THE REALIZATION OF INTELLIGENT AID TO CAD OF ARCHITECTURAL DESIGN WITH THE OBJECT-ORIENTED METHOD

Guo Haoxu
College of Architecture & Civil Engineering, South China University of Technology, Guangzhou, China

Abstract. The object-oriented analysis and design has been the principal technology of software development since the 90s and intellectualization has been the direction of development for CAD software in the architectural design. An investigation is made on the application of the object-oriented technology to the realization of the intellectualization of the CAD for architectural design.

Key words: object-oriented; CAD for architectural design; intelligent technology; design expert system; object; visual-computing integration; parameter drive; polymorphism; inherit; correlated operation

1. Introduction

The emergence of the technology of computer-aided design (CAD) has quickened the pace of transformation of achievements in science and technology into productive forces and the CAD technology has set off a revolution in design in nearly all design areas. It is a crucial topic in professional CAD software development to combine the high intelligence, great benefit, concentrated knowledge and strong combinability of the CAD technology with the technological characteristics of specific trades.

Architectural design is after all an intelligent knowledge project. With the rapid development of the software and hardware technology in the information age, the conditions for further investigation into the CAD technology of architectural design in the direction of intellectualization are already available. In his practice of software development, the author has made exploratory considerations as to how to realize the intellectual aid in the CAD of architectural design.

2. The Intellectual Realization of CAD in Architectural Design

Artificial Intelligence is a discipline that deals with the simulation of manś intelligent activities. The AI technology is in general involved in CAD in two

forms. One is the design expert system, that is, an expert system developed for a certain design with the expert system tools available. The other form of involvement is the use of the AI programming technology in certain links of CAD. Used in man-machine interaction and the process of design, such technology can increase the degree of automation of CAD. In the area of architectural design, which form the computer should select to realize intellectualization depends on the characteristics of thinking concerning the architectural design.

The graph is the main mode of knowledge representation in architectural design, and the operation of graphs plays the key role in architectural design. The architectš design is a complicated process of the mutual permeation and mutual influence of logic thinking, thinking in images and insight thinking that involves both accurate formula derivation and correlation with fuzzy experience and subconsciousness. The clear overlapping and jerkiness in the process of thinking make it difficult to organize the management of the knowledge base. At the same time, as a class of art, architectural design does not have standards for evaluating the quality of a design. It is difficult to form a well-defined reasoning mechanism.

The design expert system is good at making a judgment as to "yes" and "no" as well as the regular substitution and linear generation of symbols and its application is often concentrated in the design part mainly of logic thinking. As the current expert system is built on the basis of logic reasoning, there is as yet no effective handling method for the synthesis of shapes, one of the crucial problems in design.

What the computer is versed in is the links with distinct logic thoughts such as the very much used details of architectural legislation, symbol description, scientific calculation, parameter memorization and even thinking and intuition in the course of architectural design. Hence it is the authorš view that the more practical and feasible road to the development for the CAD of intelligent architectural design for the time being is the selection of the second form of involvement for the investigation on the intelligent realization of the CAD software of architectural design.

3. Realizing Intelligent Architectural Design with the Object-Oriented Technology

Since the mid-90s, the use of the object-oriented technology has become the hot spot of software development. As a computer abstract level that transcends the process and data, orientation to the object regards the world as a kind of entities and objects that are correlated and capable of mutual communication. The object-oriented software development technology, with the modularity of its

package, the abstractness of its classes and examples, the reusability of software provided by the inheritance mechanism, the software expandability supported by the flexibility of overloading polymorphism and dynamic binding, has exhibited its superiority in optimizing the architecture of the software system and carrying out software maintenance and management. The knowledge-based system design in the area of AI is developing from the regulation-based mechanism to the mixed mechanism of object orientation and the use of matching regulations[Figure 1].

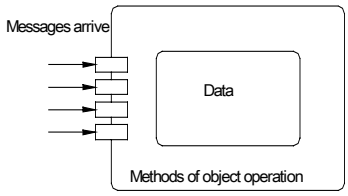

Figure 1. Structure of object

The flexible application of the object-oriented mechanism can often solve key technological problems in a convenient mode. In his CAD software development with the object-oriented method, the author has come to know that there exists a rational kernel for the realization of intelligent architectural design in the various object-oriented mechanisms.

3.1. THE OBJECT OF INTELLIGENT ARCHITECTURE

The object of architecture specialty formed by building components and architectural symbols are media by which an architect communicates and thinks. The problem the intellectualized architectural design CAD has to face above everything else is the realization of an intuitive description of the object of building by following the cognitive intuition of the architect.

The conventional architectural design CAD software is picture-oriented. Under this method, application is limited to working with the native entities and command set. Real world objects is forcibly represented by geometric entities and could seldom respond directly to commands. For instance, the operation of a wall is in effect the editing of two parallel lines. On the one hand, it is not intuitive. On the other hand, None of the physical properties involved under the symbols, the engineering design parameters and the relation between positions has a carrier of records. The software development thus described is conducive to simplifying the software architecture but limits the depth and breadth of its auxiliary design.

For the object-oriented program design, the gravity of program design is shifted from the process to the object. The object is the ADT (Abstract Data Type) mathematical modeling abstraction of specific things in the objective realistic world and of the type of abstract data with the particular attributes and

the relevant operating method packaged. When an object of building is described by an abstract mechanism, it becomes a first-order object in the database with the ability to respond directly to editing commands, displaying itself according to its inherent display characteristics and interoperating among themselves. Object-oriented technology enables applications to transform ordinary geometry lines, arcs, circles, and objects into discipline-specific modeling with intelligent design objects that contain behavior. For instance, a staircase that is described in the mode of an object is no longer a simple combination of parallel lines. Rather, it is a unified whole with such static characteristics as step length and defined as having such dynamic operating methods as 2D, 3D graphic displaying, grid point operation, movement, duplication, and modification [Figure 2]. The staircase responds to the operation of the architect as a solid. For the details of associated changes involved, self-renewal will be realized by the member functions within the objects. Following the intuition of human cognition, the architect understands and operates the staircase exactly as an object in the real world. This will no doubt greatly reduce the time spent completing routine design and drafting tasks and lay the foundation for further realizing the intellectualization of operation.

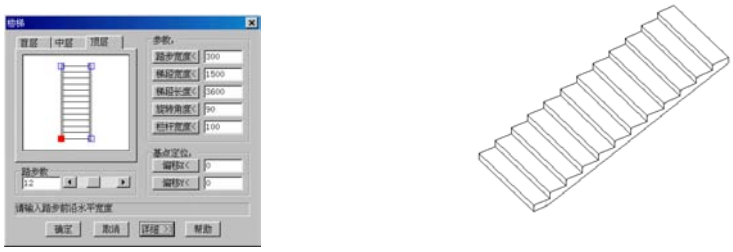

Figure 2. The dialog and the 3D graphic displaying of staircase

3.2. INTELLECTUALIZED DESIGN OPERATION

Intelligent design can be regarded as the application of intelligent engineering which is the technology of automation of decision-making in the field of design. The CAD of intellectualized architectural design should have definite scheme understanding and designing abilities. Under the precondition of recognizing the supreme design decision-making power of the architect, the CAD software should try to capture the architect's design intention, perform appropriate reasoning and make design decisions with a basis so as to raise the degree of automation of the software.

With the conventional structured procedural-based programming and software development techniques, the software invariably proceeds from the whole and expands from top to bottom. This top-down program design views a system as a layered set of subprograms. Elements of top-down design are necessary for all systems. However, when the problem becomes too large, the approach can fail because its complexity overwhelms the ability for a person to manage the hierarchy of subprograms. In addition, simple design changes in subprograms near the top of the hierarchy can require expensive and time-consuming changes in the algorithms for subprograms that are lower in the hierarchy. Hence, compared with the procedural-based programming technique, the object-oriented programming technique method is better suited to realizing intelligent aid of the architectural design process.

By employing the object-oriented technology, these operations with intellectual characteristics can be realized in multiple links and aspects as follows.

3.2.1. Visual-computing integration

The visual-computing integration technology refers to making computation by dynamically acquiring data from graphs. The design work of an architect asks for both tangible control and also a basis of numbers. The building component parameters, design modulus, design targets and architectural standards and details are all bases an architect relies on in his design. An architectš artistic thinking often starts from graphs, which are then quantized, to social products with construction feasibility. Hence the great necessity for carrying out numerical deliberation with the visual-computing integration technology.

Figure 3. The window width changes with the graphs

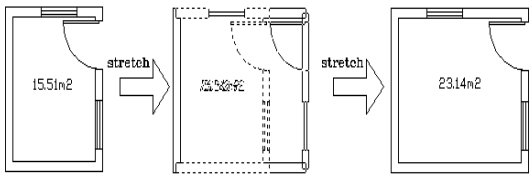

Figure 4. The habitable space is changing with the stretch operation

With object-oriented ‘s ADT mechanism, visual-computing integration can be realized conveniently during the description of object. If the member function for calculating design parameters is defined in the architectural object, then the software will take effect in runtime according to the variation of the

object's graphs to obtain calculation results other than those of the graphs so that the necessary design information will be automatically provided for the designer. For example, when a graph of a window is stretched, the window width will vary accordingly [Figure 3]. When the position of a wall is moved, the new habitable space and plot ratio will be renewed consequently [Figure 4]. The architect will obtain the key design parameters needed with ease and analyze and determine the numerical logic of a building.

Such a visual-computing integration technology is the direction of development for engineering design CAD. It can be widely used in the calculation of such indices as marking, the open space ratio, and the traffic area to support and raise the design quality with direct and accurate data.

3.2.2. Parameter drive

Parameter drive is a process contrary to visual-computing integration. It mainly refers to driving graphs to change by a change of graph parameters.

The design work of an architect is an ever-revising process. Architectural art is particular about beauty in order. In his design work, an architect is accustomed to determining dimensions with modulus and the standardized parameters of actual building components, too, affect the determination of design parameters. For this reason, it appears quite necessary to use the parameter drive technology in the CAD development of architectural design.

As messages can be transmitted between one object and another, a relation of constraint can be specified between the marking parameter and the object marked. It will be possible to realize parameter drive when the marking itself of an object also exists in the form of an object. Under parameter drive, when the marking value of the door or window is altered, the geometric figures will automatically change according to the new parameters. When the marking of the distance between axes is modified, the position of axes, too, will change[Figure 5].

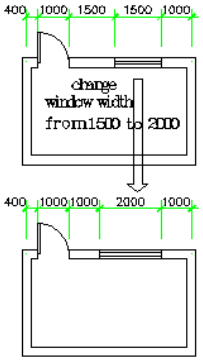

Figure 5. Driving graphs to change

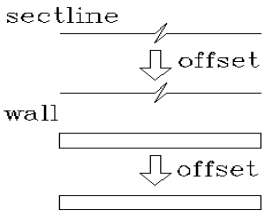

Figure 6. Using an offset command

by a change of window width to operate different classes

3.2.3. Intelligent recognition of the universal operating command

It is through the operating command that an architect communicates with the CAD software of architectural design. The more powerful the function of the software is the designer, as a rule will possess the more operating commands. Therefore, the importance of simplifying operating commands has all along been recognized in the development of architectural CAD software. Apart from continuing to use the shortcut key, menu selection, dialog box and tools bar in conventional architectural design CAD, we can also unify operating commands according to the overloading and polymorphism.

Under the overloading mechanism, some objects may include specific descriptions about certain operation. When the architect issues the operating command, the software will find out the architect's intention of operation according to the object that receives the message and search for the implementing function contained in the corresponding object to make a correct response. Hence, we can use the same **offset** command to create section line if the command is issued to the section line, while if it is the wall, then a offset wall will be created as a result [Figure6]. Such intelligent means of analysis and identification of commands reduce operating commands become more generally applicable and the absolute number of commands.

Figure 7. Use the **modify** to change different objects may lead to different behavior

The same message received by different objects may lead to totally different behavior, which is called object-oriented polymorphism. By using such polymorphism, different operations of similar behavior can be specified as capable of being performed with one and the same operating command so as to simplify commands. When the architect issues a command named **modify** to a wall, a door or an elevation label, the software will find out the architect's intention and make different response according to the object [Figure 7]. Another typical example is the command for operating the dimension marking.

There are many kinds of markings and large amounts of marking in architectural drawings. The dimensional marking under the action of polymorphism can be unified with only a few commands. If a certain marking command is issued to the axial , then the software will carry out axial marking while if it is the wall, then marking operation along the wall line will be implemented as a result. [Figure 8]

Polymorphism is encouraged by the object-oriented inheritance mechanism. The general operation for building objects of the same class can be unified with the inheritance mechanism. The more widely applicable the operating commands are, the lower the layers of class structure will they be defined in.

3.2.4. Correlated operation

A building formed by structural components, the enclosure, traffic structural parts and ornamental structural parts is linked and built into an enclosed object in accordance with definite architectural structure and constructional theory. There exists a definite logic relation between one structural part and another no matter whether these parts are treated in theory or from the reality of construction. The intelligent architectural CAD software has made it possible to realize correlated operation based on such a relation.

In an object-oriented software, the public members of an object may be accessed by clients outside the object. The control code directs an object to access its data by using one of its methods or operations. The process of directing the activities of objects is called message passing. A sender passes a message to a receiving object and asks the object to perform a task. When appropriate, the sender includes information that is used by receiver, this information is passed with the message as input data for the operation. After performing the task, the receiver may return information to the sender or pass messages to other objects requesting the execution of additional tasks. The characteristic of a message passing between one object and another is the theoretical basis of the correlated operation of architectural design CAD.

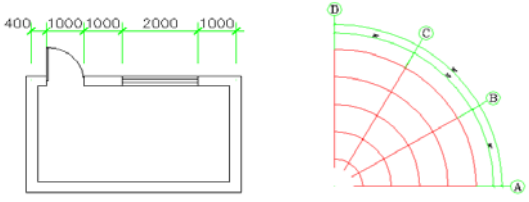

Figure 8. Use the **adim** command to unify the marking operation

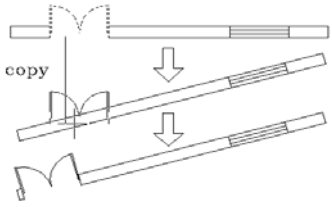

Figure 9. the correlated operation when copying a door

There invariably exists a definite design correlation between the gate, window and wall, marking data and marked entities, and columns and axial net. When a software developer has specified the structural logic of these members in the description of the architectural object of building and has them controlled in the operating command, the architectural CAD software will be able to bring its initiative into play in the design to realize correlated operation. For instance, when a door is roughly inserted in a wall with a 'copy' command, the wall and the door transmit information to each other. According to such attributes as the thickness and angle of direction of the wall and taking into account the analysis of the approximate positioning of the architect's operation, the mode the door opens and the modulus of the door buttress, the door will perform repositioning operation of its own accord, transforming the rough door inserting operation into accurate position. At the same time, the relevant wall, too, automatically renews itself as a wall with room for the door and the marked objects along the wall will also be revised owing to the insertion of the door [Figure 9]. In this case, the object is said to undergo a state change and new post conditions occur.

The correlated operation such as mutual control, mediation and coordination among the building objects makes the operation of the software an optional process.

Below is an excerpt concerning a program written in C++ language for obtaining the quasi-member function of the vector of the wall direction.

```
getRefVector(AcGeVector3d & refV){
  ads_name     entwall;
  refV = AcGeVector3d::kXAxis;
  AcDbObjectId objId;
  ……
  if((acdbGetObjectId(objId, entwall)) == Acad::eOk) {
     AcDbEntity *pEnt;
     if (acdbOpenAcDbEntity   (pEnt, objId,AcDb::ForRead,Adesk::kTrue)
     ==Acad::eOk) {
          if (pEnt->isKindOf (Walls::desc())) {
          Walls *pWall = Walls::cast (pEnt) ;
          refV = pWall->startPoint() - pWall->endPoint () ;
          refV.normalize () ;
          ……
          }
     }
```

```
        pEnt->close () ;
    }
}
```

4. The Organizational Mode of System in Conformity with the Characteristics of Architectural Design

The software organizational mode under the object-oriented mechanism coincides very well with the characteristics of the building trade. The development of the design software system according to the inheritance mechanism of an object is conducive to the sustainable development of software.

The objects of architecture are characterized by very few categories but many styles. The design classes involved in architecture are nothing more than about ten or twenty items such as the gate or door, the window, the wall, the staircase and balcony. But, if simply the pattern of gate is divided, then we have the horizontally opened gate, push-and-pull gate, and rolling gate (according to the mode it is opened); the glass gate, iron gate, wooden gate (according to the material the gate is made of), and the single-leaf gate, double-leaf gate, and triple-leaf gate (according to the number of leaves of a gate). Diversified individual description of building members and the constant innovative design of an architect asks for an open and scientific method for the architectural design software to develop the system of the members.

The software development techniques of structured procedural-based programming do not have a sufficiently effective mechanism to help the developer to organize the various design software in a unified manner. The commonality among the individual objects is overlooked more often than not and the structural members often have to be independently described, resulting in the repetition of large amounts of codes and the software architecture, too, appears redundant and inflexible.

Object-oriented languages allow a class to be derived by inheritance from other classes. This allows a designer to build new classes as refinements of other classes and to reuse the code that was previously developed. According to the object-oriented inheritance mechanism, a group of objects of identical properties should be generalized, with the common methods and data collected in one class. Thus, it is only necessary for a developer to create a new class in accordance with the variance programming of the parent class. New classes inherit the attributes of their parent classes and pass their own features to the subclasses. For instance, when creating an object of glass curtain wall, if we

directly inherit the principal code of the wall, it will only be necessary to make a special descriptive explanation about the graph, material and performance without having to repeat the complicated algorithm. And the multiple inheritance can also be used to inherit the attributes of multiple parent classes. With multiple inheritance, however, the class is derived from two or more base class. For instance, for the definition of a two-leaf glass gate, the three parent classes the gate, the double-leaf gate and glass gate can be inherited at once with supplementary explanation about the dimensions and shape.

By means of class and inheritance, the architectural design CAD software has very successfully solved the description of a diversity of identical building members. The software architecture is clear, the code concise and the cycle of development is shortened. When expanding the software, new contents can be added to the layered system without having to change the original architecture, enabling the software to always be in a dynamic state and apt to be brought into new knowledge and a varying mode.

5. Concluding Remarks

The AI technology that can be combined into CADs man-machine interaction and the design process is multi-faceted, such as the heuristic search technology, reasoning technology, constraint satisfying technology, and computer vision technology. It is the authors intention to seek a method for realizing intellectualization by employing the internal mechanism of the object-oriented technology while developing a software system of rational architecture, keeping in mind certain points applicable to both the said technology and the architectural design.

Most of the currently available architectural design CAD software are the results of secondary development on an AutoCAD platform. The AutoCAD R14 that realizes seamless link-up with the Windows95/NT operating system provides users with an object-oriented ObjectARX development environment. ObjectARX adopts the object-oriented C++ language and has the support of a complete function base. The program developed adopts the mode of communication of direct function call and runs in the same address space as the AutoCAD and with its high speed of run, it is a development environment of very wide application space. The greater part of the line of thought of the intelligent technology mentioned in this paper has been verified in the Object ARX-based architectural CAD software development the author participated in.

The development of the architectural design CAD is intended to use a computer to help with the design activity of an architect mainly armed with creative thinking to realize the automation of design as much as possible. It is

advisable for a developer to abide by the principle of humanism. He should fully understand the characteristics and method of work of an architect and provide intelligent design aid from the point of view of the architect so as to realize the perfect combination of frontier science and technology with the beautiful art of architecture.

References

Chen Qingzhang, Jiang Yuyan, 1994: *Method and Tool for System Development,* The Computer World (9)

Ma Qian, Gai Gang, Zhang Yong (editors), 1995: *Basics of the Object-Oriented Software Design*, Beijing Kehai Training Center, Beijing

Mai Zhongfan, Yan Jianxin, Liu Shuzhou, 1996: *Textbook on C++ Program Design Language*,(programming technique), Beijing University of Aeronautics and Astronautics Press, Beijing

Mai Zhongfan,Yan Jianxin, Liu Shuzhou, 1995: *Textbook on C++ Program Design Language*(programming basics), Beijing University of Aeronautics and Astronautics Press, Beijing

Pan Yunhe, 1997: The Intelligent CAD Method and Model, Science Press(3)

William Ford, William Topp, 1997: Data Structures with C++, Prentice-Hall International, Inc.

Xiao Renbin, Zhou Ji, Zhi Jianzhong, 1997: The Intelligent Design-Oriented Knowledge Describing Form, CADM (11)

Yuan Yaoming, From Visuality to Visual-Computing Interation--the Direction of Development for Engineering Design CAD, Engineering Design CAD and Automation, initial issue.