Yugoslav Journal of Operations Research
5 (1995), Number 2, 233-257

REPRESENTATION AND MANAGEMENT OF DESIGN INFORMATION IN AN INTEGRATED ENVIRONMENT IN ARCHITECTURE

Inhan KIM
e-mail adress: i.kim@strath.ac.uk

Thomas LIEBICH
e-mail adress: liebich@cab-muenchen.de
Fax No: +49-89-363801
CAB, Osterwaldstrasse 10
D-80805 Muenchen
Germany

Tom MAVER
e-mail adress: abacus@strath.ac.uk
FAX No: 041-552-3997
ABACUS
Department of Architecture & Building Science
Strathclyde University
131 Rottenrow, Glasgow G4 0NG
Scotland, UK

Abstract: A design problem cannot be comprehensively stated because the design problem has a multidisciplinary nature and the design problem itself evolves as solutions are attempted by the designer. This paper proposes a prototype architectural design environment, which aaims to integrate various aapplication for designing a building. Within an object-oriented design environment, a *unified data model* and a *data management system* have been implemented to seamlessly connect all application.

The suggested *unified data model* organises the structure of the design data to keep the design consistent throughout the design and construction process. By means of the *unified data model*, integrated CAAD systems could reepresent and exchange design information at a semantic level, i.e. the user 's way of thinking, such

as exchanging components and features of a building rather than graphical primitives. In consequence, the *unified data model* reduces the misunderstandings and communication problems among the multiple disciplines of architectural design.

The suggested *data management system* supports the consistent and straight forward mechanisms for controlling the data representation through the inter-connected modules. It is responsible for creating, maintaining, and viewing a consistent database of the design description. It also helps to perform effective data communication among the various design stages to ensure quality and time saving in the final construction of the building.

Keywords: Product model, Unified data representation, Object-oriented design system, Computer-aided architectural design [CAAD].

1. INTRODUCTION

Even in the future the quality of the design process will be influenced by both the designer's abilities and the design tools he or she has chosen Whereas the creation of design ideas and the judgement of design solution should be left to the human, the computer could provide significant support by its capability to store, maaintain, and evaluate highly complex and integrated design data.

Traditional CAAD environments are not much concerned with providing easy access to design data trough unified data description and standardised data exchange formats. In that case, misunderstanding and insufficient coopperation among different disciplines can result. An approptiate environment, however, has to ensure effective data communication among the various design stages and design actors, which requires a way to accommodate the multiple views of the discipline [8]. Thus, an integrated design environment enables various building design tools to create complex project data over the life-cycle of a design artifact and to appraaise its performance, according to several criteria, using the same basic set of project data. The design team would now be able to efficiently cooperate and to easly predict the performance of buildings in order to improve the quality of the design.

The authors suggest an object-oriented design environment following the product modelling approach. The implemented system, ID'EST[1], comprises *a unified data model* to provide semantically meaningful description of buildings, a set *of data control modules* to seamlessly connect building design tools, and a set of *computer-based design tools*. The paper further deals with product modelling and refers to the emerging international standard for data exchange andd dataa definition - STEP [5].

[1] ID'EST is an acronym for Integrated Design Environment using Step methodology.

2. INTEGRATED DESIGN ENVIRONMENT

2.1. Shortcomings of Current CAD System

Current CAD system have inherent shortcomings, whish diminish possible achievements for architectural practites. First, almost all conventional CAD systems rely on a pure geometric data model. All non-geometric information about objects of architectural interest have to be attached to these geometric entities. This restricts the ability to describe semantically dependent relationship. Second, the data exchange remains restricted, since it is based on a fairly low semantic level of a document-based exchange of information ssuch as geometric representation in DXF or IGES, rather than on a high semantic level of a model-based exchange. In consequence, the integration of different design tools for building and construction is still very limited. Thus, the different information about the same building object, such as a wall, (e.g., drawing symbols in CAAD systems, cost calculations in spreadsheet programs, tender documents in word pprocessors, and results of compliance checks in simulation tools) cannot be exchanged between the different design tools. CAD systems fail to play the integration role due to the deficiencies of their underlying database.

2.2. Basic Structure of Inegrated Design Environment

The organisation of Integrated Design Environment [IDE] is inherently complex because of many arbitrary details that must be precisely right for the system to be correct. IDE must be partitioned into smaller , more manageable pieces in order to be maintainable. Therefore, The modular structure is an essential requirement for the suggested design environment. This structure is based on the decomposition criterion known as information hiding, which allows system details that are likely to change independently to be the secrets of separate modules.

IDE consists of a *data management system* and a set of *computer-based design and assessment tools* [11] (See **Figure 1**); The *data management system* organises the structure of the design data to keep the design consisent, and forms the framework in which various tools can be built. Accordingly, it has to depend on a unified data model. The *computer based design tools* assist designers in creating and evaluating the design artifact and validates its correctness. These tools are treated as task-related editors of the unified data model. They receive aall relevant data from the unified data model, map these data into their own separate data structure, and send the modified data back to the model, in order to keep consistancy among themselves [2]. Only very private data, which are unlikely to be used by other modules, are permanently stored within individual databases of the design tools. A complete definition of an IDE is beyond the scope of this paper. All further investigations are directed towards the data management modules, as one of the main parts of the proposed environment. Moreover, product modelling, as a prerequisite for the data management core, is taken into special consideration.

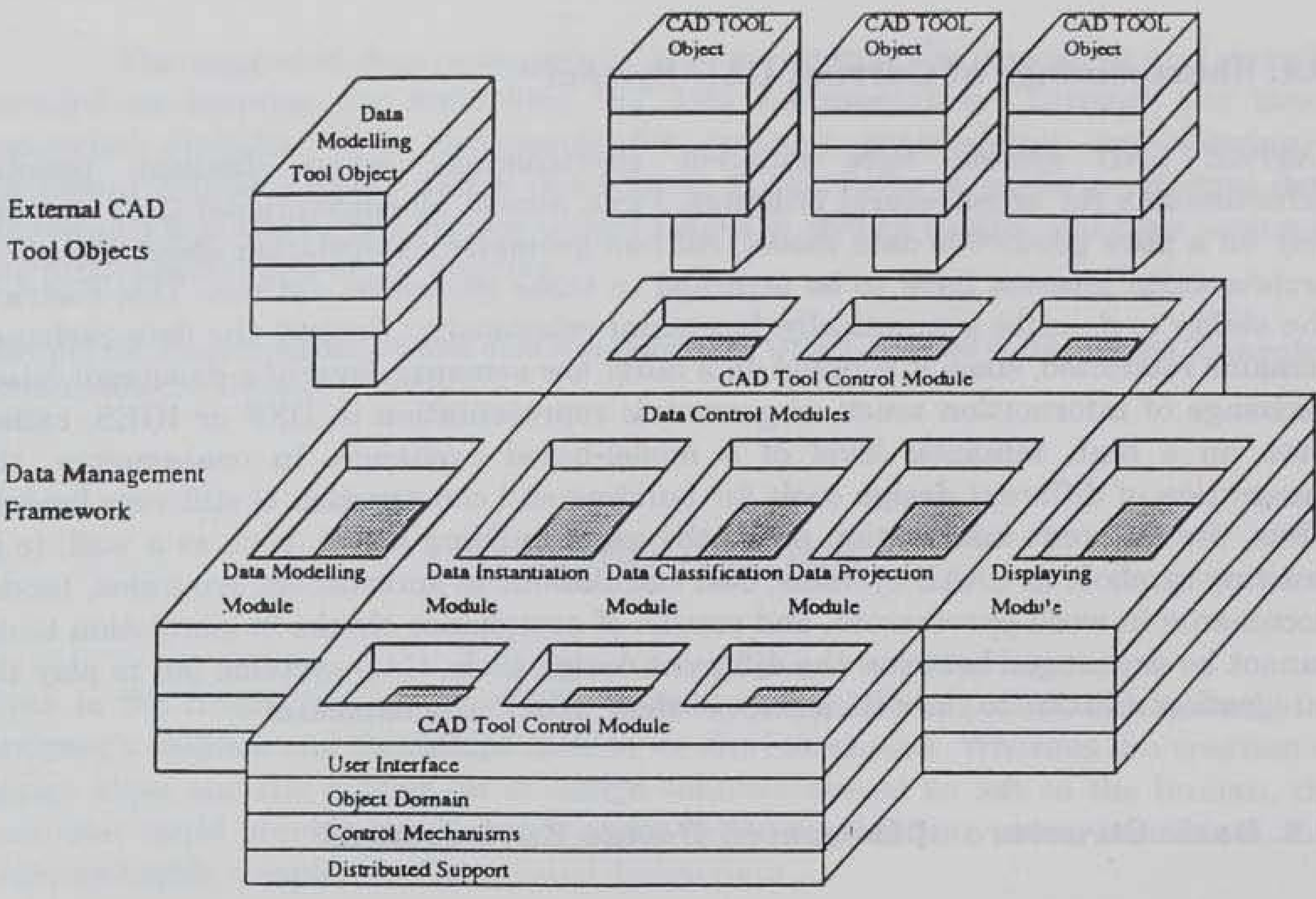

Figure 1: Integrated Design Environment

2.3. Product Modelling

Product modelling should be considered as a process leading to an information model, which provides an abstract description of facts, concepts, or instructions about a product or a set products [16]. This kind of data model, specifying the categories of information about an artifact during its complete life-cycle, is commonly known as a Product Model [PM].

The basic concept used in PM to describe objects of interest is the entity within a hierarchy of sub/super relationships. The entity is specified in terms of its attributes, relationships and contraints. Thus, PM incorporates some characteristics of the well-known object-oriented techniques, such as data encapsulation and inheritance. Furthermore, the PM methodology draws a clear distinction between the conceptual level and the instantiation level. PM specifies the structure of information in schemata with no regard to any format in which the information is later stored. In order to unveil the inherent structure of object data, modelling tools (e.g., the graphical schema definition languges NIAM [21] or EXPRESS-G and the textual data definition language to form uniform expressions about design objects.

2.3.1. STEP

The derection of research in PM is directly influenced by the international standard STEP. The primary aim of the ISO 10303 standard is the specification of a form for the unambigous representation and exchange of computer-interpretable product information throughout the life of a product. Within the industry-wide organisation STEP, the AEC Subgroup is planning to define a series of application protocols and application integrated resources for standardized data description, which will also comprise the domains of architecture and civil engineering.

Apart from that, a number of other research projects highly relate to STEP, such as GARM [9], AEC Building System Model [22], RATAS [3], and the ongoing works in European projects COMBINE [2] and COMBI [1].

2.3.2. Product Modelling as a Process

Further in this paper the authors refer to product modelling mainly as a developing process including the three phases of analysis, design, and inplementation (See **Figure 2**):

- *Analysis*: the creation of a conceptual model on a high level of abstraction (this task partly remains to be the responsibility of architects and engineers, who are holders of all relevant knowledge).

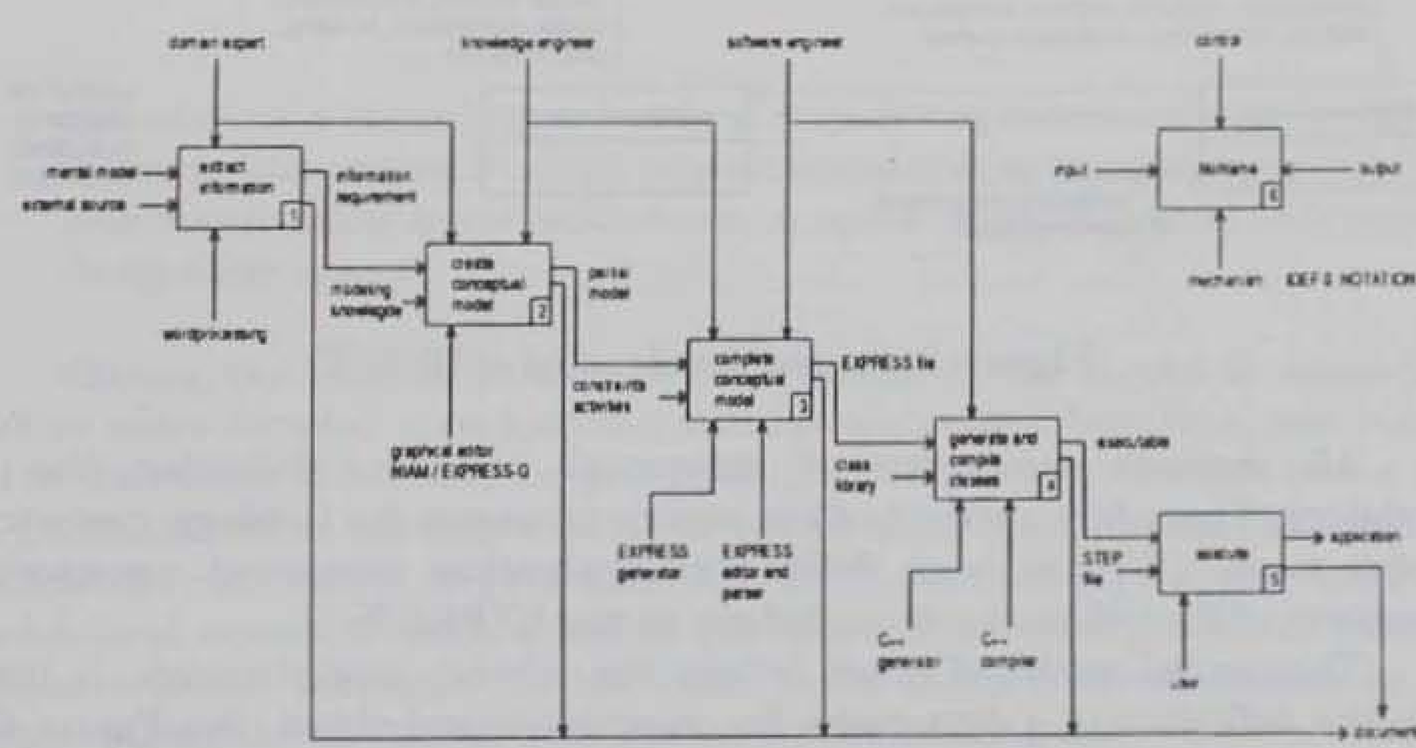

Figure 2: IDEF 0 Diagram of Products Modelling Process

- *Design*: the completion of aa aconceptual model, i.e., the derivation of a neutral data specification form,
- *Implementation*: the generation of an implementation form, which integrates and controls the design data across multiple representation of design in different design tools.

During the first phase of analysis within the ID'EST, the PM was divided into two main layers, each consisting of several schemata [2] (See **Figure 3**):

- the layer of building and construction comprises the schemata *building-project* and *building-system*.
- the layer of specific building systems comprises the schemata *enclosure-system* and *spatial-system*.

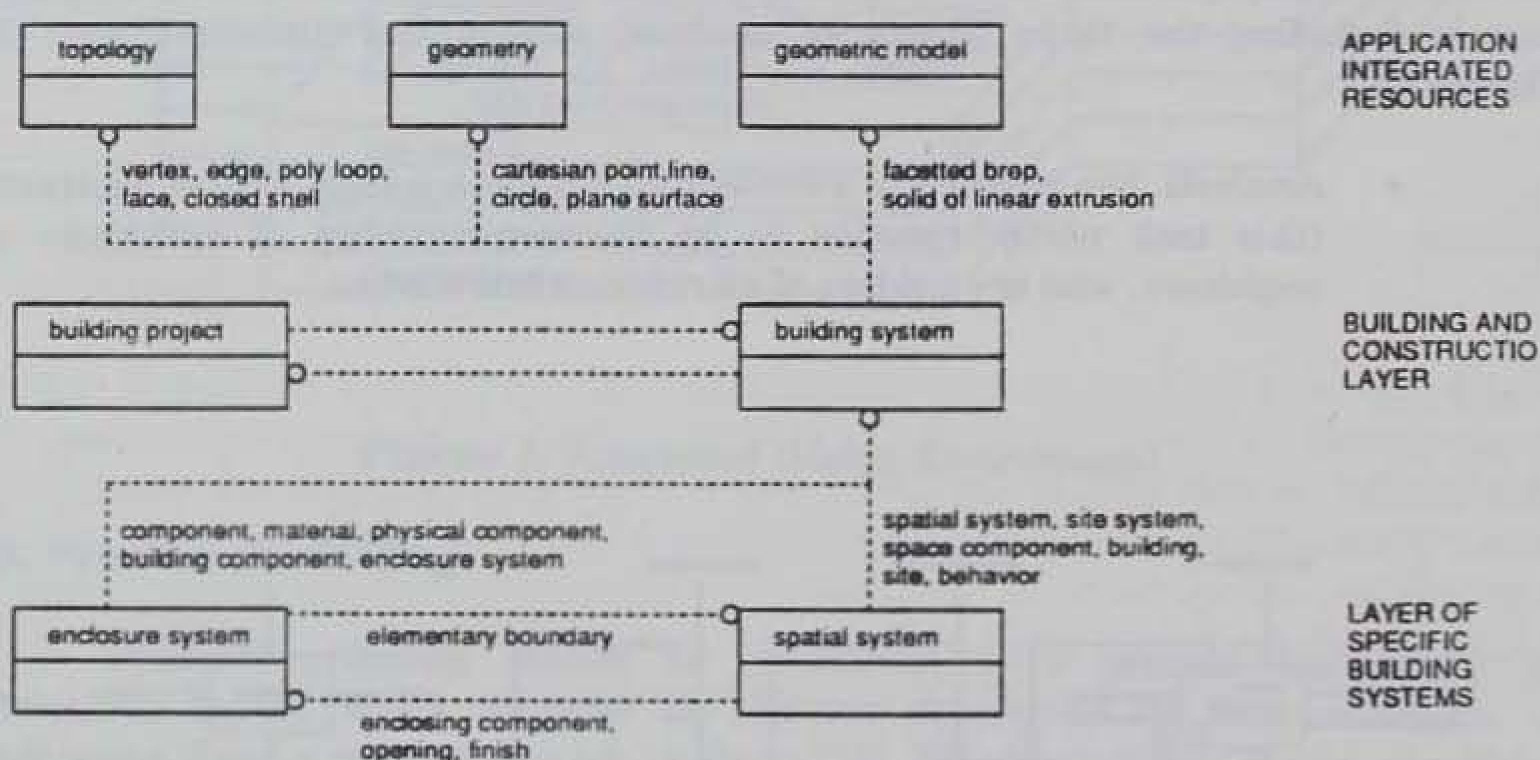

Figure 3: Schema level model of ID'EST

All shemata make use of integrated resources following the STEP methodology. There-fore a subset of the generic resources for *topology*, *geometry*, and *geometric-model* [17] has been defined as application integrated resources. The development of the PM has been carried out ussing EXPRESS-G.

The partial model of *space*, within the schema *spatial-system*, is usedd to explain the definition of a data entity for an architectural object (See Figure 4). The following explanation is given by a pruned natural language to express the relationships within the model:

[2] In addition there is often an AEC core model layer, as the common integration level for process plants, offshore, ship building and construction, as well as a layer for each specific AEC sector [4]. In this paper, however, these two uppermost layers are out of scope, because ID'EST only deals architect's view on building and construction, which again is restricted to space and enclosure.

A space is defined within the hierarchy of spaces. A space belongs to exactly one floor space and can be included in one-to-many floor spaces. A space is bounded by at least one space boundary, which is either a physical space

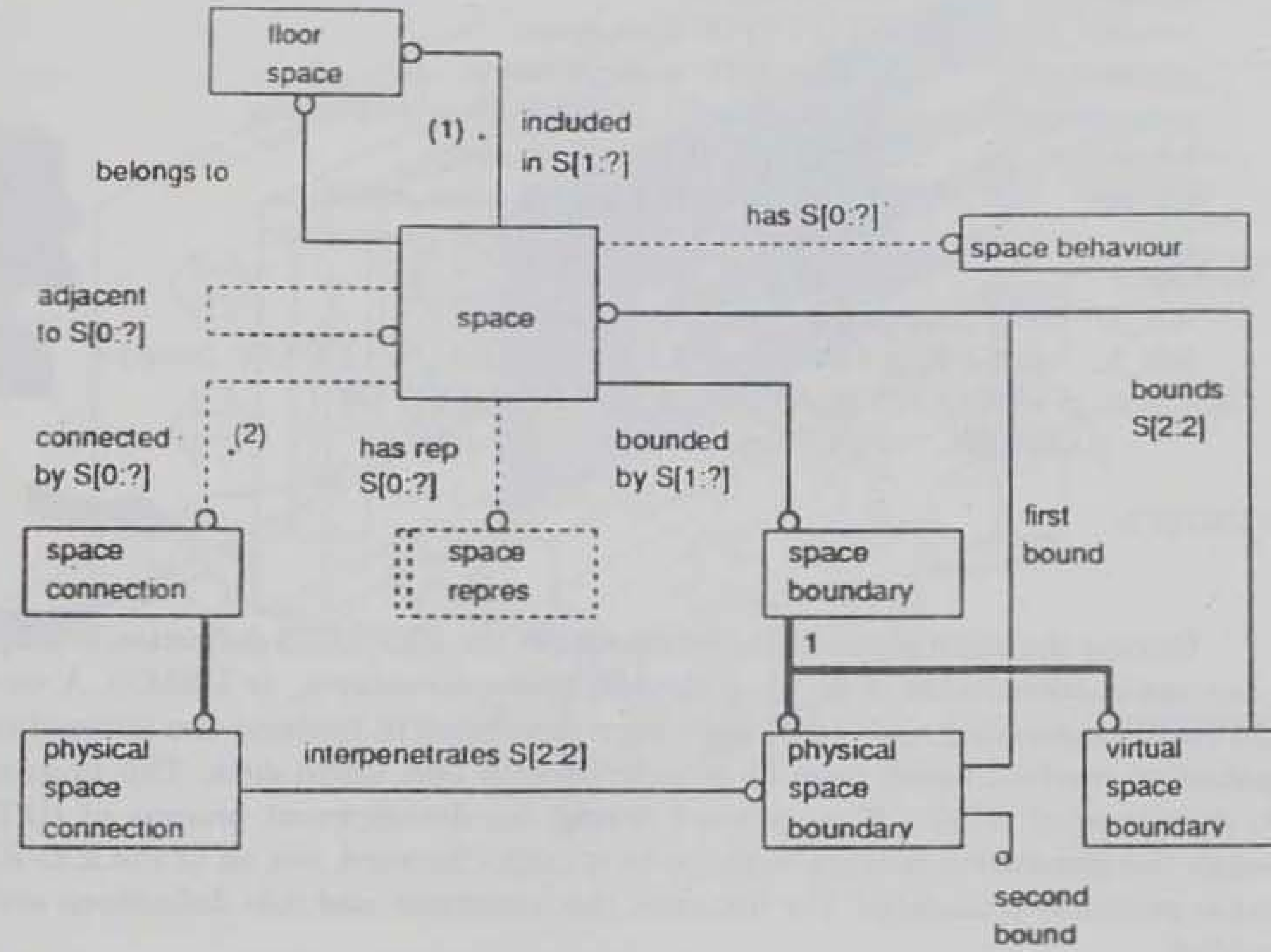

Figure 4: Partial Entity Model of Space in EXPRESS-G

boundary or a virtual space boudary. A space might be connected by a physical space connection. Further, a space has zero-to-many space reprentations and zero-to-many space behaviours. A space might be adjanced to zero-to-many other spaces.

During the seconds phase the final design of the model is accomplished. Therefore more detailed specifications, such as contraints, functions, and rules, are added to the model. Therefore, the graphical notation has to be mapped into a textual form. EXPRESS offers the functionality to include these specifications, although there are still ssome restrictions. In particular, EXPRESS does not support the definition of the behavioral aspects of objects, and in this sense it does not provide all features of the object-oriented paradigm. In our case, the contraints are described by *where* clauses, containg the additional specification for the entity *space* [3].

[3] The clause wrl stipulates, that the first floor space within the set of floor spaces within which a space is included, should be the same floor space as the floor space to which a space belongs. The clause wr2 stipulates, that either there is at least one virtual boundary among the set of space boundaries which bound a space, or the space is connected by at least one physical space connection (Physical space connections are provided by openings for doors or windows).

```
ENTITY space

    SUBTYPE OF (artificial_space);
        belongs_to        : floor_space;
        included_in       : SET  [ 1:?] OF floor_space;
        bounded_by        : SET  [ 1:?] OF space_boudary;
        connected_by      : OPTIONAL SET [ 0:?] OF space_connection;
        adjacent_to       : OPTIONAL SET [ 0:?] OF space;
        has_rep           : OPTIONAL SET [ 0:?] OF space_repres;
        has               : OPTIONAL SET [ 0:?] OF space_behaviour;
    WHERE
        WR_1:   SELF.included_in [ 1] = SELF.belongs_to;
        WR_2:   (SIZEOF(QUERY temp <* SELF.bounded _by | TYPEOF (temp) =
    ' SPATIAL_SYSTEM.VIRTUAL_SPACE_BOUNDARY ') OR
            (LOINDEX     (SELF.conecected_by >= 1);

END_ENTITY;
```

During the third phase of implementation the EXPRESS definition is mapped into any application forrm (e..g., C++classes, frame-structures, or DBMS). A variety of EXPRESS processing tools [23] have been developed to faciliate the generation of instantiation models, which caan be populated with real world data. The processing tools, developed at NIST [6] were used during the development process of ID'EST. Although the generation process ought to be straight forward, not all of the EXPRESS syntax is presently translated. For instance, the constraint and rule definitions are not supported.

2.4. Data Management in the Environment

The data management system has to control and interpret the collection of data instances within an application form of the unified data model. Its underlying data structure depends on the PM definition for the environment. The system is responsible for creating, maintaining, and viewing a consistent database of the design description. Consistency checking and constraint propagation are further task of these modules.

The data management system also has an *integrated framework* which helps the data to be processed for design tolls, which can be either realised as a file-based exchange or provided as a direct access interface [4] . The *integrated framework* facilitates to expand the design at any point of the design process. A *blackboard system* and an *integrated object-oriented database* will provide effcient means to meet the above requirements in data management system (See **Figure 5**).

---

[4] Using STEP, there are two standardized definitions for data interchange. Part 21[19] describes the implementation method to physically exchange data (often referred to as STEP physical file). Part 22 [20] describes the common access interface.

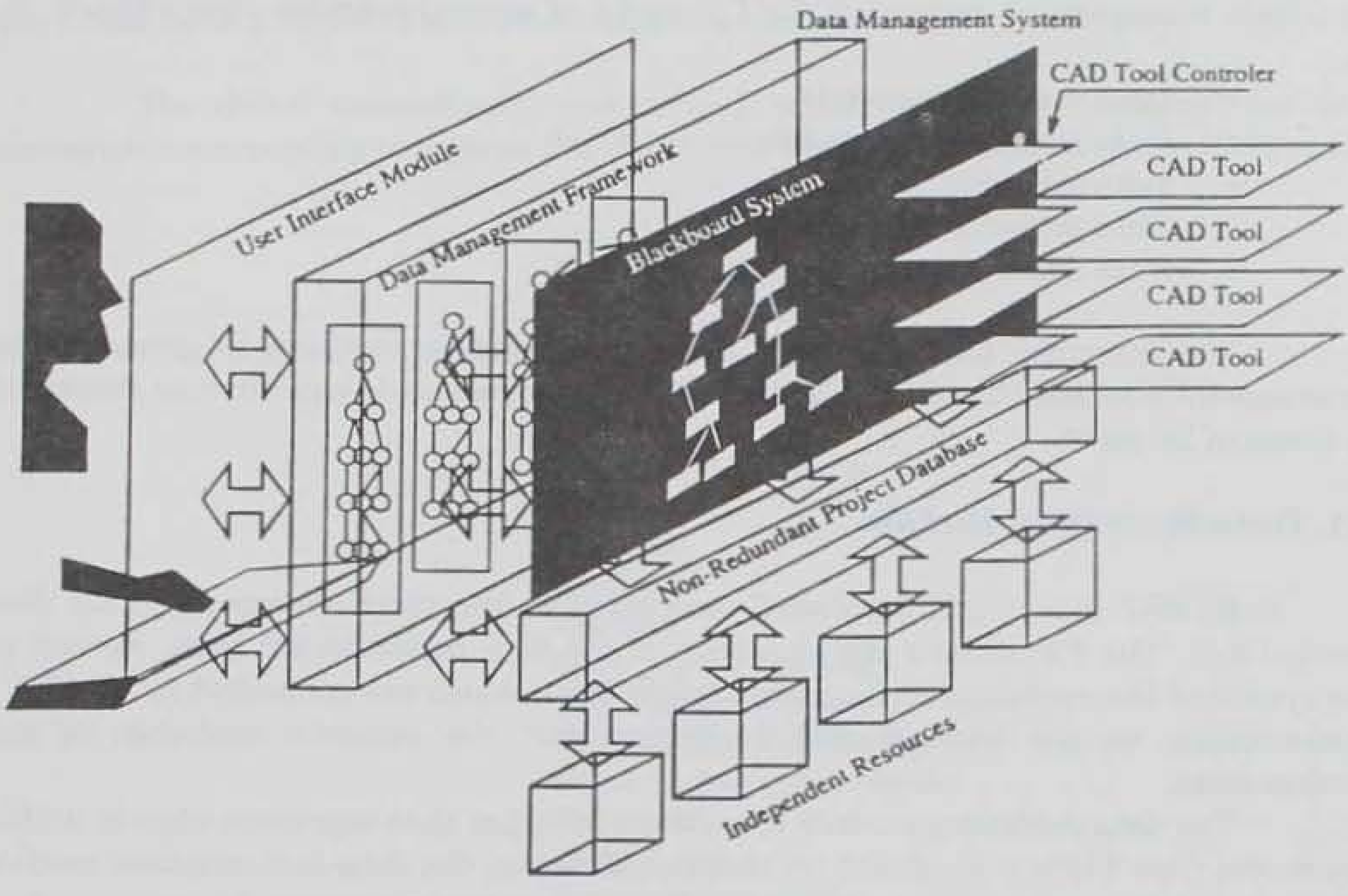

Figure 5: Data Management system in the Ennvironment

The *blackboard system* serves as an inter-resource notifier and acts as a communication centre between the various ID'EST modules, [7]. The blackboard contains the problem data and a hierarchy of hypotheses relevant to the problem. At any one time during a single hypothesis, the focus can shift to another area of the board [13].

The *integrated design database* is the core part of the data management system [5] . The integrated design database is a prerequisite for creating an integrated design environment by which design data can be shared among the different design tools of the design environment.

With the help of these software support tools, the design data management system oganises the design description within each representation, correlates equivalent descriptions across the representations, and attempts to maintain these correspodences as the design incrementaly evolves.

[5] In ID'EST, a prototype database system has been developed by the authors as existing commercial database systems are not suitable to manage semantically rich architectural design data.

3. PROTOTYPE ENVIRONMENT ID'EST

As a data management system, ID'EST consist of several modules (See **Figure 1**), i.e.:

- Data modelling module
- Data instantiation module
- Data calssification module
- Data projection module
- Data displaying module

The interplay between these rather independent modules is given in the environment's architecture (See Figure 6). The environment is now further described in terms of its parts.

3.1. Data Modelling Module

ID'EST uses a PM developed according to the methodology of STEP (See Section 2.3). The PM defines the structure of the data instantiation tools, as well as the syntax of the exchange format. The design tools, which are connected to ID'EST , communicate on the basis of STEP physical file, the common exchange of the environment.

The data modelling module specific entity types that represent objects within the model (See **Figure 7**). Based on this specification, the data instantiation module allows the definition and the exchange of unique entity instances. In consequence, every time the unified data model is changed, the data instantiation module has to be recompiled as well.

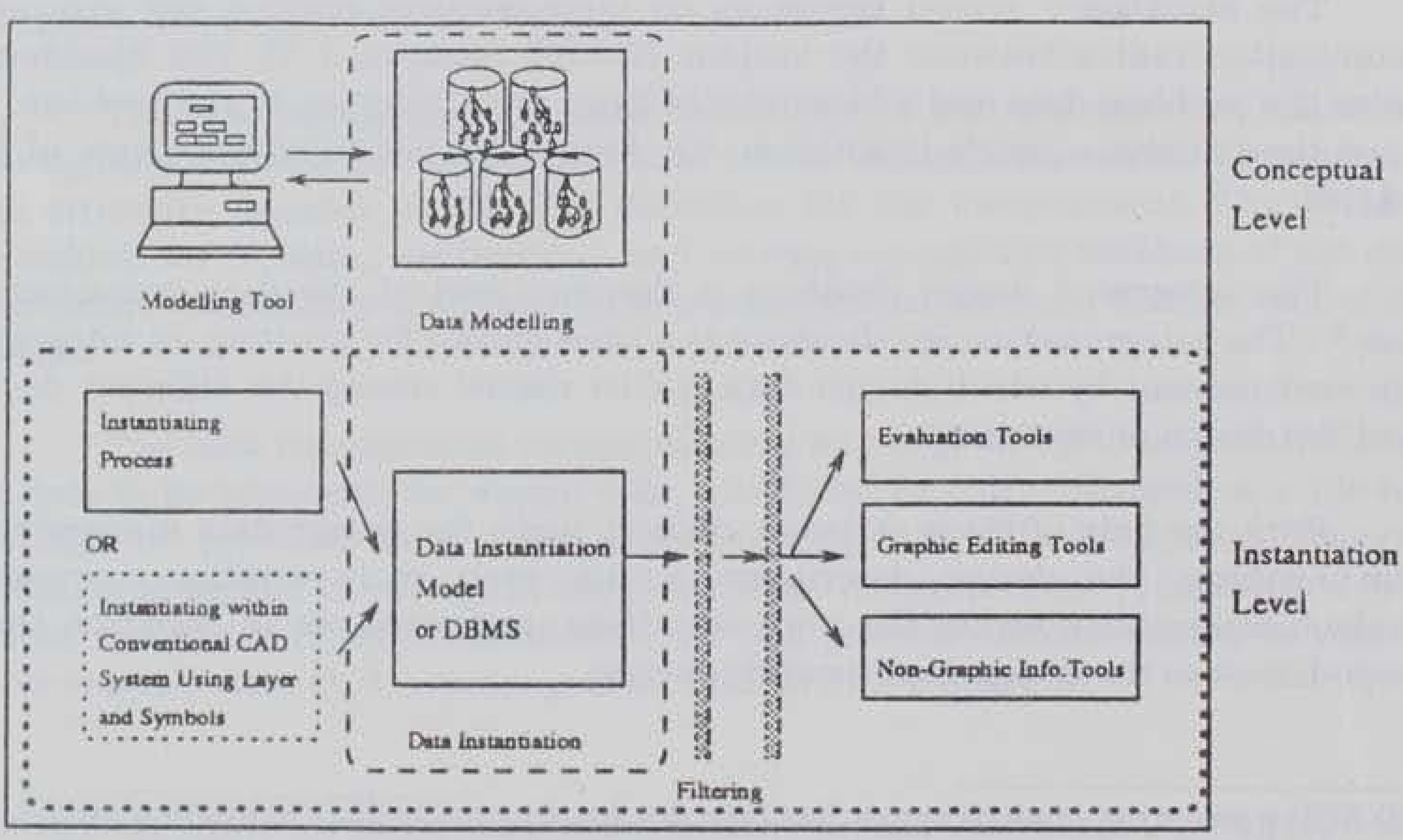

Figure 6: Conceptual Data Processes

3.2. Data Instantiation Module

The data instantiation module creates, edits or views information which is in the instantiation model. The module supports two kinds of instantiation.

The *direct instantiation* uses a tool, which is specially designed for data instantiation according to a given PM definition. The main objective of this instantiati-

Geometric_Representation_Item
Group_Of_Spaces
Horizontal_Enclosure
Inner_Space_Boundary
Line
Manifold_Solid_Brep
Material
Measure
Object
Opening
Outer_Space_Boundary
Physical_Component
Physical_Space_Boundary
Physical_Space_Connection
Placement
Polly_Loop
Representation_Item
Roof
Site
Site_Space
Site_System
Solid_Of_Linear_Extrusion
Space
Space_Behaviour
Space_Boudary
Space_Component
Space_Connection
Spatial_System
Surface
Swept_Area_Solid
System
Topological_Representation_Item
Vector
Vertex
Vertex_Point
Vertical_Enclosure
Virtual_Space_Boundary
Wall
Window

Figure 7: Part of the entity list of PM

on module is to have an instance manipulation environment for the semantically structured entities defined in the suggested data model. It is a tedious and time consuming task to actually instante values of each attribute of a data model., if every instance should be entered manually. Thus, an instantiation module should provide a mechanism to input data in a graphic way and to classify data when entering into the database. A new generation of object-oriented CAD systems which can represent information at a semantic level is needed, but unless they are available, conventional CAD systems can be used as data instantiation tools with special conventions and some restrictions as well.

The indirect instantiation uses an external CAD tool to instantiate entities. In this case, it is necessary to have an accessory tool to convert the external CAD tool data format to the STEP physical file format, in order to import data to the project database. In the ID'EST system, Data Probe[6] has been used to create, edit, or view data corresponding to the information model for which it was created. Data Probe also used to read, merge and write STEP physical files.

3.2.1. Instantiation Using External CAD Tools

The creation of building design instances is normally done using conventional CAD systems, especially for the input of geometry. Beside the geometrical representation, more semantic information is needed for input into the instantiation model. Therefore, the inherent structure of CAD systems, provided by layers, macros, and attached attributes, plays a key role in enabling the extraction of appropriate data. Naming conventions, as suggested by the AIA in the U.S. [15], which has several recommendations to classify graphic data [7], thus giving the opportunity to map this implict information into the object-oriented mode of data representation, are used in ID'EST.

The converter DXF2STEP, which was developed as a part of the proposed environment, reads a DXF file and translates it into a STEP physical file, according to the given PM. DXF2STEP makes use of a user-defined mapping table allowing other layer conventions to be easly adopted (See **Figure 8**). The mapping table defines the relation between layer names and entity types as well as between attached CAD attributes in the STEP file. Semantic information, however, which can not be created using CAD, has to be added within the instantiation model. In this case, the instantiation model is further refined within Data Probe, after the STEP file has been read.

---

[6] The software used to create Data Probe tools was written in the C++ programming language on top of the X Window System using the InterViews user interface toolkit [14]. A Data Probe can have an entity type list which contains the list of entity types defined in the information model. It also provides a funcionality for manipulating and maintaining the individual instances and the list of instances. Entity instances may be created and edited interactively within a Data Probe from its list of entity types.

[7] A layer name (according to AIA layer guide) using the short format, has the principle from *xyyzz-00*, where *x* is the major group, representing the different disciplines (e.g., A = architecture, S = structural engineering), *yy* is the minor group, designating construction system, *zz* is an user-defined minor group, and *00* reflects the level.

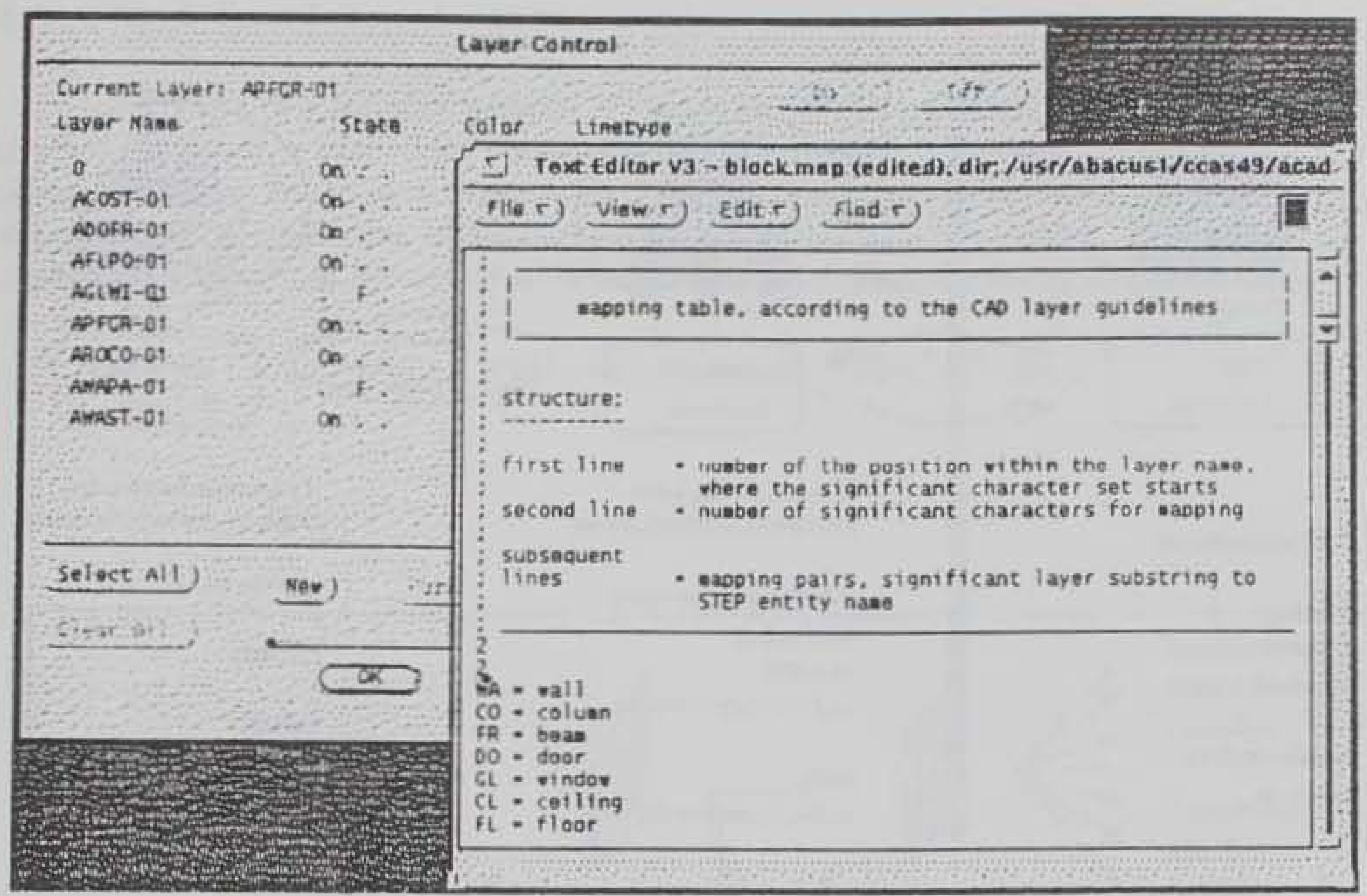

Figure 8: CAD layer structure and the mapping table of DXF2STEP

The usage of file-based exchange between traditional CAD system and an object-oriented design environment also shows a path to preserve the usability of existing tools for a certain period after the introduction of a new software generation.

Besides DXF2STEP , ID'EST communicates through other converters, such as STEP2OBJ, as well. All converters have in common that they map the unified data model, as contained in the instantiated data model in the STEP physicaal file, to the specific format, used by other modules (See **Figure 9**).

3.3. Data Classification Module

The module is designed to acquire object descriptions, both textually and graphically, and to be reflective of a user's intentions and descriptions.

3.3.1. Selected Inheritance for a Required Discipline

This module classifies the *project database* according to the design discipline and elicits necessary information for the required discipline from the database (See **Figure 10**). To elicit appropriate information from the *project database*, a highly sophisticated control mechanism is needed. The data classification process check the desired entity types, filters necessary information from the database, and sets special marks on the information to identify the necessary information (See **Figure 11**).

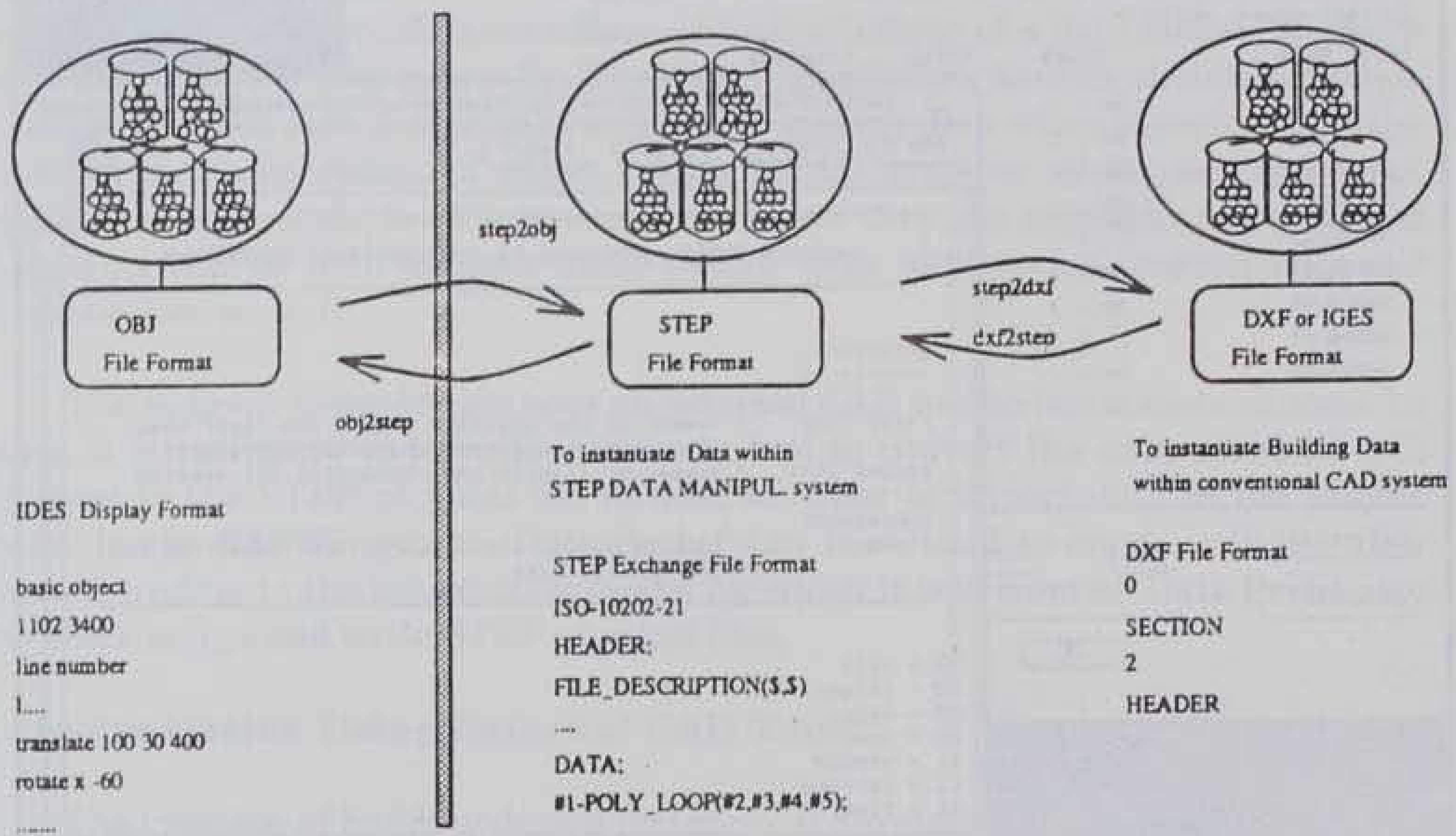

Figure 9: Data Exchange, using converters from and to the standard STEP physical file format.

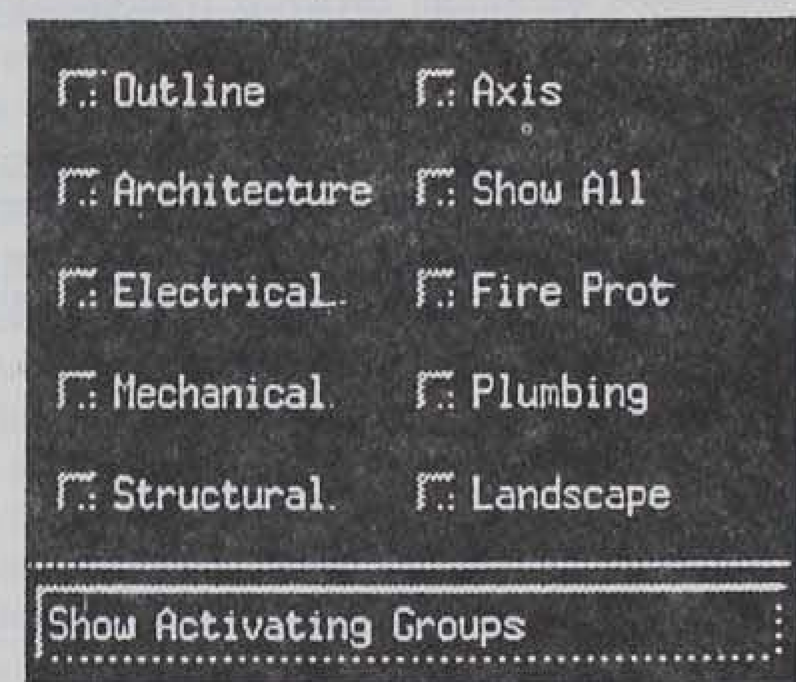

Figure 10: Data Classification Tool

3.3.2. Multiple Discipline Selection

To have an ideal integrated design environment, a designer or engineer should be able to examine information of relevant disciplines which are stored in the project database. In ID'EST, by using the *Data Grouping Toggle Tool*, the user can view or edit building data of one or more disciplines at the same time (See **Figure 12**).

3.3.3. High Level Command Interface

Within the ID'EST system, to view or edit the desired information from the project database, a high level command based interface is provided. The language like commands can be used to query entities in the project database, as an entity is described as a collection of features one associate with the object. For example, to check all walls which consist of a certain material, a simple command like:

Display : : wall : has-material : value = " onyx "

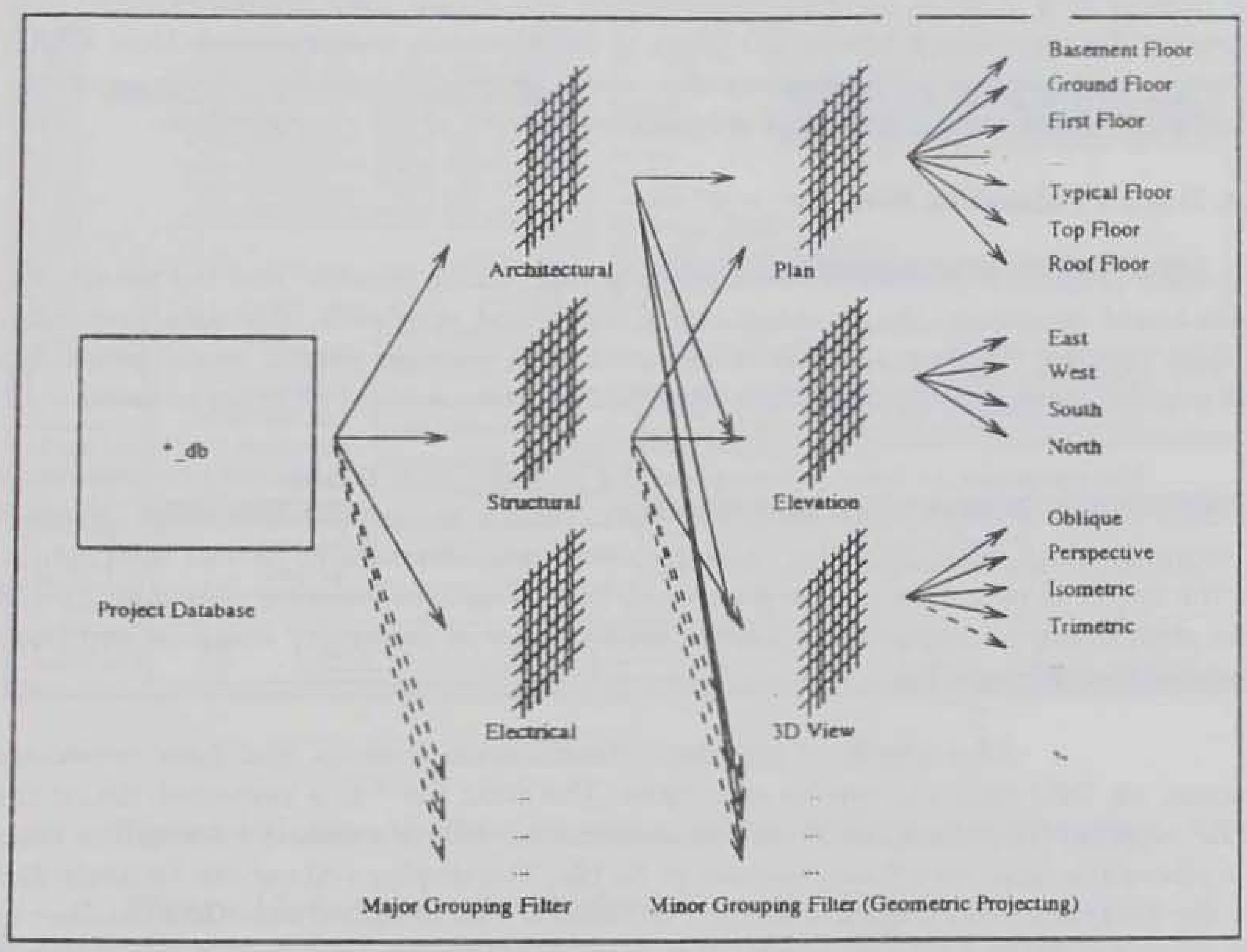

Figure 11: Filtering Process

can recursively check all walls in the project database and highlight them on the monitor (See **Figure 13**). Similarly, a bill of quantities for a building can be generated with ID'EST, providing all necessary information. An assessement tool might send a message to calculate the total cost for walls:

Retrieve : : wall : has-behaviour : name = " cost "

Figure 12: Select Multiple Discipline

3.3.4. Layer control

The layer control facility in the conventional CAAD system, AutoCAD, has beeen used to manage the different presentations of an object. To extract a series of flat views from an integrated 3D graphic database, layers can be used to differentiate information that is not seen in all views [12]. For example, on a furniture plan, only the outline of a chair is chown: the details of the chaiar base are suppressed. In a elevation drawing, detail beyond the plane of the elevation is suppressed. Most CAAD systems that support 3D drawing can also control the display depth settings, effectlively hiding information that is behind or in front of the viewing plane.

3.4. Data Projection Module

The data projection process is executed by a user dialog handler and the result of a black-board processor. By checking entity types and attributes, the data projection process (second filtering process) is executed with possible partial model projection and possible re-execution of the data classification process (first filtering process).

For example, to have a floor plan of a building, first it necessary to concetrate on data of the architectural discipline. With proper projections including selected inheritance, data deseggregation, and geometric transformation, partial information for the required plan view can be generated. In this case, re-entering of higher level of data abstraction is necessary to have a section view of necessary columns and their positions (See **Figure 14**).

As a result of the data classification process and data projection process, an OBJ data file can be generated. The OBJ file [8] ia a projected file of the STEP physical file, which can store the semantic as well as syntatic information since it is allowed to have structured entities in its file. The display setting can be controlled by the several levels of control module through a well designed interface handler, as the interface system allows selection of desired entities from the instantiated data model.

3.5. Standard Building Element As a Reusable Object

To reduce the complexity of a design, it is helpful to use standard building elements

[8] The converter STEP2OBJ reads a STEP physical file and translates it into OBJ file format, which was originally developed as a data manipulation format for Hybrid Integrated Design Environment[HIDE], in the ID'EST environment [10].

and a few rules of composition. Standard building elements are defined as Reusable Objects in the prototype environment. The Reusable Obkects is derived from "template" objects. The "template" objects are defined as object oriented classes to be used as "templates" in creating copies of a generic object. Reusable objects are stored in the secondary storage area. The location of the insertion point of the object is stored in each instantiated data model and, with appropriate projections, the objects are inserted in the drawing.

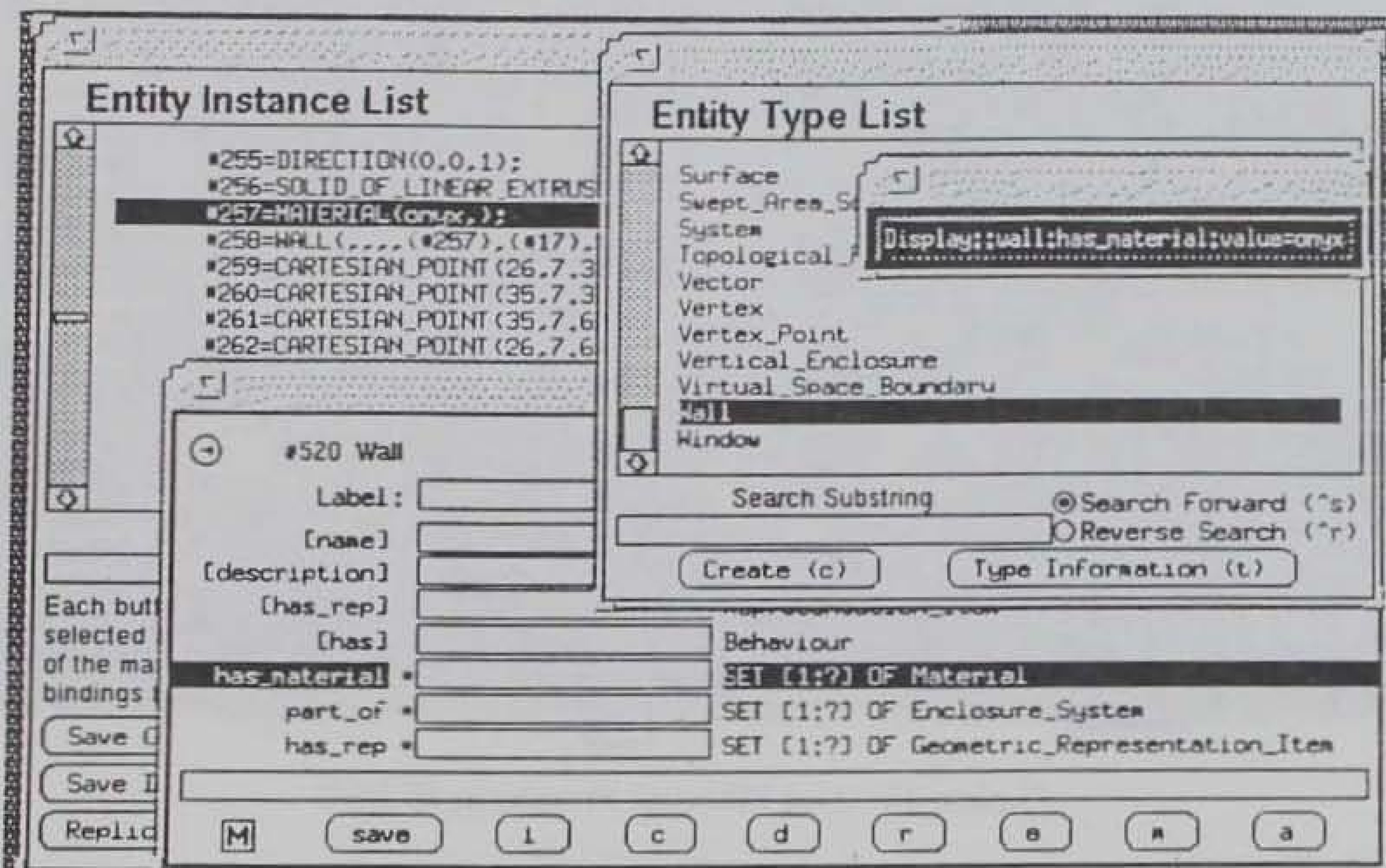

Figure 13: Data classification, using the high level command interface

Figure 15 shows a screen display of several different representation of two *Reusable Objects*, a door and a window, as a result of different projections. To use these *Reusable Objects* in the building, a user need to specify a *locational value* and, if necessary, rotational an parametric values of the object. As these *Reusable Objects* are defined as *objects*, which follow the PM definitions (See Section 2,3), and stored as separate project databases, it is possible to examine various representation of these *Reusable Objects* like the same way to manipulate a building *project database*.

3.6. Data Propagation Facility

To keep the representations of the design object consistent, the data propagation facility has to be developed in a further stage of ID'EST. To propagate changes across representations, the design propagation module will require the detailed semantic of

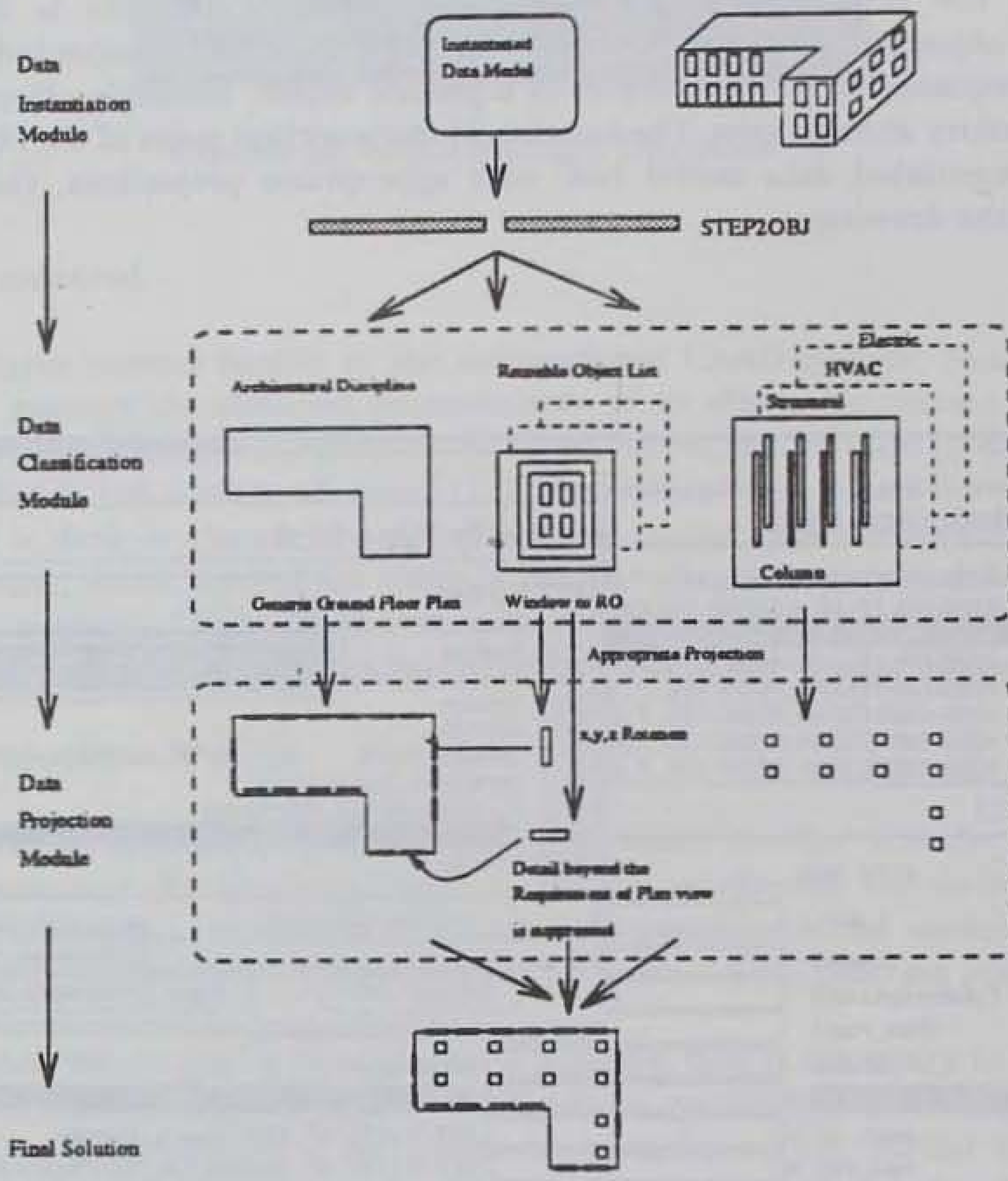

Figure 14: Data Filtering Process in ID'EST

the design representations to be interpreted by the system. When the data are changet, all relevant data are propagated accordingly.

3.7. Integration Examples

The authors have proved the interplay between the different modules of ID'EST, using two existing buildings:

- The *Barcelona Pavillion* by *Mies van der Rohe* has been used mainly to test the capability of the data modelling module and the data management system. The building has a simple structure with simple geometry which is suitable for the test purpose. Most of building elements are defined in PM, and the building has been instantiated and processed to Data Probe and ID'EST.

  The graphical representation, including additional information such as materijal, was created in AutoCAD and structured according to the given layer naming convention. The AutoCAD file was later converted by DXF2STEP and included into the data instantiation module (See **Figure 16**).

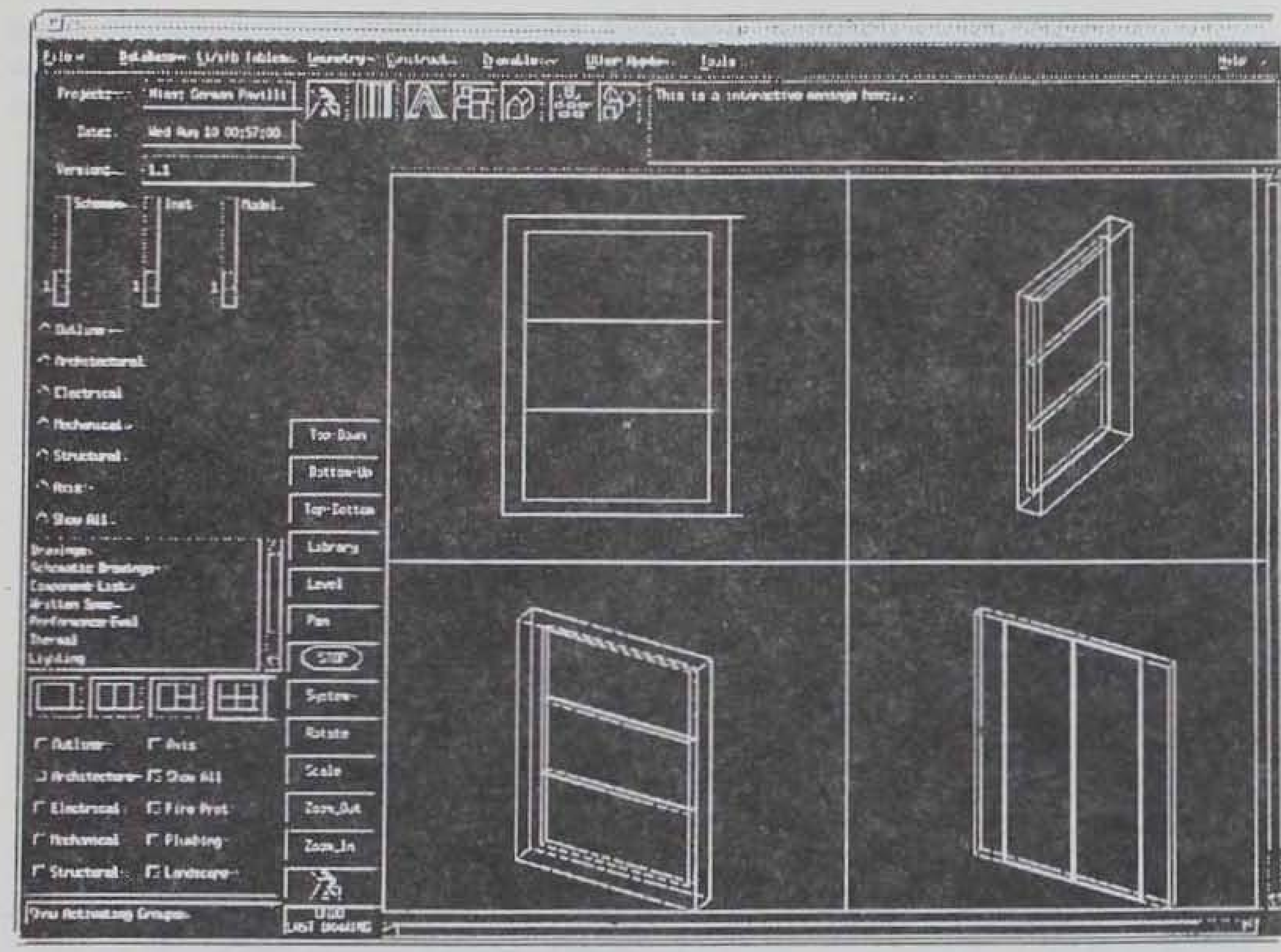

Figure 15. Screen display of several Resable Objects, a door and a window, with different representation

After more semantic information has been entered, relevant data were sent to the data classification module and the data projection module.

- The *Bloomielaw building*, a medium-rise office building located alongside the river *Clyde* in *Glasgow*, has been used mainly to test the capability of the data management system and the user interface system.

  Figure 17. shows the *ground floor plan* of the building. In the *ground floor plan* drawing, with the help of a display depth control mechanism, several different levels of semantic structures have been included. The outline of the plana view is derived from "outline" layers, section views of column objects are derived from column layers in the structural layering set, and section vews of windows or doors in the level of the floor are derived from a layer library of "Reusable Object".

Figure 18. shows several representation windows of the building. The information of the building can be accessed and changed from any of the representation windows. The change of the information affects the non-redudand *project database*. Therefore, all other representation of the building will be propagated and updated accordingly as the representations of the building is the result of the *data classification* and the *data projection* process which manipulate the *project database*.

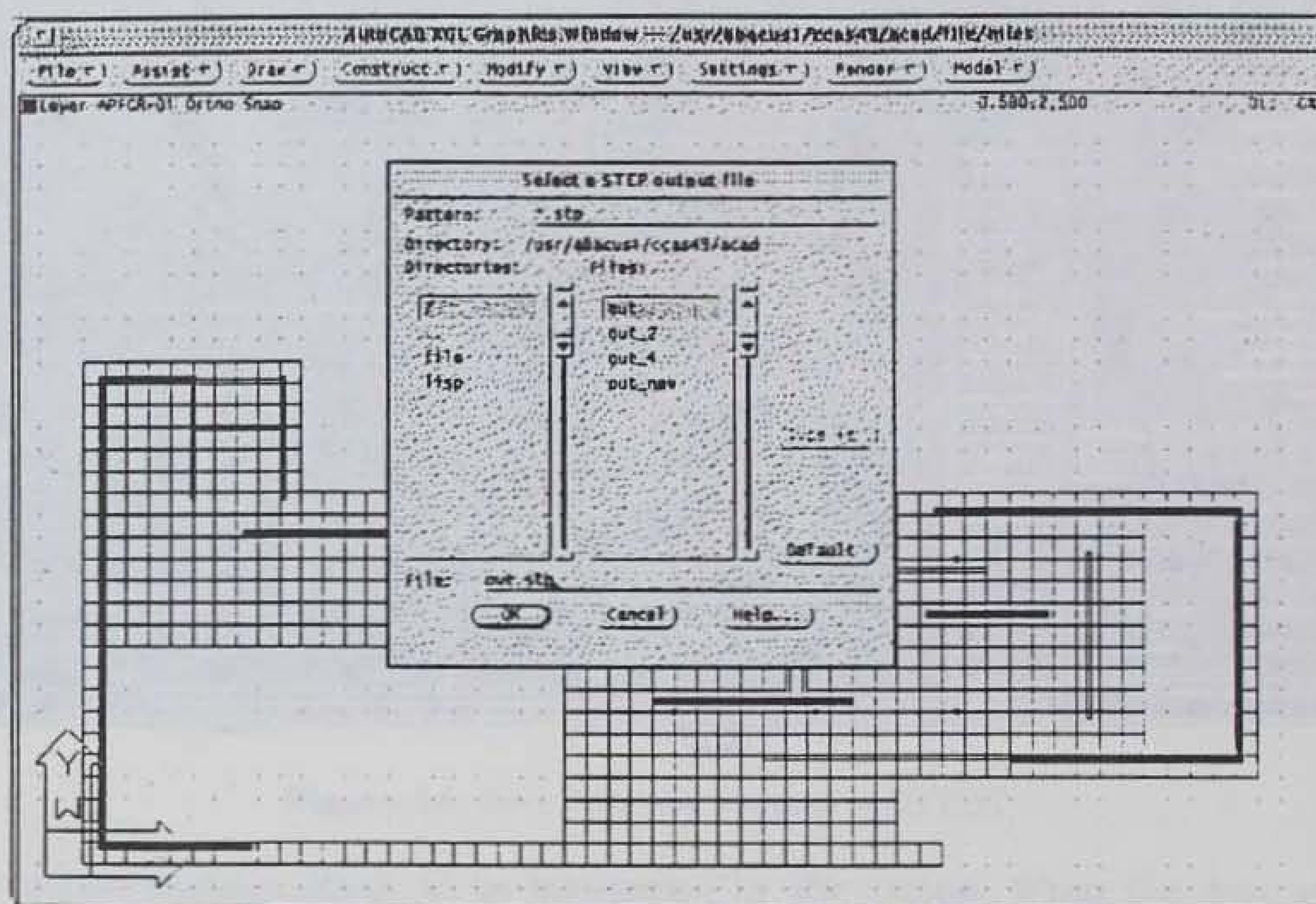

Figure 16. Instantiation Process with Converter

The screen display of the whole data managing system with explanations is given (See **Figure 19**).

4. CONLUSIONS

An Integrated Design Environment dacilitates cooperation between the different disciplines of building design and construction. If offers control and management to keep the design description consistent. Several design and evaluation tools can now access data from the IDE, thus the behaviour of the design artifact will be more predictable. These data can also be further used in later stages, such as facility management.

To conclude, the authors argue that the human designer should have all the flexibility possible for his or her work and the full responsibility for the project in any design system. The designer should receive all possible support for the crucial parts of the design work, such as collaboration and consistency. The suggested design environment should lead to the "ideal CAAD environment" a designer might want to use in future. The design environment shows the possibility of a seamless and continuous working environment for architects from the initial data modelling process to the final design solution by providing a *unified data model*, integrated design data control modules an various design tools.

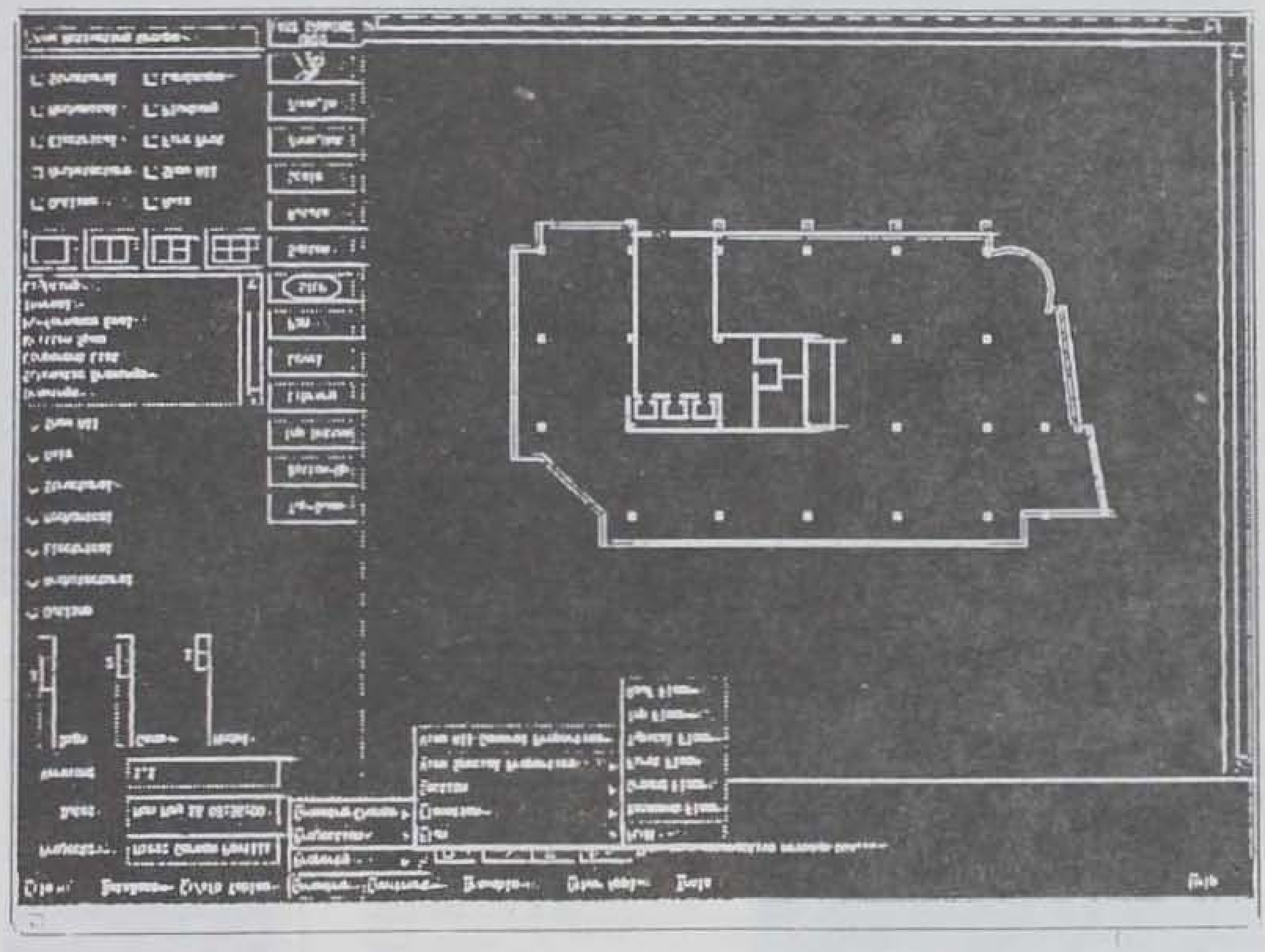

Figure 17: Display of ground floor plan of the prototype building as a result of data projection process

5. ACKNOWLEDGEMENT

The authors wish to thank the ABACUS unit in the University of Strathclyde for its support for this project.

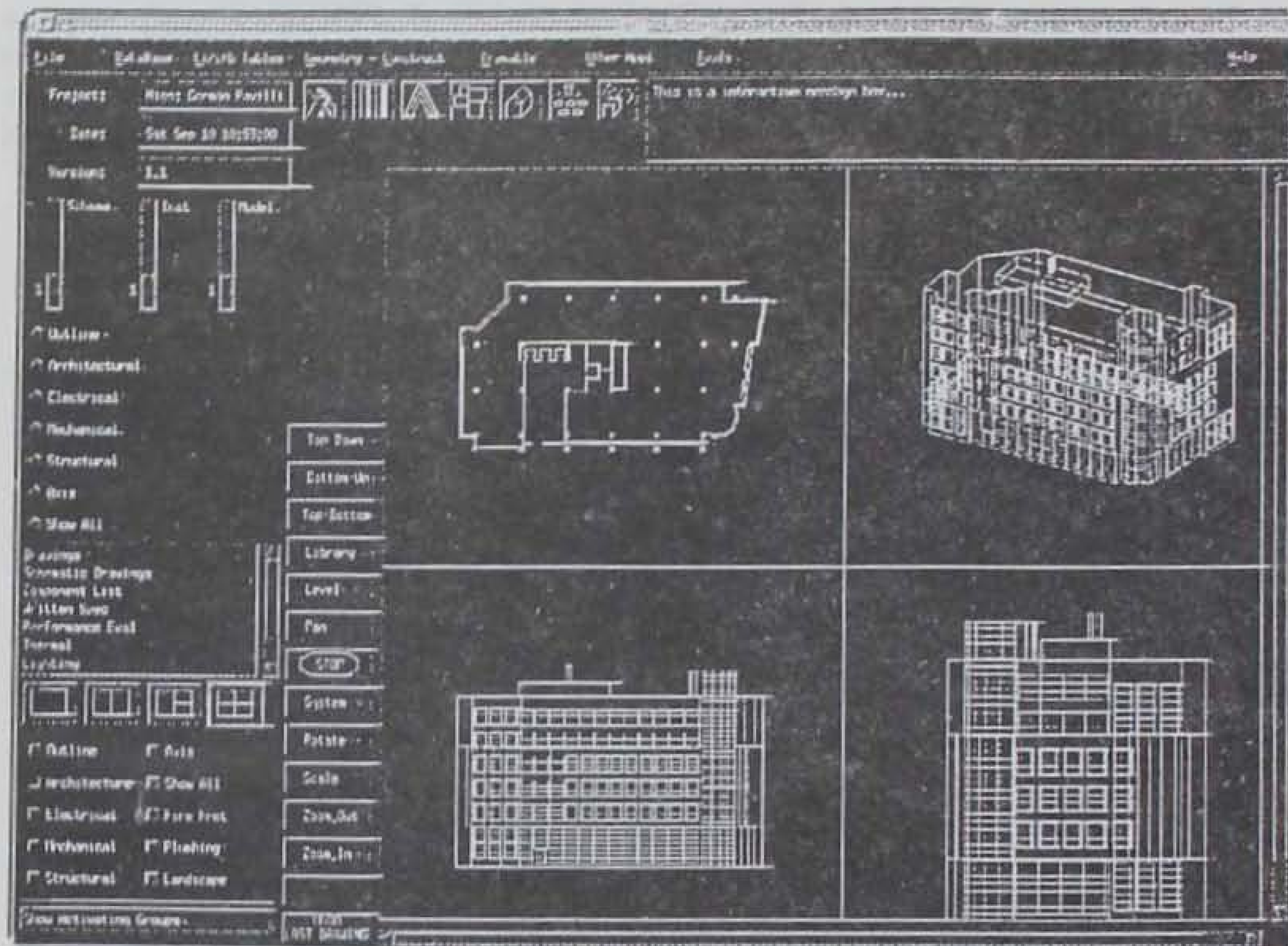

Figure 18: Several Representation Windows of Same Design Object

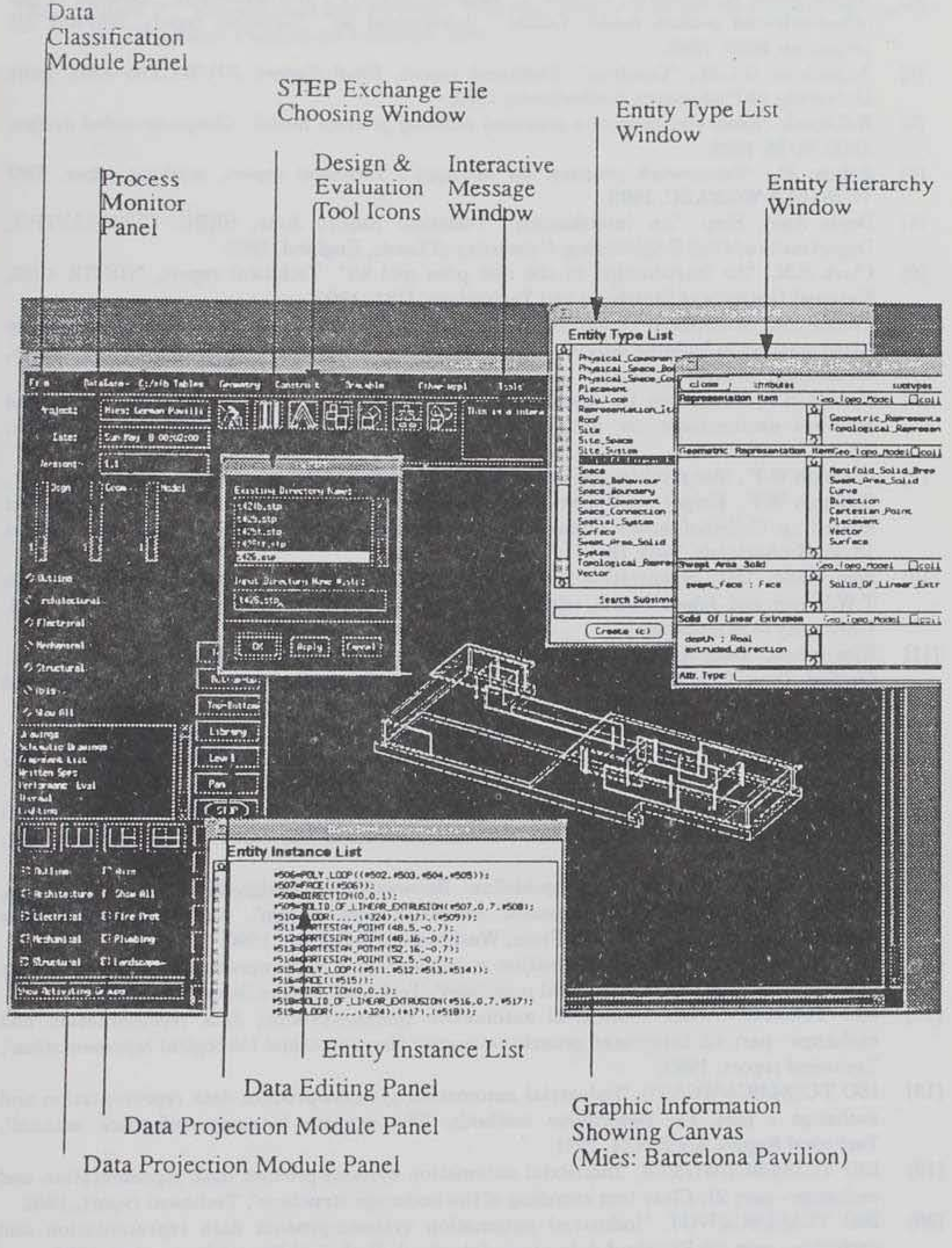

Figure 19: Screen display of data management system (Mies: Barcelona Pavilion)

REFERENCES

[1] Ammermann E., Junge R., Liebich T., Katranuschkov P., and Scherer R.. "Concept of the object-oriented profuct model: Combi - deliverables a2". Technical report, ESPRIT III project no. 6609, 1993.

[2] Augenbroe G.L.M., "Combine". Technical report, Final Report JOUE-CT90-0060, Delft University of Technology, Netherlands, 1993.

[3] B.C.Bjork, "Basic structure of a proposed building product model". *Computer-aided design*, 21(2):71-78, 1989.

[4] Bohms M., "Framework proposal for aec-appp". Technical report, working paper, ISO TC184/SC5/WG3/AEC, 1993.

[5] Boyle Alan, Step: "An introduction". Technical Report June, SERC: CIPM/LU/TP/1, Department of Civil Engineering, University of Leeds, England, 1992.

[6] Clark S.N., "An introduction to the nist pdes tool kit". Technical report, NISTIR 4336, National Institute of Standards and Technology, USA, 1993.

[7] Clarke J.A., MacRandal D. and Rutherford J.H. "The application of intelligent knowledge based systems in building design". Technical report, SERC Grant GR/E/18018 Final Report, Strathclyde University, 1989.

[8] Fenves S.J., Flemming U., Hendrickson C., Maher M.L., and Schmitt G., "Integrated software environment for building design and construction". *Computer-aided design*, 22(1):27-36, jan 1990.

[9] Gielingh W.F., "*Aec product-type modelling, or: how to use the aec-model?*" . In Tolman F.P., Gielingh W.F., Kuiper P., Willems P.H., and Bohms H.M., editors, *Four years of Product Modelling: Collected papers*, Organization for applied scientific research: TNO-report, pages 18-35, Netherlands, 1989. IBBC.

[10] Kim Inhan, "Data organization and management: In an integrated design environment". In T.W.Maver and Jelena Patric, editors, *Virtual Studio*, ECCADE '94, Glasgow, U.K., 1994. University of Strathclyde.

[11] Kim Inhan, "Data Representations in an Integrated Architectural Design Environmennt: Unified Data Organisation and Management". PhD thesis, Dept. of Architecture and Building Science, University of Strathclyde, Glasgow, United Kingdom, September 1994.

[12] Port S., "The Management of CAD for Construction". *BSP Professional Books*, 1989.

[13] Quinlan J.R., "An introduction to knowledge-based expert systems". *The Australian Computer Journal*, 22(2):56-62, may 1980.

[14] Sauder David A., "Data probe user's guide". Technical Report March, National PDES Testbed Report Series, NISTIR 5141, National Institute of Standards and Technology, USA, 1993.

[15] Schley Michael K., "Cad layer guideline: Recommended designations for architecture, egineering, and facility management computer-aided design", Technical report, The American Institute of Architects Press, Washington, D.C. USA, 1990.

[16] ISO TC184/SC4. "Industrial automation systems-product data representation and exchange - part 1: Overview and fundamental principles", Technical Report September, 1992.

[17] ISO TC184/SC4/WG3. "Industrial automation systems-product data representation and exchange - part 42: Integrated generic resources: Geometric and tolological representation", Technical report, 1992.

[18] ISO TC184/SC4/WG3/P6. "Industrial automation systems-product data representation and exchange - part 11: Descriptive methods: The express language reference manual", Technical Report April, N14, 1991.

[19] ISO TC184/SC4/WG3/P6. "Industrial automation systems-product data representation and exchange - part 21: Clear text encoding of the exchange structure", Technical report, 1992.

[20] ISO TC184/SC4/WG7. "Industrial automation systems-product data representation and exchange - part 22: Standard data access interface", Technical Report September, 1992.

[21] Thomson D., "Nyssen information analysis method (niam) ", Technical report, Control Data, 1986.

[22] Turner J. and Pitman J., "Aec building systems model", Technical report, ISO N363, AEC No. 1.1.3, Doc No.3.2.2.4, The University of Michigan, Architecture and Planning Research Laboratory, 1990.

[23] Wilson P., "Express tools and services", Technical report, obtained via pub/step @ cme.nist.gov, Rensselaer Polytechnic Institute, 1994.