An Information Reference Model for Architecture, Engineering, and Construction

G LUITEN[1]
T FROESE[2]
B-C BJÖRK[3]
G COOPER[4]
R JUNGE[5]
K KARSTILA[6]
R OXMAN [7]

ABSTRACT

This paper describes the results of the information modelling work group from the International Workshop on Models For Computer-Integrated Construction in Finland, October 1992. At this workshop, researchers from around the world presented their individual modelling efforts. These models' parallel goals and abundant similarities led the participants to combine their individual results into a single cohesive reference point to act as a basis for future work. The result is *IRMA, an Information Reference Model for Architecture, Engineering, and Construction*. IRMA can serve as the core of a conceptual project model, which can be used as a framework for data standards, as a kernel for modelling, and as a basis for implementing computer applications. It defines generally applicable relationships between products, activities, resources, and participants in a building project.

1 *Computer Integrated Construction, Department of Civil Engineering, Delft University of Technology, PO Box 5048, 2600 GA Delft, The Netherlands.*
2 *Department of Civil Engineering, University of British Columbia, Vancouver, B.C., Canada V6T 1Z4.*
3 *Laboratory of Urban Planning and Building Design, VTT (Technical Research Center of Finland), PO Box 26, 02151 Espoo, Finland.*
4 *Information Technology Institute, Salford University, Salford M5 4WT, United Kingdom.*
5 *CAD, Computer Anwendung in der Bauplanung, Osterwaldstraße 10, 800 München 40, Germany.*
6 *Laboratory of Urban Planning and Building Design, VTT (Technical Research Center of Finland), PO Box 26, 02151 Espoo, Finland.*
7 *Faculty of Architecture and Town Planning, Technion, Haifa 3200, Israel.*

Key Words
computer integrated construction; project modelling; project model; data standards; object-oriented modelling languages

INTRODUCTION

In October, 1992, a dozen researchers from around the globe who had been working in various areas of Computer-Integrated Construction (CIC) and project modelling met at VTT in Helsinki to share their experiences and results to date. The meeting split into two working groups: one initiated a framework for describing and evaluating CIC efforts (Fischer et al, 1992) and the other looked into conceptual models of Architecture, Engineering, and Construction (AEC). This paper presents the results of the modelling working group.

Three members of the Modelling working group had previously developed conceptual models for AEC projects. It seemed clear to the group that we should compare these models and, if possible, combine the best features of each into a single conceptual model, IRMA model, an Information Reference Model for AEC. IRMA identifies and formalizes some of the key concepts of projects (such as physical components, activities, resources, *etc*) and it shows the relationships between them. It is intended as the consolidation of our previous research, not as a final product. Thus, IRMA could act as a starting point for future, more-complete conceptual project models. Before developing IRMA, however, we found it necessary to establish a common understanding of the basic modelling terminology and concepts involved. We also created a list of the basic modelling and representational mechanisms that we assumed to be available for our work.

There were two major objectives. First, to advance our individual modelling work. IRMA points to improvements that can be made in each of the initial models, since it represents a selection of the best features of all of them. Second, to provide a level of integration within our modelling research work, much as our models themselves can help provide integration within the AEC industry. IRMA offers a common understanding that each of our individual efforts can refer to, which will help to synchronize our future work and lead to greater benefits from our modelling research as a whole.

In this paper, we first describe the context of IRMA, which summarizes the underlying modelling concepts for the purpose of creating IRMA. We then outline the basic modelling mechanisms that we assumed to be available. Some of these are commonly available while others are still general principles that have yet to be implemented in modelling languages. We then give an overview of the three models from which IRMA was derived. A description of IRMA itself then follows, along with guidelines for how IRMA might be used. We close with a discussion of unresolved issues and future work.

Context of IRMA

In discussing the various models that contributed to IRMA, we found ourselves using similar modelling concepts, but often with different terminology and slightly different definitions. Before we could proceed to an integrated model, we had to agree on the context, underlying concepts, and terminology. This resulted in a model that describes the context of IRMA. The aim in developing this model is to clarify terminology and to provide a context in which the role of IRMA may be understood. This section describes the basic concepts of the model and it's four elements: the modelling language, the conceptual model, the information, and the real-life object.

Basic Concepts

Project modelling primarily deals with the relationships between information about real-life objects in a project and the real-life objects themselves. Information about real-life objects must be structured in order to be stored in computers and used effectively later. Conceptual models provide this structuring, *ie*, they describe how information about real-life objects should be organized. Conceptual models themselves are described with modelling languages (more precisely, they are structured according to the specification of modelling languages). Figure 1 is an EXPRESS-G diagram that shows the relationships between real-life objects, information, conceptual models, and modelling languages (see (Spiby, 1991) for an explanation of the EXPRESS-G technique).

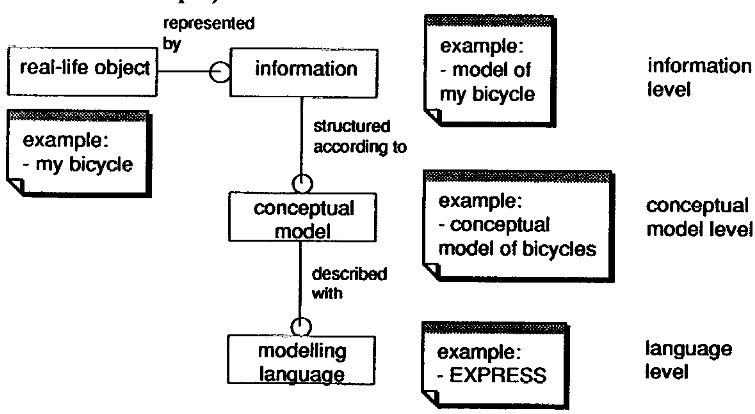

Figure 1. Basic concepts that form a context for the IRMA model, with three levels of abstraction.

An example of a real-life object is 'my bicycle'. The relevant information about that bicycle can be stored in a 'model of my bicycle'. This data is structured according to a 'conceptual model of bicycles'. This conceptual model stores a data structure that can be used for bicycles in general. The conceptual model is described using a modelling language, *eg*, EXPRESS. The main elements of the Figure 1 are described in the following sub-sections.

Modelling Language

A modelling language has a syntax and semantics. The syntax prescribes the symbols that can be used in the language while the semantics define the symbols' meaning. Syntax and semantics are described in the specification of the modelling language, as shown in Figure 2. Examples of modelling languages are EXPRESS (Spiby, 1991), NIAM (Nijssen and Halpin, 1989), and IDEF1X (IDEF1X, 1985). EXPRESS is the alpha-numerical modelling language developed by the International Organization for Standardization - STandard for the Exchange of Product model data effort (ISO-STEP). It has a graphical representation, EXPRESS-G, which can be used for presentation. NIAM is a graphical data-modelling language, widely used in CIC projects. IDEF1X is also a graphical data-modelling language, developed by the US Army. We are currently using EXPRESS and EXPRESS-G to represent IRMA.

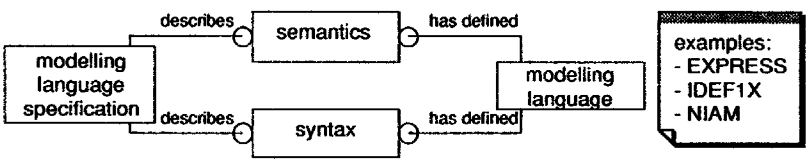

Figure 2. The Modelling Language Level

Conceptual Model

A conceptual model describes the structure of the information about real-life objects that is relevant to a certain domain, in our case AEC. A conceptual model can be stored in a physical representation, *eg*, in an ASCII file or a graphical schema (see Figure 3). This physical representation depends on the modelling language used. When using EXPRESS, a conceptual model is stored in an ASCII file (currently, the EXPRESS-G graphical representation is for presentation only).

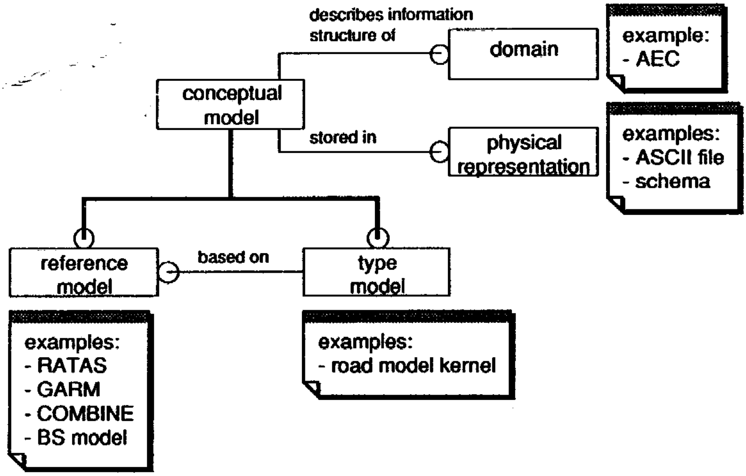

Figure 3. The Conceptual Model Level

There are two types of conceptual models: reference models and type models. Reference models are more generic and describe the information structure for a whole branch of industry, *eg*, the STEP General AEC Reference Model, GARM (Gielingh, 1988), for AEC products. Other examples of reference models for AEC products are the COMBINE Integrated Data Model, IDM (Dubois *et al*, 1992), the RATAS model (Pentillä and Tiainen, 1991), and the AEC Building Systems Model (Turner, 1988). Examples of reference models for project information are the three models that contributed to IRMA: the unified approach model, GenCOM, and BPM (these are described later in this paper). Type models are more specific and describe the information structure for a restricted type or family of objects, *eg*, the Road Model Kernel (Willems, 1990) for highways. Type models are based on reference models.

Information

Conceptual models structure information about real-life objects. Information models are instantiations of these conceptual models. "Instantiation" is the filling-in of the information structure (as defined by the conceptual model) with values that correspond to the properties of the real-life objects. These instantiations of conceptual models form information models of specific real-life objects. Information models are stored in a physical representation, *eg*, in an ASCII file or a database (see Figure 4). The physical representation of the information model depends upon the modelling language used to define the conceptual model. When EXPRESS is used to describe the conceptual model, a 'STEP physical file format' (STEP, 1991) is used to describe the data in the information model.

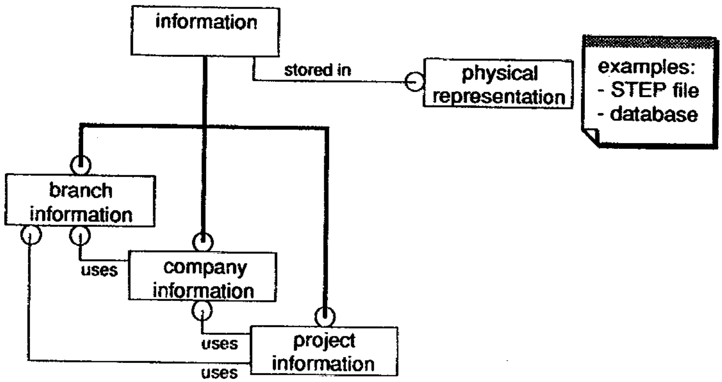

Figure 4. The Information Level

Modelling Mechanisms

The preceding section described the basic modelling concepts used to create IRMA. Beyond these basic concepts, however, both IRMA and the AEC information models that could derive from IRMA will employ various modelling mechanisms. Before we could define IRMA, we found it necessary to define the mechanisms that we assumed to be available to us. The goal of this section is to identify these mechanisms rather than to solve any technological issues relating to them. In fact, it is currently unclear how some of these mechanisms could be implemented.

Most of the modelling mechanisms we used for IRMA are provided by the object-oriented (OO) paradigm (Meyer, 1988). A modelling language such as EXPRESS has most OO mechanisms, but not all. Many of the basic OO mechanisms are generally agreed upon, including the use of object types or entities that encapsulate properties and behavior (*ie*, object types have features which may be attributes, relationships, or methods), rules on information structures (local rules on objects and global rules on models), and classification or specialization hierarchies, through which object types can inherit properties (we assume that multiple-inheritance is possible). We also assumed the availability of newer OO mechanisms, such as *multiple classification* (where an object is an instance of more then one object type) and *dynamic classification* (where objects can change their object type).

The following modelling mechanisms are necessary for IRMA but are not provided by the basic OO paradigm:

- Physical composition/decomposition hierarchies break objects down into smaller part-objects and into relationships between the part-objects.
- A versioning mechanism allows different versions of an object to coexist in the system, and it supports several different capabilities that we wish IRMA to possess. First, versioning allows the concretization of information about objects during the life of a project. That is, an object is first specified as a set of requirements, information is then added to represent proposed solutions that satisfy the requirements and, later, an actual as-built version of the object can be described. The successive versions of objects can be recorded in the system as an archive of the project life cycle. Second, versioning allows different project participants to maintain different views of the same basic objects. Third, different versions of an object can be offered as alternatives from which the best approach can be selected. Fourth, allowing new versions of objects to be defined without committing the changes to the project database allows simulation and what-if analysis of project performance.
- A variety of information inheritance mechanisms should exist to eliminate redundant information storage. For example, individual objects could inherit properties from types or classes of objects, from prototype objects, from past cases of similar objects, from the aggregate objects that they are part of, and from previous versions of the same object.
- Views should be created that map the general IRMA model into models that are customized for particular applications. These views will often simplify the general model and may introduce alternative terminology or aliases, but it must be possible to convert these application-specific information models back to the general case.

Certain conceptual models have been tried in order to define these mechanisms as they apply to construction projects. The model which most explicitly deals with them is the GARM model (Gielingh, 1988). The modelling of these mechanisms can, in fact, be handled on a very generic level, independent of the explicit modelling of specific building parts or construction activities.

Original Conceptual Project Models

IRMA was developed by combining features from the three previously developed conceptual models described in this section (some work relating to the ICON model (Cooper et al, 1992) also contributed to IRMA).

Unified Approach Model

The *Unified Approach Model* was devised as a generic model suggesting how the use of a single conceptual modelling technique for modelling all kinds of construction information would facilitate the integration of different information technology (IT) applications from very diverse domains, such as

CAD, project management, EDI procurement, *etc* (Björk, 1992). Originally the model was presented using the IDEF1X modelling technique, but later an EXPRESS-G version was been developed (see Figure 5). An exercise showing how the framework could be used for structuring information concerning the erection of partition walls was also carried out (Karstila *et al*, 1991).

One aim of the model was to provide a framework which would help explain the relationships between current building classification systems and product models. This research also high-lighted the differences between activity models, which usually are formalized using techniques such as IDEF-0 (IDEF0, 1981), and conceptual models, including object classes for activities.

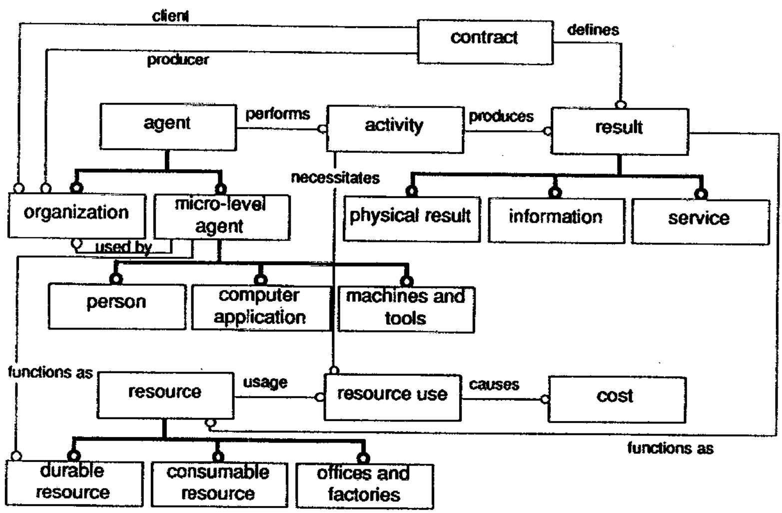

Figure 5. The Generic Model of the Unified Approach Model

GenCOM

The *General Construction Object Model, GenCOM,* was developed as part of a project carried out from 1989 to 1992 at Stanford University to improve the integration of project management software using standard object-oriented models of construction projects (Froese, 1992a, 1992b). This work investigated the role of standard models for implementing integrated software packages, for creating data exchange languages, and for providing industry-wide information categorization schemes. GenCOM's scope and conceptual development may be more limited than other models presented in this section, but the attributes, constraints, and implementation characteristics of the general objects were developed in greater detail. The GenCOM model (consisting of 36 object types) was implemented in an integrated project planning program called the *Object-model-based Project Information System, OPIS*.

Figure 6 shows some of the high-level GenCOM object types. At the heart of the model is the representation of the project's physical components, and of the activities that operate on the components. Specifically, activities represent the application of some action to some component using a particular method and a set of resources (thereby changing the state of the component). The actions, methods, and resource-uses are all explicitly defined, as is the project participant responsible for the activity. GenCOM shares with this section's other models the consistency among the high-level object types that motivated the IRMA work. Future work in adding greater depth to the IRMA model could draw further upon the detail of the GenCOM project.

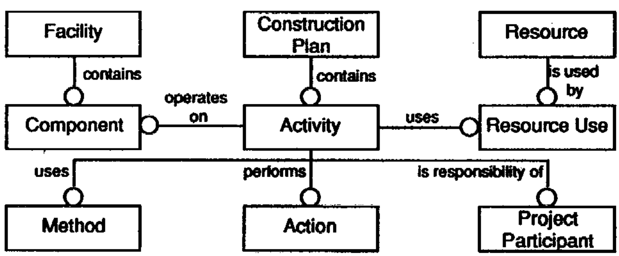

Figure 6. Some of the high-level object types from the GenCOM conceptual model of construction

Building Project Model

The *Building Project Model, BPM* (Luiten and Bakkeren, 1992), comprises a kernel in which project, product, activity, resource, and actor information is related. BPM is based on the General AEC Reference Model, GARM (Gielingh, 1988), and the IMPPACT Reference Model (Gielingh and Suhm, 1992). The information in the kernel is specified with modelling mechanisms that divide entities into function and solution, that identify both specification and realization views, and that make decomposition possible. For IRMA, only the kernel is of importance.

Figure 7 shows an EXPRESS-G schema of the kernel. A project includes five types of 'project objects': 'projects', 'products', 'activities', 'resources', and 'actors'. A 'project' contains the general project information and consists of 'products', activities', 'resources' and 'actors'. An 'activity' has an initial and an end-state. The information about these states is described in a 'project object'. For example, an 'activity' starts with a 'product' in a certain state and ends with a 'product' in a new state. These states are described with the aid of modelling mechanisms. An 'activity' is performed by an 'actor' and uses 'resources', which are at the disposal of the 'actor'.

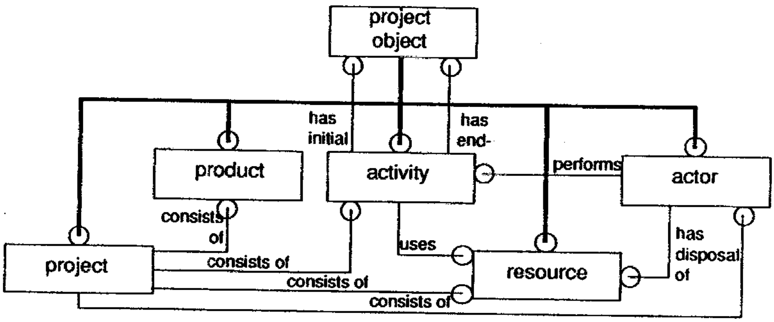

Figure 7. Conceptual Project Model, Kernel of the Building Project Model

IRMA

We combined the three original conceptual project models into a new *Information Reference Model for AEC* called *IRMA*. IRMA is a conceptual project model that focuses on the relationships between the products, activities, resources, and participants in a building project. IRMA treats these as fairly generic concepts, it does not pursue great depth of information for any of them.

Figure 8 depicts the inheritance relationships in IRMA. There are four sub-types of 'project objects': 'products', 'activities', 'resources', and 'contracts'. For the modelling of these sub-types, we refer to existing conceptual reference models, such as RATAS (Pentillä and Tiainen, 1991), GARM (Gielingh, 1988), COMBINE IDM (Dubois *et al*, 1992), AEC Building Systems Model (Turner, 1988), and the Integrated Building Process Model (IBPM) (Sanvido *et al*, 1990). A special type of 'resource' is an 'agent', a participant in the building project. An 'agent' can be an 'organization' or a 'micro-level agent'. Examples of micro-level agents include individuals, machines, robots, or computer applications.

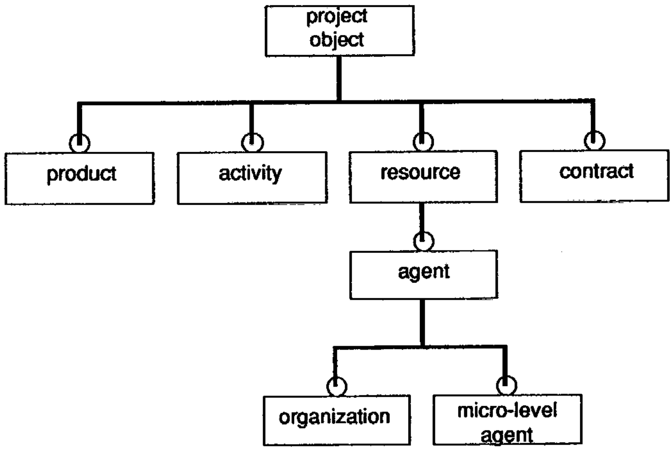

Figure 8. Inheritance Relationships in IRMA

Figure 9 shows some of the major relationships between the 'project object' sub-types shown in Figure 8. 'Activity' is the central object type. An 'activity' results in a 'state' of a 'project object'. To be able to reach that new 'state', the 'activity' requires a number of 'project objects' to be in a certain initial 'state'. An 'activity' uses 'resources'. A 'resource use' describes the utilization of a resource for a particular activity. An 'activity' is performed by an 'agent' and its result is defined in a 'contract' between the client and the producing 'agent'.

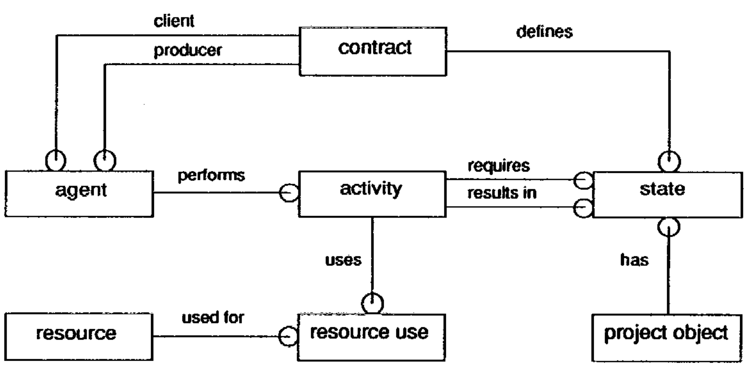

Figure 9. Relationships in IRMA

The notion of 'state' is essential to IRMA. A 'project object' can exist in different states that can be explicitly described. For example, a column can be in an 'as required' state, an 'as designed' state, an 'as built' state, *etc.* Activities may require certain states to exist before they can occur (prerequisites) and will result in new states (results). A 'state' that is the result of one 'activity' can be a prerequisite for the next, in which case the 'state' is called a dependency. Thus, 'states' are used for relating activities to other project objects, such as building components, as well as for representing activity sequencing logic which is used for scheduling or simulating construction processes. For example, Figure 10 shows a set of activities involved in the production of a prefabricated concrete beam: 'design concrete beam', 'plan construction', 'make formwork', and 'place concrete'. This figure uses the IDEF-0 process modelling technique (IDEF0, 1981). The activity 'design concrete beam' requires the specifications (*ie*, the 'as required' state of the product) as a prerequisite. This results in a design for the beam (*ie*, the product in the 'as proposed' state). A completed design is a prerequisite for the construction planning activity. The result-the construction activities in the 'as proposed' state-forms a prerequisite for the final production of the beam, represented here by the concrete placement activity. Aside from the production activity's information prerequisites (*ie*, the design and the plan), the activity also requires that the resources are in an appropriate state. That is, the formwork must be produced before the concrete placement can begin. This example shows that all project object subtypes (not only products, but also activities and resources) have states and can be altered by activities.

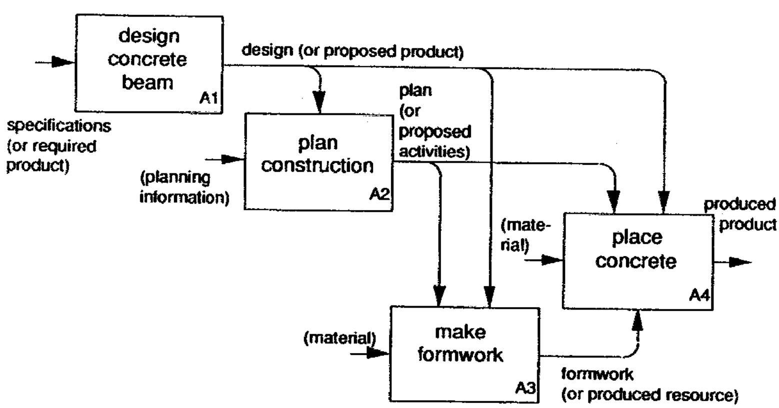

Figure 10. IDEF-0 Activity Model of the Activity 'Produce Prefab Concrete Beam'

Guidelines for Using IRMA

IRMA can be used on three levels: as a framework for standardization, as a kernel for modelling, and as a basis for implementation.

As a general reference model for building projects, IRMA offers a sound basis for standardization efforts such as classification systems, Electronic Data Interchange and Product Data Interchange (EDI-PDI) developments, and general construction information systems. IRMA could thus be useful in the work of international organizations such as the AEC subcommittee of STEP, ISO's working group for building classification systems (ISO TC59/SC13) and European committees dealing with EDIFACT messages as well. IRMA's object types, though few, underlie much of the construction process and their definitions represent a consensus of opinion among several contributors.

IRMA can also be used as a modelling kernel by specializing all or part of it to create conceptual models for specific types of building projects, *eg*, a conceptual model for the creation of office buildings. Specialization involves defining new object types by sub-typing IRMA's object types, and adding or altering their properties, relationships, and behavior.

Finally, IRMA can be used as a basis for implementing computer programs. This implementation should use an OO programming environment and will likely involve a relational or object-oriented database. Applications should implement not only IRMA's object types and their properties (as shown in Figure 9), but also their behavior. Implementations of IRMA and its specialized type models will be used by data modellers, who work at the data level (see Figure 1) and instantiate the conceptual models (to produce and utilize shared construction databases, for example).

There should be no need to start from scratch in implementing IRMA: partial models from existing implementations can be reused. Partial models are conceptual models for products, activities, *etc*. The focus of IRMA implementations should be on implementing the relationships between existing partial models. To enable system-independent communication between users at different locations, communication should be based on ASCII files with a standardized format. STEP provides such standardized formats for both conceptual modellers and data modellers. Systems can import and export EXPRESS schemata to communicate conceptual models, or STEP physical file formatted ASCII files to communicate information models.

Issues Relating to IRMA

Although some important steps are made towards a combined reference model, IRMA is not yet completed: there are still many issues to be resolved. The main issues are enumerated below:

1. IRMA assumes many modelling mechanisms to be available. However, the actual availability of some of these mechanisms is questionable. For

example, the versioning mechanism is very complicated to model and to implement; it is questionable whether one will ever be able to fulfill all our versioning requirements. A second example involves the OO notions of multiple instantiation and dynamic classification. These notions are not currently supported in most OO languages and, in particular, not by the STEP's EXPRESS and physical file format.

2. Section 3 introduced 'project objects' states and posed some requirements. This mechanism must be worked out in much greater detail. The final form of this mechanism has great influence on IRMA.

3. We envision the final version of IRMA to include approximately forty object types. The model presented in this paper is only a starting point. Although we believe it was necessary to agree on this starting point first, we must now continue to address issues such as costs, shape, time, *etc*.

4. The distinction between generic information (such as branch or company information) versus specific information (such as project information) must be worked out in greater detail, since it is vital for information management.

CONCLUSIONS

Several conclusions can be drawn from the preliminary IRMA work presented in this paper. The high degree of similarity among the independently produced models that contributed to IRMA clearly points to the possibility of a single, generally applicable reference model for AEC projects. Equally notable, however, was the amount of work still required before such a standard model can be produced satisfactorily. Several key modelling issues have yet to be resolved and IRMA's validity and practical applicability have yet to be proven. Nevertheless, experience with similar models, such as reference models for product data, have shown the value of our approach as a basis for a framework for standardization efforts and a practical reference model for project specific models. We believe that the remaining issue can be solved, and that the IRMA work could eventually lead to robust standard reference models of projects. Finally, the IRMA work confirms the value of coordinating independent research. IRMA has already helped to improve our individual modelling work, to broaden the scope of our research, to provide a unifying reference point, and to protect each of us (and others) from "yet another discovery of a nearly square wheel".

References

Björk, B C (1992), A Unified Approach for Modelling Construction Information. *Building and Environment*. Vol. 27, No. 2. pp 173-194.

Cooper, G S et al (1992), Incorporating Alternative Perspectives in a Single Information Model - The ICON Project Analysis Method. *Workshop on Models for Computer Integrated Construction*, Finland:VTT, October.

Dubois, A M, J C Escudié and L Laret (1992), *Combine IDM.* Ver. 3.3.Tech. Rep.

Fischer, M A, M Betts, M Hannus, Y Yamasaki and J Laitinen (1992), CIC Framework. *Workshop on Models for Computer Integrated Construction*, Finland:VTT, October.

Froese, T M (1992a), *Integrated Computer-Aided Project Management Throught Standard Object-Oriented Models.* Ph.D. thesis, Stanford University.

Froese, T M (1992b), GenCOM, A Domain Model for Project Management and Construction. *Workshop on Models for Computer Integrated Construction*, Finland: VTT, October.

Gielingh, W F (1988), *General AEC Reference Model.* Tech. Rep., P.O. Box 46 2600 AA, the Netherlands, BI-88-150, October.

Gielingh, W F, and A Suhm (eds) (1992), *MPPACT Reference Model: An Approach for Integrated Product and Process Modelling of Discrete Parts Manufacturing.* Springer-Verlag.

IDEF0 (1981), *Integrated Computer-Aided Manufacturing (ICAM), Architecture Part II, vol. IV - Functional Modelling Manual (IDEF-0).* SofTech, Inc. Waltham, MA, USA, June.

IDEF1X (1985), *Information Modeling Manual Idef1 - Extended (IDEF1X).* Manhattan Beach, California: D. Appleton Company, Inc., December.

Karstila, K, B C Björk and M Hannus (1991), *A conceptual framework for design and construction information.* Proceedings 1st International Symposium on Building Systems Automation-Integration, Madison, Wisconsin, USA.

Luiten, G T and W J C Bakkeren (1992), A Layered Modelling Methodology applied to the Building and Construction Industry. *Workshop on Models for Computer Integrated Construction*, Finland: VTT, October.

Meyer B (1988), *Object-oriented Software Construction.* Prentice Hall.

Nijssen, G M and T A Halpin (1989), *Conceptual Schema and Relational Database Design: A fact oriented approach.* Prentice Hall.

Pentillä, H and P Tiainen (1991), RATAS - Building Product model proptotype quantity take off from a building project model representation of design data. *Proceedings ARECDAO91 Conference, ITEC,* Barcelona, April.

Sanvido, V et al (1990), *An Integrated Building Process Model,* Tech. Rep. 1, CIC Research Program, Dept. of Arch. Eng., Pennsylvania State Univ.

Spiby, P (1991), *The EXPRESS Language Reference Manual,* STEP document: ISO TC184/SC4/WG5 N9, March.

STEP (1991), *Product Data Representation and Exchange - Part 21: Clear Text Encoding of the Exchange Structure,* STEP document: ISO TC184/SC4 N78, March.