An Integrated Software Environment **for**
Building Design and Construction

by

SJ. Fenves, U. Flemming, C. Hendrickson, M.LMaher and G. Schmitt

EDRC-12-21-88

An Integrated Software Environment for Building Design and Construction

by
S.J. Fenves, M. ASCE,[1] U. Flemming,[2]
C. Hendrickson, *AM.* ASCE,[3] MX. Maher, AJVI. ASCE,[4] and G. Schmitt[5]

Abstract

The design and construction of buildings involves many different professionals and large volumes of diverse information. Presently, the major media for communicating design decisions are drawings and specifications. Along with the ability to automate portions of the design process comes the need to communicate design decisions in other forms. Electronic communication to date has largely taken the form of files of numerical data. As the use of the computer moves towards symbolic reasoning, communication by simple files of primarily numerical data become inadequate.

This paper describes an integrated environment of processes and information flows for the vertical integration of architectural design, structural design and analysis, and construction planning. The integrated environment makes use of a number of Artificial Intelligence techniques. The processes are implemented as Knowledge Based Expert Systems. A Blackboard Architecture is used to coordinate communication between processes. The global information shared among the processes is hierarchically organized in an object oriented programming language.

The paper describes the current state of the integrated environment and its major knowledge-based processes. Emphasis is placed on the nature and structure of the information communicated among the processes. Implications are drawn on the nature and contents of a global database needed for computer-integrated construction.

1 Introduction

The building construction industry is characterized by an organizational structure in which a number of different organizations participate in the planning, design and construction of each building project This distributed project organization requires the communication of large volumes of diverse information between the participants in a timely fashion. Presently, the major media for communicating this information are drawings and written specifications. As increasing portions of the design-construction process become computer-based, the need for appropriate forms of electronic communication becomes increasingly apparent Furthermore, as computer use shifts from purely numeric calculations towards symbolic and knowledge-based reasoning, there is additional need to communicate symbolic information as welL

The dispersed nature of the construction industry raises further communication issues. On the one hand, there is no pool of common knowledge: since architects and structural engineers only perform design, they do no possess first-hand knowledge of construction practices and what con-

[1] Sun Company Professor of Civil Engr, Dept of Civil Engr, Cirnegie Mellon University, Pittsburgh, PA 15213

[2] Assoc. Professor, Dept. of Architecture, Carnegie Mellon University, Pittsburgh, PA 15213

[3] Professor, Dept. of Civil Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

[4] Asst. Prof., Dept of Civil Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

[5] Asst. Prof., Dept of Architecture, Carnegie Mellon University, Pittsburgh, PA 15213

stitutes a constructible desiga On the other hand, there is no direct way for constructors to provide "feedback" to designers on what changes in the design may improve constructability and thus reduce cost

The prototype integrated environment described in this paper is intended to serve as a testbed for examining these communication issues. The integrated environment addresses the issue of planning, structural design and construction planning of office buildings. In order to address a broad range of communication and control issues, attention is primarily focused on three aspects: (1) the representation of the project information as the project progresses, (2) the communication of that information among the participating computer-based processes, and (3) the control of the overall process.

The Integrated Building Design Environment (IBDE) system is implemented in the form of five vertically integrated knowledge based processes (Figure 1): an architectural planner (ARCHPLAN), a preliminary structural designer (HI-RISE), a component designer (SPEX), a foundation designer (FOOTER), and a construction planner (CONSTRUCTION PLANEX).

The processes communicate with each other in two ways: (1) a message blackboard is used to communicate project status information such as whether a process is ready to execute, has successfully performed its task, or has encountered a failure; and (2) a project datastore used for storing the information generated and used by the processes.

A controller uses the information posted on the blackboard to initiate the execution of individual processes. The controller also directs the data manager to provide and receive the information shared between the processes. Since the different processes may reside on different machines, the data manager and the blackboard rely on a local area communication network.

2 Knowledge-Based Processes

2.1. ARCHPLAN

ARCHPLAN (Schmitt, 1987) is a knowledge-based ARCHitectural PLANning expert system which assists in the development of the conceptual design of a building. The input describes the site, the client's program, budget, and geometric constraints. The output provides three dimensional functional, circulation, and structural information. ARCHPLAN subdivides the conceptual architectural design process into four distinct, but interrelated decision modules dealing with building site, function, circulation, and structure. The program establishes at the beginning a generic prototype of an office building which the user and the built-in heuristic and algebraic knowledge can refine.

Program interaction starts with the the *site, cost, and massing module* (SCM) by default. After a building site is determined, preliminary design begins with the development of a massing model that will fit a given budget and a range of other parameters listed in Figure 2 (top). Cost and massing options are inter-dependent concerns. Site characteristics are considered to be facts and therefore fixed, whereas building requirements are more flexible. The user describes the degree of commitment to a certain requirement, such as floor-to-floor height or ground floor area, by entering a certainty factor between 1 and 10. The SCM module also contains simple optimization options: minimum cost, maximum daylighting, or a combination of the two. The user is able to graphically compare the optimized design with her or his "manual" design and choose either one for further evaluation.

The *function module* assists in the vertical and horizontal distribution of building functions within the basic massing volume (see Figure 2, bottom). Examples of building functions are office, retail, atrium, mechanical and parking space. Each function has particular requirements and affects the layout, appearance, and cost of the building. ARCHPLAN contains a set of function distribution prototypes which can be manipulated and refined by the user. Typical

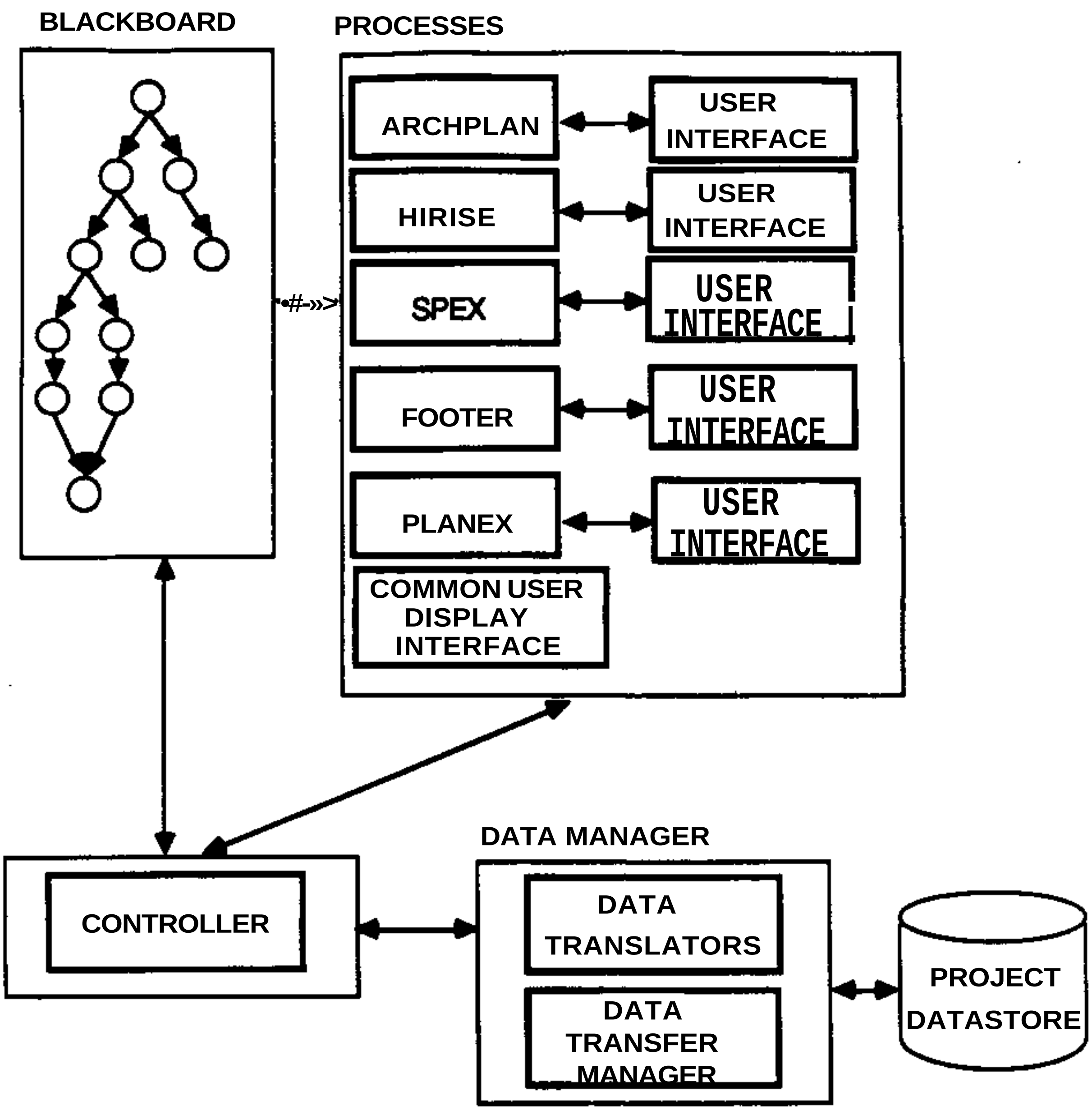
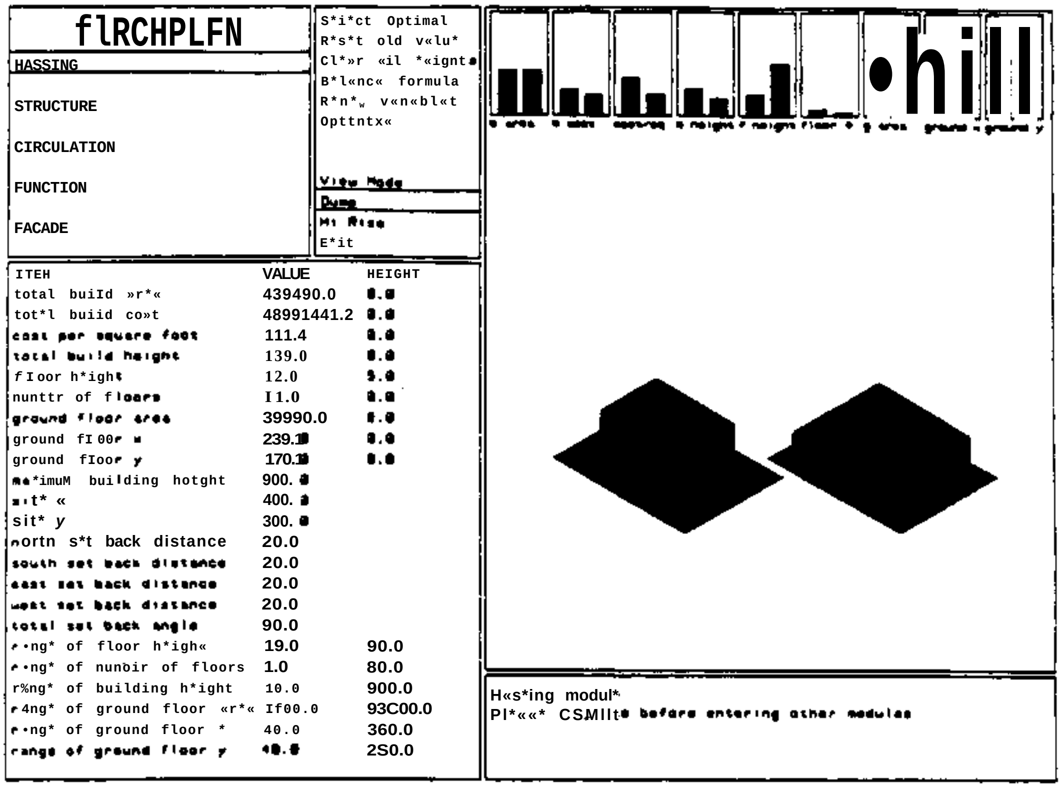


Figure 1: Architecture of the Integrated Building Design Environment

manipulations are the percentages of each of the functions, and the location of parking and mechanical floors. The resulting three-dimensional schemes are displayed as solids or as wire frames. Functional decisions are made and reflected locally, unless the constants in the global building description object are violated. In this case, the program backtracks and control is passed back to the SCM module. In the SCM module, the user can choose either to automatically adjust the design description to the information received from the Function module, or make changes manually.

The *circulation module* addresses the problem of moving occupants and equipment from floor to floor and within floors and of guaranteeing the safe evacuation of the occupants in emergencies. Circulation also has a major impact on the internal functioning and on the architectural



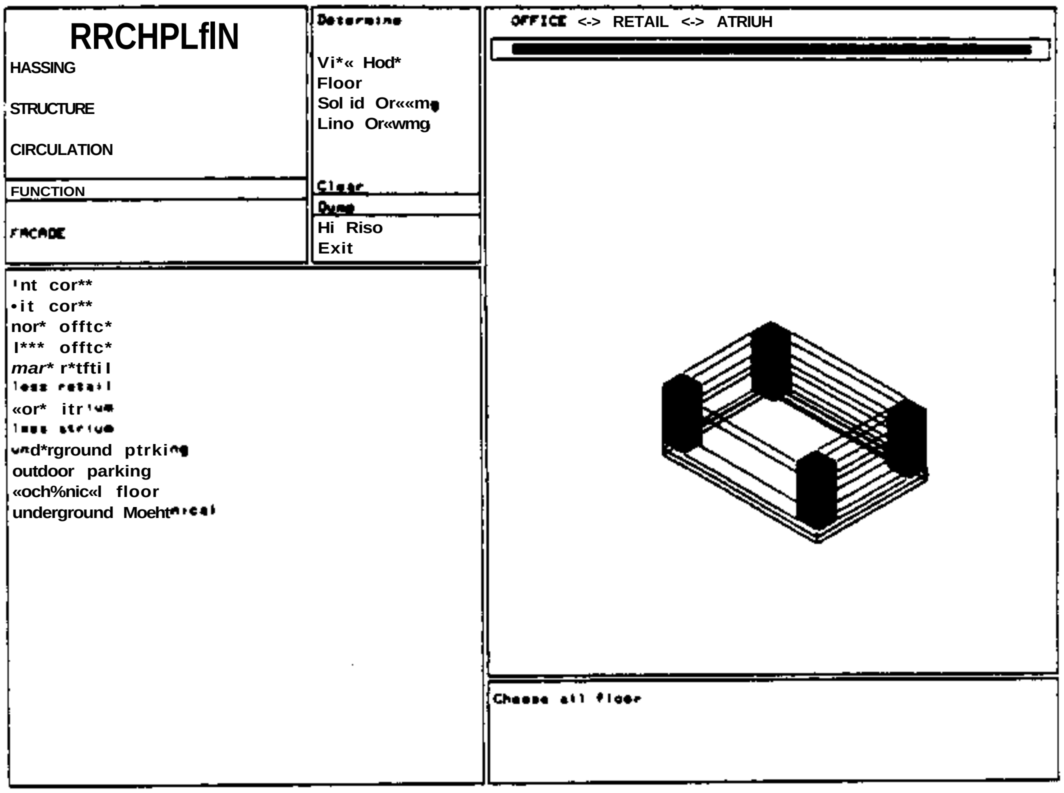


Figure 1: Top: user view of the SCM module. The graphic window shows the manually constructed massing model on the left and the building, optimized for minimal initial cost on the right.
Bottom: user view of the function module. The circulation towers are shown as solids, the other building functions as wire frames.

expression of a high-rise building. The two extreme cases for the placement of vertical circulation are the internal (service and elevator core in the center of the building) or the external solution (service and elevator cores attached to the outside of the buildings). ARCHPLAN concentrates on creating vertical circulation proposals based on variations of these two prototypes and considers a range of factors which influence the placement of the circulation elements. The relative importance of each factor is determined by moving horizontal bars.

All design decisions in the previously described modules have an impact on the type and performance of the building's structural system. The *structure module* is intended to give the designer an overview over possible structural types appropriate for the building design. The synthesis of a structural system for a design developed with ARCHPLAN is reserved for the HI-RISE process in the integrated design environment The Structure Module considers up to eight structural systems. If the building has been defined through the previous design decisions, the options are limited. If the Structure Module is executed early in the design process, the set of selectable structural types is larger. After the user has accepted the proposed structural type for the given building or has made an independent choice, the structure is displayed three-dimensionaily.

2.2. HI-RISE

HI-RISE (Maher, 1984) is a knowledge based expert system that performs the preliminary structural design of high rise buildings. HI-RISE generates and presents alternative feasible structural systems. The input to HI-RISE is the three dimensional structural grid produced by ARCHPLAN, specifying potential locations for beams, columns, and walls. The topology of the grid is defined by the number of stories and the number of bays in each direction. The geometry is defined by the dimensions of die bays and the minimum required clearance for a typical story. Other spatial constraints, such as the location of service shafts or internal spaces, are specified in terms of their location on the input grid. Other input information required by HI-RISE are the intended occupancy of the building, the wind load, and the live load.

In HI-RISE, the preliminary design process is divided into two major tasks; each task addresses the design of a functional system. The functional systems are designed in a fixed order: first the lateral load resisting system, followed by the design of the gravity load resisting systems. Each of the two major tasks are decomposed into a set of similar subtasks. The subtasks have the same goals for each functional system, however, the details of reaching these goals differ. The subtasks are described below.

- *Synthesis.* The first task is to synthesize alternatives for the functional system under consideration. The synthesis is performed as a depth first search through the appropriate generic subsystems stored in the knowledge base. Each subsystem has a set of synthesis levels with itemized alternatives at each leveL For example, the levels and alternatives for the lateral load resisting systems are as follows- 3D-lateral: core or 2D orthogonal; 2D-lateral: rigid frame, braced frame, or shear wall; material: steel or reinforced concrete. Synthesis is implemented as sets of rules that represent the decisions to select an alternative at a given level, move to a lower level, move to a higher level, or eliminate an infeasible design alternative.

- *Analysis.* The purpose of the analysis subtask is to define component groups, so that preliminary design is performed only for one component in a group, and to estimate the required load capacity of components by means of an approximate analysis.

- *Evaluation.* Evaluation of a structural design may be based on many features of the design. Features considered include cost, efficiency, and structural integrity. HI-RISE evaluates alternatives with a linear evaluation function. There is a distinct function for each of the functional systems.

- *System Selection.* HI-RISE presents all structurally feasible systems to the user, indicating which system has been determined to be the "best", selected as the system with the minimum value assigned by the evaluation function. The user may either accept die recommended design or override the decision of HI-RISE and choose one of the other structurally feasible systems.

23. SPEX

SPEX performs the preliminary design and structural components for the structural system selected by HI-RISE (Garrett, 1986). SPEX implements a structural component design strategy in which components are designed automatically by applying three types of knowledge: knowledge contained in design standards; "textbook" knowledge of structural, material and geometric relationships; and designer-dependent design expertise.

The knowledge contained in design standards is represented by a formal model. Standards *requirements* are represented as decision tables, with each requirement having a status value of "satisfied", "violated" or "not applicable". The complete set of requirements is represented by an organizational system which uniquely describes each requirement in terms of the type of object, stress type, limit state and object composition it addresses. Each unique path through the taxonomy to a requirement is described in a standard-independent fashion (e.g., object - beam; stress * flexure; limit state * plastic yielding; component - flange) and is called a *behavior limitation.* The requirements of a particular standard are linked to the standard-independent behavior limitations.

The "textbook" knowledge supplies relations and definitions referenced but not explicitly found in the standard (e.g., a standard may specify the plastic yield capacity in terms of the plastic section modulus, Z, only, and textbook knowledge is needed to express Z as a function of the element's dimensions).

The designer-dependent knowledge consists of rules for generating a *design focus,* which identifies one or more behavior limitations that the designer hypothesizes as governing the design of a specific component

In the IBDE implementation, SPEX receives as input the design parameters for each component group, including: type of component (e.g., beam, column), length, material (steel or concrete), and estimated loads. The SPEX interface supplies the material grade, the name of the design standard, the design focus, and an opbmality criterion.

The design process in SPEX is implemented in die following six design process modules:

- *Design Focus Generation.* The designer-supplied rules base is used to select a design focus.
- *Requirement Retrieval.* The requirements of the named standard corresponding to the design focus are retrieved
- *Constraint Set Generation.* The constraints resulting from the requirements are assembled.
- *Constraint Set Satisfaction.* An optimal solution to the constraints is generated using either mathematical optimization or database lookup.
- *Conformance Verification.* In the case of detailed design the solution based on the initial design focus is verified against *all* requirements pertaining to the component If some requirements are violated, SPEX backtracks either to generate a new set of constraints or a new design focus.

The output of SPEX is the description of the optimal component (e.g., square concrete column 12" wide with reinforcement ratio of 0.025, or steel beam designation W 12 x 84).

In the IBDE implementation of SPEX, two shortcuts have been introduced for efficiency: a) constraint set satisfaction is performed by table lookup, from pre-generated tables of optimal component configurations; and b) only preliminary design is performed, so that conformance verification, and thus the possible need for backtracking, is omitted.

2.4. FOOTER

FOOTER is an expert system that performs a preliminary design of the foundation of a building. FOOTER is implemented in EDESYN (Maher, 1987), an expert system shell for engineering design synthesis. The input to FOOTER includes: (1) soil conditions, such as the presence of obstructions, location of water table, depth of bedrock, and soil classification, and (2) imposed load conditions from the structure provided by HI-RISE. The knowledge base includes problem decomposition, alternative partial solutions, and constraints on illegal combinations of partial solutions. The design strategy provided by EDESYN is a constraint directed search for feasible solutions.

The problem of designing a foundation is decomposed into several subproblcms: selection of foundation type, casting type, excavation type, width, thickness, and percentage of steel. The alternatives considered for each of these are given below:

- **The foundation type is limited to individual footings in the current version. This is being extended to include piles and mats.**
- **The materials considered include reinforced concrete and prestressed concrete. With the extension above, steel, timber and composite material alternatives will be added.**
- **The casting type is either precast or cast-in-place. Planned extensions are as-is (for timber and steel piles), cast-in-place cased, and cast-in-place uncased.**
- **The excavation types include pnuematic driven, excavated and bored.**
- **The width, thickness and percentage of steel are approximated using the input information and the selections from the above subproblems.**

The constraints in FOOTER include heuristics on the appropriateness of a specific foundation design for the given input conditions. For example, a timber pile is not appropriate if the column load exceeds 270 kips. Other constraints eliminate alternatives that are inconsistent combinations of partial solutions.

2.5. CONSTRUCTION PLANEX

CONSTRUCTION PLANEX (Hendrickson, 1987a) is a knowledge intensive expert system intended to assist the construction planner. The system suggests technologies; generates necessary activities and precedences; estimates durations and required resources; and develops a construction schedule. The system will either generate a plan automatically or a planner can review and modify decisions during the planning process. The input to PLANEX consists of specifications of physical elements in the design (provided by other processes) which may be grouped by section; site information (such as soil type and elevations); and resource availability (such as number of crews or equipment types). The output from FLANEX consists of a complete plan of construction activities including a provisional schedule and cost estimates.

The operation of CONSTRUCTION PLANEX relies heavily on a large number of distinct *knowledge sources* (Zozaya, 1987). Each knowledge source is applied to a small portion or task in the overall planning process. An example knowledge source is illustrated as a decision table in Figure 3. In this case, the knowledge source suggests different sets of activities required for the construction of a footing depending upon soil characteristics. While knowledge sources can be represented as a set of rules or a decision table, they are represented internally as frame schemas. By supporting numerical functions, calls to other knowledge sources and binding, CONSTRUCTION PLANEX knowledge sources perform as small expert systems themselves.

In the initial creation of a construction plan, PLANEX performs the following sequence of operations:

- ***Create element activities* for design elements. This operation identifies a set of element activities required to construct each design element. Decisions about precedences among activities, technologies to employ, required resources, etc. are made by other operators.**

Figure 3: Illustration of a CONSTRUCTION PLANEX Knowledge Source

Name: KS-Example			Type: first			
Object	**Slot**	**Op**	**Value**	**RULES**		
current-object	type-element	is	cast-in-place concrete column-footing	t	t	f
soil-characteristics	backfill	is	yes	t	f	i
excavate-column-footing				x	x	i
dispose-excavation-column-footing				x	x	i
pile-up-excavation-column-footing				x	i	i
borrow-material-column-footing				i	x	i
place-forms-column-footing				x	x	i
reinforce-column-footing				x	x	i
pour-concrete-column-footing				x	x	i
remove-forms-column-footing				x	x	i
KS-other-elements				i	i	x

- *Group element activities* of common characteristics in order to have a hierarchy of element activities similar to that of MASTERFORMAT. Thus, element activities are associated with particular physical design elements (such as a column or a beam) and aggregations of activities called *project activities* and *project activity groups.*

- *Determine amounts of work* for element activities. Geometric information for the quantity take off is inherited from design element frames in the central data store.

- *Select units of measure* for element activities. Crew productivities or material quantities may be expressed in different units (e.g. days instead of hours).

- *Determine material packages* for element activities based on design specifications.

- *Create project activities* that aggregate element activities and provide summary information on the underlying element activities.

- *Determine precedences* for project activities. In CONSTRUCTION PLANEX, scheduling is performed at the project activity level, reflecting the homogeneity of resource use and the small granularity of detail contained in the underlying element activities.

- *Compute lags* for project activities. Element activities of several project activities are structured into an element activity subnetwork. A simple critical path algorithm is used to determine relevant lags among project activities based on this subnetwoik.

- *Select technologies* for project activities. Technologies are chosen at a macro-scopic or project level since consistency in this regard will reduce costs. Structuring activities into a hierarchy facilitates this imposition of overall constraints on the construction plan.

- *Estimate durations* for project and element activities. This estimation process relies on a hierarchical decomposition similar to that described in (Hendrickson, 1987b).
- *Schedule* project activities using CPM, resource allocation and constraint satisfaction;
- *Estimate costs* by computing activity costs and project costs using unit costs and scheduling information.

3 System Issues

3.1. Control and Communication

The processes described in the previous section communicate with each other by means of *messages* and *data.* The blackboard serves as the communication medium between the processes.

Control. The *controller* is responsible for activating each process. The controller maintains the following static description about each process:

- *preconditions* for its execution, namely, the process(es) that must have successfully completed before the current process can be activated;
- *backtrack target,* namely, the process that is to be re-activated if the current process is unsuccessful; and
- *machine* on which the process runs.

When the preconditions of a process are satisfied, the controller causes that process to be executed. When the process terminates, it sends a message which the controller posts on the blackboard. The message signifies whether the process was successful in producing a feasible solution or unsuccessful. If the process was unsuccessful, the controller causes the system to backtrack at the indicated target process, initiating a new version of the design.

Communication. The *data manager works* in conceit with die controller and is responsible for supplying the input data to the processes and retrieving their output data. In the current implementation, data is communicated by means of files. Prior to initiating a process by the controller, the data manager transmits the input data file to the machine on which that process resides. When a process terminates, it leaves its output file on its own machine; when its termination message is received, the controller causes the data manager to retrieve the file from the processes' machine and store it on the machine on which the controller and data manager reside.

Control and communication are implemented on top of the DPSK (Distributed Problem Solving Kernel) system developed at CMU (Cardozo, 1987). DPSK provides an environment for distributed problem solving on multiple machines by programs written in several languages. DPSK provides utilities for sending messages and signals between processes running on different machines, generating and responding to events, and communicating between processes by means of a Shared Memory accessible to all the processes. DPSK was designed to facilitate the implementation of a variety of cooperative problem-solving architectures; the current IBDE implementation, with fixed precedence ordering between processes, is a relatively simple application of DPSK. However, because of the computational expense of communicating large volumes of data via DPSK's shared memory, IBDE uses that memory only for inter-process messages and for maintaining an index or directory of the data files.

3.2. Global Representation of the Building

The global representation of the building is hierarchically organized as a tree of related objects residing in the project database. Objects may represent very high-level abstractions, such as the entire building site, or very detailed information, such as individual groups of building elements (walls, beams, etc.). The hierarchy represents two kinds of relations:

• *part-of*, where the current object is a part or component of a higher-level parent object; or

• *is-cdternative*, where the current object is an alternative design solution of the parent object.

Each object is represented by a frame, consisting of a number of slots, each slot holding the value of one parameter or attribute of the object. The hierarchical organization among the object types is shown in Figure 4.

With the representation described, each process is aware of the contents of the project database relevant to it, but not of the segments relevant to the other processes. This organization has supported the concurrent development of the processes and provides complete data and process independence among the processes. The project database provides at any time a complete snapshot of the current state of the building design and construction planning process.

The representation currently implemented is the result of two sets of design decisions. First, it was decided that the global representation contain some, but not all, of the *conceptual* relations among the objects representing the building. For example, the STRUCTURAL FUNCTION and FRAME objects are only used internally by HI-RISE; its successor processes (SPEX and FOOTER) operate without referring to these objects. Conversely, the functional relations between the WALL GROUP, etc. objects and the PROJECT ACTIVITIES objects reside entirely within PLANEX. The decision to include more than the union of the processes' input and output requirements in the project database was made to facilitate the subsequent addition of critics. Related to this is the second design decision that the frame representing each object contain all of the object's attributes, rather than creating separate "input" and "output" frames for each object/process combination. As a result of this decision, processing is highly localized (e.g., SPEX never creates any frames), and die status of any object can be seen at any time by examining which slots currently contain a non-null value.

3.3. User Interfaces

A common user display interface is currently under development. The user interface will reside on the same machine as and interact with the controller and data manager. It will provide a uniform set of interface facilities for the following functions:

• graphical display of data at any level of the project database representation;

• graphical "animation" of the construction process,

• graphical "navigation" to select specific objects to display; and

• textual display of selected objects.

In addition, each process has its own interface for

• run-time interactions with the process, including the controller, to provide process-specific input, interactions and user decisions (e.g., select a HI-RISE alternative for further processing or override the controller's agenda); and

• run-time process display.

4 Closure

The experimental integrated environment described here has many features required for a new generation of computer aids in construction. Current computer aided design systems have a variety of algorithmic and graphic display processes making use of a central database. In the environment described, the central data store is extended to support knowledge based application processes. A flexible controller permits automation of the entire design process and permits modifications over time. At the same time, existing processes such as graphic facility representations can be maintained.

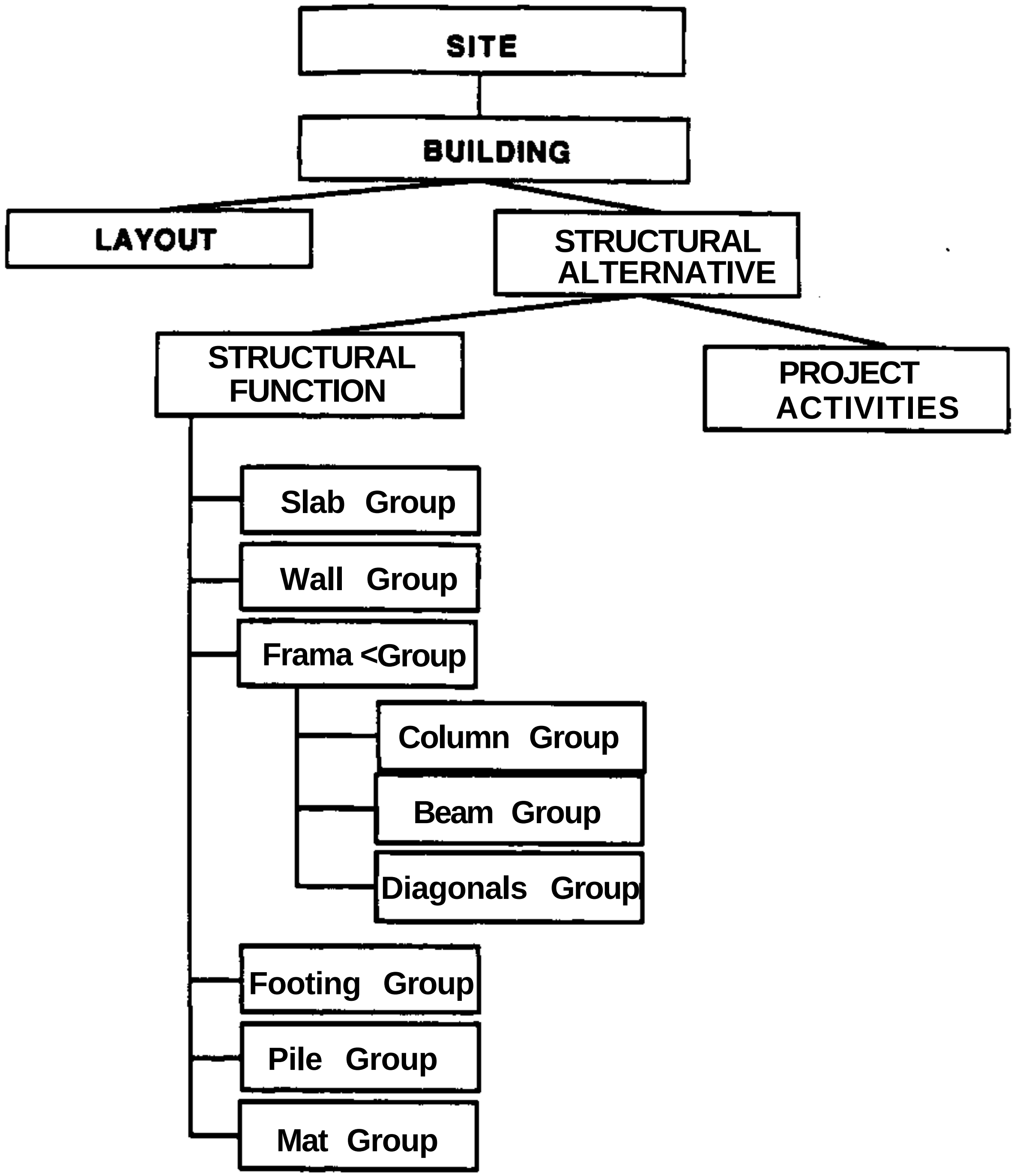


Figure 4: Illustration of the Object *Class* Hierarchy in the Data Store

The system described is currently (October 1987) in the last phase of implementation. Several complete designs have already been generated. Future plans are to add additional parallel processes to explore issues of horizontal integration and to increasingly focus attention to issues of feedback.

Acknowledgments

The Integrated Building Design Environment project has been sponsored by the Engineering Design Research Center (EDRC) at Carnegie Mellon University. Development of the individual processes has been supported by the EDRC and the National Science Foundation.

5 References

(Cardozo, 1987) Cardozo, Eleri, *DPSK: A Kernel for Distributed Problem Solving,* Technical Report EDRC-02-04-87, Carnegie Mellon University, Engineering Design Research Center, Pittsburgh, PA, 1987.

(Garrett, 1986) Garrett, Jr., J.H., and Fenves, S.J., *A Knowledge-Based Standards Processor for Structural Component Design,* Technical Report R-85-157, Carnegie Mellon University, Department of Civil Engineering, Pittsburgh, PA, September 1986.

(Hendrickson, 1987a) Hendrickson, C.T., Zozaya-Gorostiza, C.A., Rehak, D., Baracco-Miller, E.G. and Lim, P.S., "An Expert System for Construction Planning/' *ASCE Journal of Computing,* October, 1987.

(Hendrickson, 1987b) Hendrickson, C.T., Martinelli, D. and Rehak, D., "Hierarchical Rule-based Activity Duration Estimation," ASCE *Journal of Construction Engr. and Mgmt.,* 1987.

(Maher, 1984) Maher, M.L., and Fenves, S.J., *HI-RISE: An Expert System For The Preliminary Structural Design Of High Rise Buildings,* Technical Report R-85-146, Carnegie Mellon University, Department of Civil Engineering, November 1984.

(Maher, 1987) Maher, M.L. and P. Longinos, "Development of an Expert System Shell for Engineering Design," *IntL Journal of Applied Engineering Design,* Pergamon Press, 1987.

(Schmitt, 1987) Schmitt, G., *ARCHPLAN - An Architectural Planning Front End to Engineering Design Expert Systems,* Technical Report, Carnegie Mellon University, Engineering Design Research Center, September 1987.

(Zozaya, 1987) Zozaya-Gorostiza, G, C. Hendrickson, E. Baracco and P. Lim, *Decision Tables for Knowledge Representation and Acquisition,* Technical Report, Carnegie Mellon University, Department of Civil Engineering, Pittsburgh, PA, 1987.