MODELLING MULTIPLE VIEWS OF DESIGN OBJECTS IN A COLLABORATIVE CAD ENVIRONMENT

M. A. ROSENMAN AND J. S. GERO
Key Centre of Design Computing
Department of Architectural and Design Science, University of Sydney
NSW 2006 Australia
Tel: 61-2-351 4610 Fax: 61-2-351 3031 Email: mike@arch.su.edu.au

ABSTRACT
Collaboration between designers in different disciplines is an increasingly important aspect in complex design situations, as exemplified in the AEC domain. CAD systems are essential for handling this complexity but current CAD modelling technology is directed towards the production of a single product model. In the AEC environment, many disciplines are involved, each with its own concept of the design object. Each such concept must be accommodated in any representation and this paper argues that a single model approach is inadequate. The ideas in this paper are based upon an assumption that different concepts of an object are based on different functional contexts. Thus the representation of the functional properties of design objects is the underlying basis for the formation of different concepts.

Keywords
conceptual modelling, multiple abstraction representation, CAD modelling, collaborative design, functional representation

INTRODUCTION

Large scale design projects involve many different disciplines each with their own area of concern and expertise. A large amount of information concerned with the representation of a design object is processed among each such discipline and between these disciplines. At various stages, this information represents different kinds of information and different abstractions and representations are used but eventually a consistent representation emerges which allows for the realization of the object. Currently, paper-based representations are the conventional method used for representations in AEC. Drawings are used to represent buildings and other structures. These drawings, actually contain only unstructured graphic entities such as lines, text and symbols. Through agreed conventions, structure and meaning is added by humans and these graphic entities are interpreted as a coherent structure of physical (or conceptual) elements. However, since the graphic entities are essentially unstructured and different kinds of agreements (knowledge) exist, these drawings may be interpreted in many ways. This is both a weakness (ambiguity) and a strength (flexibility). The use of CAD systems for representing design objects brings into focus these aspects of explicit/implicit representations and especially the requirement of different views and representations of the same design object by different design disciplines.

Systems Automation and Integration through CAD Modelling

It is being accepted that only through increasing automation of the design and construction process can the quality and efficiency of the design process in the AEC domain improve[1]. The key to success in achieving automation is seen as the integration of the information processing required by the various disciplines involved at the various stages of the design process. This is recognized in the research on concurrent engineering and collaborative design.[2,3,4] Computer-Aided Design (CAD) is accepted as the vehicle for providing this integrated information processing. An agenda of this view is the provision of a single CAD model from which various aspects of the model can be obtained as necessary. These aspects may be graphic as in plans, perspectives, etc or textual as in specifications, schedules, etc.

There is much current work concerned with producing conceptual modelling schema for the representation of design objects, for use in CAD.[5,6,7,8,9,10] There are also product modelling efforts within the work on data exchange standards.[11,12,13] concerned with achieving product descriptions. In building design, there are models such as the RATAS model[14,15] and the GARM model.[16,17] However, so far, these models seem to be extremely difficult to put into use, especially where different disciplines are involved. One of the main reasons for this is that they are based on producing a single fixed model of a building rather than on accommodating the different views that the different participants in the AEC disciplines may take.

In order to make CAD modelling useful to designers in a collaborative environment, such as the AEC domain, each designer's view and representation must be accommodated and integrated within a comprehensive representation of the design under concern. This paper argues that a single model view is inadequate and that a multiple view approach is essential for any meaningful representation in a multidisciplinary environment. Since views and representations depend upon a functional context, i.e. a particular set of functional concerns, the representation of functional properties is the essential aspect of the modelling of multiple representations.

Multiple Disciplines in AEC

In the AEC design environment many disciplines are involved, each dealing with a specialized aspect of the building design and each with its own concept and interpretation of the object. The fragmentation of the design and construction disciplines in the AEC domain is due to the specialization of each discipline according to functional concerns. Architects are mainly concerned with providing sufficient, efficient and aesthetic spatial environments for a given set of activities. They are thus concerned with the form and organization of spaces and those elements relevant to those purposes and with concepts such as spatial sufficiency, spatial organization, comfort, aesthetics, weatherproofness, rooms, storeys, facades, floors, walls, etc. Structural engineers, on the other hand, are concerned with providing stability by resisting or transmitting forces and moments. They are concerned with concepts such as gravity/lateral loads, support, bending, shear, deformations, beams, columns, shear walls, etc. Mechanical engineers are concerned with providing functions such as transportation and climate control through the provision of mechanical facilities, such as transportation systems and mechanical HVAC systems. They are concerned with concepts such as flow, capacity, time, energy and power, elevators, escalators, motors, coolers, heaters, piping, etc. Contractors, on the other hand, are concerned with the constructability of a design and hence with the relationships between the physical elements and the operations and sequence of operations required to construct the building. That is, they are concerned with concepts such as availability, composability, time and place, stability, walls, windows, beams, pipes, etc. Some aspects are the concern of more than one discipline, e.g. environmental aspects are the concern of both the architect and the mechanical engineer.

MULTIPLE VIEWS, MODELS AND INTERPRETATIONS OF A DESIGN OBJECT

There are two aspects to the notion of multiple views. The first is that given a syntactic description, different viewers will interpret that description differently, i.e. derive different meanings, while the second is that different viewers build different syntactic descriptions in the first place. This second aspect is the main thrust of this paper although the approach taken allows for the first aspect.

Multiple Views

We are not concerned with what an object is in absolute terms. What is of concern, is our perception, conception and representation of that object. Our view of an object depends on our collective experiences and concerns. A view is a predisposition towards certain conceptions. We build a conceptual model of an object based on that view, a representation, and manipulate that representation when we communicate. That is we communicate using representations. Within certain common groupings, such as design disciplines, there are, generally, common views and common understandings and agreements regarding the interpretation and description of objects thus leading to common representations. In a design context, the view that a person takes depends on the functional concerns of that person, where functional concerns include non-technical functions such as aesthetics, symbolism, psychological effects, etc. Given a design object, such as a building, there are many views that we may take, leading to different conceptual interpretations. For example, a building may be viewed as a set of activities that take place in it; as a set of spaces; as sculptural form; as an environment modifier or shelter provider; as a set of force resisting elements; as a configuration of physical elements; etc. In fact, a building is all of these.

Multiple Models

A model or abstraction of an object is a representation of that object resulting from a particular view taken. Since there are many different views of a building there will be many corresponding models, Figure 1.

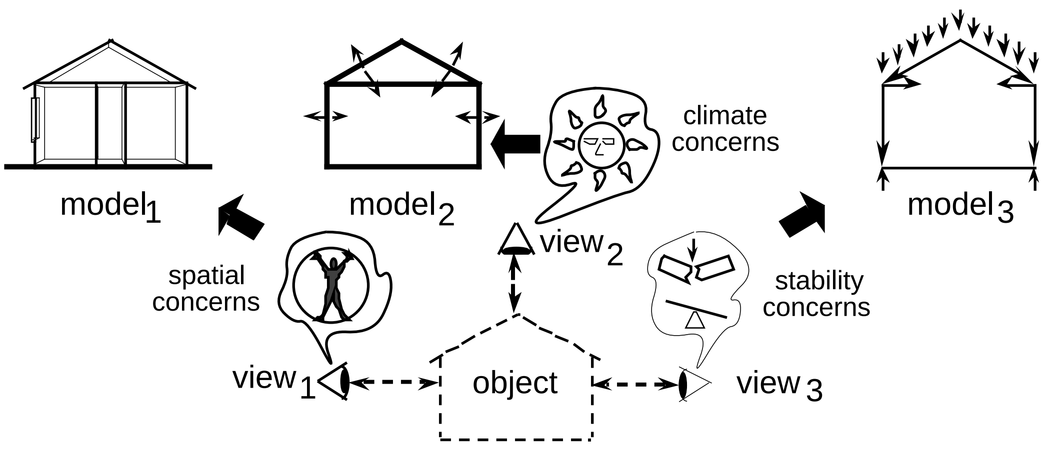


Figure 1. Multiple Views and Models

Depending on the view taken, certain properties and descriptions of the object become relevant. The sound insulating properties of a wall are not relevant to a structural engineer's description of that wall. In fact, certain walls may not be relevant at all to a structural engineer if they do not contribute to the building's stability (or instability through the addition of significant loads). Architects will model certain elements such as floors, walls, doors and windows. For the architects, these elements are associated with the spatial and environmental qualities with which architects are concerned. Structural engineers, however, see the walls and floors as structural elements capable of bearing loads and resisting forces and moments. While architects may model walls on different floors as separate elements the structural engineers may model only a single shear wall. While the 'architects walls' obviously form part of the engineers 'wall' the 'floor part' of the wall is missing from the architects' model and an aggregation of the 'architects walls' on there own will not produce the 'engineers' wall'. Both models must coexist since the structural engineers will need to carry out calculations based on their model while the architects may need to ascribe different properties to their separate wall elements, e.g. different finishes. The engineers may modify some of the properties assigned to these element by the architect and may add some new elements, such as beams and columns. The addition of such new elements may affect the architect's model and vice versa. While the architects are concerned with the spatial and environmental implications and the structural engineers with the structural implications, the contractors' view is an elemental one related to the erection of the building within given time and cost frames.

Thus any representation schema must allow for a dynamic evolution of models and must be capable of accommodating multiple concepts of a design unambiguously and consistently so that elements are not duplicated.

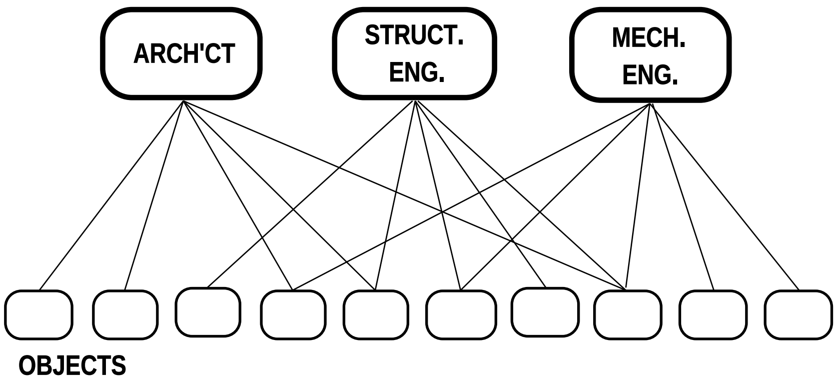
Multiple Interpretations

Given a syntactic representation of a design object, such as a graphic representation, different viewers will interpret that syntactic representation in different ways and arrive at different semantic interpretations of that object. An architect will interpret a given wall as a space-bounding element or as a noise barrier; a structural engineer as a load-bearing element; an HVAC engineer as a heat flow control element and a security engineer as a security-providing element. Clayton et al.[18] discuss this aspect of multiple interpretations and put forward mechanisms for the interpretation and critique of shared graphic objects. While this is an important aspect of multiple representation, the focus of this paper is on the multiplicity of the syntactic representation itself.

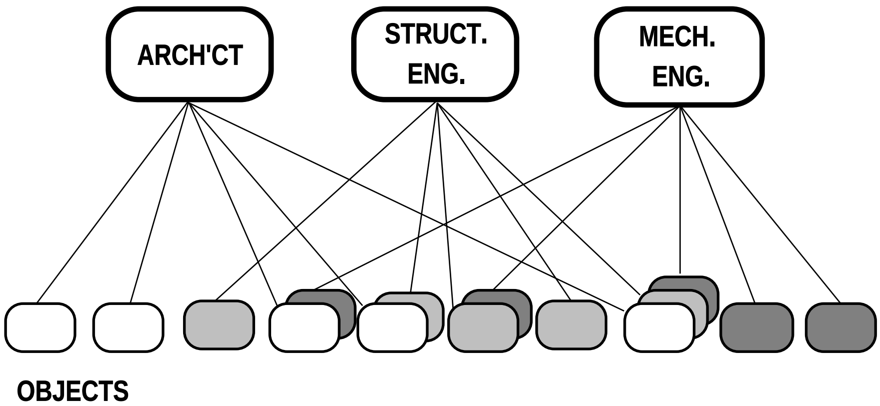
Representing Multiple Models

Using traditional methods of communications, each discipline represents its model in its own set of drawings (blueprints). Each such set of drawings represents that discipline’s model of the building using that discipline’s set of representation conventions. Any inconsistencies between the various models have to be discovered and corrected. This is done, traditionally, by marking the appropriate drawings and sending them back to the appropriate discipline. This process may go through several iterations. The result is a number of sets of drawings, one per discipline, where, although each set represents the building using a different model, the comprehensive representation is consistent. There is no attempt to integrate the various sets of drawings into one drawing. In CAD environments, the focus to date has been to construct a single unified model. The approach put forward in this paper is that this single model is inappropriate and that the traditional approach, in fact, represents a necessary approach which has to be dealt with in any electronic communicating medium such as CAD.

While Mackellar and Peckham[19] point out that there exist a single model approach and a multiple approach, their work is based on the single model approach. Clayton et al.'s SME[18] model is based on an initial model from which the different abstractions are made. However, it is not clear who produces this initial model and how the elements of that model can always form part of other disciplines' models. Howard et al.[20] put forward a data model using the primitive-composite approach. Multiple abstractions are formed much in the same way as views are formed in database management systems. A similar approach is taken by Amor and Hosking.[21] While we accept the basic premise that multiple abstractions can be formed though different compositions of these primitive elements, and indeed use that as a fundamental basis for our model, we question the fact that a single fixed model of primitive elements can be built. We argue that the primitive elements themselves are subject to the views taken by the different viewers and that different primitive models are constructed by each such viewer. That is, the basic description of an object differs from viewer to viewer. Each viewer may represent an object with different elements and different composition hierarchies. So that not only is the interpretation of the meaning of a design object different from one viewer to another but also the description of the structure of the object differs. No one model contains a comprehensive description of the object but each model must be consistent vis-a-vis the object being described. This approach is similar to that taken by Nederveen and Tolman[22] and Nederveen.[23] Figure 2 shows the differences between the two approaches.



(a) Primitive-Composite Approach



(b) Multiple Representation of Objects

Figure 2. Comparison of the Primitive-Composite and Multiple Representation Approaches

Thus, there exists no single unified model nor even a single set of unique elements but rather different descriptions of the same elements and different subsets of these descriptions in different models.

In most design situations, the models are not constructed concurrently but usually in an iterative sequence. For example, architects may construct a model followed by the structural engineers. The structural engineer's model may require modifications to the architect's model and the expression of relationships between elements in the architect's model and the engineer's model. Similarly, for other disciplines. Based on this new information, the architect may decide to make certain modifications to the design and so an iterative process ensues until a satisfactory consistent representation consisting of the various models is obtained.

Since the various models constructed by the various disciplines are representations of elemental models as seen through views based on functional contexts the representation of functional properties of design objects is the underlying basis for the formation of different concepts.

PURPOSE, FUNCTION, BEHAVIOUR AND STRUCTURE

Definitions

The essential factor in a description of any design object allowing for the formation of multiple interpretations is a description of its functional properties in addition to its structural properties. This is because the functional properties associated with a design object reflect the concerns of the various designers and their intent. There have been various attempts at defining the concepts of purpose, function, behaviour and structure.[24,25,26,27,28]. The approach taken here follows the definitions put forward in Rosenman and Gero for disambiguating these concepts.[29] The relations between purpose, function, behaviour and structure and the human socio-cultural environment and the physical object environment within the overall design object environment can be seen in Figure 3.

In summary:

> *structure* exhibits *behaviour* effects *function* enables *purpose*
> *purpose* is enabled by *function* is achieved by *behaviour* is exhibited by *structure*

Since, human needs are translated into required functions, functional concerns become the essential factors in satisfying those needs by mapping those required functions to the functional properties of proposed objects.

A design object may be described in terms of its structure, behaviour, function, or purpose e.g. a pencil may be described in structural terms as a cylinder (with certain dimensions) of graphite inside another cylinder (with certain dimensions) of wood, or in behavioural terms as something which controls the amount of graphite transferred to a contact, or in functional terms, as something which makes marks on paper, or in purposeful terms as an instrument for writing. In essence, a design object is all of these although, at the early stages of its design, we may only be able to describe it in terms of purposeful, functional and behavioural properties. Only after some identification of these as requirements can some embodiment take place and finally a detailed structural description. A design object may effect several functions. A wall separates two spaces (visually, physically and acoustically) and hence serves a space-partitioning function but it may also support another element and hence serve a structural or stability-providing function. Additionally, if it is an external wall, it prevents air and water penetration and inhibits thermal transfer and hence serves a climate control function.

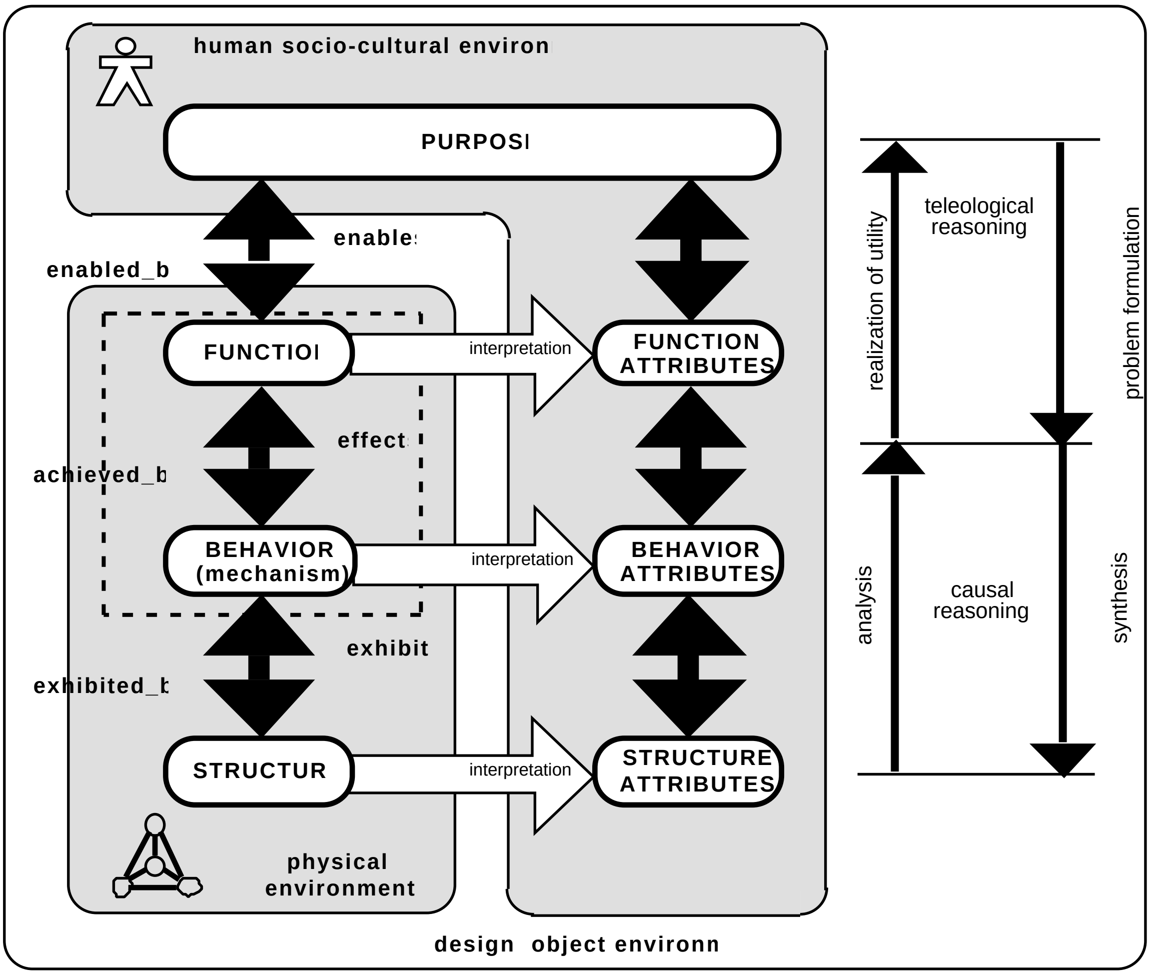


Figure 3. Concepts, Environments and Processes

The current practice in CAD systems is to represent merely the structural properties of an object. In many cases only the graphical representation of the object is described. The information regarding the object's functions is lost. It may be not always be possible to infer this information. For example, one cannot determine that a wall is loadbearing from topological relations alone. Experience in acquiring information from drawings in a case-based reasoning project at the Key Centre of Design Computing,[30] has shown that it is not possible to determine information such as whether a beam is part of the lateral force-resisting system, from the structural drawings, without recourse to the designers. Thus, the functional properties of a design object must be represented in any CAD information system. The recognition that graphical properties, while important, are not the only properties that need be described in an object's representation forms the underlying basis of the STEP effort for electronic data exchange of product information.[13,31]

Decomposition, Formulation and Specialization

Each of the concept of purpose, function and behaviour may be decomposed into more detailed concepts. Decomposition can be carried out along as many levels as it is felt necessary to explain the concept. When a satisfactory level of decomposition is reached in one category, the concepts thereby represented must be translated into concepts in a more operational category. Figure 4 shows the decomposition-formulation structure.

Each concept may be achieved by more than one (sub)concept and each concept may contribute to more than one (super)concept. For example, the function (of a car) of moving forward is achieved by the functions of the left and right rear wheels of rotating, while the function of providing (electrical) power of a car battery contributes to many functions.

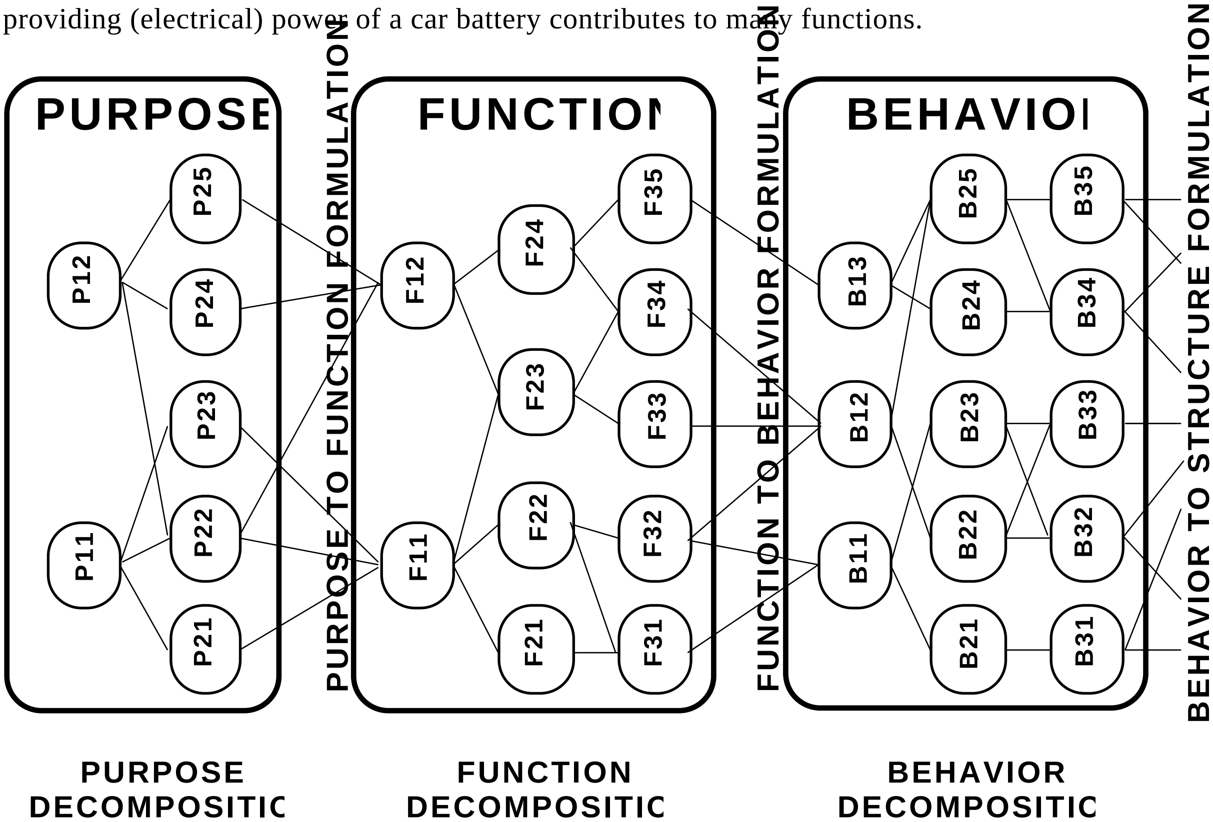


Figure 4. Decomposition and Problem Formulation

Another issue of importance is that of generalization/specialization. The class-subclass relationship among descriptions of objects means that the functions of subclasses must specialize the functions of the class. For example, all walls separate space in general. External walls separate internal space from external space whereas internal walls separate internal space from internal space. A particular internal wall (an instance) separates a particular internal space or room from another particular internal space or room. This class-subclass and class-instance specialization must be taken care of in any representation of function.

The above abstractions allow for relationships between the different properties, between levels in each property and between specific and general abstractions to be made. Thus implicit functional relationships can be found from e.g. behavioural properties.

DESIGN PROTOTYPES AND FUNCTIONAL SUBSYSTEMS

Design prototypes[6,7] describe classes of design elements. As such they encompass function, behaviour and structure properties as well as context and the relationships (in the form of knowledge) between these different factors. They are object-centred schemas similar to object-oriented programming objects but specifically dealing with design objects through their categorization of function, behaviour and structure properties. In a fragmented environment, such as AEC, each discipline has its own set of design prototypes with its own concepts, terminology and visual representation which are not necessarily shared between the disciplines. For, example, the structural engineer need not necessarily know about the concept 'wet-zone'. Specific examples of design prototypes, i.e. instances, are described using the design prototype schema and by instantiating all relevant properties to specific values.

While design prototypes describe a class of design objects, a complex design object (composed of more than one element) can also be regarded as a functional system composed of various functional subsystems, each of which carries out or contributes to the intended functions of the whole[33]. Eastman et al.[5] recognize this in their definition of a design object as a functional entity (FE) in the EDM model. Unlike a design prototype, a functional (sub)system, e.g. the climate control FS, is a purely functional concept without embodiment. It is represented by the functions it carries out and the behaviours required for those functions. For example, while beams, columns and walls are objects, the lateral force-resisting system is a functional subsystem which will itself not be found in any CAD graphic database. A similar approach is taken in the GARM model where Functional Units (FUs) and Technical Solutions (TSs) are differentiated.[16,17] An FS may be composed of other FSs, e.g. the lighting subsystem may be composed of the natural lighting FS and the artificial lighting FS.

At any time, new FSs may be formed by specifying new combinations of functions and/or FSs without restructuring of existing concepts. Design prototypes and functional subsystems form part of the general domain knowledge rather than project specific knowledge.

VIEWS AND MODELS

Views

A view is defined by a functional context, i.e. a given set of functions. A view prescribes the relevant FSs which in turn prescribe a particular model of a design object, i.e. which design prototypes, design elements and properties are relevant to that view. A view of a complex design object can therefore be formed by either directly selecting the relevant FSs or, alternatively, stating which functions a view(er) is concerned with. In the first case, the embodiment of these FSs w.r.t. their associated functions is the model associated with that view, while in the second case, the model is comprised of all those elements which have functions associated to those given. Such functions may have to be determined through implicit relationships in as described previously.

Models

Eventually, in any embodiment, a functional subsystem is embodied as a set of design elements whose functions contribute to those of the FS. For example, the natural lighting FS may be composed of the windows, light shafts and skylights. This relation between the FSs and the design elements, either design protoypes or specific instances, is achieved through the function properties of the design elements. The relation between FSs and the design elements may have to be found using decomposition and generalization hierarchies. No design element can (or should in a design representation) exist without being part of a functional subsystem. Otherwise it is redundant. The set of design elements and their properties and relationships resulting from the embodiment of a set of functional subsytems specified for a given view is the model associated with that view, Figure 5. Any design element may form part of several FSs if it carries out multiple functions, as shown in Figure 5.

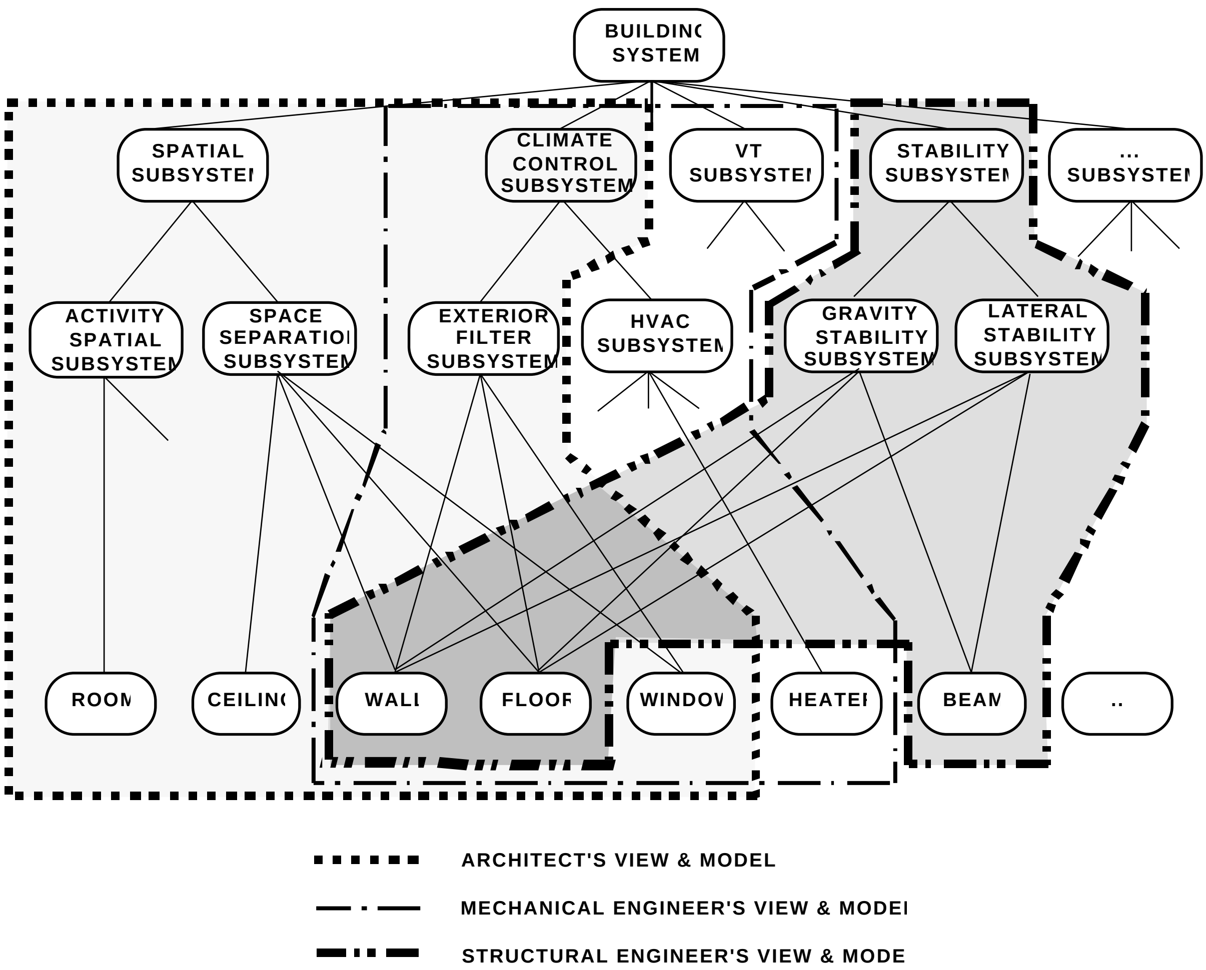


Figure 5. Functional Subsystems, Elements and Views

So that more formally:

given

$S = FS_1 \quad FS_2 \quad \ldots \quad FS_n$

$V_a = \{FS_k, \ldots\} \mid \{F_v, \ldots\}$

$FS_i = \{F_1, F_2, \ldots F_t\}$

$\mathbf{V} = \{V_1, V_2, \ldots, V_v\}$

then

$M_a = (E_a, R_a)$

such that

the set of functions of the elements in M_a, set of functions in V_a

where

S = system (design object)

FS_i = ith functional subsytem

$FS_i = FS_p \quad FS_q \quad \ldots \quad FS_w$

e_j = jth element or assembly

V_a = a particular view

M_a = particular model based on view Va

E_a = $\{e_g, ..., e_k\}$, i.e. the set of elements
R_a = $\{r_p, ..., r_w\}$, i.e. the set of relationships between the elements in E_a
$\mathbf{V}$ = set of all views
F_j = jth function

The above states that views are defined by specifying a set of relevant subsystems or, alternatively, a set of relevant functions. Views may be associated with particular viewers or viewer types. For example, the architect may specify a view, (V_{arch}), as {spatial, climate control} FSs and the structural engineer's view $V_{st\text{-}eng}$ as {stability} FS. Alternatively, views may be specified as a set of functions, e.g. V_{arch} may be specified as {enable_activity, separate_spaces, provide_access, control_climate, ...} while ($V_{st\text{-}eng}$) may be specified as {support_element, support_live_loads, resist_lat_loads, ...}. Views may be general to disciplines or specific to particular viewers. It is possible to construct a class hierarchy of views with inheritance from superclass to subclass. Any number of views of a design object can be formed at any time. New views may be formed by new combinations of functions and/or FSs. According to each different view there will be a model of the object as an embodiment of the functional subsystems. There is no necessity to attempt to consolidate these models into a single model. The totality of the representation does not become invalid as long as consistency is kept between the various abstractions of the same design elements.

Note the use of the union operator to ensure that, in any aggregation of functional subsystems, duplication of elements does not occur.[33]

Consistency Between Different Models

Although Figure 5 represents only the same single elemental concepts, it is possible for the different disciplines to refer to essentially the same element using different terminology, e.g. floor (architect), slab (structural engineer). In that case, the elements must be related through explicit relationships in each of the elements. Such relationships may be:

same_as: the element has all the properties of the named element or if applied to an individual property applies only to that property
element_of: the element is a component of the named 'element' (which in fact becomes an assembly)
part_of: the element forms part of the named element
constrained_by: a property of an element is constrained by a property of another element

Note the important difference between the *element_of* and *part_of* relationships.[32] A component forms part of an assembly but has properties which may be different from other components of the assembly, e.g. a wall as a component of a room assembly, whereas a part of an element has all the physical properties of the element and only differs in its geometric extent, e.g. floor of room1 is a part of the floor of storey1. Although, a part of an element is not strictly a design object, in a CAD database it is required to be a labelled entity for its identification and representation. In the *part_of* relation, any changes in one or other of the 'elements' vis-a-vis their properties other than some dimensions cannot be made without a corresponding change in the other. Such relationships between elements in the different models will have to be maintained through the use of constraints using procedures and demons. Mackellar and Peckham[19] use update rules for each view derivation relationship and constraints expressed as IF-THEN rules for integrity maintenance.

The result is a set of models where each model has its own concepts and elements. Some elements may be common to more than one model although some of their properties may differ and some elements in one model are related to elements in other models. This is shown in Figure 6.

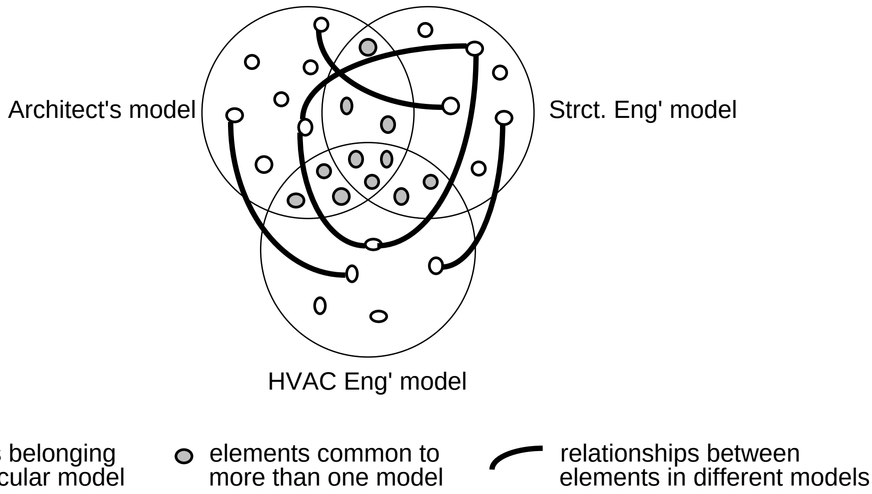


Figure 6. Models and Relationships

REPRESENTATION OF DESIGNS WITH CAD SYSTEMS

Integration of Graphic and Non-Graphic Properties

Most CAD systems only represent graphic entities such as points, lines, polylines and solids. These are stored in the CAD systems' graphic databases. Such a graphic representation allows for the interpretation of geometric properties such as shape, areas, volumes and relationships, such as adjacencies, based on distances. In many cases, graphic entities can be grouped and labelled so as to be interpreted as design objects and, in some cases, attributes can be attached to these entities. Where these attributes are not graphic, they are usually for notational purpose only and cannot be used in any reasoning process, although some physical properties, such as material, may be used in the calculation of quantities in systems such as ARCHICAD[33]. Other, non-graphic, properties are usually stored in databases, such as relational databases managed by database management systems (RDBMS). Any links that exist between such RDBMSs and the graphic databases of CAD systems are no more than relating an element in the graphic database to a record in the database. For example the AES system[34] provides communications with several RDBMS, such as INGRES.[35] Pairing is provided between the graphic elements and records in the database which allows identification of graphic objects in the graphic representation through the names of entities and, vice versa, the identification and display of properties of graphic entities picked from the graphic model. While this allows some association between the two, it does not guarantee consistency since information in one can be entered and modified independently of the other. While RDBMSs are common and powerful in storing and manipulating data about object instances, they cannot easily handle the class relationships and information required in design representation nor provide such mechanisms as inheritance through generalization/ specialization/instantiation constructs. Object-oriented databases (OODBMS) are more suited to this purpose

An integrated approach is required for the modelling of design objects whereby all aspects of the design object are related. Given the nature of current CAD systems, this is usually implemented by some control process written in some external language which can communicate with the CAD system and the DBMS (RDBMS or OODBMS). In the case of AES this can be written in the command language provided by AES.[36,37] Another alternative is to use the C interface as allowed by AES and AutoCAD.

Graphic Representation of Models

Each model based on a different view will require a different graphic representation of elements. This includes representation for applications, such as analysis, and also for visualization. For example, for one application, an element may need be represented by solid geometry whereas, for another application, a surface model may be required or even a centre line representation. More than one such representation may be needed for the same discipline. Architects may use a line representation at the early stage of design and a surface model at a more developed stage. Each representation must be consistent vis-a-vis the other one. Eastman et al.[5] show how a wall may be represented by several geometry FEs, based on predefined FEs of multi-lines, Symbols, Poly-lines and B-shapes. Consistency is maintained through the use of FE constraints and accumulations. Another way would be to have a single fundamental representation, e.g. a surface approach, from which other representations could be derived through the use of procedural transformations.

A BUILDING EXAMPLE

Figure 7 shows an example of a two-storey apartment block, BLDG1. This example is a simplified one but is sufficiently general in its demonstration of the need for the representation of multiple concepts to allow for multiple abstractions of a design object. At the beginning of each CAD session users will identify themselves by their view which must be predefined. Thus, only the relevant design prototypes and functional contexts will be addressed. Figure 8 shows part of BLDG1 as represented by an architect using a CAD modelling system to represent objects. Figure 7 also shows those entities that are being modelled by the architect as may be stored in a database.

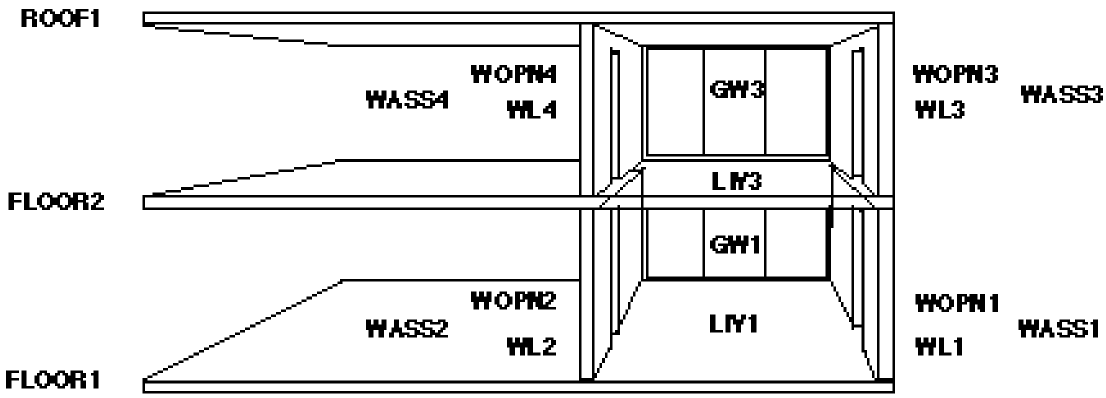


Figure 7. Architect's CAD model

The part model as shown in Figure 7 contains 13 building elements, namely FLOOR1, FLOOR2, FLOOR3, WL1, WL2, WL3, WL4, WOPN1, WOPN2, WOPN3, WOPN4, GWL1, GWL2 and 4 element aggregations, namely, WASS1, WASS2, WASS3, WASS4, created through an aggregation of the elements (WL1, WOPN1), (WL2,WOPN2),....Other entities will be defined by the architect, e.g. STOREY1, STOREY2, FLAT1, ..., FLAT4 and relations defined between these and the building elements and spaces. Figure 8 shows some of these entities with some properties as defined by the architect during the modelling process, also stored in the database. This instance information follows the schema as defined in the appropriate design prototypes.

WL1

AN_INSTANCE_OF	internal_wall
FUNCTION:	separate_space (STAIR1, LIV1)
BEHAVIOUR:	transparency, sound transmission, ...
STRUCTURE:	
ELEMENT_OF:	WASS1
SHAPE:	rect_prism
LENGTH:	7200
HEIGHT:	2400
THICKNESS:	200
MATERIAL:	concrete block
LOCATION:	...

WOPN1

AN_INSTANCE_OF	wall _opening
FUNCTION:	provide_access (STAIR1, LIV1)
BEHAVIOUR:	ease of passage, ...
STRUCTURE:	
ELEMENT_OF:	WASS1
SHAPE:	rect_prism
WIDTH	900
HEIGHT:	2100
THICKNESS:	200
LOCATION:	...

WASS1

AN_INSTANCE_OF	wall _assembly
FUNCTION:	separate_space (STAIR1, LIV1) provide_access (STAIR1, LIV1)
BEHAVIOUR:	ease of passage, ...
STRUCTURE:	
ELEMENT_OF:	FLAT1
ELEMENTS:	WL1, WOPN1
SHAPE:	rect_prism
LENGTH:	7200
HEIGHT:	2400
THICKNESS:	200
LOCATION:	...

GW1

AN_INSTANCE_OF	glass_wall
FUNCTION:	separate_space (EXT, LIV1) allow_light (LIV1)
BEHAVIOUR:	transparency ...
STRUCTURE:	
ELEMENT_OF:	FLAT1
SHAPE:	rect_prism
LENGTH:	4000
HEIGHT:	2400
THICKNESS:	100
LOCATION:	...

Figure 8. Instance information from architect's model

On the other hand, the structural engineer models the elements shown in Figure 9.

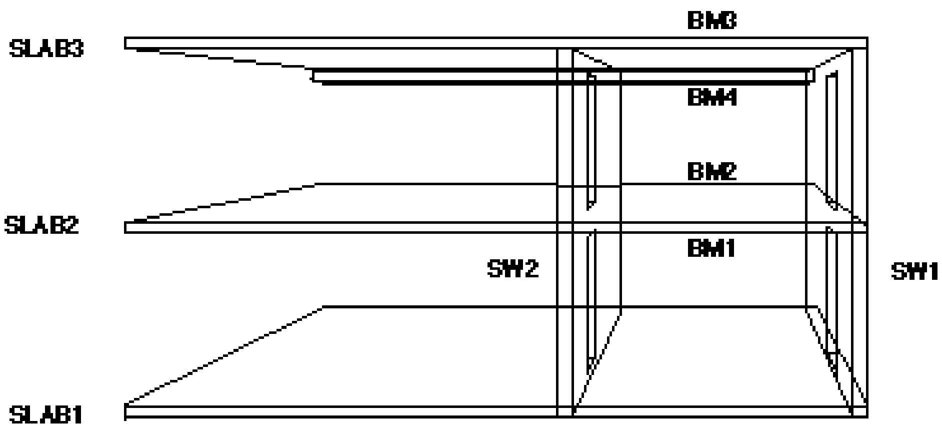


Figure 9. Structural engineer's CAD model

This model contains only 9 elements, namely SLAB1, SLAB2, SLAB3, SW1, SW2, BM1, ..., BM4, where SW1 and SW2 are shear walls whose properties, as defined by the engineer, are given in Figure 10(a). Based on the view of the building as a force-resisting/force-transmitting object, the structural engineer does not see WASS1 and WASS3 as does architect but rather SW1. S/he may modify some of the properties of this wall, e.g. the thickness and material. This must then be reflected back in the architect's model. Links must be made to the fact that WASS1 and WASS3 are related to SW1, so that any modification to one or the other causes a modification to the properties of the others. Thus WASS1 and WASS3 need be defined as *part_of* SW1 rather than as *element_of*. SLAB3 is synonymous to ROOF1 as an element and must be noted as such using the *same_as* relationship. The addition of the edge beams, BM1, ..., BM4, will cause modifications to the height of the glass walls GW1, ..., GW4, and a relationship between the height of the glass walls, the depth of the beams and the storey height has to be noted using the *constrained_by* relationship. In addition, the beams will be included in the exterior_filter subsystem since their waterproofness, thermal transmittance, etc are relevant factors. The changes to the architect's elements are shown in Figure 10(b).

SW1

AN_INSTANCE_OF	shear_wall
FUNCTION:	support (SLAB2) support (SLAB3) resist_lateral_force (50)
BEHAVIOUR:	strength, shear, ...
STRUCTURE:	
ELEMENT_OF:	BLDG1
PARTS:	WASS1, WASS2
SHAPE:	rect_prism
LENGTH:	7200
HEIGHT:	5200
THICKNESS:	200
MATERIAL:	r.c.
LOCATION:	...

SLAB1

AN_INSTANCE_OI	floor_slab
SAME_AS:	floor1
FUNCTION:	support_live_loads (50)
BEHAVIOUR:	..., bending, shear ...
STRUCTURE:	
ELEMENT_OF:	BLDG1
SHAPE:	rect_prism
LENGTH:	10200
WIDTH:	7200
THICKNESS:	200
MATERIAL	r.c.
LOCATION:	...

(a) Structural engineer's instances

WASS1

...	...
STRUCTURE:	
ELEMENT_OF	FLAT1
ELEMENTS:	WL1, WOPN1
PART_OF	SW1
...	...

GW1

...	...
STRUCTURE:	
ELEMENT_OF	FLAT1
SHAPE:	rect_prism
LENGTH:	4000
HEIGHT:	2100 : CONSTRAINED_BY : BM1: depth
...	...

WL1

...	...
STRUCTURE:	
ELEMENT_OF	WASS1
...	...
MATERIAL:	SAME_AS: SW1
...	...

(b) Modified architect's instances

Figure 10. Instance information from structural engineer's model

Figure 11 shows part of the resulting functional system model from which the architect's and structural engineer's models can be constructed. Only some of the elements and concepts are shown for clarity. Furthermore, contractors may construct their model according to an elemental functional decomposition based on completing construction stages. For example, they may model SL2, BM1 and BM2 as a single channel aggregation, CH1 if they intend to pour that as a single element.

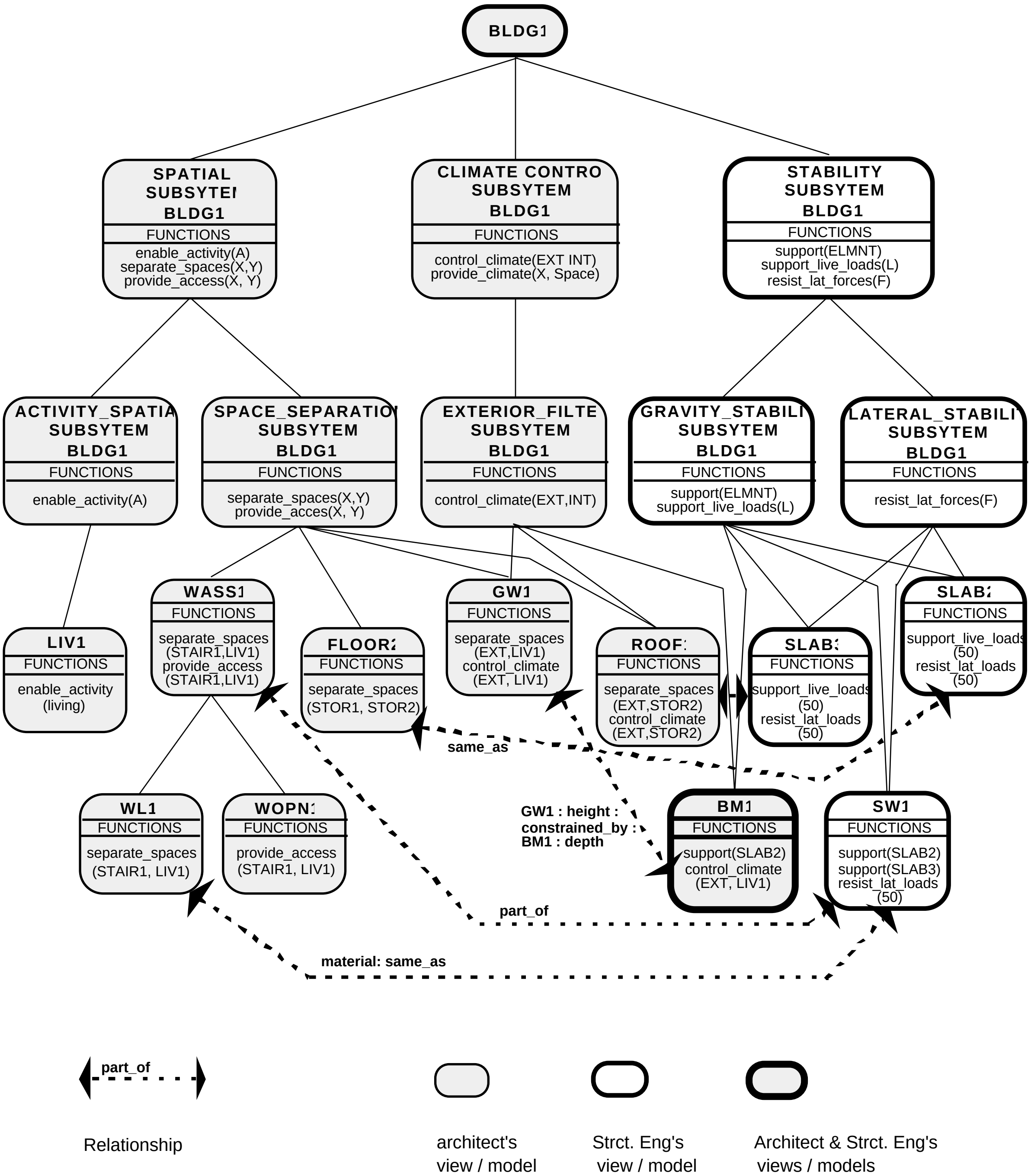


Figure 11. Combined architect's and structural engineer's functional subsystems and models

COMPUTER-SUPPORTED COLLABORATIVE DESIGN

In a computer-supported collaborative design environment there will be a need for several models of the design object, e.g. building, to be represented at any time. For example, the architect and the structural engineer may collaborate on a building with each having both models on view. The models include both graphic and non-graphic representations. When one model is manipulated, corresponding effects may have to be made in the other. At the very least, some form of alert must occur, whether it be in textual, graphic or audial form. When, for example the structural engineer selects a shear wall for discussion, the corresponding walls in the architect's model should be highlighted. If the dimensions of that wall or its material are changed by the structural engineer, this should be reflected in the architect's model.

In synchronous situations, the notification of changes or proposed changes can be notified by direct visual or audial notification. However, consistency must be maintained even in asynchronous exchange of information. Moreover, the above example is for cases where explicit relationships exist between elements in the different models. In the type of case where, for example, the structural engineer adds some columns to a space in the building, this must also be reflected in the architect's model. This can only be so if columns exist as concepts in the architect's domain and are related to some functions with which the architect is concerned, such as space occupancy in this case. In that case, when instances of columns are created by the structural engineer, even though they will only be ascribed structural properties, they will be related to column concepts in the architect's model and be added to the architect's model.

IMPLEMENTATION

The modelling of multiple views has been implemented using the CAD system, AES, and the INGRES RDBMS under the AIX environment on IBM RISC systems/600 workstations.[37] AES is a 3-D surface modeller where design objects may be represented a symbols. In addition, it has an interface capability for associating graphic entities in its graphic database with design objects stored as records in the INGRES database and allowing queries on the database using a subset of the SQL language.

The implemented system has three subsystems, namely the graphic system, the design object database system and the semantic query system.

Graphic System

The graphic system uses the graphic module of AES. The graphic system also includes the interface to the system through control commands written in the AES command language. These establish interface links between the graphic objects and the design objects.

Design Object Database System

The INGRES database contains all the non-graphic information represented by all design disciplines. It contains both class and instance information and relates the representations of abstractions of the design objects according to different views (disciplines).

Semantic Query System

The semantic query system allows for semantic information to be displayed for design objects selected either by name or picked in the graphic representation. In addition it displays the various graphic (and non-graphic) representations of the different models according to the different views.

Hwang[37] has demonstrated the formation of multiple semantic interpretations and multiple models with a townhouse example. Three models are demonstrated, namely architect's, structural engineer and HVAC engineer. Given a particular graphic object, e.g. as picked in a graphic representation, the various semantic interpretations according to the different views are given. Alternatively, given a particular viewpoint, a model, i.e. a graphic and semantic representation is produced.

CONCLUSIONS

This paper has shown that current single fixed representations are inadequate to model the various concepts that are present in multidisciplinary design situations. It has put forward concepts and demonstrated a methodology for the construction of a flexible and dynamic representation of multiple views of a design object based on functional contexts. The essential factors are the representation of functional properties of design objects and the definition of functional subsystems allowing different interpretations of design objects to be constructed through the definition of views as functional contexts. This allows for a dynamic construction of models rather than a static explicitly defined membership. The addition of relations between the same elements in different models is critical for consistency.

The work to date has demonstrated the potential for CAD systems to allow the modelling of different views through the linking of graphic and non-graphic databases using a graphic database a relational database and an interface command language. However, as pointed out previously, relational databases have limited capabilities in representing class information. Future work will look at the use of OODBMSs or the design prototype schema for representing such domain class information. For this, the interface between the various components of the system will need to be written in the C language. This should also provide more general capabilities than at present are available under the AES command language.

ACKNOWLEDGEMENTS
This work is supported by the Australian Research Council.

REFERENCES

1. Madison (1991). *Conference papers, 1st Int. Symposium Building Systems Automation-Integration, June 2-8, Madison, Wisconsin*, Dept. of Eng. Professional Development, College of Engineering, University of Wisconsin-Madison/ Extension.
2. Sriram, D., Logcher, R., Wong, A. and Ahmed, S. (1991). Computer-aided cooperative product development: a case study, *Int. Journ. of Systems Automation: Research and Applications (SARA)*, **1**:91-114.
3. Prasad, B. Morenc, R. S. and Rangan, R. M. (1993). Information management for concurrent engineering: Research issues, *Concurrent Engineering: Research and Applications (CERA)*, **1**(1):3-20.
4. Saad, M. and Maher, M. L. (1993). A computational model for synchronous collaborative design, *Preprints of AAAI-93 Workshop on AI in Collaborative Design*, Washington , D.C., pp.191-206.

5. Eastman, C. , Bond, A. and Chase, S. (1991). A data model for design databases, *in* J. S. Gero. (ed.), *Artificial Intelligence in Design '91*, Butterworth-Heinemann, Oxford, pp.339-365.
6. Gero, J. S. (1990). Design prototypes: a knowledge representation schema for design, *AI Magazine*, **11**(4):26-36.
7. Gero, J. S. and Rosenman, M. A. (1990). A conceptual framework for knowledge-based design research at Sydney University's Design Computing Unit, *Artificial Intelligence in Engineering*, **5**(2): 65-77.
8. MacKellar, B. K. and Ozel, F. (1991). ArchObjects: design codes as constraints in an object-oriented KBMS, *in* J. S. Gero (ed.), *Artificial Intelligence in Design '91*, Butterworth-Heinemann Ltd, Oxford, pp.115-134.
9. Nguyen, G. T. and Rieu, D. (1991). Representing design objects, *in* J. S. Gero (ed.), *Artificial Intelligence in Design '91*, Butterworth-Heinemann Ltd, Oxford, pp.367-386.
10. Myers, L., Pohl, J., Cotton, J., Snyder, J., Pohl, K. J., Chien, S-F., Aly, S. and Rodriguez, T. (1993). Object representation and the ICADS-Kernel Design, *Design Institute Report CADRU-08-93*, CAD Research Center, College of Architecture and Environmental Design, Cal Poly State University, San Luis Obispo, CA.
11. Fowler, J. (1991). STEP modelling methods SADT, NIAM, IDEF1X, EXPRESS-G and EXPRESS, *Product modelling and the STEP standard-seminar*, Technical Research Centre of Finland, Espoo.
12. Nijssen, G. M. and Halpin, T. A. (1989). *Conceptual Schema and Relational Database Design*, Prentice-Hall, New York.
13. Spiby, P. (1991). Product data representation and exchange - *Part 11: The EXPRESS language reference manual, CD 10303 -11, ISO TC 184/SC4 N83*, National Institute of Standards and Technology, Gaithersburg, MD.
14. Bjork, B-C. (1987). RATAS: A proposed Finnish building product model, *Studies in Environmental Research No. T6*, Helsinki University of Technology, Otaneimi, Finland.
15. Bjork, B-C. (1989). Basic structure of a proposed building product model, *CAD*, **21**(2):71-77.
16. Gielingh, W. F. (1989). General AEC Reference Model (GARM), *ISO TC 184/SC4/WG1 Document N329*.
17. Nederveen, S. V., Plokker W. and Rombouts, W. (1991). A building data modelling exercise using the GARM approach, *COMBINE Report working draft*.
18. Clayton, M. J., Fruchter, R., Krawinkler, H and Teicholz, P. (1994). Interpretation objects for multi-disciplinary design, *in* J. S. Gero and F. Sudweeks (eds), *Artificial Intelligence in Design ‘94*, Kluwer Academic Publishers, Dordrecht, Netherlands, pp.573-590.
19. Mackellar, B. K. and Peckham, J. (1994). Specifying multiple representations of design objects in SORAC, *in* J. S. Gero and F. Sudweeks (eds), *Artificial Intelligence in Design ‘94*, Kluwer Academic Publishers, Dordrecht, Netherlands, pp.555-572.
20. Howard, H. C., Abdalla, J. A. and D. Phan, D. H. (1992). Primitive-composite approach for structural data modelling, *Journal of Computing in Civil Engineering*, **6**(1):19-40.
21. Amor R W and Hosking J G (1993). Multi-disciplinary views for integrated and concurrent design, *in* K. S. Mathur, M. P. Betts and K. W. Tham (eds), *Management of Information Technology for Construction*, World Scientific, Singapore, pp.255-267.
22. Nederveen. G. A. van and Tolman, F. P. (1992). Modelling multiple views on buildings, *Automation in Construction*, **1**:215-224.
23. Nederveen, S. V. (1993). View integration in building design, *in* K. S. Mathur, M. P. Betts and K.W. Tham (eds), *Management of Information Technology for Construction*, World Scientific, Singapore, pp.209-221.
24. Rodenacker, W. (1984). *Methodisches Konstruieren*, 3rd. ed. Berlin, Springer Verlag.
25. DeKleer, J. and Brown, J. S. (1984). A qualitative physics based on confluences, *Artificial Intelligence*, **24**:7-83.
26. Bobrow, D. G. (1984). Qualitative reasoning about physical systems: an introduction, *Artificial Intelligence*, **24**:(1-5).
27. Umeda, Y., Takeda, H., Tomiyama, T. and Yoshikawa, H. (1990). Function, behaviour, and structure, *in* J. S. Gero (ed.), *Applications of Artificial Engineering in Engineering V, Vol1: Design*, Computational Mechanics Publications, Southampton, pp. 177-193.
28. Hundal, M. S. (1991). Conceptual design of technical systems, *Proceedings of the 1991 NSF Design and Manufacturing Systems Conference*, Society of Manufacturing Engineers, Michigan, pp.1041-49.
29. Rosenman and Gero (1994). The what, the how, and the why in design, *Applied Artificial Intelligence*, **8**(2):199-218.

30. Balachandran, M., Villamayor, R. and Maher, M. L. (1992). Using past design cases to support structural system design for buildings, *Progress Report for Acer Wargon Chapman Associates*, Design Computing Unit, Department of Architectural and Design Science, University of Sydney, Sydney.
31. STEP (1991). Part 1: Overview and fundamental principles, Draft N14, ISO TC 184/SC4/ WG6.
32. Rosenman, M. A. (1993a). Dynamic decomposition strategies in the conceptual modelling of design objects (with special reference to buildings), *Concurrent Engineering: Research and Applications (CERA)*, **1**:21-29.
33. Graphisoft (1989). *ArchiCAD, Version 3.4, Users' Manual*, Graphisoft.
34. IBM and SOM (1991). *The IBM Architecture and Engineering Series*, International Business Machines Corporation and Skidmore, Owings and Merill, Kingston, NY.
35. Relational Technology Inc. (1990). *INGRES/SQl Reference Manual*, Release 6.3, UNIX, Alameda, CA.
36. Rosenman, M. A. (1993b). *Design Object Modelling Using AES and INGRES*, Paper distributed at the Asia Pacific CAD Operators' Exchange South Conference, August 15-18, Sydney, Department of Architectural and Design Science, University of Sydney.
37. Hwang, Y. S. (1994). *Design Semantics and CAD Databases*, PhD Thesis, Department of Architectural and Design Science, University of Sydney, (unpublished).

This paper is a copy of: Rosenman, M. A. and Gero, J. S. (1996). Modelling multiple views of design objects in a collaborative CAD environment, *INCIT 96 Proceedings*, The Institution of Engineers, Australia, ACT, Australia, pp.49-62.