An Integrated Simulation Network

Godfried Augenbroe, Peter Bijvoet, Leo Nederlof, Wouter Rombouts, Peter de Vries

Delft University of Technology
Faculty of Civil Engineering
P.O. Box 5048, 2600 GA DELFT, The Netherlands

ABSTRACT

The paper reports on the NODES project as a spin-off implementation of the EC-funded COMBINE project. NODES targets efficient data exchange in a LAN-connected engineering design team, where each member of the team, representing a separate node, communicates building and performance data with other members through a central data repository. The central conceptual building model and its implementation in an ODB (Object Data Base) is discussed. The focus will be on the CAD-tool supported "design node" and two "simulation nodes", i.e. one for a set of global evaluations and the other for solar irradiation and daylighting assessments. The global performance node deals with a variety of building physics criteria ranging from acoustical (traffic noise) to moisture and energy evaluations, thus confronting us with node-internal integration issues as well. Each node is equipped with a GUI, built from a generic view and selection module, with the aid of the COMBINE interface kit. All actual data exchange is accomplished by file transport over the network and fully adheres to the STEP standard. Both implementation issues as well as user aspects of the simulation network are highlighted.

INTRODUCTION

NODES (Network Object-based Data Exchange System) was started as a spin-off implementation of the EC-funded COMBINE project (Augenbroe 1992). Phase one ('90-'92) of COMBINE (Computer Models for the Building Industry in Europe) was a first step towards the development of intelligent integrated building design systems (IIBDS) through which the energy, services, functional and other performance characteristics of a planned building can be analyzed. The research has concentrated on data integration, based on the concept of a set of separate actors grouped around a central common data repository. The set of actors is as yet still limited as their selection was dominated by the need to address energy and HVAC performance aspects. The next section will give an overview of the COMBINE project. The remaining sections will deal with the setup and components of NODES.

OVERVIEW OF COMBINE FIRST PHASE RESULTS:

A COMBINE pilot study and the ensuing call for proposals led to the joint COMBINE project, carried out by 14 partners from 8 countries. Its long term goal is the development of a future generation of Intelligent Integrated Building Design Systems (IIBDS). Realistically, the short term goal is to concentrate on integration, rather than "intelligence" and thus target an IBDS (dropping the first I) in a limited domain. The first phase of the project was thus meant to take a first step towards the development of an IBDS. A final report of the first phase will be available shortly (Augenbroe 1993). All the data integration is accomplished through the use of tools and methods developed in the international STEP standardization effort (ISO-STEP 1991).

The conceptual architecture of the "delivered" system of the first phase of the project is diagrammed in figure 1. It can be viewed as an off-line data exchange system (DES) for a limited number of design tools. The DES prototype consists of a set of design tool prototypes (DTP), logically shared around the common conceptual data model. The Application Interface executes the mapping between the IDM and the aspect model of the design tool. The six DTP's that are developed address the following tasks:

- DTP-1: Construction design of external building elements
- DTP-2: HVAC-design
- DTP-3: Dimensioning and functional organization of inner spaces
- DTP-4: Thermal simulation tool in the late design stage
- DTP-5: LT-method in the early design stage
- DTP-6: Radiator network design

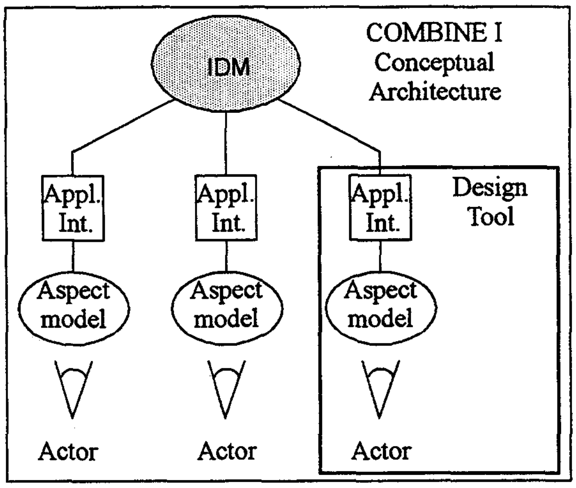

Figure 1: COMBINE1 prototype architecture

At this stage of the overall research strategy no attempt was made to include an architectural design actor. This will be one of the targets of the second phase. The data exchange is realised through the following IT-tools:

- a conceptual building model (IDM) that integrates all DTP-actor views.
- Implementation of the conceptual IDM in a powerful software platform.
- data exchange facilities through which the DTP's can communicate with the IDM, delivered as a STEP-based interface kit. The actual communication is accomplished through STEP neutral file (ASCII) exchange.

Main deliverables are:

Conceptual Building Model

A key output, as the resulting integrated data model (IDM) enables the integration of all communicating actors. The IDM is available in NIAM diagrams as well as in EXPRESS (STEP modeling languages (Spiby 1991)). It should be stated that the present IDM meets only a few of the long-term requirements on ultimate building models; the output reflects the present state of the art. For easy access to the resulting model, an EXPRESS browser is provided. The browser enables easy navigation between entity definitions, with access to EXPRESS text and dynamic updates of user-defined concepts and creation of new entities.

IDM implementation

The implementation of the IDM relied on the XPDI software environment used by the French group in charge of this task (Poyet, 1993). The translation from conceptual schema to the XPDI implementation involves the definition of classes and class-hierarchies, the direct translation of the relationships from the NIAM diagrams and the addition of the implementation structures (frame- slot approach) within the XPDI station. The following additional activities were carried out:

- development of a XPDI-function for exporting the schema in EXPRESS format and producing STEP-neutral format files for instantiated models, according to subschema-specification in EXPRESS.
- development of the COMSET system (as part of XPDI) which supports rapid definition of NIAM structures. The tool supports NIAM diagram generation, interactive update facilities, NIAM to EXPRESS translation.

The present implementation supports the exchange of STEP-files according to predefined subsets of the complete schema.

IDM-Data Exchange facilities

The COMBINE approach is at this stage dominated by the view that DTP's act as "remote" tools, in a heterogeneous (with respect to hardware and software) system of loosely coupled actors. Thus the basic data exchange is accomplished through ASCII file exchange with full adherence to the STEP-standard (off line data integration). The off-line data exchange facility is provided as a development kit enabling external DTP-developers to interface their DTP to the COMBINE IDM. The interface kit provides:

- an EXPRESS parser delivering C++ software (i.e. class definitions) that together with added own mapping software (DTP-specific) forms the mapping module.
- C++ software that handles generic functions that should be available in any mapping module, like import and export of STEP-files. No generic solutions for the DTP-specific mapping (e.g. a mapping language) has been attempted as yet. At this stage every interface development has to deal with DTP-specific mapping functions from scratch. Depending on the requirements of the DTP, the outcome of the mapping process may be a plain ASCII file, a database or a neutral "object dump" that can be moved to another software environment.

DTP's

The six DTP's were built in the hardware/software environment of choice. No standardization of implementation approaches is required, as heterogeneity across DTP's is one of the starting points. Each application contains:

- an application interface for two-way data exchange with the IDM, handling the mapping/translation functions. This interface will in most cases be a separate module built with the help of the interface kit, distributed by the IDM team. The standard interface (a minimal option for each DTP) will be based on EXPRESS/STEP file-based data exchange.
- a user interface, giving a design-actor access to and control over the DTP- functions. In most cases the user interface will also allow access to the interface module, e.g. for sub-model selection.
- the application (run) module itself, which may use existing BPE-modules.

Demo Prototype

The first phase of the project ended with a COMBINE seminar and workshop where a prototype of the data exchange system was demonstrated to practitioners (COMBINE consortium 1992). Moderate goals were reached, in the sense that the demo showed the set of actors exchange data in the context of a simulated project, according to a hardwired predefined scenario. The primary aim was to be able to show one support layer of cooperative group work, i.e. the data exchange layer, in action, be it in a laboratory setting.

Parallel to the COMBINE project the local NODES project was started to explore implementation of the COMBINE architecture in the design/engineering office environment.

NODES project

NODES is concentrating on providing efficient data exchange between a number of "nodes" in a network, each node being identified as an actor at a workstation in a LAN- connected design team. Each node performs certain functions, using a suite of software tools. The inclusion of a CAD-supported "design actor" is regarded as an essential requirement in this set-up. The implementation of these facilities is accomplished through the definition of a common conceptual building model and its implementation in a central ODB (Object Data Base), whereas the actual data exchange is based on file transport over the network, Each node is equipped with a NODE-interface to handle the translation and mapping to the local actor-view. These graphical user interfaces are developed with the aid of the "interface kit", a COMBINE-deliverable.

Although NODES employs the same conceptual architecture as COMBINE, there is a number of significant differences in the approach, namely:

- the integrated building model has been developed from a different view point than the IDM. Strong topology/geometry support was deemed essential, whereas the set of Nodes covers a smaller subset compared to the information in the IDM. This led to a new conceptual building model, which will be discussed below.
- the implementation is targeted towards mainstream robust software platforms, based on C++.
- the inclusion of a CAD-actor, which explains the emphasis on shape description in the central model.
- tailoring towards use in an office setting with LAN connected PC's and one central server (UNIX).
- uniform GUI development under MS-Windows for all Nodes.

Figure 2 shows the four nodes (actors) presently incorporated in the network.

- 3RM: a conceptual CAD tool, for early architectural design.
- SIBE: a solar irradiation and daylighting package.
- MAT: a materialisation and instantiation module.
- GTS: a global toolset building physics simulation.

All nodes interact through the central NBM-node, holding the NODES Building Model.

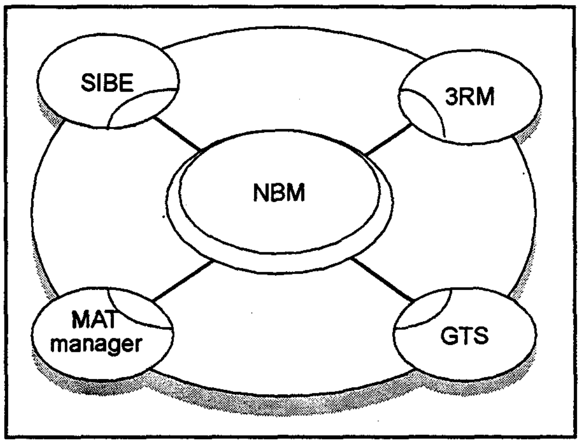

Figure 2. Nodes in NODES

In this paper we will focus on two nodes: the architectural design node (supported by a conceptual CAD tool) and a so-called global tool set (GTS) node. The GTS node provides a suite of "global" building physics simulation tools in the area of heat; moisture, acoustics and daylighting.

Three of the four Nodes (3RM, GTS, SIBE) consist of existing software packages, so interfaces had to be built external to each of them. These interfaces were kept uniform through the creation of a number of generic modules. The next sections will describe these and other components in the NODES network.

INTRODUCING THE ACTORS IN NODES.

3RM-Node

3RM is a prototype 3D CAD tool developed in the Netherlands. This tool is well suited to define an urban site along with a particular detail of a building, which for instance can be used for a solar evaluation.

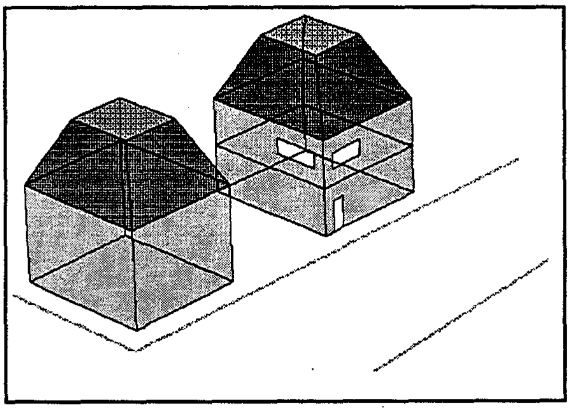

Fig 3. Buildings in 3RM, used as input for solar evaluation

Although 3RM lacks a great deal of functionality, provided by other popular CAD tools, it supports a more powerful topologic model. It supports a representation called 3RMREP, which offers shape describing capabilities beyond BREP.

Like BREP, faces must be bounded by lines, spaces by faces and lines by points.

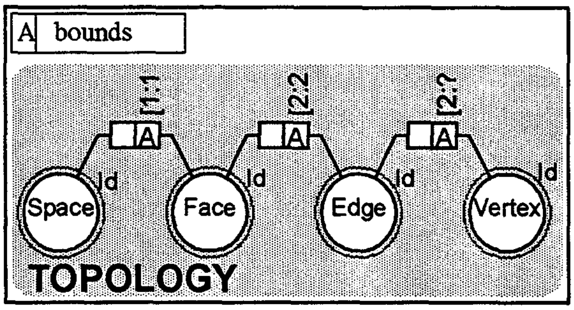

Fig 4. Boundary Representation (geometry not shown)

In 3RMREP however a point bounds 0, 1 or more lines, a line bounds 0, 1 or more faces and a face bounds 0, 1 or 2 spaces. It is not a solid but a "hollow" model. 3RMREP allows a single line, face or point as a legal instance of the model. This is quite important because in reference topology, all topologic entities have to be able to exist without the need of bounding other entities. In this way a face can be used as representation of a wall without the need of specifying a solid that is bounded by that face (this is necessary in BREP). Later it is always possible to define a space (not solid!), possibly as representation of a bathroom, bounded by that face. The boundary relation wall and room can then be derived from topology.

3RM supports only a low level of geometry, i.e.cartesian points, straight lines and flat faces.

A feature of 3RM is its ability to link "objects" to points, lines, faces and spaces of the 3RMREP model. Thus references from walls, roofs, and living rooms etc. can be made by arbitrary topologic entities.

The 3RMREP model may be an advanced model from the current CAD-tools perspective, it is not suited for use as topological backbone of a building design system. It does, for instance, not enable definition of walls with different characteristics on either side. 3RMREP is therefore only a local actor's view in the system.

The NODES-interface of 3RM performs as an offline connection to the the Object DataBase (ODB) which maps the local 3RM actor view to the richer (superset) topology/geometry in the central building model. It works only one way (from 3RM to ODB). After a drawing session, a file is saved in a 3RM specific fileformat. The 3RM-Interface maps from 3RMREP to the shape representation in the NBM-schema and produces a neutral STEP file according to this schema.

The data in the STEP file is then loaded into the ODB through straightforward creation of the instances found in the STEP file into the database

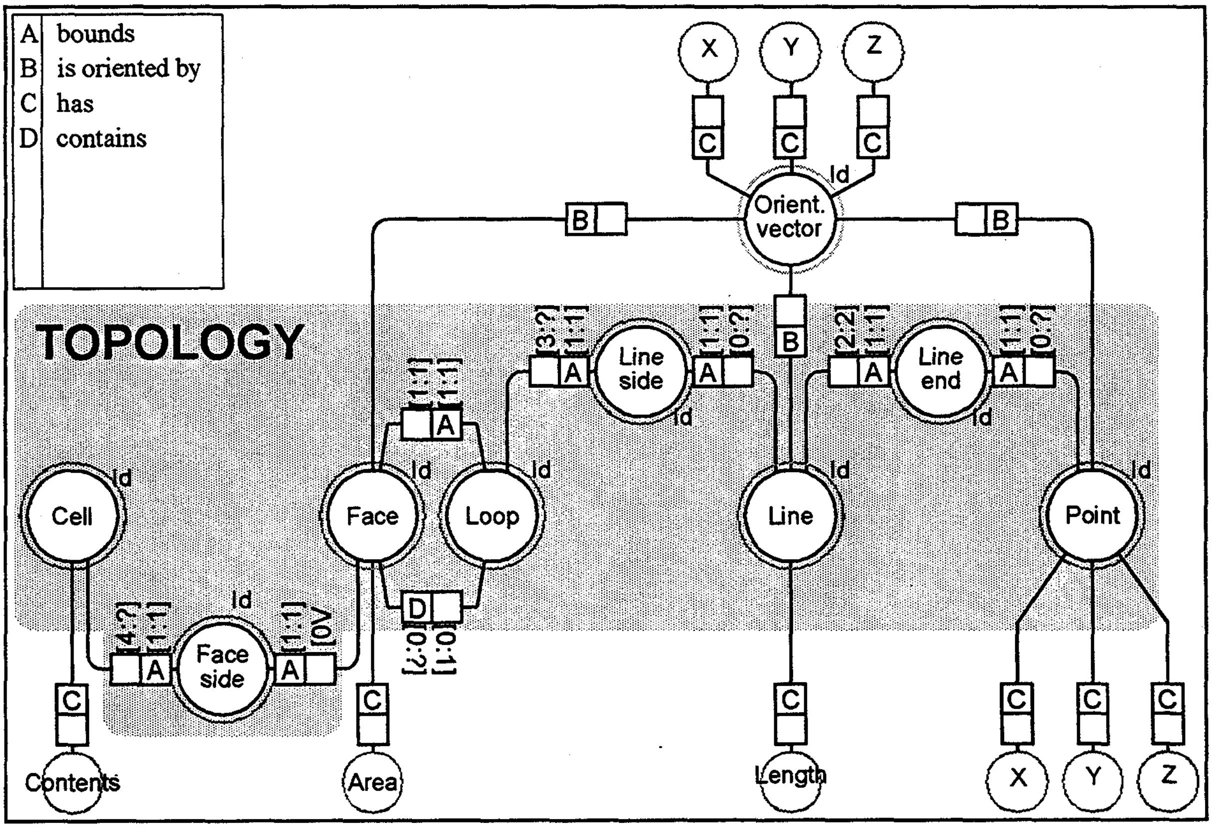

Fig 5. RMREP Topology used in NODES with all geometry shown. Geometry for line and face is implicitly defined as straight and flat.

CENTRAL NBM-node

The shape representation in the NBM is a superset of 3RMREP, as it will be required to support multiple shape views i.e. from all participating actors. We will refer to this superset as RMREP.

RMREP is very alike 3RMREP but is enhanced with line-end, line-side, loop, face-side

The conceptual modeling task consisted of interpreting the shape description above (the enriched view of the 3RM actor) with all other actors views. As an initial approach a bottom up model was made by adding all the conceptual models of the nodes in the system together. Then all the overlapping data was removed. After examining the model, data used by one node only, was deleted. The resulting model contained only entities that were necessary for any actor communication. This model turned out to be very "narrow" and still application flavored. New actors would be hard to integrate and the need for a more neutral conceptual model was obvious. A new conceptual model was made with the help of some concepts from the GARM (Gielingh 1988), improving the structure of the original model. Some observations can be made:

- Only two abstraction mechanisms were used (Composition & Aggregation).
- RMREP topology was embedded.
- The modeling effort was confined to the set of actors we wanted to integrate.

Figure 6 shows the relevant concepts and decomposition levels in the NBM.

SIBE-Node

SIBE (Voorden 1985) consists of separate kernels for direct solar irradiation (SIBE1) and daylight assessments (SIBE2). We will briefly describe some characteristics of SIBE1.

SIBE1 calculates solar irradiation on arbitrary planes (planes of interest) in arbitrary shaped building configurations. One of the major advantages of the SIBE kernels is that it determines irradiation data mathematically. This in contrast to ray tracing techniques. This quite unique approach has some reflections on its external appearance. Some limitations to validity of its input result from the programming language it is written in (FORTRAN), others relate to the constaints on allowed shapes of objects.

SIBE1 allows multiple ways of defining the build environment.

- The concept of building contour (of rather simple shape) enables a very compact input to the program. A building is described as a 3Dloop which defines the edges of the roof. Walls are implicitly assumed from the edge vertical to the ground. It is very convenient, if the inputfile is generated in a simple editor, to use this concept. Although the inputfile will be compact, there is no extra gain in terms of memory use or speed of the application, compared to the following input definition.
- The alternative way of defining a building is definition via faces (this definition is used in NODES). In this way more complex shapes can be handled. The input to the program is bigger this way but it is of no concern because the input is generated automatically, using the description of the building in the ODB. A disadvantage of this type of input is that SIBE1 just receives a series of "scattered" faces void of a building concept, consequently it cannot generate output related to a building. Building related output, like the role of a building for the solar irradiation, however is retrieved by using the

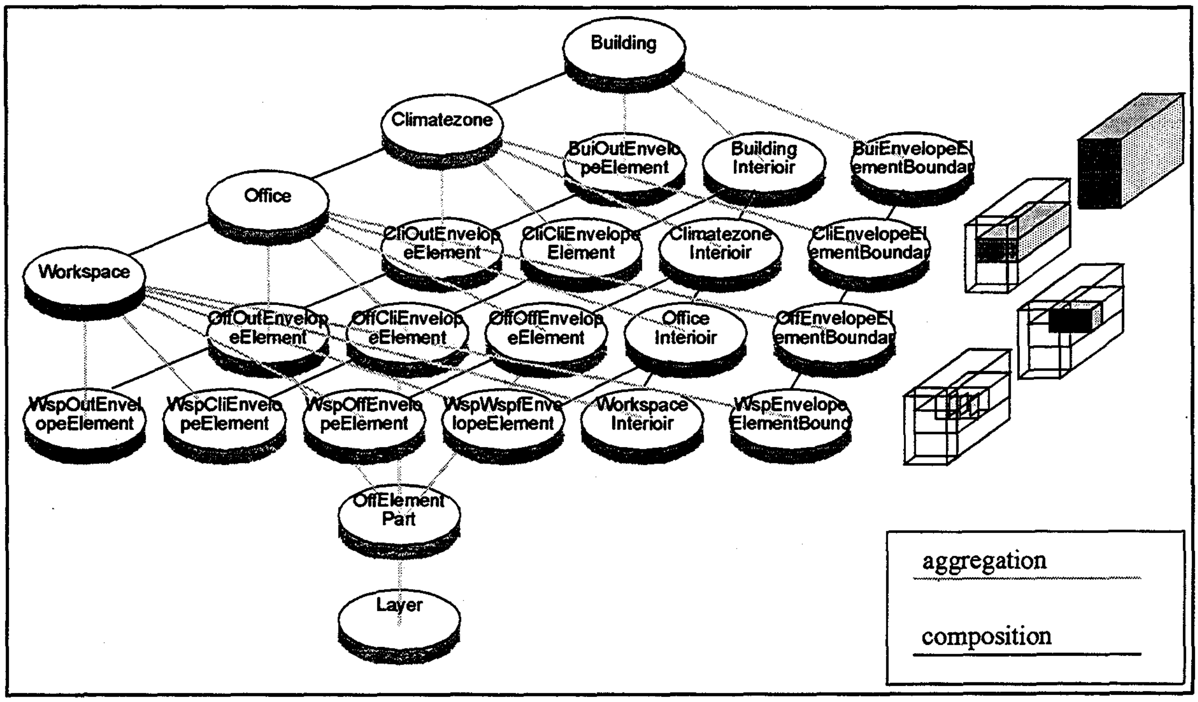

Fig 6. The conceptual building model of NODES (smoothed version).

knowledge of which faces 'compose' a building. This information is available in the ODB.

The output SIBE1 provides is direct exposure data of planes of interest. Information about which entities influence the radiation of the plane of interest is also available.

GTS-Node

The global toolset contains a suite of tools for day to day use by the environmental physics consultant. The commercially available set comprises the usual tools that help to assess the performance of the planned object in the following areas:

- heat (Fanger comfort, yearly energy consumption, heat transmission, overall loss factor, solar gain factor)
- moisture (external/internal condensation risk)
- light (daylight penetration, shadowing assessment)
- acoustics (transmission, facade characteristics, industrial sources, mass laws, prescribed calculation procedures for traffic noise).

The variety of tools requires that we define one Node-specific aspect model, which integrates the different data requirements of internal tools (figure 7).

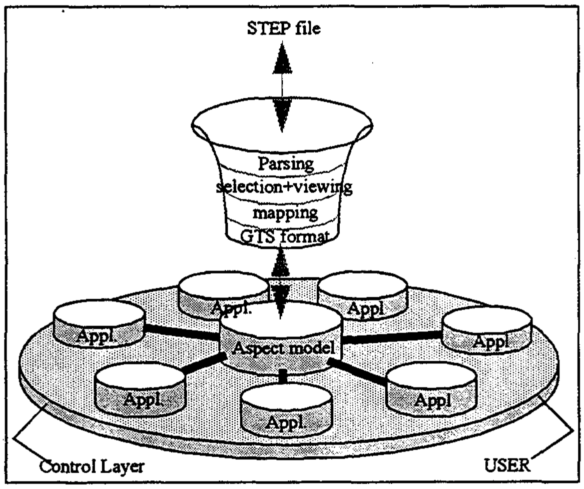

Fig. 7.GTS Node + Internal architecture

All internal applications exchange data with the local aspect model. The user is allowed to access the control layer which gives him permission to run the different applications and access the aspect model. If necessary the user is prompted for additional information, e.g. native input data and run control parameters. The control layer supervises the sequence in which local applications are used, certain sequences are not allowed, unless the use intervenes and adds data to the aspect storage manually.

NODES INTERFACE

Figure 8. shows the role of the interface module in the system.

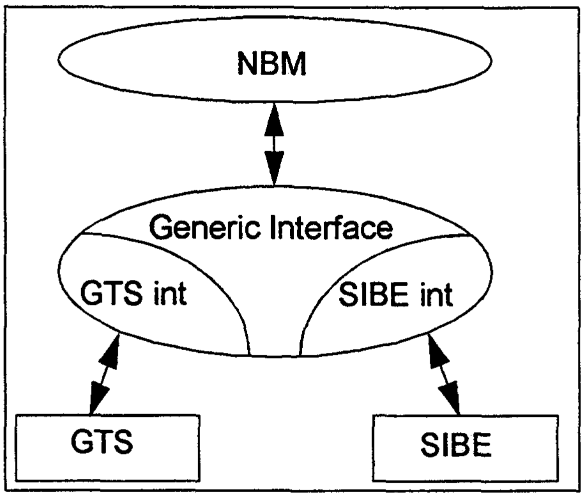

Figure 8. Role of Interface Module

The following generic functions are performed:

- communicating with the database in terms of EXPRESS Schema and related STEP files (containing the objects)
- graphical presentation supporting the selection of objects (view and selection module)

The Node-specific function consists of:

- export and import of (selected) objects in the application specific format, and according to the application specific data model (mapping modules).

The generic module is built with the COMBINE Interface kit, which provides the NODES Interface access to data contained in a STEP file and enables it to produce STEP files.

This "STEP part" of the interface has both EXPRESS schema dependent and schema independent structures. The schema independent provide general read write functionality, while the schema dependent part enables the reading and writing of STEP files according to a specific EXPRESS schema, diagrammed in figure 9. The interface kit consists of two basic elements (both based upon NIST PDES toolkit elements (Clark and Libes 1992)):

- The Combine Class Generator (CCGen). CCGen generates the schema dependent parts of the interface. It reads an Express file and generates C++ classes representing entities described in it.
- The Combine Class Library (CCLib). CCLib consists of general parser classes. All CCGen generated code is related to CCLib class elements: all classes have subtype relations with the general

CCLib STEP_Entity, other sourcecode is produced to initialise CCLib class instances of STEP_Parser.

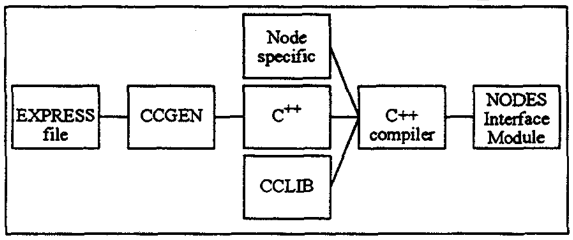

Figure 9. Interface building

The STEPfile data is available in the NODES Interface by means of instances of STEP_Parser (reading, writing and validation of exchange files) and STEP_Entity (reading, writing and validation of instances in these files). The STEP_Entity based instances are equiped with basic "meta-schema" information (they have a "Type" and a set of "Attributes"). This makes general access and selection procedures possible.

The view and selection module is built with C++ in the MS-Windows environment. After loading all STEP file objects, those that have an "attached" shape representation are drawn in multiple presentation windows allowing different view angles. Selection of objects is mouse supported, whereas several pop-up windows show picked entity type and picked instance ID. Selections can be marked to be exported. Figure 10. shows the screen in a view/ selection session.

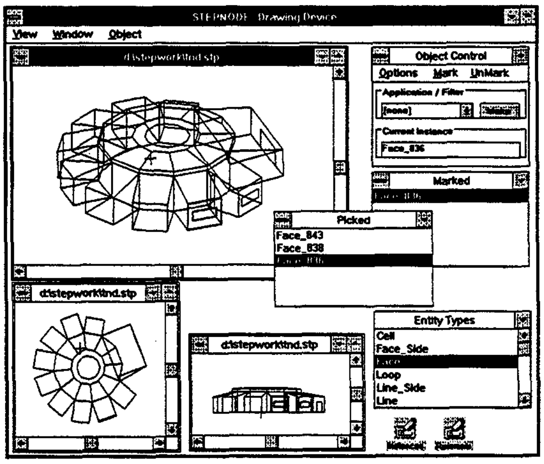

Figure 10. Interface screen during selection and marking.

System use

The actual system consists of a thin ethernet LAN connecting all nodes (PC's) to the central node (UNIX). The communication protocol uses PC-NFS. This allows all nodes to open the ODB and download or send STEP-files. At this moment a simple executive module is being developed, able to control and manage the operating sequence of the data interchange. This module logs all node actions and controls permission and ODB locking.

In principle we do not prescribe a particular sequence for the execution of the applications. The simulation procedure allows one node at a time to evaluate the project. The applications may be scattered over a network randomly and applications may occur on more than one place at a time. All application's feedback to the ODB is monitored and only after successful checking the database is updated and freed for other applications.

Conclusions and future developments

We have shown that the basic technology is available to build simulation networks, where multiple simulation tools can co-exist in one simple data environment.

But it must be recognized that data integration is not the ultimate goal, but only one step in the direction of future integrated systems.

If we enlarge our scope from simulation to design, we conclude that the system lacks the support for real building projects. This is the main theme of the current second phase of COMBINE, which is now in progress. This phase will build upon the available deliverables by combining them into an operational IBDS according to functional specifications resulting from particular building project settings.

REFERENCES

Augenbroe, G. 1992. "Integrated Building Performance Evaluation in the Early Design Stages". *Building and Environment,* Vol. 27, 2 pp 149-161.

Augenbroe, G. 1993. *"COMBINE Final Report First Phase"*. CEC- DGXII Report (to be published).

Clark, S.N. and D. Libes. 1992. *"Fed-X: The NIST EXPRESS Translator"*, NISTIR 4822, NIST Gaithersburg.

COMBINE Consortium. 1992. *COMBINE Seminar Report.* Stuttgart, 20 November.

Gielingh, W.F. 1988. *"General AEC Reference Model (GARM)"*, TNO Report, Delft.

ISO-STEP. 1991. *"Product Data Representation and Exchange, STEP"*. ISO-TC 184/SC4.

Poyet, P. 1993. *"XPDI Manual"*. CSTB Report.

Spiby, P (ed.). 1991. *"EXPRESS Language Reference Manual"*. ISO TC184/SC4/WG5, Doc. N14, NIST Gaithersburg.

Voorden, M. van der. 1985. "Determining the Rate of Solar Irradiation in the Built Environment". In *Proceedings of the Ninth Biennial Congres of ISES,* Vol. 1, pp 425-430.