An Integrated Design and Appraisal System

N. D'Cruz and M. Vuong[1]
Curtin University of Technology

The design of buildings involve specialists from different disciplines, each performing tasks specific to their purpose, using data that describes the building in a particular way. This paper describes a prototype of an integrated building design system which incorporates design of the building, thermal analysis, daylight appraisal and artificial lighting design. The integration is based on the EAGLE computer aided design system, a database generator to convert the building model to an object database with interfaces to the CHEETAH thermal analysis program and programs written specifically for daylighting and artificial lighting appraisal.

INTRODUCTION

There are many people involved in the design and construction of a building; the owner, the designers, the tenderers, the suppliers of the various components for the building and its services, the tradespeople that assemble these components and the people that maintain the building in fully functional and operating condition. These people embody a number of different disciplines such as architecture, the various engineering disciplines, the quantity surveyors, amongst others. Each of these professionals perform tasks specific to their disciplines, using data that describe the building in a particular way for their own purpose. There is thus the possibility of repetition of data entry and errors in translation, as data is accessed by the various members involved. An integrated building design system would facilitate transfer of data, saving time, money and errors, as data and information is entered once and held in a common data base and made available to all people involved in the project as required.

This paper describes a prototype of an integrated building design system which incorporates design of the building, thermal analysis, daylight appraisal and artificial lighting design. Unlike the AEDOT project (Brambley and Bailey 1991), which aims to develop energy design-support tools integrable into comprehensive building design environments, it was developed specifically to facilitate the use of analysis as part of the design process, by students of architecture in the University. The system is targeted at the design of small buildings, typically houses. Thus the integration is based on tools currently used, namely the EAGLE computer aided design system, the CHEETAH thermal analysis program (Delsante 1987) and programs written specifically for daylighting and artificial lighting appraisal.

THE AIMS OF AN INTEGRATED SYSTEM

The notion of using an integrated system (IS) for building design is that there is a central repository of data for a project that is accessible to various users of the data. Such a development is usually based around a computer aided design (CAD) system, for the generation of the three dimensional form of the building. It provides a most effective human interface, as visualisation is an important aspect here. The user receives from the computer pictorial responses of the building being analysed. These responses are used as both data verification aids between phases of work and design decision aids. It can also allow the designer to return to their design for reusing and modifying earlier ideas.

Allied with the CAD system are programs for the extraction and storing of the required geometric data in a neutral database format, that is amenable for use by those requiring the data. Interfaces that convert the neutral data to application specific requirements, the running of the applications and

1 Department of Architecture, GPO Box U1987, Perth, W. Australia 6001.
Tel. +61 9 351 7312, Fax. +61 9 351 2711,
Email: nev@puffin.curtin.edu.au

feedback to the designer, complete the integration circuit.

Thus the three main aims of IS development are (Latombe 1977):

a) *Design automation* - any sub process of the design process that can be efficiently represented by a computer program could be programmed to become a helpful resource for solving design problems.

b) *Man machine interaction* - several parts of the design process are difficult to program and are more efficiently performed by designers; man-machine interaction is aimed at blending the best qualities of the computer and the designers.

c) *Database management* - its two main aspects concern the representation of models and the access to large amounts of data.

Most IS incorporate techniques from these three topics and integrate resources into sets able to satisfy a variety of designers' needs within well-defined areas (architecture, quantity surveying, engineering, among others). These integrated sets, usually called applications, are characterised by an internal logic that determines all the control paths between the designers and the resources and among the resources themselves. Some IS are built around a simple application (special purpose systems), others can support different applications in various domains (general-purpose systems).

The central core of the development is the geometric building database. A database may be defined as a collection of interrelated data stored together without harmful or unnecessary redundancy, to serve multiple applications. The data are stored so that they are independent of programs that use the data, though structured to provide a foundation for future application development. A common and controlled approach is used in adding new data and in modifying and retrieving existing data within the database.

Although an integrated system includes a graphics database, all the project information is not necessarily incorporated in this data base. Much of the information used in the various design tasks is discipline specific, that is non-graphic and not required by any other discipline. Such data is usually stored separately, being accessible for the particular program for which the information is needed.

Such an integrated system brings the following benefits (Gott 1973):

a) The system can provide more comprehensive facilities.

b) The constant feedback of user requirements provides powerful control on the direction of the building development.

c) Application software may be developed as part of the system while also being made generally available. It thus avoids or at least minimises software duplication resulting in reduced cost and time.

d) Application software packages have had limited success, mainly because of re implementation costs and on going support costs. The use, through the interface, of applications software "where it lies" is a powerful alternative use of an IS.

There are difficulties in developing such systems for various reasons.

- The differences in the nature of information representation between applications have to be resolved.
- There remains little available standard data that can ease the integration path.
- The cost of development and maintenance of a fully integrated system.

DIFFICULTIES AND PROPOSED SOLUTIONS FOR THIS INTEGRATED SYSTEM

CAD data format

To communicate between various components of the IS, the data needs to be intelligible and useable by the various parties during the design process. Therefore the need of a standard data structure that can be used by the whole system to exchange information between the CAD system and different applications and between applications themselves is essential. There are many approaches that may be used by an IS.

One approach is to adhere with the format of data used by the existing CAD system as a common data structure for the IS. The form of data model used in most CAD systems is a passive numerical model, in which items are represented by a structure of numerical values which is considered pertinent to the process to be carried out. This simple form tends to be rather primitive, since every item to be processed must be capable of representation in this form. An object such as a building can be represented in this form by entities such as a line, arc, circle, face, solid, pipe or sphere. Each entity is represented in the CAD system as a set of vertices. Such information can greatly simplify certain operations such as calculating areas or volumes of an object. But the modelling of all items by this form often loses a great deal of information about special properties of particular items. For example, a building is formed by walls, floors, windows, doors, etc. They can be represented perhaps as solids or faces. A user looking through the data cannot distinguish which ones are walls, which ones are windows, overhangs, etc.

The second approach is to represent the data in an international standard data format such as IGES (Initial Graphics Exchange Specification), or to follow the STEP protocol (STEP 1991), to exchange data from dissimilar applications. IGES includes entities such as points, arcs and ruled surfaces and annotation features such as dimensions and notes. The IGES exchange process is divided into two stages Figure 1 (Meguid 1987).

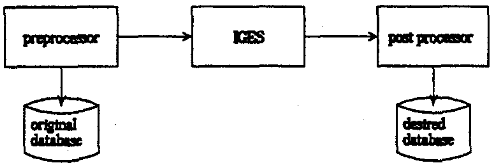

Figure 1 IGES data exchange process

Here

a) A pre processor maps the CAD database into an IGES file.

b) A post processor maps the IGES file into the desired databases.

Unfortunately the IGES data interchange is currently very confusing. There are many versions of IGES that exist: SET in France, VDAFS in Germany, PDES in US, and EEC-ESPRIT in the European Community. The data format used is still very primitive and it still does not convey information associated with the type of intended data objects such as walls, windows, etc.

The approach used in this work is to build a higher level of data structure on top of the CAD system primitives (line, string, solid, surface, etc), converting these in to object models. Here the EAGLE data structure is converted into a database of objects as walls, windows and other building elements that are pertinent to the design for energy efficiency. In an object model, each item is represented not by a structure of numerical values, but by an object (stored in the database as a record). For example, a wall may be represented by parameters such as location, length, width, thickness, and orientation. Each type of object is stored in a different data base. The retrieval of information from such a model is achieved by addressing a request to the relevant database for the required information. This object model, can be very useful in exchange of information between different applications and between CAD systems.

This approach is implemented by the creation of a database generator that reads data from a CAD numerical model and converts it into databases which have a well-defined database structure. The process of data converting can be illustrated as shown in figure 2.

Figure 2 Data conversion process

Costing issues

The development of a fully integrated system would be an enormous task, which would involve many people from all disciplines, and would cost millions of dollars because of the following reasons:

- An integrated system involving all the disciplines needs to be well thought out and planned. The system and the information that is transferred from discipline to discipline needs to be identified and analysed in detail.
- Software system design usually begins by breaking a large effort into manageable tasks. These tasks are assigned to individuals, who will implement their part through completion. Very large programming efforts usually have many individual tasks.
- A small program can generally be developed and made fully operational with less time and effort to eradicate problems, while with a larger program, the proportion of time and effort required increases almost exponentially.

The proposed solution to the above difficulties is to use as much as possible available software application packages and to limit the development of application software to the ones that cannot be met with an off the shelf purchased package. By doing so we can minimise the development cost, the re implementation cost and on going support cost. A purchased package is often well supported and highly reliable, and the cost is a lot less than for developing a package. Therefore, the only requirement for each purchased package is the need of development of an interface to link the package to the IS. The size and the cost of development of the interface are small compared to the size and cost of developing the entire package for the same purpose.

With regard to portability, it is suggested an application run "where it lies", which means that if the package is developed for the PC then let it run on a PC and do not re implement on a dedicated machine, unless required. The package can easily communicate with the IS via an inexpensive communication link, (such as RS-232 serial link) or a network.

With regard to the durability of software, if there is new software that is more reliable and is of better quality than the current one, then the new one can simply replace the old one. The only requirement is to modify the current interface (if possible), or to develop an interface for the new software.

Communication between sub systems

It is rare that a user will be able to solve all problems by means of a single sub system. There are many analyses (eg. costing, structural calculations, thermal analysis) involved in the design of the building. Thus the user must be able to exchange information readily between various sub systems. In theory it should be possible to chain together the many discrete application programs, which are normally designed to cope with the many special situations that occur in traditional building, and thus form an integrated computer aided building system. Such an endeavour tends to be difficult due to the anomalies that arise between separate application techniques as:

- The programs are generally limited to the solution of a particular problem.
- These programs require specific input data, eg temperature or cost.

Implementing several of these partial systems ultimately means that it is necessary to interpret the results and to prepare the data for transformation to the next program to solve another partial problem

For the connection of a number of program systems, which each partially solve a problem within the design process, three principal ways are possible (Eversheim, Fau and Rothenberg 1977):

Type 1: Between the separate program systems, adapting programs or interfaces are inserted to perform data transformation from one system into the required format of the other (Figure 3).

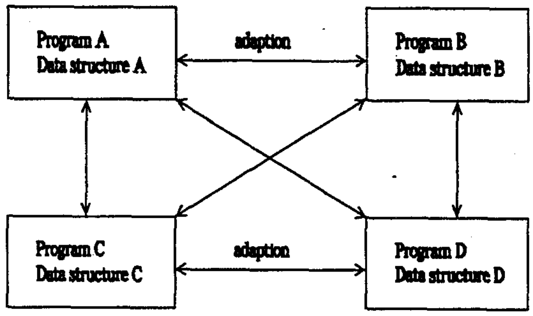

Figure 3 Direct connection of systems

Type 2: A central data structure is used as a standard interface when transforming data from one system to the other (Figure 4).

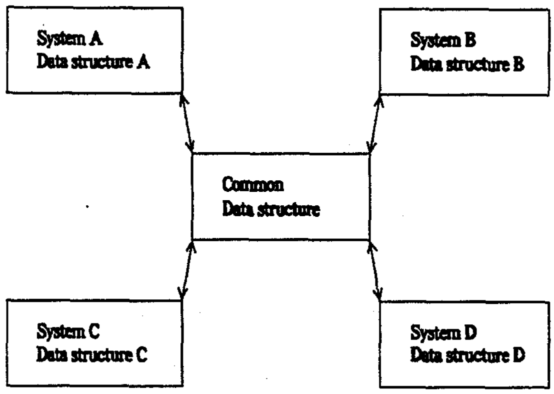

Figure 4 Connection of systems by common data structure

Type 3: All systems have access to a common data structure in a way that no data has to be transformed (Figure 5).

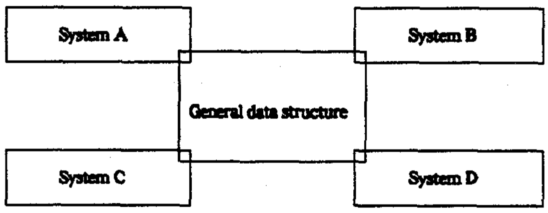

Figure 5 Connection of systems by a General data structure

The advantage of Type 1 of coupling is:

- Early implementation.
- Efficient program sequences with rapid processing.

Some disadvantages are:

- Inflexible linkage.
- An alteration or extension of the connected programs always demands an adjustment of the adapting program.

Types 2 and 3 connecting systems have in general a superior data structure and prove to be a more efficient concept, valid for a longer time.

The data structure chosen has to comply with the following requirements:

- Generally applicable structure, which allows an adaptation to already existing systems.
- Immediate retrieval of the data required in order to obtain time saving in processing.
- The possibility of adding geometric information which can be assigned to specified workpiece elements.

As Type 3 can only be implemented on a fully integrated system, Type 2 has been chosen for implementation in this work, to demonstrate the application and as it is relatively low cost.

The requirements to be met for a common data structure to be used as a link between sub systems are as follows:

- Logical and simple structure.
- Simple handling and easy learning.
- Simple diagnostics.
- Possibilities for extensions.

Application interfaces

There are many approaches to producing user algorithm interfaces (Gero and Julian 1977). Here a direct question, numeric answer with a menu selection interface is implemented. The reasons for their implementation are:

- The EAGLE CAD package (used in this IS) provides a friendly graphics environment for developing the interface. It has menu facilities that are quite simple and very easy to use in creating the menu. It also supports textual data input and output which is required by the question/answer interface. These characteristics of EAGLE greatly simplify the tasks of writing the applications interfaces.

- The nature of the input data is mostly rigid in which data needed to be entered is very precise, with no alternative. Thus menu selection and direct question answer interfaces, limit the possibility of errors in the input data for the applications programs.

SYSTEM CONCEPT

The implementation of an IS according to Leinemann and Schlectendahl (1977), consists of:

- a nucleus.
- a sub system for database management.
- a number of application sub systems

The nucleus itself consists, as indicated, of a CAD system, with systems for:

- menu management.
- database generation.
- error message handling.

An application sub system consists of:

- an interface.
- a set of error messages.
- a sub system database.

In an Integrated System huge amounts of data must be stored in a database, but as indicated, there is no need for the database itself to contain all sorts of data used by the system, but only the core data that is used by most programs. In simple cases a sequential file may be a sufficient database for such data, but a facility to combine such sequential files into hierarchical groups and to manipulate such groups is highly desirable. A first step therefore towards providing a data management system is the implementation of building databases.

THE IMPLEMENTATION OF THE IS

Given the above, the implementation of a prototype IS for this work, consists of:

- The nucleus.
- The database generator.
- Application sub systems for thermal, daylighting and artificial lighting evaluation.

It differs in concept from the COMBINE project (Augenbroe 1992), in that it is based on a commercial CAD package as the nucleus.

The Nucleus

The nucleus consists of the EAGLE CAD system. EAGLE is a 3-D computer aided design package that can be used in a variety of applications and has proved to be quite suitable for architectural applications. This CAD package has facilities to create, edit and display 3D models. Any 3D object can be modelled as a combination of lines, arcs, circles, solid objects, flat surfaces, with or without any annotation. The EAGLE system allows a sequence of its commands to be grouped and stored in a text file called a macro, wherein the whole sequence can be executed by a single command. EAGLE also provides a graphic environment with menu facilities which is easy to program and provides a very convenient user interface. Commands can be programmed in to a single menu or there can be a hierarchy of sub menus.

To facilitate the operation of the IS, design rules are enforced in the creation of building models. For example, components of a model have to be created using certain EAGLE primitives and assigned to specific layers.

The database generator

This is a sub system containing programs that decodes an EAGLE CAD model and generates the databases, one for each type of element of the EAGLE 3D model. It includes a data management sub system that handles all accesses to the database. It contains programs to retrieve the data from databases. However, most of the data retrieving and acquisition is embedded in the interface routines of the application sub systems.

A building can be considered as an object. Many authors consider that a relational database is not well suited for handling objects. Nevertheless the existence of commercially available relational databases is attractive enough to justify some attempts at extending them or at least integrating them to an object handling environment.

In the future, a proprietary database could hold the encoded data from a CAD model. Such databases usually include data management systems of their own which could facilitate enquiries to and from the database.

Application sub systems

Each application sub system is implemented independently as a separate module. By this approach any number of applications can be added into the system without interrupting and interfering with the operation of the other application sub systems.

Each application sub system contains the following components:

- An interface that receives data from input devices such as keyboard, mouse, etc.; extracts data from the system database or from a local database; and produces the output which is the required input in the format that is acceptable by the application program.
- The application program itself (it can be on the same system as the nucleus or can be on different computer system and its relevant database (local databases or data files), help messages system, and error handling system.

IMPLEMENTATION

A prototype of the integrated system described has been developed. It requires, as indicated, that a house to be analysed, is modelled using the EAGLE CAD software, running in a workstation environment, following certain conventions. These are that walls, overhangs and sunbreaks are built using the solid primitive on specific layers. Roofs, ceiling and floors are drawn as faces, again on defined layers. Openings for windows and doors are created by punching holes in walls. EAGLE macros and C programs access the CAD model and create database records of each element, in a predefined record structure.

No attempt has been made to modify the requirements of the CHEETAH thermal analysis program, which currently runs on a personal computer (PC). Rather, a front end has been written within EAGLE, which mimics the input requirements (with its default settings). An interface converts the geometric data that describes the building from the system database into the format required by CHEETAH. When all data input requirements are complete, an input file is transferred to a PC to run CHEETAH's calculation routine. Work is in progress to run the complete analysis on the workstation.

Similar processes take place for daylight and artificial lighting analysis, the difference being that here all the synthesis takes place on the workstation.

The integrated system developed thus aims to avoid or at least reduce the hassle of the lengthy process of data entering and preparation. The system is also expected to speed up the process of modification of the design and give the designer an opportunity to view the designed building visually via the 3d modeller and viewing facilities provided in the CAD package EAGLE. The availability of this integrated system will simplify the process of appraisal of the building in the three measures considered and hopefully encourage architects and building designers in undertaking these analyses.

The integrated system is intended to be independent of the CAD system to pave the way for further integration of other building design applications from other disciplines.

REFERENCES

Augenbroe, G. 1992. "COMBINE Project: Project Overview." In *Proceedings of the COMBINE Seminar: Computer Models for the Building Industry in Europe*, (Fraunhofer Institut fur Bauphysik, November 20).

Brambley, M.R. and M.L. Bailey. 1991. "The U.S. Department of Energy's Advanced Energy Design and Operation Technologies (AEDOT) Project." In *Proceedings of the Building Systems Automation - Integration 91*, (Madison, Wisconsin, USA, June 2-8).

Delsante, A.E. 1987. "CHEETAH A Thermal Design Tool for Small Buildings." Computer User Manual for Program CHEETAH. CSIRO Division of Building Research, Highett, Australia 3190.

Eversheim, W.; D. Pfau; and R. Rothenberg. 1977. "Application of suitable data structures for CAD in the machine tool industry." In *Cad systems*, J. J. Allan eds., North Holland, Amsterdam, 327-338.

Gero, J.S. and W.G. Julian. 1977. "High level non-graphical interaction in Computer-aided design." In *Cad systems*, J. J. Allan eds. North Holland Amsterdam, 241-259.

Gott, B. 1973. "The scope of Computer-aided Design." In *Computer Aided Design*, Vlietstra Wielinga eds. North Holland, Amsterdam, 1-16.

Latombe, J. C. 1977. "Artificial intelligence in computer-aided design: The tropic system." In *Cad systems*, J. J. Allan eds. North Holland, Amsterdam, 61-67.

Leinemann, K. and Schlechtendahl, E. G. 1977. "The Regent System for CAD." In *Cad systems*, J. J. Allan eds. North Holland, Amsterdam, 143-156.

Meguid, S.A. 1987. *Integrated computer-aided design of mechanical systems*. Elsevier Applied Systems, London.

STEP 1991 "Part 1: Overview and Fundamental Principles." Draft N14, ISO TC 184/SC4/WG6.