A SEMANTICALLY RICH REFERENCE MODEL FOR BUILDING DESIGN

Reference model for building design

M.J. O'BRIEN and A. BAIG
The School of Computer Science and Information Technology, University of Nottingham, Nottingham, United Kingdom

Durability of Building Materials and Components 8. (1999) *Edited by M.A. Lacasse and D.J. Vanier.* Institute for Research in Construction, Ottawa ON, K1A 0R6, Canada, pp. 2609-2618.

Abstract

Much effort has been expended by software developers attempting to build databases suitable for use by those working within the construction industry. Various models from the original RATAS relational database model through to sophisticated process models have been proposed, developed and evaluated. It is probably fair to say that these research efforts have only recently begun to effect the practices of professional construction engineers. This, in part, is due to the need for more sophisticated systems. This paper describes a database that is usable throughout the design and construction processes in the construction industry. The method uses the well-established idea of generic components that can be combined to create a large scale artefact. The novelty of the approach described herein allows the components to embody facts and rules that allow design knowledge to be modelled, captured and retrieved. The facts and rules encapsulate not only the interactions of the various products but also the processes involved in their use. In effect, the atomic primitive elements (both components and rules) can be combined to create complex elements which are semantically rich. The basic ideas and fundamental philosophy of this approach have been described elsewhere. This paper is devoted to describing the detailed implementation of this approach. The content is technical and thorough; it describes how a passive relational database management system, Oracle, has been used to create a new metadata structure for the creation, control and management of the components - both simple and complex. In effect, the relational database becomes active. Thus, the database reacts to design decisions by firing rules which then govern the interaction of the components. The paper presents a detailed description of the underlying architecture and the data model which has been developed. The paper is interesting not only to construction engineers but also to software designers in that it shows how existing database models can be extended by using their predefined data types to create new, and more complex, ones. While this is an old, well-established trick, this application to a real-world problem is a good test of its viability. Finally, a brief review puts this particular approach into the context of the other myriad attempts to create product and process reference models with an evaluation of its strengths and weaknesses.

Keywords: Active database, Component model, Design database, Product model, rules

1 Introduction

The design, development, and evaluation of reference models has become a popular activity amongst researchers working in the area of construction information systems. Product models, process models, standardisation efforts, and system architectures - all of these have become standard fare. One might cite the work of Wix and Bloomfield (1997) and Wix and Liebich (1997) in the area of standards, Brown *et al* (1996), Hazelhurst *et al* (1997), Björk (1997) and a host of others in proposing various system architectures and paradigms. This paper is no different. Yet, this somewhat flippant exposition should not hide the usefulness and necessity of all these efforts. On the one hand, the realisation that this application domain is one of the most difficult to computerise is now well established. Construction projects combine a depth of complexity, a massive volume of data - and high volatile data at that - and a fragmented organisation framework that have made it difficult to build truly useful systems. Whereas self-contained systems have been successful in their own limited domains, CAD in particular being such an example, the unification and sharing of data across systems has yet to occur. On the other hand, there has been a slow but growing realisation that the computer and software tools at our disposal have been inadequate to the task. In this regard the application domain has acted as a spur to the systems developers to come up with the tools that can do the job. If the linkage is not direct one might posit that the growth of object-oriented systems, with their emphasis on objects - that is to say, encapsulated abstract data types - has come to the aid of construction system developers who have tried to reconcile the differing approached characterised by the product modellers and the process modellers.

This paper takes forward both the ideas behind construction reference models and those behind current thinking on databases. Thus as an application this paper describes a reference model that is active and thus semantically rich. Whereas its contribution to database theory is to show how triggers and actions can be used to create an active database. While this latter achievement is rather clumsily engineered, the ideas should prompt database designers to think of better ways to achieve the desired effects.

A number of papers have already been published that are based on the underlying research for this paper. This paper takes a small step into the build of the proposed system and defines its structure, inspiration and an overall decomposition of the system that is build based on the research.

This paper provides an overview of the model proposed and the scope of the research undertaken to develop the active component model. It then shows an overall functional decomposition of the model. The paper is based upon an earlier one (O'Brien and Baig, 1998) which laid down some of the guiding principles behind this work. Much of the justification for this work has been previously published; this paper therefore describes some of the more technical aspects of the work. However, a full description of the model and the system that has been developed based on the model is outside the scope of this paper.

1.1 Overview

The proposed model illustrates how components are structured, built and managed, how semantic rules about the design components are incorporated into a database and how these rules can be used to elicit active database behaviour. The model defined here relates to the construction of a building, the principles underlining the model can however be applied to any database that is used for design purposes. The model is not a full-scale design database; a prototype system has been developed to create a scenario for developing semantically rich components that can be used to elicit active database behaviour.

1.2 Design overview and requirements

The aim of this section is to provide the overview and requirements document from which the first cut of the ERD can be developed. Concepts mentioned in this section are discussed in details in the thesis by Baig to be published in the near future. A systematic approach is developed that allows the creation of components, which hold not only their properties but also the rules that effect their behaviour. These components are then used to build a semantically rich design model. (From hence fourth a design will refer to building construction design unless otherwise stated.). In order to achieve this the following facilities need to be developed.

- A way of defining atomic components,
- a process of assigning properties and rules to the components,
- a way of associating lower level components to build higher level components,
- methods of defining the general environment within which the components are placed and interact, such as site conditions, space and time, and
- method for hold design history and permitting changes arising from future eventualities or pre timed events.

A component is either an object or a construct it has factual properties and forms a part of the physical world. A construct is an abstract component that also has properties or features. The factual features can be inherent to a component or interdependent between many components. Fore example the features of an office building and all its parts can be considered to be internal while the relationship between the office and the workers is an interdependent relationship.

The features themselves can be internal or external. The internal features are independent of any outside influence where the facts are applicable to the component and only the component. External features are dependent on the component interaction with the outside world. Fore example the stress tolerance is an internal feature of a construction beam while its inability to rest directly on a wall is an external feature because it refers to the components interaction with the outside world. The external features can be objective or subjective. Subject features refer to feelings, opinions and intuition. To say that a straw roof ‘feels warmer’ on country cottage than a roof made from black slate is a subjective feature. External features are rules that apply to a particular component they determine its behaviour and its relationship with other components and the environment within which it resides. Rules are based on the internal features of components. They can be constant or dynamic. Dynamic rules vary and adapt according to their interaction with other components. The fact that the energy requirements from a hot water tank to heat the water to 30°c will vary according to the external temperature of the tank means that the rules must consider dynamic factors in their evaluation before carrying out the appropriate action.

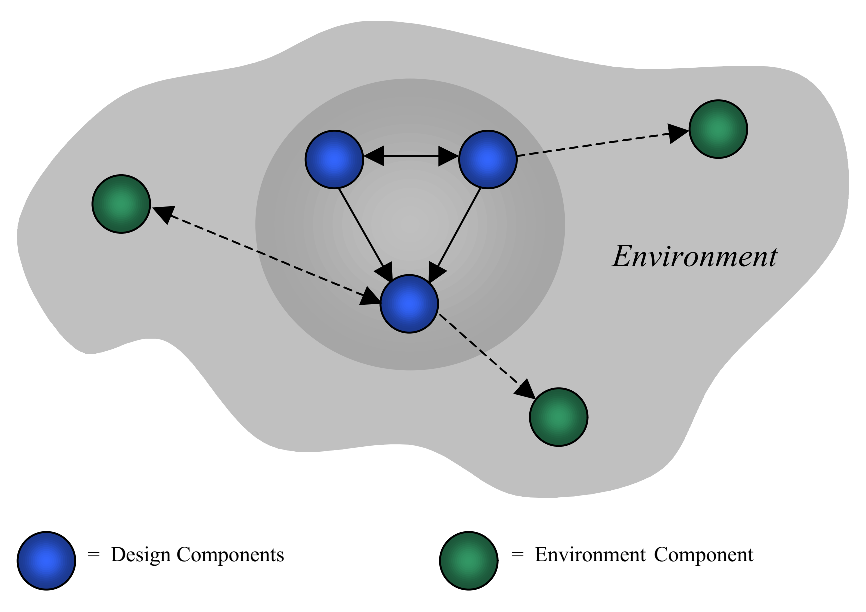


Fig. 1: General design concept using components

A component will change either due to some interaction with other components or with time, thereafter its properties will change accordingly. This refers to the state of the component, hence, the state of a component is the factual data and state changes can be bought about by a change in the rules. The firing of legitimate rules can create a state change. For example a door has different states of openness that can be represented by the function for the angle between the wall and door. The function can take values from 1-180°, other values are not allowed because the relationship rules of the door and wall, hence, any other state of openness is not permitted. The state change can be easily controlled and measured but in some cases a state change that would normally not be allowed may be required. In this case the properties of the components need to change to accommodate the fulfilment of the new state, however, the percussion for 'impossibility states' must exist. In the above example it is impossible for a door to open beyond 360°. The impossible state could also be governed by the rules of the environment. The process of a state change is considered as an event, multiple events or an operation. The firing of events and operations lead to 'indirect' state and/or property change to other components, and some that may not be obviously related. A class is the collection of all properties that make up the component. A component is a member of a class if it has properties, which define a class. A component 'kind' is determined by a set of properties. Fore example a house is a member of the class residential duelling and it also belongs to a kind of accommodation. Classification is a way of grouping the external features into equivalent classes but varying degree of granularity. Within the process of classification individual members are related to some common set and set are related to sub-sets.

New component created by combining existing components has rules that belong to it self even though its properties are based on the properties of the existing components. The environment also influences the emergence of rule within the new component which means that design can not be considered without considering its environment (Ackoff, 1979). An overall design is a collection of the necessary components, an environment that acts upon the design and is acted upon by the design and a method of organising and relating the design.

As shown in Figure 1. The overall design also behaves like a complex component that has internal and external features and states. The word overall design is used loosely as it can itself form a larger design. In this case it is referred to as a sub-design or even a sub-components, the higher level design is referred to as a super-design or super-component in relation to its sub-components. The 'whole design' can therefore not be a sub-component of any thing. A sub and super design introduces the concept of levels. Unlike a hierarchy the higher level design is not linked to a lower level design they are all designs in themselves. The higher level design is made from lower level design and its components. The lowest level design is called an atomic design. Any particular level does not act on its parts. The elements used to link components are regarded, as components them selves and will be referred to as linking components (links), which also have internal and external features. The linking components can be those that physically join two components and generic components that associate components these will be referred to as 'specific fasteners' and 'generic fasteners'. Operations such as moving, close, new may be regarded as generic fasteners. In object oriented terminology these may be referred to as polymorphic objects. The use of a link will produce a new component. This may be of a components-link-component relationship or component-link relationship the latter will normally produce a variation to the participating component. In Figure 2 component 'A-fixed to-B' consists of Component A, Component B and a 'Fixed to' link component. The type of link will determine the result and hence the nature of the new component

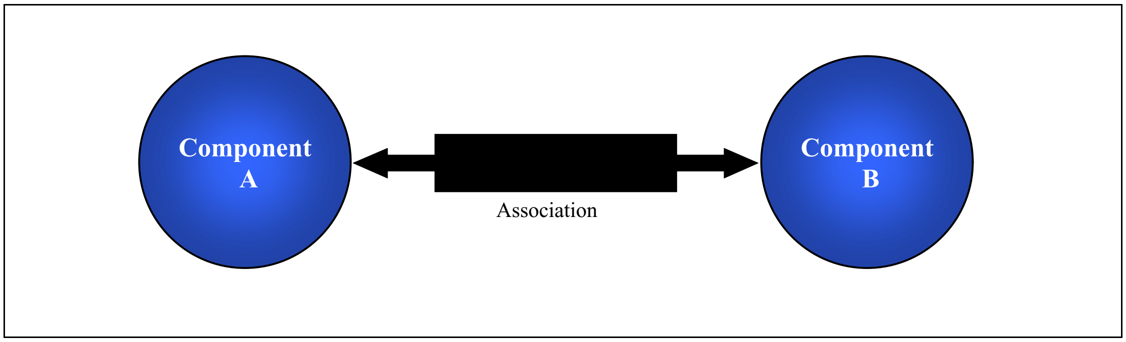


Fig. 2: Component A-fixed to-B

1.3 Rules in the active component model

The rules within the active component model are ECA production type rules and are based on the component facts as discussed in the thesis to be pubised by Baig. The events, conditions, and actions are explicitly declared and are encapsulated into the rule. In addition rules can have attributes and carry other context information, such as coupling mode. Separation of the events, conditions and actions and the use of declarative rule management on the RDBMS have several advantages these are:

1. Provides a unique interpretation of the rule,
2. permits the coupling of rules with triggering transaction in several ways: event-condition, condition-action, and event-condition-action grouping is possible,
3. knowledge of the action portion of the rule as well as the event portion of the rule can be used for optimising the rule,
4. perform transformations by pushing computations from action to condition and from condition to event without changing the execution semantics,
5. two or more rules can be combined into a larger rule,
6. rules can be related to application areas, hence, grouped by context, thus reducing the scope of searches. It is also possible to create sub classes of rules that have their own attributes, and
7. rules can be modified, inserted and deleted in the same way as other data in the database. They are subject to the same transaction semantics as other data objects, such as concurrency controls, therefore rule that is being fired by one transaction cannot be modified, deleted or disabled by another.

1.4 Other systems and product models

This section provides a brief outline of three related systems that relate to the design for the active component model proposed in the underlying research. These are General AEC Reference Model (GRAM), The AEC building model and the RATAS model. These are all rather venerable predecessors but they have effectively established the various developmental lines of research.

1.4.1 GRAM

GRAM is a method of defining a product model and its decomposition for the purposes of design. Within GRAM a product is represented as a PDU, Product definition Unit (Gielingh, 1988). A PDU can be a whole product, sub-system, element, component, part or a feature of a product. The information is given as a collection of attributes of the product. Each attribute of a PDU is related to an aspect. An aspect may be cost, tolerance, and functionality. The description of a PDU is created from four basic aspects called ‘abstraction mechanisms’ and they are as follows:

- Specialisation. This separates different application areas like building, civil engineering, process plant, shipbuilding and terrain mapping.
- Decomposition. Represents how a product can be decomposed into smaller units.
- Life cycle. Distinguishes the stages into requirements, design, planning, build, usage, alterations, and demolition.

- Classification. Identifies occurrences, specific and generic PDUs.

GRAM has had considerable influence as an example of a product modelling method and has been considered as an ISO standard. A full description and evaluation of GRAM is outside the scope of this paper, for more details refer to Gielingh (1988).

1.4.2 AEC building systems model

The AEC Building System Model has been developed within the IGES/PDES ACE Committee, which is a part of the US-national Graphical Exchange Specification (IGES) committee within the Product Data Exchange Specification (PDES) project. The aim of the model is to present high level conceptual schema of an AEC product model. In this case product data refers to the totality of data elements, which completely defines a product for all applications over its expected life cycle (Turner, 1990). The basic concepts of the model are the object. An object may have one or more properties. Most objects go through three phases.

- Generic phase. Exact attributes are not specified.
- Specific phase. Most attributes are known.
- Occurrence. When the location and orientation of a specific object has been determined, exactly or approximately.

An AEC product model is a representation of an AEC project. AEC project types include buildings, process plants, ships, civil projects, and space habitats. AEC project phases include programming; concept design, preliminary design, or planning or design development; detail design; construction documentation; construction planning or construction scheduling; construction; maintenance and operation; redesign, or design for re-use; demolition. A full description and evaluation of ACE is outside the scope of this paper. Refer to Turner (1990) for more details.

1.4.3 RATAS model

The RATAS project attempts to develop a national Finnish system for computer aided design in the construction industry in Finland. The RATAS building model was developed as part of the project to achieve a national building product data model standard. Refer to Björk (1989) and Björk and Penttiala (1993). The RATAS Model describes a building using objects and relations between objects. Two types of relations are involved, the “part-of” -relation and the “connect-to” -relation. An attribute represents properties of the building or parts of the building. An attribute has a domain of values. Objects belong to classes specified with attributes. Lower level classes inherit attributes from higher level classes. To limit the need for data in a representation specific views of the model can be taken. A full description and evaluation of RATAS is outside the scope of this paper. Refer to Björk (1989) and Björk and Penttiala (1993) for more details.

1.5 System Scope

This section places boundaries on the requirements of the active component model. The purpose of the research on which this paper is based is to illustrate how semantics are incorporated within a construction design project by using semantically rich components and, how these components can induce active database behaviour. Ideas developed within the underlying research can be extended to develop a working system. The research carried out focuses on developing: a) an infrastructure to create and manage components, b) an infrastructure to create and manage rules, c) a method of assigning rules to components, and, d) build a design using semantically rich components. Since any object in a construction project can be considered as a component, it is necessary to define the lowest level of granularity for components defined in the underlying research. It is possible to consider the sand, cement and water that make up the bricks and mortar as components. However, to illustrate the active component model on a more practical level the lowest level of component will be lowest level of manufactured object that are used to develop a building such as windows and doors.

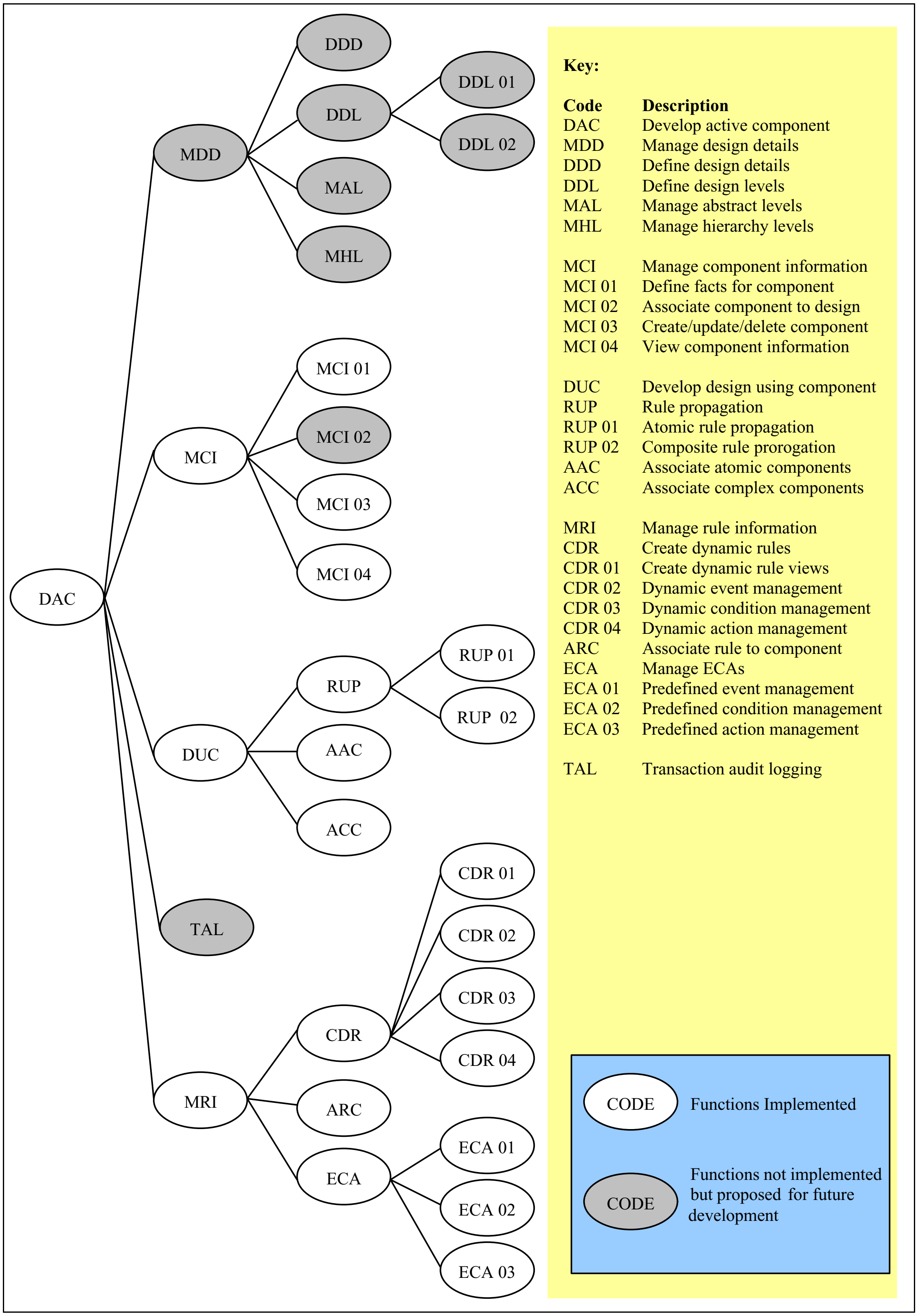
1.6 Functional decomposition

Figure 3 shows the functional decomposition of the active component model. There are five main functional areas: manage design details (MDD), manage component information (MCI), develop design using components (DUC), manage rule information (MRI) and transaction audit logging (TAL). MDD and TAL functions are not implemented but are recommended for future development.

2 Conclusion

This paper has described a successful research effort to build an active database. The database was built on top of an existing mainstream database management system: Oracle. The effort has been mainly directed towards showing that such an approach is indeed feasible. In this regard it has been successful. It must be admitted that it has never been the intention to produce a fully fledged system which could be used in the offices of working construction engineers: principles and theory have been the motivating forces rather than practice and pragmatism.

Finally, this exercise has shown the form that future developments in database theory that will be needed to satisfy complex problems domains such as the construction industry. In this regard this paper points the way forward not only for construction engineers but also for computer scientists.



Key:

Code	**Description**
DAC	Develop active component
MDD	Manage design details
DDD	Define design details
DDL	Define design levels
MAL	Manage abstract levels
MHL	Manage hierarchy levels
MCI	Manage component information
MCI 01	Define facts for component
MCI 02	Associate component to design
MCI 03	Create/update/delete component
MCI 04	View component information
DUC	Develop design using component
RUP	Rule propagation
RUP 01	Atomic rule propagation
RUP 02	Composite rule prorogation
AAC	Associate atomic components
ACC	Associate complex components
MRI	Manage rule information
CDR	Create dynamic rules
CDR 01	Create dynamic rule views
CDR 02	Dynamic event management
CDR 03	Dynamic condition management
CDR 04	Dynamic action management
ARC	Associate rule to component
ECA	Manage ECAs
ECA 01	Predefined event management
ECA 02	Predefined condition management
ECA 03	Predefined action management
TAL	Transaction audit logging

Fig. 3: Functional decomposition

3 References

Ackoff, R. L. (1979). The future of operational research is past, *General systems* Vol. 24.

Björk, B. (1989*)*. Basic structure of a proposed building model, *Computer aided design*, **21**, 2, pp 71-78.

Björk, B. and Penttial H. (1993). A scenario for the development and implementation of building product model standard. *Advance Engineering Software*, **11**, 4, pp176-87.

Björk, B. (1997). Informate: a framework for discussing information technology applications in construction, *Information Technology Support for Construction Re-engineering, CIB Workshop W78, Cairns, Australia,* published in proceedings pp 5-22.

Brown, A., Cooper, G., Rezgui, Y. Brandon, P. and Kirkham, J. (1996). The architecture and implementation of a distributed computer integrated construction environment, *Construction on the information highway, CIB Workshop W78, Bled, Slovenia,* published in proceedings pp 95-108.

Gielingh, W. (1988). General AEC Reference model. External representation of product definition data, *Document no. 3.2.2.1. TNO-report BI-88-150*.

Hazelhurst G., Pitt, T. and Buxton, R. (1997). Implementation of single buiding modelling technologies into the management phase of the property cycle, *Information Technology Support for Construction Re-engineering, CIB Workshop W78, Cairns, Australia,* published in proceedings pp 159-170.

O'Brien, M. and Baig, A. (1998). An active knowledge base for building design, *First International Conference on New Information Technologies for Decision making in Civil Engineering, Montreal, Canada,* published in Vol 1 of proceedings pp 379-388.

Turner, J. (1990). AEC Building system model. ISO TC/184SC4/WG1. Document 3.2.2.4. (Working paper).

Wix, J. and Bloomfield, D. (1997). Standards for information exchange and sharing: business benefits, *Information Technology Support for Construction Re-engineering, CIB Workshop W78, Cairns, Australia,* published in proceedings pp 409-418.

Wix, J. and Liebich, T. (1997). Industry foundation classes: architecture and development guidelines, *Information Technology Support for Construction Re-engineering, CIB Workshop W78, Cairns, Australia,* published in proceedings pp 419-432.