Representation of Multiple Concepts of a Design Object Based on Multiple Functions

M A ROSENMAN[1]
J S GERO[2]
Y-S HUANG[3]

ABSTRACT

Current representation schemas for design objects in CAD environments make assumptions regarding particular representations of that design object. In the AEC environment, many disciplines are involved, each with its own concept of the design object. Each such concept must be respected and accommodated in any representation. This paper presents the ideas behind the representation of multiple concepts from an underlying description of a design such that the inter- and intra-discipline views of that design can be formed dynamically. These ideas are based upon an assumption that different concepts of an object are based on different functional contexts. Functional subsystems are introduced as an adjunct to design prototypes. An example shows how these functional subsystems are related to the design elements and how they allow for the formation of the various concepts. Thus the representation of the functional properties of design objects is the underlying basis for the formation of different concepts.

Key Words

information technology; conceptual modeling; multiple abstraction representation; building design; function

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. In conventional AEC, drawings are used to represent buildings and other

1 *Research Fellow, Key Centre of Design Quality, University of Sydney, Australia 2006.*

2 *Professor, Department of Architectural and Design Science, University of Sydney.*

3 *PhD Student, Department of Architectural and Design Science.*

structures. These drawings, in fact 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 contain the ability to be interpreted in a multitude of ways. This is both a weakness (ambiguity) and a strength (flexibility).

Systems Automation and Integration through CAD Modeling

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, (Madison, 1991). 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.

There is much current work concerned with producing conceptual modeling schema for the representation of design objects. However, these models seem to be extremely difficult to put into use in a general sense. In CAD databases, all representations of entities have to be explicitly stated. This includes both graphic representations and representations of other properties, whether in a single database or in separate graphic and relational databases. In contrast to conventional paper drawings, the descriptions in a CAD system make assumptions regarding particular representations of that design object and produce fixed and static representations. 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.

Multiple Disciplines in AEC

In the AEC design environment, where many disciplines are involved, each discipline will have its own concept of the object. Each such concept may be formed incrementally over a period of time but must be respected and hence accommodated in any representation.

For example, architects will lay out certain elements such as floors, walls, doors and windows. These elements may or may not have certain material and dimensions assigned to them. For the architects, these elements are associated with the spatial and environmental qualities with which they are concerned. Structural engineers, however, see the walls and floors in a different light, namely as structural elements capable of bearing loads and resisting forces and moments. They may see a set of walls on different floors as a single shear wall. 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. Thus any representation schema must allow for a dynamic model capable of accomodating multiple concepts of a design in an unambiguous and consistent model so that elements are not duplicated.

This paper presents the ideas behind the representation of multiple concepts from an underlying description of a design such that inter- and intra-discipline views of that design can be formed dynamically. These ideas are based upon an assumption that different concepts of an object are based on different functional contexts. Thus, the representation of functional properties of design objects is the underlying basis for the formation of different concepts.

MULTIPLE VIEWS OF A DESIGN OBJECT

Given a design object, such as a building, there are many views that one may take. For a particular 'viewer' only some (or one) of these views may be relevant. Depending on the view taken, certain properties and descriptions of the object are 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).

The fragmentation of the design and construction disciplines in the AEC domain is due to the specialization of each discipline according to functional aspects. 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.

Notwithstanding the above, in order for CAD to be useful in the AEC domain, a comprehensive representation of a building must be able to be built from which various views of it can be formed depending on the particular need. Howard et al (1992) put forward a data model using the primitive-composite approach. 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. No one model contains a comprehensive description of the object but each model must be consistent vis-a-vis the object being described. 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 engineers' model. Similarly, for other disciplines. Based on this new information, the architect may decide to make certain modifications to the design and so an interative process ensues until a satisfactory consistent representation consisting of the various models is obtained.

It will be shown that the various models constructed by the various disciplines can be achieved through an approach based on representations of elemental models as seen through views based on functional contexts.

STRUCTURE, BEHAVIOUR AND FUNCTION

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. Bobrow (1984) defines the following: function is *what* an object does, behaviour is *how* the object does what it does and structure is what the object *is*. Using those definitions, the function of a clock is to tell the time, a behaviour is that the hands rotate with a fixed periodicity and its structure is that is a particular configuration of metal, wood and glass, etc. Artificial objects are conceived and realized to satisfy given human needs. Thus, function is related to the intended purpose or utility of a design object, ie to the reason for its existence (although functions may be found which were not intended). Behaviour is the totality of the properties of an object which emerge as a result of the interaction between the object's structure and its environment. The structure of a physical object is its physical embodiment, ie in terms of material, toplogy and geometry. A structural description includes those properties which are necessary and sufficient to allow the object to be realized. Required functions give rise to required behavioural properties which enable those functions to be carried out. Required behaviours are satisfied by various structural properties. The structural properties include those factors about which designers make decisions in order to realize a design so that the actual behaviours of the object will satisfy the required behaviours and, as a result, satisfy the intended functions.

A design object may be described in terms of its structure, behaviour or function, eg 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 makes marks on paper, or in functional 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 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 fulfill 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. Current practice is to use a CAD system to represent merely the structural properties of an object. The information regarding the object's intended functions is lost. While, in some cases, it may be possible to infer this infomation this cannot always be done. For example, one cannot determine that a wall is loadbearing from topological relations alone. Thus, the functional properties of a design object must be represented in any CAD information system.

Each function of a design object is related to various structure properties of that object through particular behaviour properties of that object. These relationships can be expressed in a function-behaviour-structure network (Gero et al, 1991) as shown in Figure 1. It can be seen, from Figure 1, that given particular functions, different behaviour properties and hence different structure properties become relevant.

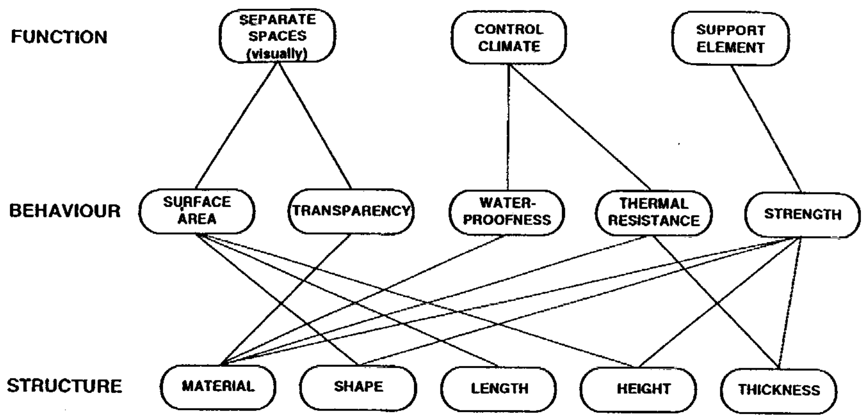


Figure 1. Function-behaviour-structure Dependency Network

DESIGN PROTOTYPES AND FUNCTIONAL SUBSYSTEMS

Design prototypes (Gero, 1990, Gero and Rosenman, 1990, Rosenman and Gero, 1989) 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, ie 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. Eastman (1991) recognizes this in his definition of a design object as a functional entity (FE) in the EDM model. The difference between a functional (sub)system (FS) and a design prototype is that a functional (sub)system, eg 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 (Gielingh, 1989; Nederveen, 1991). An FS may be composed of other FSs, eg the lighting subsystem may be composed of the natural lighting FS and the artificial lighting FS. 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. No design element can (or should in a design representation) exist without being part of a functional subsystem. Otherwise it is redundant.

Any design element may form part of several FSs if it carries out multiple functions, see Figure 2. Although Figure 2 represents only the same single elemental concepts, it is possible for the different disciplines to refer to essentially the same element using different terminology, eg 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 (Rosenman, 1993). A component forms part of an assembly but has properties which may be different from other components of the assembly, eg 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, eg 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.

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.

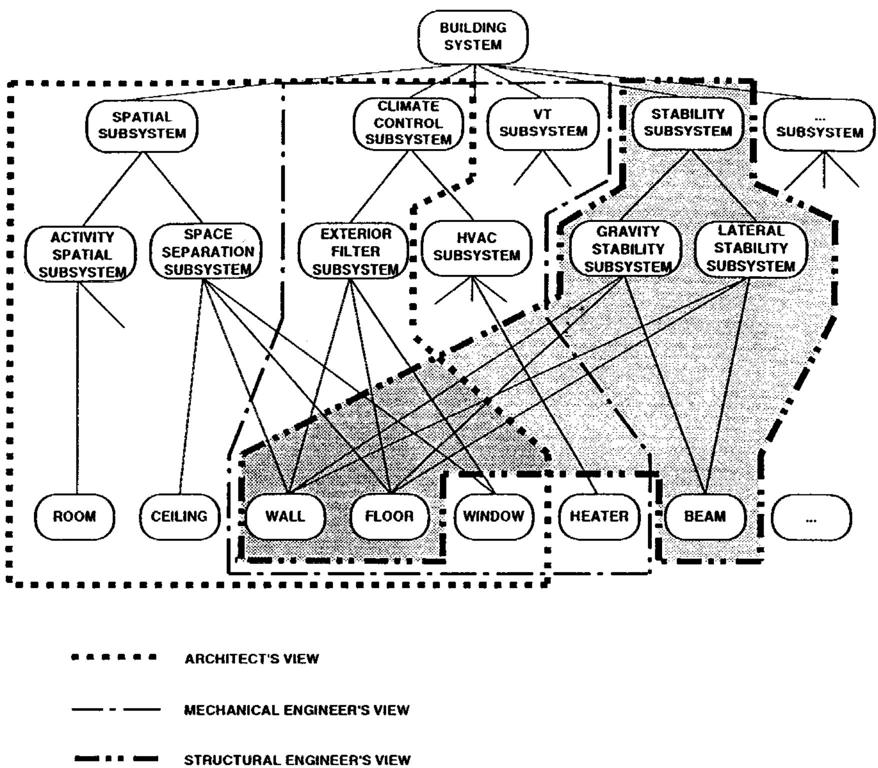


Figure 2. Functional Subsystems, Elements and Views

VIEWS AND MODELS

A view is defined by a functional context, ie a given set of functions. A view prescribes the relevant FSs which in turn prescribe a particular model of a design object, ie 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. So that:

given

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

$V_a = \{Fv, ...\}$

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

$V = \{V_1, V_2, ..., Vv\}$

then

$V_a = FS_i$

$M_a = \{e_g, ..., e_k\}$

such that

the set of functions in V_a, i.e. $\{Fv, ...\}$ set of functions in FS_i

set of functions in FS_i the set of functions in M_a

the set of element representations in M_a is unique

where

S = system (design object)

FS_i = ith functional subsytem

$FS_i = FS_p \quad FS_q \quad ... \quad FS_w \mid \{e_1, e_2, ..., e_m\}$

e_j = jth element (or assembly)

V_a = a particular view

M_a = particular model based on view Va

V = set of all views

F_j = jth function

The above states that a view is defined by specifying a set of relevant functions. Those FSs which include those functions (as well as others) are selected. The embodiment of these FSs w.r.t. to those functions is the model. This means that as FSs are decomposed into more specific FSs, only those FSs that are relevant w.r.t. the specified functions are retained. Alternatively, a view can be defined directly as a set of functional subsystems. In this case, all the FSs forming part of the specified FSs will be retained.

Note the use of the union operator to ensure that, in any aggregation of functional subsystems, duplication of elements does not occur, (Rosenman, 1993).

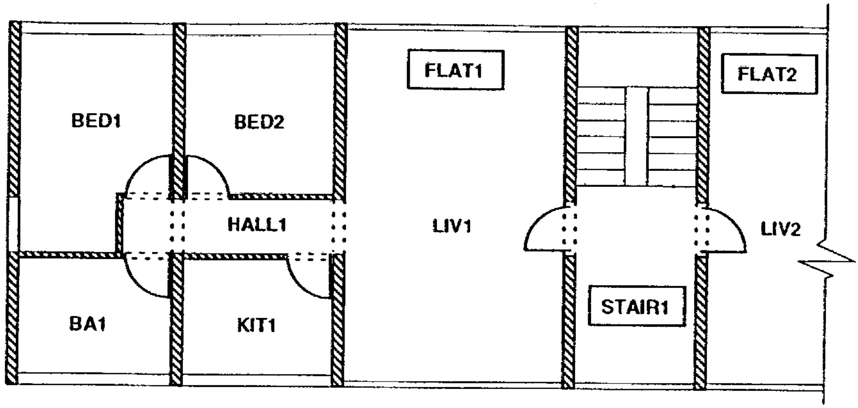
Views are formed by the various design participants specifying those functions relevant to their discipline. For example, the architect may specify a view (V_{arch}) as {enable_activity, separate_spaces, provide_access, control_climate, ...} while the structural engineer's view ($V_{st\text{-}eng}$) may be specified as {support_element, support_live_loads, resist_lat_loads, ...}. Alternatively, V_{arch} may be specified as {spatial, climate control} FSs and $V_{st\text{-}eng}$ as {stability} FS. 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. The totality of the representation does not become invalid as long as consistency is kept between the various abstractions of the same design elements.

Graphic Representation

Each model based on a different view will require a different graphic representation of elements for efficient visualization. For example, in the structural engineer's view, non-stuctural walls may need to be shown, for contextual reasons, even though they are not part of the stability FS. This means that any FS which requires the representation of elements which are not part of its functional context will have to make note of those elements. They should appear in a subdued representation, ie using dashed lines and/or lesser intensity and/or other colour. Thus, different graphical representations will either have to be stored for elements for different views or be able to be generated under instructions in those views.

A BUILDING EXAMPLE

Figure 3 shows an example of a floor plan of a two-storey appartment 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 4 shows part of BLDG1 as represented by an architect using a CAD modeling system to represent objects. Figure 4 also shows those entities that are being modeled by the architect as may be stored in a database (eg relational database). The room spaces, LIV1, BED1, etc are simply spaces. How they are modelled (explicitly or derived) is not an issue in this paper.



BLDG1
PLAN STOREY1

Figure 3. Part of Plan of Building Example

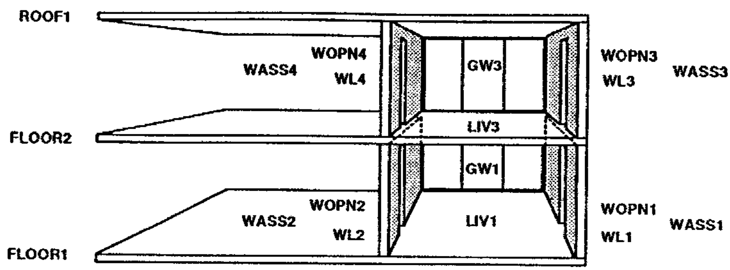


Figure 4. Architect's CAD Model

The part model as shown in Figure 4 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, eg STOREY1, STOREY2, FLAT1, ..., FLAT4 and relations defined between these and the building elements and spaces. Figure 5 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

WL1

A_TYPE_OF:	internal_wall
FUNCTION:	separate_space (STAIR1, LIV1)
BEHAVIOUR:	transparency, sound transmission, ...
STRUCTURE:	
ELEMENT_OF:	WASS1, BLDG1
SHAPE:	rect_prism
LENGTH:	7200
HEIGHT:	2400
THICKNESS:	200
MATERIAL:	concrete block
LOCATION:	...

WOPN1

A_TYPE_OF:	wall _opening
FUNCTION:	provide_access (STAIR1, LIV1)
BEHAVIOUR:	ease of passage, ...
STRUCTURE:	
ELEMENT_OF:	WASS1, BLDG1
SHAPE:	rect_prism
WIDTH	900
HEIGHT:	2100
THICKNESS:	200
LOCATION:	...

WASS1

A_TYPE_OF:	wall _assembly
FUNCTION:	separate_space (STAIR1, LIV1) provide_access (STAIR1, LIV1)
BEHAVIOUR:	ease of passage, ...
STRUCTURE:	
ELEMENT_OF:	FLAT1, BLDG1
ELEMENTS:	WL1, WOPN1
SHAPE:	rect_prism
LENGTH:	7200
HEIGHT:	2400
THICKNESS:	200
LOCATION:	...

GW1

A_TYPE_OF:	glass_wall
FUNCTION:	separate_space (EXT, LIV1) allow_light (LIV1)
BEHAVIOUR:	transparency ...
STRUCTURE:	
ELEMENT_OF:	FLAT1, BLDG1
SHAPE:	rect_prism
LENGTH:	4000
HEIGHT:	2400
THICKNESS:	100
LOCATION:	...

Figure 5. Instance Information from Architect's Model

On the other hand, the structural engineer models the elements shown in Figure 6.

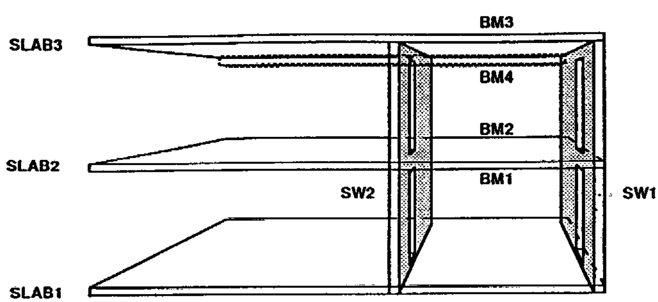


Figure 6. Structural Engineer's CAD Model

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 7(a). The structural engineer has a different perspective of the building 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, eg the thickness and material. This

SW1

A_TYPE_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

A_TYPE_OF:	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, BLDG1
ELEMENTS:	WL1, WOPN1
PART_OF	SW1
...	...

GW1

...	...
STRUCTURE:	
ELEMENT_OF:	FLAT1, BLDG1
SHAPE:	rect_prism
LENGTH:	4000
HEIGHT:	2100 : CONSTRAINED_BY : BM1: depth
...	...

WL1

...	...
STRUCTURE:	
ELEMENT_OF:	WASS1, BLDG1
...	...
MATERIAL:	SAME_AS: SW1
...	...

(b) Modified Architect's Instances

Figure 7. Instance Information From Structural Engineer's Model

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 noted 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 must 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 7(b).

Figure 8 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.

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. The addition of relations between the same elements in different models is critical for consistency.

Work is currently proceeding at Sydney University on developing programs to demonstrate the above concepts in a CAD environment. The CAD system, AES, has been chosen as a suitable system since it has a command language allowing the manipulation of design elements linked to an INGRES database. The implementation will include the different graphic representations required for elements in the different views.

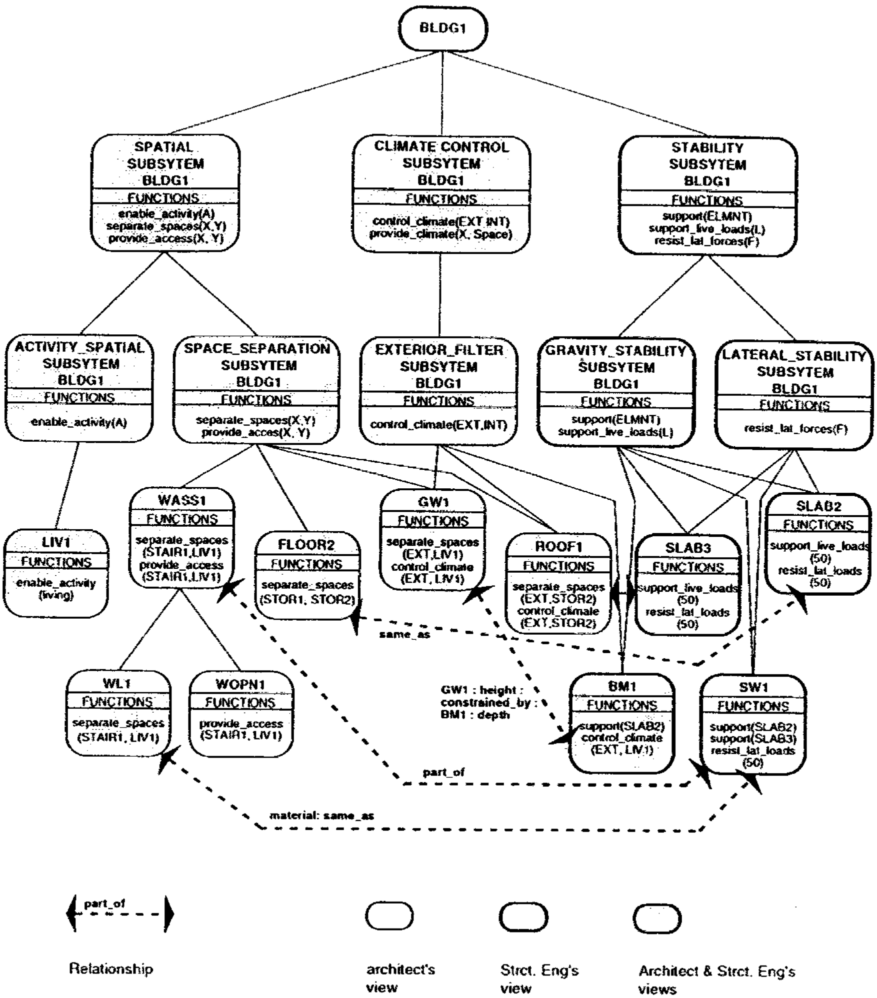


Figure 8. Combined Architect's and Structural Engineer's Functional Subsystems and Models

References

Björk, B C (1987), RATAS: A proposed Finnish building product model, *Studies in Environmental Research No. T6*, Helsinki University of Technology, Otaneimi, Finland.

Björk, B C (1989), Basic structure of a proposed building product model, *CAD*, **21**(2):71-77.

Bobrow, D G (1984), Qualitative reasoning about physical systems: an introduction, *Artificial Intelligence*, **24**:(1-5).

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.

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.

Gero, J S (1990), Design prototypes: a knowledge representation schema for design, *AI Magazine*, **11**(4):26-36.

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.

Gero, J S, Tham, K W and Lee, H S (1991), Behaviour: a link between function and structure in design, *in* D.C. Brown, H. Yoshikawa and M. Waldron (eds), *IntCAD'91 Preprints*, IFIP, Ohio, pp.201-230.

Gielingh, W F (1989), General AEC Reference Model (GARM), *ISO TC 184/SC4/WG1 Document N329*.

Howard, H C, Abdalla, J A and Douglas Phan, D H (1992), Primitive-composite approach for structural data modeling, *Journal of Computing in Civil Engineering*, **6**(1):19-40.

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.

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.

Nederveen, S V, Plokker W and Rombouts, W (1991), A building data modelling exercise using the GARM approach, *COMBINE Report working draft*.

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.

Nijssen, G M and Halpin, T A (1989), *Conceptual Schema and Relational*

Database Design, Prentice-Hall, New York.

Rosenman, M A (1993), Dynamic decomposition strategies in the conceptual modelling of design objects (with special reference to buildings), *Concurrent Engineering: Research and Applications (CERA)*, (to appear).

Rosenman, M A and Gero, J S (1989), Creativity in design using a prototype approach, *Preprints Modeling Creativity and Knowledge-Based Creative Design*, Design Computing Unit, Department of Architectural and Design Science, University of Sydney, pp.207-232.

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.

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.

c. 1993, Management of Information Technology for Construction, K. Mathur et al (Eds), World Scientific Publishing Co., Singapore.