-model. Next, the planner defines the construction activities needed to reach these states and the required construction resources. The constructor acquires the construction resources and performs the activities, which results in the product. The project-model can also be used to exchange feedback information from the constructor to the designer. For example, the constructor can send the designer a proposal for an improved design of a part of the building.

In addition, the product-activity model might support types of communication that are rarely seen in today's practice. For example, sometimes a design choice implies a certain construction method. With the product-activity model, the designer can add this information to the project-model and transfer it to the constructor. Constructors can also use the model to exchange information about available construction resources to designers. Another new possibility is the exchange of the building's construction states that have to be taken into account during design. For example, a structural engineer often takes into account the loading cases that occur during transportation and assembly. Finally, with the product-activity model it is easier to switch between design and construction, and it is easier to make (preliminary) schedules and cost estimates in early phases of design. These results can be used to see (and react to) the consequences of design choices quickly.

Abstraction mechanism 2: Concretization

Concretization, the second abstraction mechanism in the BPM, supports communication during design and construction management. These processes can be seen as problem-solving processes: designers must find a solution for the client's functional requirements and construction managers have to find a solution if the design is to be realized on time and within budget. The concretization mechanism supports the information needs of a problem-solving process by modeling and relating the evolving states of information during this process.

Most problem-solving strategies require a clear *distinction between problem and solution.* First the problem has to be specified, with criteria to be fulfilled by the solution. Then a number of alternative solutions is proposed and worked out in such detail that the alternatives can be compared to the criteria and to each other. Finally, one of the alternatives is selected and realized. Afterwards, the realized solution can be compared with the criteria specified in the beginning. During the process an object (i.e., a product, an activity, or a construction resource) becomes increasingly concrete. Generally, objects occur in three evolutionary states of concretization:

- *as required* objects (representing function),
- *as proposed* objects (representing solution), and
- *as realized* objects (representing final results).

When problems are large, a second problem-solving principle is often applied: *divide-and-conquer*. Here the main idea is that a large problem can be solved by dividing (or decomposing) it into sub-problems. Normally, the sub-problems are related to each other.

The two principles can be combined with each alternative solution decomposed into sub-problems. Criteria for the solution of these sub-problems can be derived from criteria of the main problem. For each sub-problem an alternative solution is proposed, which can again be divided in sub-sub-problems. This approach can be applied until the problem is small enough to be solved. On the lowest decomposition level solutions are pre-defined and can be specified by parameter values.

Fig. 6 shows how the information needed for the two problem-solving principles can be modeled in the concretization model. A required object is fulfilled by a proposed object. This proposed object decomposes into required objects that are related to each other. The object that is finally realized should conform to the proposed object (and thereby fulfill the required object). This model uses the same interpretation of the life cycle stages of the GARM (Gielingh 1988) as used in the IMPPACT reference model (Gielingh and Suhm 1993). The concretization model applies to products, activities, and construction resources. When applied to products, it supports the function-form-behavior paradigm (Howard et al. 1992) and explicitly relates design intent to chosen form.

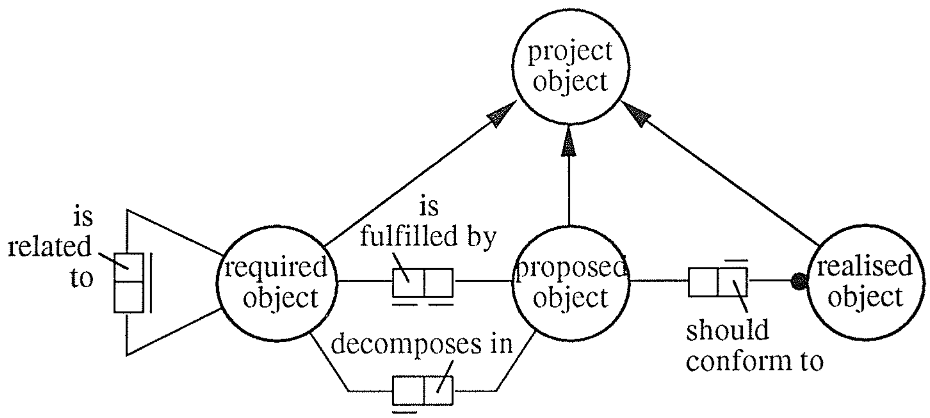

Fig. 6. The concretization model in the BPM: supporting information needs in problem-solving processes.

In contrast to currently used means of communication, which focus on chosen solutions, the concretization model enables the exchange of requirements, solutions, and their relations. It is therefore more flexible with new construction techniques, which can be added easily to the system. It also makes it possible to define (and relate) tasks of participants in a work team, and to relate successive phases in a project. This could enable a shift of some design tasks to construction management, for example, to suppliers who design their part of the building. In addition, the same means of communication can now be used during less detailed, early phases as well as detailed, later phases of a project. Moreover, because the intent of a design is communicated, constructors know the reasons behind design decisions and thus are able to propose alternatives that still correspond to the requirements, but also conform better to their construction possibilities.

Prototype implementation of the BPM

In (Luiten 1994) we described requirements for a project-modeling environment by identifying the needs of the developers of the project-type models, the programmers that implement these models, and the end-users that apply the models in practice. In close cooperation with TNO-Building and Construction Research, we developed a prototype implementation that fulfills some of these requirements (Luijten and Luiten 1993). and called it PMshell, which stands for Project-Modeling environment. For the first version of PMshell we chose the object-oriented language Eiffel[1] (Meyer 1988), because the conceptual modeling notions of object-types with attributes correspond closely to the object-oriented notions of classes with properties and behavior.

With PMshell, developers and programmers of conceptual models are able to use a graphical user interface (based on NIAM or EXPRESS-G) to define information structures and a menu-driven interface to define behavior to manipulate information. PMshell deals with conversion between modeling languages and database management, and provides tools to relate conceptual models with different scopes and formalize (constructability) knowledge rules. With these features, PMshell can be seen as a CASE[2] tool for the development of integrateable Project Specific Modelers.

The end-users in practice can use PMshell to generate information about one particular building project by applying the conceptual models implemented by the developers. With

[1] TNO's current version of PMshell is implemented in C^{++}.

[2] CASE = Computer-Aided System Engineering

PMshell they are able to generate and manipulate project information, exchange it, and apply (constructability) knowledge rules to it.

Fig. 7 shows the layered architecture of the modules of the implementation of the BPM in PMshell. It shows the modules in boxes and their relations by arrows. Modules implement conceptual model and are structured in layers that correspond to the scope layers proposed by Tolman (1991). The kernel layer is provided by PMshell and deals with general functionality such as database management, language conversions, and generating documentation. The other modules inherit this functionality. We discuss each layer and its modules briefly below.

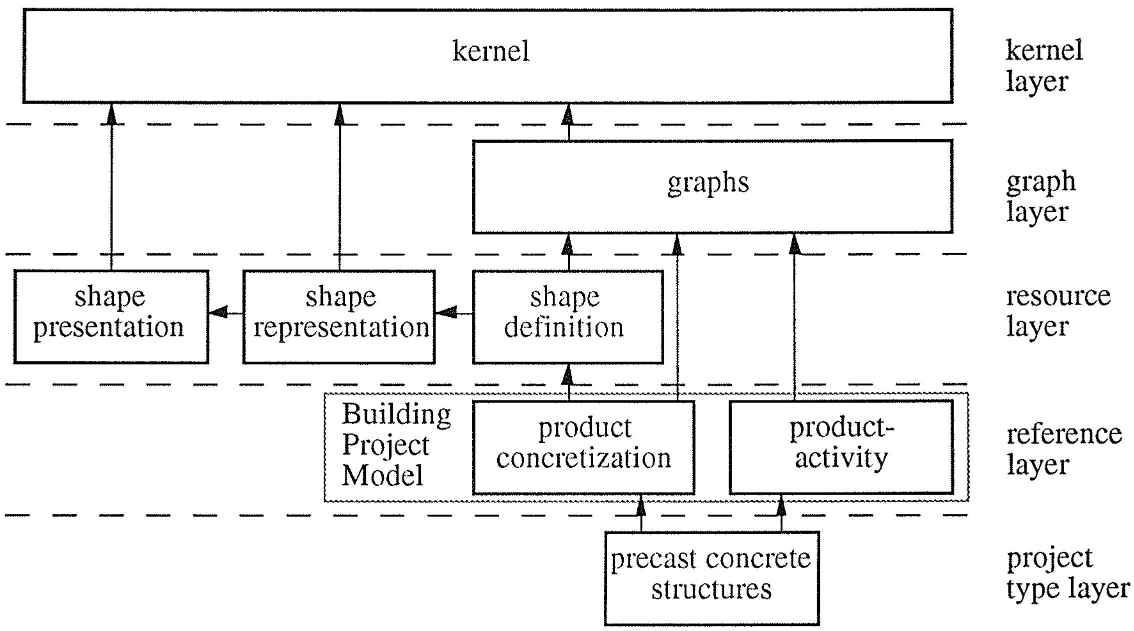

Fig. 7. Layered architecture of the computer implementation of the project-modeling approach developed.

The graph layer contains general graph functionality that can be used as a basis for the implementation of network-like models, such as the decomposition tree of a building, the activity network in a schedule, or the organizational network in business relations. Fig. 8 shows the basic elements of a graph: a graph has nodes, connected by directed links (Sowa 1984). Functionality implemented in the module is, for example, finding the shortest path in a graph, determining connectivity between nodes, and determining sub-graphs.

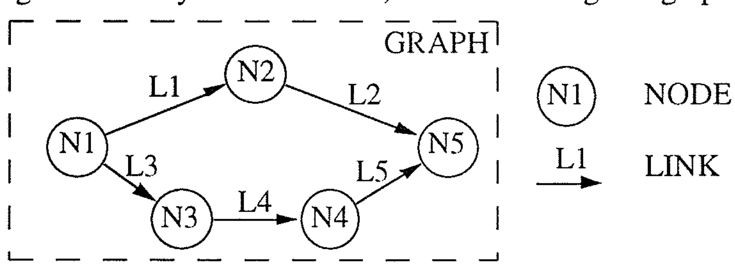

Fig. 8. Graphs, a basis for the implementation of many network-like models.

The resource[1] layer focuses on shape. Shape is defined in three modules: definition, representation, and presentation (as described by Tolman and Gielingh (ISO/TC184 1989)). The *definition* defines a shape independent of the computer applications used. For example, the shape of a steel beam can be defined as IPE400 with a length of 3.00 meters. The *representation* of a shape contains part of this information in a format suited for computer manipulation. For example, the representation of the shape of the steel beam in a CAD system can be a B-rep with vertices, edges, faces, a volume, and topological relations. The CAD system can, for example, calculate the volume of the beam. The *presentation* of a shape contains part of the information in a format suited for human interpretation, e.g., a 3D picture on a CAD screen, or (electronic) 2D drawings of top and side views. For communication, ideally only the definition needs to be exchanged, because computer applications can derive the other on the fly. Our implementation of these three modules builds on work of Willems (1993). The shape representation model uses graph functionality to model topological networks.

The reference model layer implements the BPM in two modules: product concretization and product-activity. The product concretization module implements the concretization relations (as shown in Fig. 6) for product information. It uses graph functionality twice: (1) to model the decomposition network of required and proposed products and (2) to model the network of related required products that are part of one proposed product. It uses shape definition to model the shape of products. It contains functionality to build the required-proposed network easily and to derive the shape of products by composing the shapes of their parts. The product-activity model contains functionality to build a model as corresponding to Fig. 5. It uses graph functionality to model the activity network.

The project-type layer specializes the BPM modules for different types of projects. To illustrate this, we implemented a project-type model for a type of precast concrete structures, as we discuss in the next section.

EKON, A TYPE OF PRECAST CONCRETE STRUCTURES

To assess its practical value, we tested the developed approach--that is BPM, PMshell, and BPM's implementation--in close cooperation with Spanbeton. Spanbeton is a Dutch supplier of precast concrete elements that developed EKON, a new structural system for office buildings. We selected the EKON system for our test because of its high level of standardization and relatively simple elements (see Fig. 9). We focused on preliminary design and assembly on site. This section describes a project-type model for EKON, an implementation, and a test case.

[1] These resources denote general conceptual models that can be used in other conceptual models, not models of construction resources.

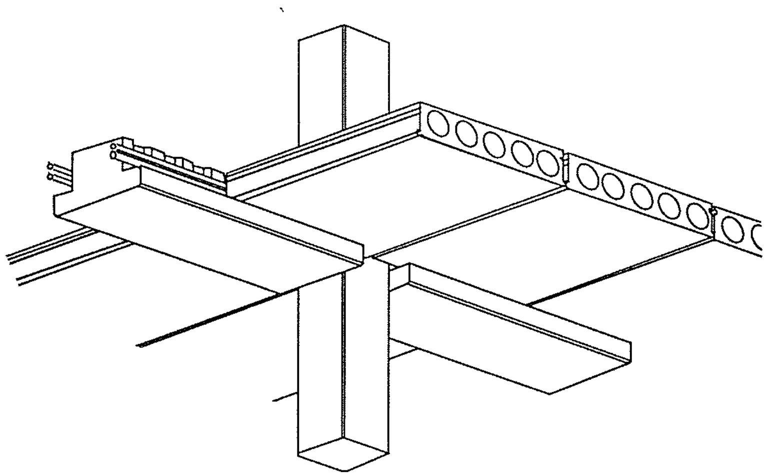

Fig. 9. **The EKON precast concrete structural system (Bennenk and Boom 1992).**

Project-type model

We developed a EKON project-type model as a specialization of the BPM. The product part of the model is a specialization of the concretization model of Fig. 6. To model the structural system and its parts we followed the preliminary design process as closely as possible, using a top-down modeling approach. In the model, a load bearing structure is decomposed in structural areas, each of which can be fulfilled by a different structural system, e.g., in-situ cast concrete or the EKON system. The EKON system is detailed further: it decomposes into floor fields and column rows (Fig. 10). A floor field decomposes in three steps into beams, hollow-core floor slabs, and joints. A column row decomposes into columns and joints. For the assembly process on site we modeled the construction states of the structure, the construction activities, and the construction resources using the product-activity model of Fig. 5.

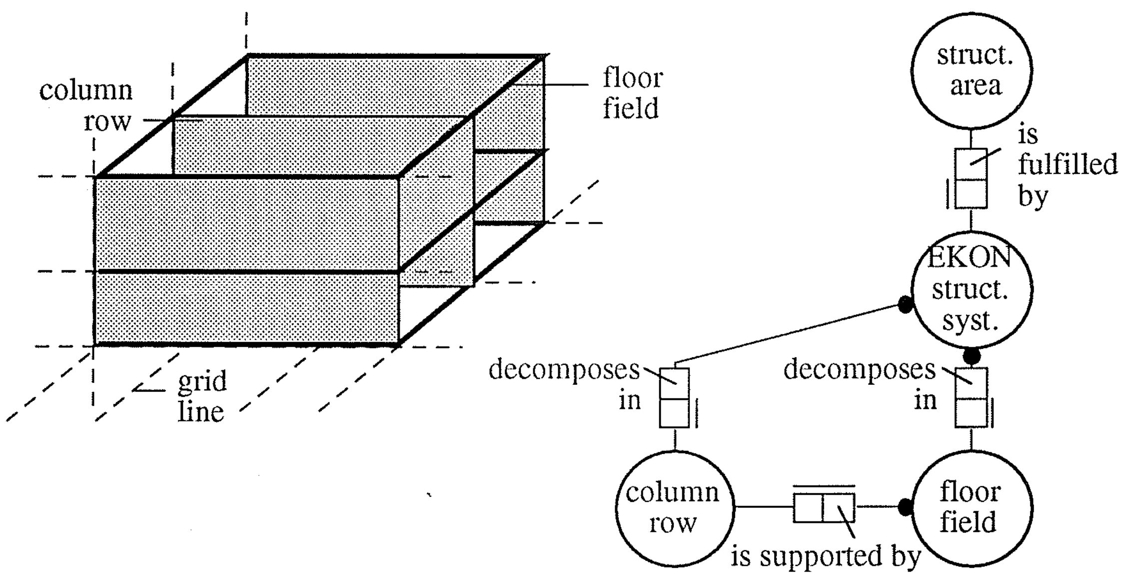

Fig. 10. **Part of the project-type model for EKON structures.**

Implementation

We used the BPM implementation in PMshell to implement the EKON project-type model. Because there was no design system in use at Spanbeton that supported the high semantic level of project-models, we had to develop a simple design system. In this design system, column rows and floor fields are derived automatically from the grid lines and the end-user can define loads on the floor fields. Floor fields and column rows can be decomposed easily with a few parameters into surface and line elements and connections. The elements are fulfilled by EKON elements. The system calculates the shape and material parameters of the elements based on the loads on the floor fields and the dimensions.

For the second part of the implementation, we used PMshell to formalize, exchange, and apply constructability knowledge. We implemented rules to check whether the preliminary design conformed to assembly knowledge. For example, we implemented a rule that warns the designer when the weight of a designed element is larger than the capacity of the largest crane that will be on site. Another rule warns the designer when columns are connected in length direction in situations where one longer column can be used. Even though longer columns are more expensive because they are difficult to transport, on site they are much easier and faster to assemble.

Test case

We tested the implementation of the design system and the knowledge rules in a research environment. We designed a simple structure, exchanged information and knowledge using the STEP languages, and applied constructability knowledge to the designed structure.

The test case (both developing the system and its application) showed the potential value of the approach. Better means for formalizing and exchanging information and knowledge are within reach of the building industry. Knowledge does not have to fade away anymore when projects are finished. In addition, the large amounts of high-level information that will be available when industry uses this approach will have many possible other areas of use besides Design for Construction such as integration with other life-cycle stages, (robot) automation on site, linking technical and financial departments, better control of production, better change management, and use of cast-in transponders for easy element identification on site.

CONCLUSIONS

One of the building industry's answers to the growing demands of society is further automation and integration of design and construction management, the main information transformation activities in a building project. This eliminates wasteful activities such as manual copying of data and human reinterpretation of documents. It also leads to better use of the large amounts of information and knowledge that have become available since the introduction of computer applications. Further automation implies the development of more and better computer applications. Integration implies communication between these applications. This makes automation and integration two sides of the same coin. Integration is only possible when information and knowledge are available electronically, and further automation is only economically feasible when information and knowledge input is automated. This paper has outlined an approach that might support further automation and integration of design and construction management.

As discussed in this paper, we based our integration approach on project-modeling, which is an extension of product-modeling. The approach focuses on two areas of interest: Project Specific Modelers and communication mechanisms. The paper argues the importance of using the same high semantic level of representing knowledge for the modelers and the transfer mechanisms. Both should use the semantic level of professionals in practice. Product-modeling, as being developed by STEP (ISO/TC184 1993), provides this high semantic level. It is always possible to derive lower level information (e.g., 2D drawings) from a product-model automatically, but, deriving a product-model from drawings requires human interpretation, which is time consuming, costly, and open for misinterpretation.

It is important to follow international standardization developments when developing a Project Specific Modeler in a company: computer applications will sooner or later conform to the standards and conformity with the standards ensures communication with future (and yet unknown) project participants. Using a project-modeling system that supports standardized modeling languages and modularity enables professionals to develop and apply project-models. This paper briefly discussed PMshell, a prototype for such a system. PMshell is by far not ready for commercial use, but it indicates the feasibility and relevance of such systems.

The standardization of project-modeling is also not yet completed. Here lies a task for the building industry as a whole. In the paper we suggested an approach for developing such a standard. We showed a layered architecture of project-models to ensure modularity and integration of models (and applications). We also presented the BPM, a first version of a reference model for building projects. In the BPM we explicitly modeled the relations between products, activities, and construction resources and the evolving states of information during design and construction management. This model should be extended, refined, and tested in further detail. Companies should develop conceptual models for their own types of projects. After implementing and applying these models internally, they can propose the models for standardization. This ensures a bottom-up, industry-driven development of these rather abstract new technologies. This approach is already followed in many of the European Union-sponsored ESPRIT research projects, which are proposing their results for standardization as Application Protocols in STEP.

With further automation and integration based on (standardized) project-modeling the building industry has an important tool with which to cope with the demands of society. Whether the building companies will pick it up and dare to take the challenge, only the future will tell.

ACKNOWLEDGMENTS

We thank Ballast Nedam Engineering and the Applied Mechanics section of the Civil Engineering department of Delft University of Technology for sponsoring the research. During the research the authors received much support from colleagues at the university, TNO, Ballast Nedam, and Spanbeton. In particular, we thank Martin Fischer for his critical comments on earlier versions of this paper and Mary McDevitt for her efforts to improve the readability of our writing.

APPENDIX 1. REFERENCES

Appleton Company (1985). *Information Modeling Manual IDEF$_1$ - Extended (IDEF$_{1X}$).* D. Appleton Company, Inc.

ARTB (1993). *ARTB Vision on the Building Industry in 2010.* Technology Policy Advisory Board for the Building Industry (in Dutch).

Bennenk, H.W. and Boom, P. van (1992). "The EKON system gives space to users and designers (I)." *Cement*, 44 (2), 16-21 (in Dutch).

Björk, B.C. (1991). "A Unified Approach for Modelling Construction Information." *Building and Environment*, 27 (2), 173-194.
Björk, B.C., Penttilä, H., and Hannus, M. (1989). "A scenario for the development and implementation of a building product model standard." *Advances in Engineering Software*, 11 (4), 176-187.
Boothroyd, G. and Dewhurst, P. (1987). *Product Design for Assembly*. Boothroyd Dewhurst, USA.
Fischer, M.A. (1989). "Design Construction Integration through Constructibility Design Rules for the Preliminary Design of Reinforced Concrete Structures." *1989 CSCE/CPCA Structural Concrete Conference*, Montreal, Canada, 333-346.
Froese, T.M. (1992). *Integrated Computer-Aided Project Management Through Standard Object-Oriented Models*. Ph.D. thesis, Stanford University, CA, USA.
Gielingh, W.F. (1988). *General AEC Reference Model (GARM)*. TNO Building and Construction Research, BI-88-150.
Gielingh, W.F. and Suhm, A.K. (eds.) (1993). *IMPPACT Reference Model: An Approach for Integrated Product and Process Modelling of Discrete Parts Manufacturing*. Vol. 1, Research Reports ESPRIT: Project 2165 - IMPPACT, Springer-Verlag, Darmstadt, Germany.
Howard, H.C., Abdalla, J.A., and Phan, D.H.P. (1992). "Primitive-Composite Approach for Structural Data Modeling." *Journal of Computing in Civil Engineering*, 6 (1), 19-40.
ISO/TC184 (1989). *STEP Planning Model*. Industrial automation systems and integration - Product data representation and exchange, Tolman, F.P. and Gielingh, W.F. (eds.), TNO, Delft, the Netherlands, ISO/TC 184/SC 4/WG 1/N 328.
ISO/TC184 (1993). *Part 1: Overview and fundamental principles*. Industrial automation systems and integration - Product data representation and exchange, Draft International Standard, ISO, Geneva, Switzerland, ISO DIS 10303-1.
Luijten, B.F.M. and Luiten, G.T. (1993). "A layered-modelling tool for the integration of computer applications." *Object-oriented applications*, Meyer, B. and Nerson, J.M. (eds.), Object-oriented series, Prentice Hall, Hertfordshire, UK.
Luiten, G.T. (1994). *Computer Aided Design for Construction in the Building and Construction Industry*. Ph.D. thesis, Delft University of Technology, the Netherlands.
Luiten, G.T., Froese, T.M., Björk, B.C., Cooper, G., Junge, R., Karstila, K., and Oxman, R. (1993). "An Information Reference Model for Architecture, Engineering and Construction." *Management of Information Technology for Construction*, Singapore, Mathur, K.S., Betts, M.P., and Wai Tham, K. (eds.), 391-406, World Scientific Publishing Co. Pte. Ltd.
Meyer, B. (1988). *Object-oriented Software Construction*. Series in Computer Science, Hoare, C.A.R. (ed.), Prentice Hall, Cambridge, UK.
Nijssen, G.M. and Halpin, T.A. (1989). *Conceptual Schema and Relational Database Design: A fact oriented approach*. Prentice Hall.
Smith, B.M. (1986). *PDES Initiation Activities*. NIST.
Smith, J.M. and Smith, D.C.P. (1977). "Database abstractions: Aggregation and Generalization." *ACM Transactions on Database Systems*, 2 (2), 105-133.
SofTech (1981). *Function Modeling Manual ($IDEF_0$)*. Vol. 4, Integrated Computer-Aided Manufacturing (ICAM), Architecture Part II, SofTech, Inc., Waltham, USA.
Sowa, J.F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. The systems programming series, Addison-Wesley Publishing Company, Inc.
Tolman, F.P. (1991). *Integration Core Model for Product Modelling*. CAM-I, C90-PMP-02.

Willems, P.H. and Tolman, F.P. (1993). "Semantic Topology: the Management of Shape Definition." *Management of Information Technology for Construction*, Singapore, Mathur, K.S., Betts, M.P., and Wai Tham, K. (eds.), World Scientific Publishing Co. Pte. Ltd (not in proceedings).

Womack, J.P., Jones, D.T., and Roos, D. (1990). *The machine that changed the world: the story of lean production.* HarperCollins Publishers, New York, USA.

Wright, G.H. von (1963). *Norm and Action.* International Library of Philosophy and Scientific Method, Redwood Burn Limited, London, UK.