A 3D MODELING FOR DETAILED QUANTITY TAKE-OFF FOR BUILDING PROJECTS

Bing Tong

A Thesis

in

The Department

of

Building Civil & Environmental Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

December 2005

Your file Votre référence
ISBN: 0-494-14242-1
Our file Notre référence
ISBN: 0-494-14242-1

ABSTRACT

A 3D MODELING FOR DETAILED QUANTITY TAKE-OFF FOR BUILDING PROJECTS

Bing Tong

Quantity Take-off, the process of obtaining the dimensions and calculating the quantities from given plans and specifications, is perhaps the most basic and important phase in the detailed estimation of construction costs. The pricing strategies of cost estimation vary greatly in different situations. However, as the pricing changes, the quantity of the work involved in a particular project remains unchanged. A relatively accurate quantity take-off is thus a crucial element to successful bidding and project management.

Preparing the quantity take-off using conventional manual methods can be a time-consuming, tedious and costly process. However, since the 1980s, advances in computer technology have assisted the estimator by helping to measure, count, compute, and tabulate quantities, lengths, areas and volumes of objects found in the plans and specifications. Spreadsheets have assisted the estimator in calculating the quantities for items in the take-off and then extending the pricing for the items faster than what an estimator could do manually. Digitizers have greatly decreased the amount of time required for estimators to obtain the dimensions needed to perform a take-off. Nowadays, with the electronic documents becoming readily available and popular, AutoCAD-generated

drawings offer us an incredible amount of potential information to help estimate the construction costs.

This thesis presents a new methodology that can integrate AutoCAD with the quantity take-off during the detailed cost estimating process. Based on the designer's 2D AutoCAD drawings, and through interaction with a user, a customized AutoCAD module generates a 3D model of a project and exports the components' information into an external Access database which contains detailed quantity take-off information. The 3D model improves project visualization by allowing estimators to visualize the construction process as it would be actually built. Moreover, this customized module makes the quantity take-off much more efficient and effective.

ACKNOWLEDGEMENTS

I take this opportunity to express my deepest gratitude and sincere appreciation to my supervisor Dr. S. Alkass, for his support, valuable advice, and constructive guidance throughout all the stages of this study.

I would also like to express my deepest appreciation to my wife, for her love, patience, constant support and encouragement without which it would have been impossible to complete this endeavor.

I would also like to give special gratitude to my son who gives me so much joy all the time.

My thanks go to all those at the Department of Building, Civil and Environmental Engineering, the staff and faculty members who helped in various ways to carry out this research.

Last and certainly not least, I wish to express my deepest gratitude to my father, my mother and my brother who continue to provide me with unconditional love and support. To you, I dedicate this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 3

Chapter 4

Chapter 5

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 General

When the phrase 'Quantity Take-off' appears before your eyes, I am sure most of construction professionals think of a tedious, time-consuming process: reading the plans, details often spread over multiple construction drawings, performing numerous calculations to determine the quantities of supplies and work, checking the numbers and units over again to make sure they agree with each other. However, it is an inevitable and perhaps the most important phase in detailed construction cost estimation. Each estimator must face it, because a well accomplished quantity take-off usually is half of a successful construction bid, and crucial measure to control costs.

The traditional, manual method of quantity take-off is to read the 2D paper drawings, calculate the size of each component, and list the result in a pre-designed format. Spreadsheets are used to assist the estimator in formatting and calculating the items of the list, and manual quantity take-off is still the most common practice among most contractors nowadays. With the personal computer's prevalence over the past twenty years, we have seen a lot of efforts of apply computer applications to quantity take-off. With combination of scanners and OCR software, text and numbers can be converted into an electronic format. Although it is not 100% reliable, it is much faster than retyping entire documents. Another application is connecting a large digitizing tablet to the computer, laying the paper drawing on the tablet, and then using a light stick or highly accurate cursor/keypad to count and measure the items on the drawing. These applications are all improvements for paper drawing-based quantity take-off.

Today, many drawings are created using computer-aided design (CAD), which has freed designers and drafts people from the chains of cumbersome drawing tables, giving them more flexibility than ever. However, the full potential of CAD has been neglected, especially in the area of quantity take-off. Even though almost all design companies are currently using AutoCAD, few estimators are using CAD to help them with quantity take-offs. In addition to the graphical information created by AutoCAD, there is a lot of non-graphical information: areas, perimeters, lengths, etc. Transferring this information to an estimator's quantity take-off list directly from the AutoCAD drawings can help estimators with the tedious, time-consuming process. Therefore, this study will put the emphasis on how to use the full potential of AutoCAD in construction quantity take-off.

1.2 Research Objectives

The main objective of this research is to develop a methodology to make the quantity take-off more efficient, accurate and facilitative. The research sub-objectives are as follows:

1. Develop a methodology to establish 3D models of project components based on the original electronic drawings (AutoCAD dwg files).
2. Design a database to serve for the above mentioned 3D modeling. The

database will act as a supplier to provide necessary data to control the modeling process in AutoCAD, and it will also be a warehouse where the detailed quantity take-off data is to be saved.

3. Develop a methodology to retrieve and calculate the non-graphical data of each component in the established 3D model, and export the data to the external database.
4. Implement the suggested methodologies in a software system to assist users in the quantity take-off process.

1.3 Research Methodology

To achieve the foregoing objectives, the following steps were taken:

1.3.1 Literature Review

An in depth literature review was carried out in the areas of quantity take-off, 3D CAD modeling, database designing and computer integrated construction.

1.3.2 AutoCAD Construction Drawing Studying

The most widely used format of Electronic Construction Drawing, the "dwg" formatted AutoCAD file, was chosen to be the original document format for this research and AutoCAD was selected to be the platform of the research.

1.3.3 Development of the Model

This stage includes the selection of appropriate tools and the development of the computer modules which make up the final integrated model.

1.3.4 Application

Here, the proposed model was applied to an actual project and the results analyzed for further improvements.

1.4 Thesis Organization

Chapter 2 presents a literature review of the present methods for quantity take-off; IT application in construction; 3D modeling in AutoCAD and its advantages. In addition, it introduces a description of database applications in construction.

Chapter 3 explains and analyzes the architecture of the proposed model and its design process.

Chapter 4 presents the implementation of the proposed model as a prototype software system, including the introduction of its components.

Chapter 5 uses an example to verify the practicality and efficiency of the developed system.

Chapter 6 summarizes the results of this research, its contributions and recommendations for future work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

Quantity take-off is an essential aspect of detailed construction cost estimating; in fact, it is the fundamental phase of the whole estimating process. The success of a project often relies on an accurate and fast quantity take-off. This chapter presents a literature review of quantity take-off methods; computer applications in construction cost estimating and 3D modeling and database applications.

2.2 Quantity Take-off

Quantity take-off is the process of reading the plans and determining the quantities of work required to build the project (Peurifoy and Oberlender, 2002). Other commentators agree that perhaps the most basic as well as the most important aspect of the contractor's estimating and bid functions is the taking off of work quantities from the two-dimensional drawings representing a project (Adrian, 1982).The process of performing a take-off is an opportunity for the estimator to fully understand the work that must be estimated. Failure to review the take-off thoroughly could result in the missing of items that should be included in the bid price but will still incur costs during construction of the project, and may cause an unprofitable or unsuccessful contract. Therefore, quantity take-off is perhaps the most important phase in detailed construction estimating.

Quantity take-off can be performed manually or by using computers.

Conventionally, quantity take-off is based on paper drawings. An estimator first reviews blueprints, specifications and all pertinent documents thoroughly, and then determines the take-off by calculating, measuring and listing items in standard, pre-formatted sheets. This process includes many tedious clerical data collection tasks such as repeatedly reviewing of the plans; charting details which spread over multiple drawings and documents; accordingly, errors are common. Halpin (1985) thinks some of the most common errors are:

1. Arithmetic errors in addition, subtraction, and multiplication;
2. Transposition mistakes in copying or transferring figures, dimensions, or quantities;
3. Errors of omission, overlooking items called for or required to accomplish the work;
4. Poor reference, scaling drawings rather than using the dimensions indicated;
5. Unrealistic waste or loss factors.

In addition to the foregoing list, Foster (1995) has added two more common errors:

1. Using incorrect formulas;
2. Using the wrong conversion factor.

2.3 Computer Aided Quantity Take-off

Since quantity take-off is one of the most time-consuming activities in the estimating process, contractors and owners are more frequently using computer systems (Westney, 1997). In the 1980s PCs became more widely used because of their rapidly declining costs, and a new generation of purposely built cost estimating software emerged. Computer-based estimating programs are quite good at the data collection, computational and clerical aspects of estimating. They archive and retrieve large volumes of resources, cost and productivity information, perform calculations quickly and accurately, and present results in an organized, neat and consistent manner. All these virtues are of tremendous help in the high-pressure environment in which most construction estimators often find themselves (Sun and Howard, 2004). There are two main types of computer-aided estimating programs:

1. Spreadsheet-based programs.
2. Off-the-shelf estimating packages.

2.3.1 Spreadsheets

The first electronic spreadsheet application was developed by Dan Bricklin and Bob Frankston in 1979. Called VisiCalc, it was one of the earliest practical applications on personal computers. But by far the most successful and popular spreadsheet is Microsoft Excel, which provides contractors and project managers with a powerful and convenient analytical and presentation tool.

A spreadsheet is actually an electronic file that consists of hundreds of columns and thousands of rows. The intersection of a row and a column is called a cell, where textual and numerical data can be entered. The most powerful feature of a spreadsheet is that it allows users to enter formulas for desired calculations across the rows and columns. The user has considerable flexibility in formatting the input and output, and the powerful graphic package enables estimators to provide meaningful reports to management.

Spreadsheet programs are particularly effective in reducing repetitious arithmetic. They also enable the estimator to precisely duplicate the format and appearance of manual reports that were used before computers. This is an advantage to construction companies because most managers and owners are more comfortable with computer printouts that look exactly like the handwritten reports they have used for many years. However, for most companies spreadsheets are only an interim step between a paper and a computer estimating system. Most estimators still simply write down the items in a spreadsheet exactly as they did on paper. The spreadsheet is just used as an advanced calculator.

2.3.2 Cost Estimating Programs

Cost engineers need a tool that not only forecasts and tracks costs, but is able to considerably reduce the burden of data entry as well as provide instantly reliable financial reports (Jurkiewicz, 1999). One of the estimator's most important tools is the computer and associated software, and a used by a vast majority of estimators (Dysert and Elliott, 1999). Once a user learns to build an electronic estimate on the screen, he will be able to turn out more estimates, faster, with fewer errors, and also be able to analyze and manipulate numbers before finalizing the bid (Feldman, 1996). The use of computers with cost estimating can simplify and facilitate rapid consideration of many costing alternatives

(Duncan, 1996). According to Kitchens (1996), some of the activities that computers can be used for are:

- Performing quantity take-off using a digitizer;
- Performing extensions of units of labor, material, equipment, and other expenses to develop the estimate; and
- Developing spreadsheets for bid preparation.

With digitizers, contractors can electronically make measurements from blueprints and prepare detailed estimates within a very short time. Hence computer estimating programs can greatly reduce the time required to perform quantity take-off calculations and prepare bids. Also, accuracy is greatly increased, and professional looking estimates are easy to create. Moreover, computerized cost estimating software is an easy way to evaluate job profitability.

In the 1960s and 70s, only a few large companies used computerized cost estimating programs on mainframe computers; due to the high cost, their use was not widespread in the construction industry. With the rapid improvement of computers, construction companies of all sizes can now afford to purchase the computers needed to run estimating programs. These programs allow an estimator to quickly produce cost estimates by linking Bills of Quantity with cost information from standard and user-defined cost resource databases. There are usually four steps in using estimating programs (Sun and Howard, 2004):

- Entering bills of quantities: This can be achieved in a number of ways, such as:

    - Input data from a text file;
    - Input data using an optical scanner and optical character recognition software;
    - Use a digitizer to take dimensions from drawings;
    - Through a link with CAD software;
    - Manual input.
- Linking to cost libraries: This step links items of the bill to unit prices from standard or user-defined libraries.
- Analyzing estimates: At this stage, the estimators update the estimated costs by replacing some standard costs with accurate quotes from suppliers and subcontractors. They also carry out cost analyses by adjusting elements, such as waste, contingencies, profit margins, etc.
- Producing reports: After an estimate is finished, different types of reports – e.g. bidding documents – can be produced.

2.4 3D CAD modeling

Three-dimensional CAD models are three-dimensional computational representations of objects drawn in the *x*, *y*, and *z* axes and illustrated in isometric, perspective or axonometric views.

Before the development of computers, a variety of conventional media were used to represent buildings during the design process. In fact, ancient architects used text to abstractly describe the design process (Hewitt, 1985). 2D drawings were later introduced, but they still only expressed abstract visual thinking. Then, with the widespread use of physical models in the Renaissance, the form of space and architecture was given better precision. The evolution of design tools continued with the development of perspective drawing and other attempts at representing three dimensional objects using the two dimensional medium of paper, e.g., isometric and axonometric projections. More recently, digital technology has developed and matured rapidly. This has to an erosion of the traditional boundaries of computing. It is commonly accepted that it is worth employing computer technology rather than the conventional design media in the process of design and maintenance. The conventional approach involves the use of drawings and models to represent buildings. The type of models used in the design process can either be physical or digital – both can be used as means of solving complex problems that 2D drawings are unable to handle (Lin, 2001).Three-dimensional CAD models provide the following advantages over conventional drawings:

- ❖ An object can be drawn once and then can be viewed and plotted from any angle.
- ❖ A 3D CAD object holds mathematical information that can be used in engineering analysis and computer numerical control technology.
- ❖ A 3D CAD object can be shaded, rendered and assigned various materials and finishes for visualization.

These models can be generated by the use of various types of CAD software systems such as AutoCAD, ArchiCAD and many more.

3D CAD building models are useful across the entire spectrum of architecture, engineering and construction (AEC) practices. Architects and their clients can use 3D models to observe and evaluate building designs before construction begins. Engineers use 3D building models for energy, lighting, acoustics and fire simulations, which provide valuable insight into building usability and safety. Professionals in the construction industry utilize 3D models to estimate costs and to plan cost-effective construction sequences. Even for an existing building, it is often desirable to have a 3D model to facilitate analysis of the energy properties of the building, to predict how a potential fire might spread, to study potential changes made to the building, or to identify alternate uses of existing building spaces (Lewis and Sequin, 1998).

As one of the most widely used computer-aided design/drafting (CADD) products on the market of AEC, AutoCAD is an ideal tool for 3D modeling. With its open architecture, AutoCAD can be customized to fit the needs of almost any CAD user – whether that means reconfiguring menus and toolbars, adding specialized commands, or developing add-on programs that can be added to the main CAD functions. Many research works have been conducted in modeling the design procedure of all aspects of construction through the application of 3D CAD modeling. For example, Diez, et al. (2000) developed a system called Automod3 for the automatic design of modular building. The system focuses on the design phase; it adapts and links the methods of traditional architectural design with new construction methods. Specifically, it calculates the 3D modules and 2D panels needed to construct a block of flats; afterward, these will be pre-fabricated in a factory and then transported to the site for automatic assembly. Kunigahalli (1997) presented a 3D geometric modeling scheme that takes into account the reinforcement details for rectangular/square columns designed using one-way and two-way slab theories. Some other researchers have gone further by developing 4D CAD models, which include a temporal factor. 4D CAD models are being used to visualize the transformation of space over time, by combining 3D CAD models with a schedule for the construction activities.

Although 3D CAD systems are extremely powerful, they are not being used to their full capability in the AEC industries. Zahnan (2001) believes that the use of 3D CAD models empower the accuracy of designs and minimize the field changes caused by design errors and omission. Xu (2000) and Marir et al. (1998) think that specialists in the AEC industry have used today's CAD systems merely as automated drafting tools, limited to their own narrow areas of specification rather than extended to create objects that could later be utilized in applications. Kim et. al. (2000) considers that 3D CAD applications are used for effectively representing graphic information in the design phase. The basic quantity information obtained from 3D object models is then directly used as parameters for the functions established in creating specific quantity information tables. On the contrary, Bouvrie (2000) thinks that current 3D and intelligent 2D CAD systems do not provide an effective method of controlling the production of a design. Although data can be extracted from them, it is a laborious process that is also hard to manage.

Kunigahalli (1997), however, believes a 3D modeling scheme that facilitates efficient computer based storage and manipulation of geometric and topological information is required in order to develop powerful CAD systems. 3D CAD modeling is important and necessary in all phases of the project life. It helps in eliminating design errors and omissions before starting the physical implementation. Accordingly, it reduces costs. Furthermore; retrieving the required quantity information automatically from the 3D CAD model reduces the time consumed in quantity take-off.

2.5 Database Applications

During the quantity take-off process, a vast amount of data may be generated. The data must be stored, retrieved, and continually updated. This could be done using database software. Databases and database technology have had a major impact on the growing use of computers. It is fair to say that databases will play a critical role in almost all areas where computers are used (Elmasri and Navathe, 2001). A database is a collection of interrelated stored data that serves the needs of multiple users within one or more organizations; in other words, the database consists of interrelated collections of many different types of tables (Teorey, 1994); (Elmasri and Navathe, 2001); (Whitten and Bentley, 1998). Durvel and Schmidt (2002) describe a database as a collection of tables, queries, forms, and reports.

The structures of databases can vary widely. Al-Hussein (1999) states that different database structures are available, such as hierarchical and networked, which represent data and data interrelation using different predefined structures. Such database structures are difficult to modify. Instead, 'relational' databases are widely used; they allow data modeling using simple structures (tables) without having to predefine the data interrelations. Kibert and Hollister (1994) state that a relational database is characterized by its simplicity of data management, the independence of logical user views from the physical data storage structure, and by the availability of simple but powerful relational operators. These characteristics translate into a collection of tables that are composed of rows and columns.

The data stored in the relational database can be accessed and queried by using Structure Query Language (SQL). SQL is a language used to interrogate and process data in a relational database. It allows sophisticated data management processes to be performed on databases that are based upon highly orthogonal yet simple principles (Kibert and Hollister, 1994). SQL allows users to add, change, or delete information in tables and also formulate queries that allow them to categorize an structure data in different ways. With SQL, users have complete control of databases.

2.6 Summary

In this chapter, a literature review has been presented on quantity take-off methods, computer applications in construction cost estimating, 3D CAD modeling and database applications. The findings from this review are as follows:

- Preparing the quantity take-off with conventional procedures is a time-consuming and costly process. This process actually includes many tedious clerical and data collection tasks such as repeated reviewing of the plans, details which spread over multiple drawings and documents, etc.

Errors are thus frequent.

- Computer-based estimating programs are good at the data collection, computational and clerical aspects of estimating. They archive and retrieve large volumes of resources, cost and productivity information, perform calculations quickly and accurately, and present results in an organized, neat and consistent manner.
- 3D CAD systems are powerful and can be used as a means of solving complex problems that 2D drawings are unable to handle. However, they are not being used thoroughly in the AEC industries. Especially in quantity take-off. If they can be used to provide quantity take-off calculations, more accurate and detailed estimates may be attainable. Instead of manually measuring the drawings for quantities, the AutoCAD system could provide the required information.
- Databases and database technology have had a major impact on the growing use of computers. Databases will play a critical role in almost all areas where computers are used. They will also be crucial for the full implementation of CAD functions in the cost estimations of the design process.

The above findings are critical for the development of the proposed model which is described in the following chapters.

CHAPTER 3

PROPOSED MODEL

3.1 Introduction

This chapter presents the development of the proposed model, "A Detailed Quantity Take-off Model Based on AutoCAD". The model is designed to meet the needs of actual projects' quantity take-off requirements. Based on the forgoing findings in the related literature and practical working experience, the below development consists of three stages: analysis, design, and implementation.

3.2 Analysis

In this stage, the reasons for choosing AutoCAD as the platform of the proposed model are described. Some techniques that will be employed in the model development are also illustrated.

3.2.1 Introduction of AutoCAD

23 years ago, Autodesk first revolutionized the software industry with the AutoCAD® product, which introduced drafting to the PC. The widespread popularity of AutoCAD forged Autodesk's reputation of practical innovation that enhances productivity and profitability. AutoCAD also paved the way for Autodesk technology leadership in industries such as building, infrastructure, manufacturing, media and entertainment, and wireless data. Today, AutoCAD has become the world's most accessible and widely used design software tool. The Autodesk

community encompasses:

- 6 million users
- 1,700+ channel partners worldwide
- 2,500+ registered third-party developers
- 1,100+ training centers

Some competencies of AutoCAD include:

❖ **Conceptual Design**

More than two decades ago Autodesk introduced AutoCAD software, the design platform that enables conceptual design with powerful and quick drafting processes. Continuously innovated, the latest AutoCAD releases have added greater intelligence into the platform and a more intuitive interface, which means fewer repetitive tasks and more time available for the creative parts of the design process.

❖ **Visualization**

True-to-life, walkthrough visualizations provide outstanding previews of the ideas the clients want to sell and make real. Visualizations can be used for selling designs to clients, visually demonstrating how a machine can produce a product, or discussing the details of a future building site.

❖ **Advanced Modeling**

An advanced approach to modeling enables architects, designers, and engineers to work, think and innovate more freely. Ideas are virtually built into a model, and the advanced model's intelligence causes any changes to automatically ripple through all dependencies. This results in substantive reductions in errors and rework, as well as providing a more accurate sense of the viability of the modeled entity in the real world.

- **Collaboration**

  Efficient collaboration can take place among all phases of project information in digital format. This enables project data to be created, managed, and shared simultaneously across the enterprise, as well as among extended teams across geographic boundaries. The resulting project visibility also provides for more efficient workflow, more time to market and greater control and harmonization of project information and activity.

- **Lifecycle Management**

  Lifecycle management solutions make it possible to give everyone involved in a project or in an organization easy access to essential information over the life of a project and its developed assets. Keeping entire projects in digital form lays the groundwork for extracting maximum value from the project and its assets' lifecycles.

Perhaps one of the most important reasons for AutoCAD's popularity is its flexibility. With its open architecture, AutoCAD can be customized to fit the needs of almost any CAD user—whether that means reconfiguring menus and toolbars, adding specialized commands, or developing add-on programs that can run within AutoCAD. This is one of the main reasons why AutoCAD was chosen to be the platform of this research.

3.2.2 The Potential of Electronic Documents

One of the driving factors for this research is that electronic documents are becoming readily available and popular. The manual paper-based drafting is increasingly being replaced with precise electronic drawings. In today's AEC industry, with the rapid development of personal computers, most of the construction drawings are created by AutoCAD. AutoCAD drawings offer speed, exactitude, and a lot more features such as easiness of editing, modeling and filing etc. And AutoCAD is suitable and affordable, even for small and mid-sized contracting companies.

As for quantity take-off, the very specific and detailed information stored in AutoCAD drawing files offers considerable potential to the construction estimator. If this information can be used to provide quantity take-offs, more accurate and detailed estimates may be attainable. Instead of manually measuring the drawings for quantities, the AutoCAD system could provide the required information. Transferring information directly from the AutoCAD drawings would insure that the architect's intentions for material usage will be clearly relayed to the estimating team. Clear communication between the designer and the estimator will eliminate uncertainty, confusion and delays from the construction process.

3.2.3 Layers in AutoCAD

Layers (Figure 3.1) are the equivalent of the overlays used in paper-based drafting. They are the primary organizational tool in AutoCAD, and they are used to group information by function and to enforce line type, color, and other standards.

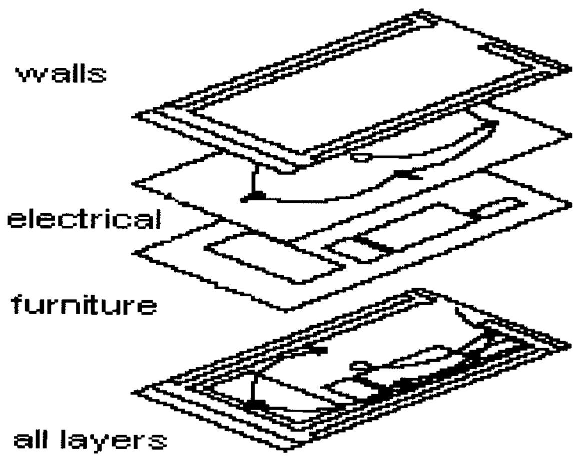


Figure 3.1 Layers in AutoCAD

By creating layers, the similar types of objects are assigned to the same layer, and can interact with other layers. For example, construction lines, text, dimensions, and blocks can be put on separate layers for easiness of editing. The user can then control:

- Whether objects on a layer are visible at any time
- Whether and how objects are plotted
- What colours are assigned to all objects on a layer
- What default linetype and lineweight is assigned to all objects on a layer
- Whether objects on a layer can be modified

When we begin a new drawing, AutoCAD creates a special layer named 0. By default, layer 0 is assigned color number 7 (white or black depending upon your background color), the CONTINUOUS linetype, a lineweight of Default (the default setting is .01 inch or .25 mm), and the NORMAL plot style. Layer 0 cannot be deleted or renamed.

Inspired by the definition and characteristics of layers in AutoCAD, we can also apply this concept to quantity take-off models. Corresponding to each of the real-world building's floors, layers can be created to represent their counterparts.

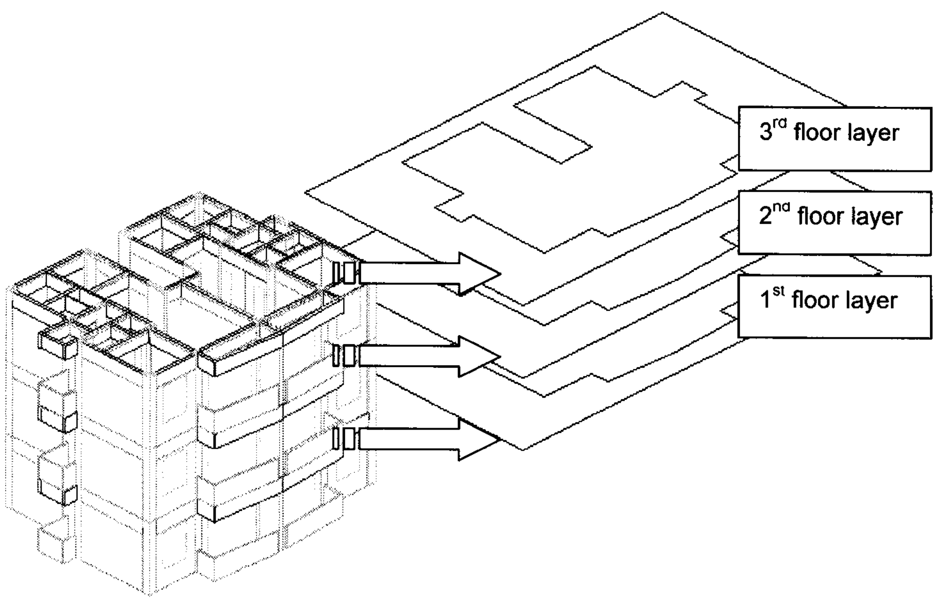


Figure 3.2 Floor Layers

As shown in Figure 3.2, the first, second and third floor layers in AutoCAD represent respectively the first, second and third floors of a real building. Components on each floor can then be drawn on their corresponding layers in AutoCAD.

3.2.4 Three Dimensional Modeling

3D AutoCAD modeling can be used in structural, lighting, acoustic, thermal and spatial analysis. This research focuses on the use of 3D AutoCAD modeling to support detailed quantity take-off. There is still a common misconception that AutoCAD is just for use in the post-design stages of work rather than having a role to play during construction management. This is simply because most designers in practice today were not formally trained to use computers as a tool, and they are therefore unfamiliar with its capabilities.

An important aspect of 3D CAD models is that they can be used as electronic geometric kernels to which non-graphic information can be added (Teicholz and Fischer, 1994); another is their ability to serve as the core model for other applications, such as database applications (Fischer and Froese, 1996). The linking of electronic information to 3D models is becoming both less difficult and more powerful. Labels can be attached to particular objects and a selection of an object can thus trigger established links to databases. In theory, all forms of relevant information may be directly accessible from the model itself. This information may include material costs, availability, location, description and so on. Versions or copies of the model can be used as models for structural analysis, for lighting and acoustic analysis, for accurate quantity estimates (Clayton et al., 1994), and as a shell to which even the most detailed construction elements, – a bolt and nut, if desired – can be added.

When performing 3D modeling, the architect and some of their consultants supply two-dimensional (2D) drawings of buildings in AutoCAD's dwg format. These drawings are then used as templates for 3D modeling. Wireframe models are the earliest and simplest type of 3D model. In a similar way to 2D drawings, a 3D model represents a building, or any other object, using straight and curved lines. These lines show the edges of the model. There is no concept of surface in the wireframe model. At first, wireframe models do not look like solid objects (Figure 3.3), however, when they are shaded in the AutoCAD environment, they better give the impression that they are solid objects (Figure 3.4).

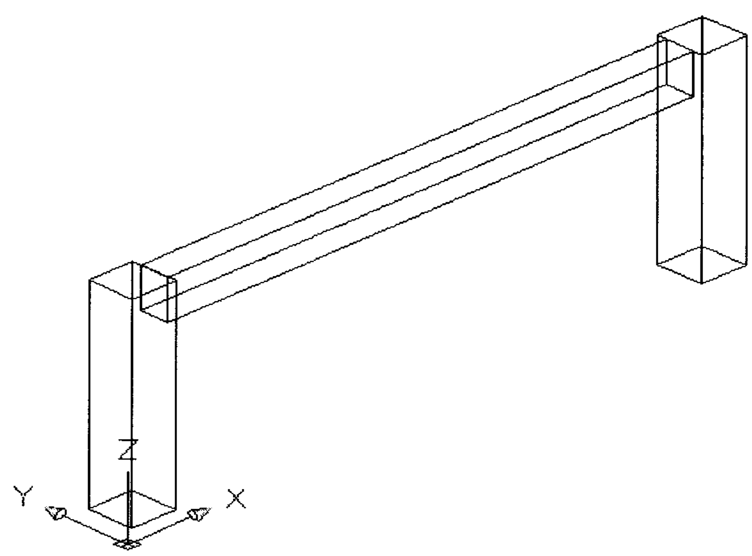


Figure 3.3 Wireframe Objcts

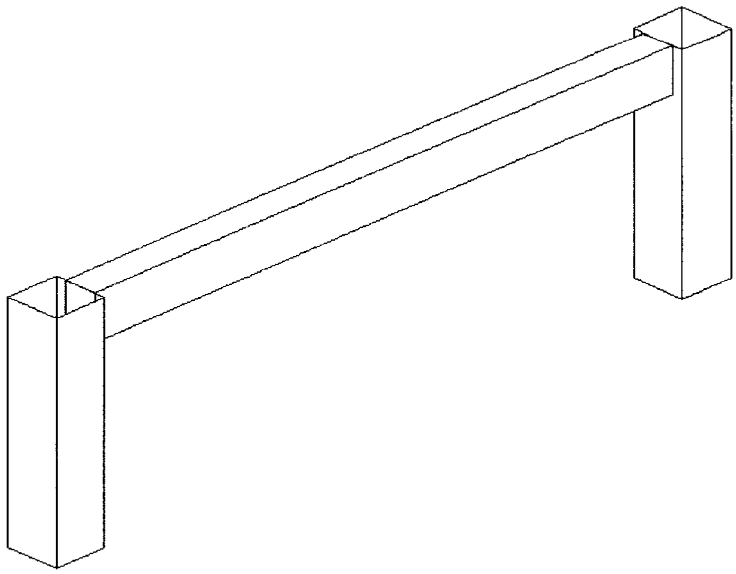


Figure 3.4 Shaded Wireframe Objects

Compared with other types of 3D modeling, such as surface and solid models, wireframe models have their own advantages:

- They are easy to create.
- They have a fast display speed, because the calculations involved are minimal.
- They provide accurate geometric information of objects being modeled.

These features make wireframe models the ideal type of 3D modeling for this research.

3.2.5 The Basic Element for 3D Modeling

Every piece of geometry you create in AutoCAD is a distinct AutoCAD object type. Most AutoCAD object types can take several forms. For example, polygons and rectangles are both polyline objects. Boxes, cones, and cylinders are all three-dimensional solid (3DSolid) objects. Considering the non-graphic information that needs to be extracted from drawings, normally we need length, width, height, area to calculate formworks, concrete volumes, floor areas, window and door areas, finishes, etc. The question is: which object type has the function of containing all the information of length, width, height, area and in the mean time, has the capability of 3D modeling? The answer is Lightweight polyline.

Lightweight polyline is a 2D line of adjustable width composed of line and arc segments. Lightweight polylines are optimized to display more quickly than standard polylines and save system resources. Table 3.1 displays some properties of the selected 2D or lightweight polyline.

Table 3.1 Properties of 2D Polyline

Property name (by category)	Description
Geometry	
Vertex	Specifies the current vertex of the lightweight polyline; enter the number of a vertex, or click arrows to cycle through all vertices
Vertex X Vertex Y	Specify the X,Y coordinates of the current vertex
Elevation	Specifies the elevation of the polyline (Z coordinate value of the current vertex)
Start segment width	Specifies the width at the start point of the polyline
End segment width	Specifies the width at the endpoint of the polyline
Global width	Specifies the width for the whole polyline
Area	Specifies the area of the lightweight polyline
Fit/Smooth	Specifies the type of line or surface curve fitting; valid settings are None, Quadratic, Cubic, and Curve Fit
Closed	Specifies closed or open polyline
Linetype generation	Determines whether linetype generation is on or off for the polyline; does not apply to polylines with tapered segments

With lightweight polyline, construction components can easily be represented in AutoCAD:

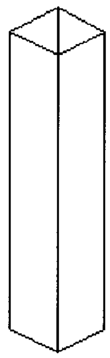


Figure 3.5 Column

A column can be represented using a closed lightweight polyline, with its thickness standing for the height of the column and the outline standing for the shape of the column (Figure 3.5).

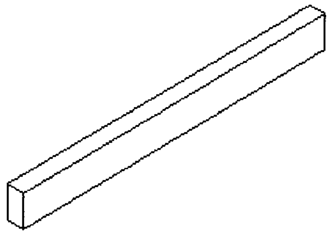


Figure 3.6 Beam

A beam can be represented using a lightweight polyline, with its thickness standing for the height of the beam, length standing for the length of the beam and global width stand for the width of the beam.

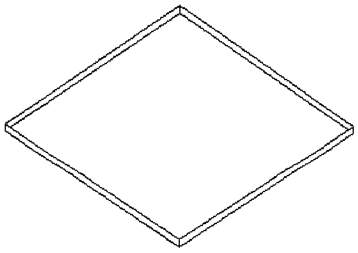
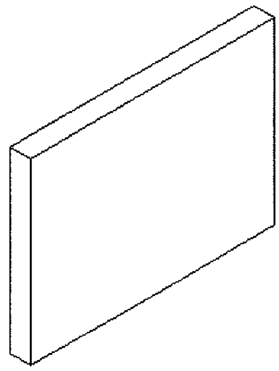


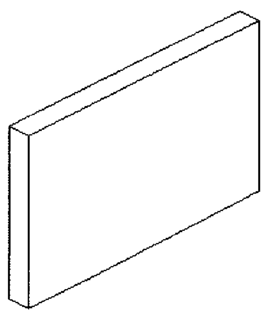
Figure 3.7 Slab

A slab can also be represented using a closed lightweight polyline, with its thickness standing for the thickness of the slab and the outline standing for the shape of the slab.



A wall can be represented using a lightweight polyline, with its thickness standing for the height of the wall, length standing for the length of the wall and global width stand for the width of the wall. (Figure 3.8)

Figure 3.8 Wall



A window or a door can be represented using a lightweight polyline as well, with its thickness standing for the height of the window or the door, length standing for the length of the window or the door and global width stand for the thickness of the window or the door.

Figure 3.9 Door or Window

Therefore, in this research, lightweight polyline is selected to be the basic element for 3D modeling.

3.2.6 Database and AutoCAD

3D models have let us see the benefits of using AutoCAD to closely represent the physical world around us. However, instead of merely producing 3D models, our focus will now shift to making full use of the 3D model. By linking the 3D model to an external database, the drawing could be made more intelligent by providing additional non-graphical information such as measurements, locations, and materials. Using an external database to link additional information to 3D model allows us to use the data as if it were a part of the drawing, while allowing the data to be manipulated outside AutoCAD.

According to the help files of AutoCAD, before we can access an external database from within AutoCAD, we must configure it using the Microsoft ODBC (Open Database Connectivity) and OLE DB programs. Using ODBC and OLE DB, AutoCAD can utilize data from other applications, regardless of their format or the particular database platform that is used. The configuration process involves creating a new data source that points to a specific collection of data and provides information about the drivers necessary to access it.

A data source may be either an individual table or a collection of tables stored in an environment, catalog, or schema. Environments, catalogs, and schemas are hierarchical database elements used by most database management systems to help organize data. They are in many ways analogous to a Windows-based directory structure: we can think of the environment as a folder that holds additional subdirectories of catalogs; catalogs in turn contain additional subdirectories of schemas; each schema is a collection of tables.

The configuration process varies slightly for different database systems. For example, server-based databases such as Oracle and Microsoft SQL Server™ require that you enter a valid user name and password and specify the network location where the database is located. File-based systems such as Microsoft

Access and dBase III do not require this information. Because of these variations, it is not possible to provide a generic configuration procedure that works for all databases. Individual procedures for configuring all the databases that AutoCAD supports are provided in the acad_dpg.hlp file in the AutoCAD Help system. The connectivity feature in AutoCAD supports the following external applications:

- Microsoft Access®
- dBase®
- Microsoft Excel®
- Oracle®
- Paradox®
- Microsoft Visual FoxPro®
- SQL Server

Once a user has configured the databases, he or she can access the data from AutoCAD, even if they do not have the database system that created the data installed on their system. AutoCAD can access data from the same applications as above.

After successfully configuring a database to be used with AutoCAD, a configuration file with the extension .udl is created that contains all the information AutoCAD needs to access the configured database. By default, .udl files are stored in the Data Links folder of AutoCAD. The user can specify a different location for the .udl files from the Options dialog box.

Several database management systems supported by AutoCAD have direct drivers available for OLE DB. If these drivers are used, users don't need to set up configuration files from within both ODBC and OLE DB; they only need a single OLE DB configuration file. Direct database drivers are available for the following database systems:

- Microsoft Access®
- Oracle®
- Microsoft SQL Server®

Microsoft Access is a component program of Microsoft Office, and it is very popular in companies around the world. Therefore, this research will use Microsoft Access to be the external database system.

3.2.7 Customization

Why customization? Though AutoCAD offers a wealth of commands right out of the box, its customization capabilities are what make it a truly effective and powerful program. A customized AutoCAD installation can enable us to better accomplish specific tasks by saving time, increasing productivity and improving quality. We can automate tedious tasks, eliminate extra keystrokes, and generally improve our work environment by empowering the software to take some of the drudgery out of our job.

The benefits can be immensely gratifying as the users begin to realize time savings, and the resulting cost savings, and free themselves up to become more creative and productive in other areas of their work. Customization can also make them better CAD operators, as they begin to think in terms of maximizing efficiency and pushing the software to do what they want it to do instead of modifying their work procedures to fit the software's capabilities. If someone has to perform a unique but laborious procedure for a specific project, it might not make sense to invest months of valuable time writing a program that will automate the process but never be used again. On the other hand, if he can reuse even part of an automated procedure over and over again, the time savings can be appreciable. This rule is suitable for the present research because the general types of building structures are limited. Even if someone develops a model for one type of building; the possibility of reusing it (or parts of it) is high. There are also intangible benefits that customization can provide such as the improvement of the users' technical skills.

3.2.8 The Selection of the Programming Environment

Before making a decision on the specific programming environment to be used for programming AutoCAD, it is necessary to understand the technology of ActiveX Automation.

ActiveX Automation is a technology developed by Microsoft and based on component object model (COM) architecture. It can be used to customize AutoCAD, share drawing data with other applications, and automate tasks. Through automation, AutoCAD exposes programmable objects described by the AutoCAD Object Model, these objects can be created, edited, and manipulated by other applications. Any application that can access the AutoCAD Object Model is an automation controller, and the most common tool used for manipulating another application using automation is Visual Basic for Applications (VBA). This form of Visual Basic is found as a component in many Microsoft Office applications. We can use these applications, or other Automation controllers, such as Visual Basic and Delphi, to drive AutoCAD.

The advantage of implementing an ActiveX interface for AutoCAD is twofold:

- Programmatic access to AutoCAD drawings is opened up to many more programming environments. Before ActiveX Automation, developers were limited to an AutoLISP or C++ interface.
- Sharing data with other Windows applications, such as Microsoft Excel and Microsoft Word, is made dramatically easier with ActiveX.

For example, a user might want to prompt for inputs, set preferences, make a selection set or retrieve drawing data. Users can also select their preferred controller, depending on the type of manipulation. Using ActiveX, a user can create and manipulate AutoCAD objects from any application that serves as an automation controller. Thus, automation enables macro programming across applications, a capability that does not exist in AutoLISP. With Automation the user can combine the features of many applications into a single application.

The displayed objects are called automation objects. Automation objects make various methods and properties available. Methods are functions that perform an action on an object. Properties are functions that set or return information about the state of an object. An object is the main building block of any ActiveX application. Each exposed object represents a precise part of AutoCAD. There are many different types of objects in the AutoCAD ActiveX interface. For example:

- Graphical objects such as lines, arcs, text, and dimensions are objects.
- Style settings such as linetypes and dimension styles are objects.
- Organizational structures such as layers, groups, and blocks are objects.
- The drawing displays such as view and viewport are objects.
- Even the drawing and the AutoCAD application are considered objects.

Virtually any type of application can access the displayed automation objects within AutoCAD. These applications can be stand-alone executables, dynamic linked library (DLL) files and macros within applications such as Microsoft Word or Microsoft Excel.

Another key concept we need to explore in Visual Basic programming is COM, which stands for Component Object Model. This should not be confused with older uses of the file extension *.COM (command), the COM (communications) port of a computer or the .com (commercial) extension to some Internet addresses. The COM which we are concerned with here is the technology that enables object-based programming. It is the foundation on which languages like Visual Basic are built. Objects are the fundamental building blocks in Visual Basic programming. The programs to be created will contain objects like command buttons and text boxes. The AutoCAD entities are also considered objects. To provide order, an object model serves as a road map in defining how the various objects are structured and manipulated through programming. AutoCAD has its own object model. COM plays a broader role, essentially defining how objects can "talk" to each other. A COM object implements the interfaces that support object interaction and allow cross-application programming.

In this research, AutoCAD 2004 is selected as the platform of the proposed model and the embedded VBA is employed as the primary programming environment. Microsoft VBA is an object-oriented programming environment designed to provide rich development capabilities similar to those of Visual Basic (VB). The main difference between VBA and VB is that VBA runs in the same process space as AutoCAD, providing a very fast and AutoCAD-intelligent programming environment.

Since its inception, AutoCAD has offered a variety of customization options. Some have enabled users to make simple configuration adjustments, and others have provided powerful development tools but have required sophisticated programming skills. Not until the arrival of Microsoft's Visual Basic language has a product offered customization opportunities for such a wide range of users in all types of disciplines. Shortly after its introduction in the early 1990s, Visual Basic began gaining wide popularity as a programming language. The object-based programming environment places a versatile set of programming tools in the user's hands without requiring programming expertise. As a stand-alone product, Visual Basic has been successfully used to develop a wide spectrum of software applications, ranging from simple games to large business applications used throughout the world. It is often referred to as a rapid application development (RAD) tool, due to the relative ease with which applications can be developed.

With the release of R14, Autodesk announced it would provide full support within AutoCAD for Visual Basic for Applications (VBA). Various VBA editions had already been included in Microsoft's Office products, and Autodesk was one of many independent software vendors (ISVs) to license the technology for inclusion with its products. Now, in addition to being able to develop VB programs to work with AutoCAD, developers can develop programs to run *inside* AutoCAD. In fact, VBA allows users to *develop* the programs within their AutoCAD session.

Without formal programming experience, users are likely to encounter a series of errors, bugs, and unexpected results when starting out. Although VBA will not necessarily avoid these problems, it should at least greatly decrease them and assist in the development of operational programs among non-professionals. Because it uses a graphical user interface (GUI), VBA helps us develop programs visually, instead of requiring the users to type numerous lines of code.

VBA also provides application integration with other VBA-enabled applications. This means that AutoCAD, using other application object libraries, can act as an automation controller for other applications such as Microsoft Word, Excel or Access. There are four advantages to implementing VBA for AutoCAD:

- The Visual Basic programming environment is easy to learn and use.
- VBA runs in-process with AutoCAD. This translates into fast program execution.
- Dialog construction is quick and effective. This allows developers to prototype applications and quickly receive feedback on designs.
- Projects can be standalone or imbedded in drawings. This choice allows developers great flexibility in the distribution of their applications.

Besides, the standalone development editions of Visual Basic must be purchased separately, whereas VBA is included at no extra cost with AutoCAD R14.01 and newer versions.

VBA sends messages to AutoCAD through the AutoCAD ActiveX Automation interface (Figure 3.10). AutoCAD VBA permits the VBA environment to run simultaneously with AutoCAD and provides programmatic control of AutoCAD through the ActiveX Automation interface. This coupling of AutoCAD, ActiveX Automation and VBA provides an intuitive and extremely powerful interface not only for manipulating AutoCAD objects, but for sending data to or retrieving data from other applications.

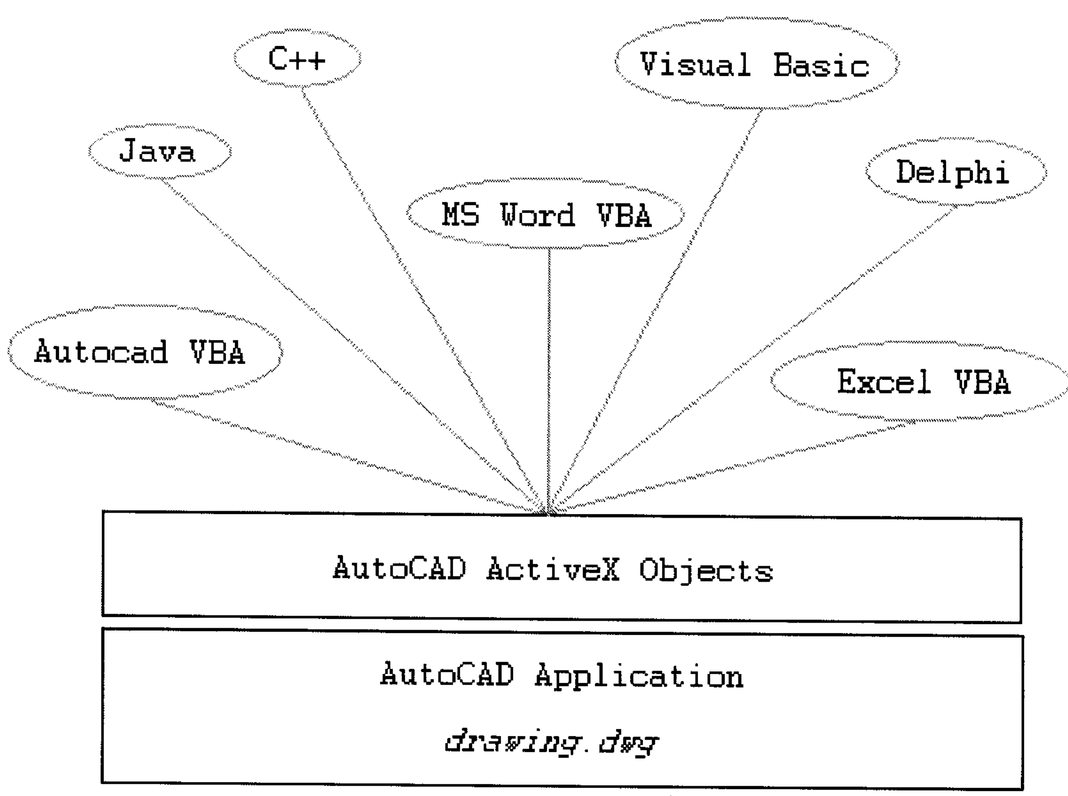


Figure 3.10 AutoCAD ActiveX Automation Interface

There are three fundamental elements that define ActiveX and VBA programming in AutoCAD. The first is AutoCAD itself, which has a rich set of objects that encapsulates AutoCAD entities, data, and commands. Because AutoCAD was designed as an open-architecture application with multiple levels of interface, familiarity with AutoCAD programmability is highly desirable in order to use VBA effectively. The second element is the AutoCAD ActiveX Automation interface, which establishes communication with AutoCAD objects. Programming in VBA requires a fundamental understanding of ActiveX Automation. A description of the AutoCAD ActiveX Automation interface can be found in the ActiveX and VBA Reference. Even an experienced VB programmer will find the AutoCAD ActiveX Automation interface invaluable for understanding and developing AutoCAD VBA applications. The third element is the VBA programming environment (IDE) which has its own set of objects, keywords, constants, and so forth that provides program flow, control, debugging, and execution.

The AutoCAD ActiveX/VBA interface represents several advantages over other AutoCAD API environments:

- Speed

  Running in-process with VBA, ActiveX applications are faster than either AutoLISP or ADS applications.

- Ease of Use

  The programming language and development environment are easy to use and come installed with AutoCAD.

- Windows Interoperability

  ActiveX and VBA are designed to be used with other Windows applications and provide an excellent path for communication of information across applications.

- Rapid Prototyping

  The rapid interface development of VBA provides the perfect environment for prototyping applications, even if those applications will eventually be developed in another language.

- Programmer Base

  There are millions of Visual Basic programmers around the world. AutoCAD ActiveX and VBA technology open up AutoCAD customization and application development to these programmers and many others who will learn Visual Basic in the future.

In a nutshell, VBA has injected more enthusiasm into the already popular sport of customizing AutoCAD. The AutoCAD version of VBA includes extensive application scripting features, full Visual Basic language syntax, a debugger, and a comprehensive Integrated Development Environment (IDE). For this research, VBA is an ideal development tool.

3.3 Design Stage

The proposed model will be applicable to real, concrete buildings. Based on the analysis stage, the architecture of the model is designed as shown in Figure 3.11. It consists of four major modules: the project database module, the project electronic drawing preprocessing module, the 3D AutoCAD modeling module and the data export module. Each module is an AutoCAD VBA project. And each

AutoCAD VBA project is a collection of code modules, class modules and forms that work together to perform a given function. Projects can be stored within an AutoCAD drawing, or as a separate file. Global projects are stored in separate files and are more versatile because they can work in, open, and close any AutoCAD drawing, but are not automatically loaded when a drawing is opened. The following section explains the design of each module and their functions.

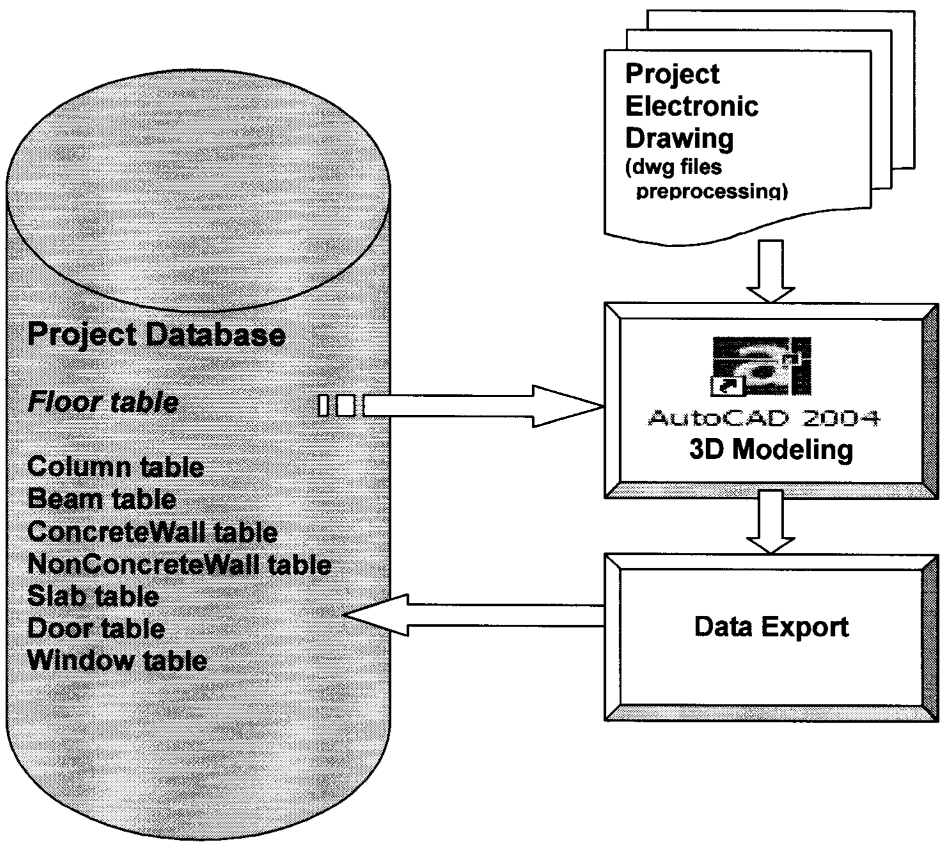


Figure 3.11 The Architecture of Proposed Model

3.3.1 Project Database Module

This module is designed to allow the user to establish a new external Access database for a new project. The database serves as a data supplier for 3D modeling and as a data warehouse for data exporting.

AutoCAD imposes no restrictions on the number of rows that can be linked to an object nor does it restrict the number of entities that can be linked to a single row. As a result, it is easy to choose a linking scheme that fits best with the specific needs of an application. There are four possible ways to establish links between AutoCAD objects and external database tables:

- Many-to-One (several objects are linked to a single row in the table)
- One-to-Many (each object has links to many rows in the table)
- Many-to-Many (multiple objects are linked to multiple rows in the table)
- One-to-One (each object is linked to a single unique row in the table)

In this research, quantity take-off requires that one component in the drawing has only one unique corresponding entry in the final list indicating quantity; therefore, the most desirable configuration is a one-to-one relationship between the drawing objects and the table. In this scenario, for every row in the table there is a graphical object and for every object there is a row in the table.

This database consists of one Floor table and the other seven tables, namely the Column table, Beam table, ConcreteWall table, NonConcreteWall table, Slab table, Door table and Window table. The following paragraphs briefly illustrate the structures and functions of these tables.

3.3.1.1 Floor table

The design view of the Floor table is shown in the below Figure:

Field Name	Data Type	Description
FloorName	Text	
Floor Height(m)	Number	
Numbers of Floor	Number	
Bottom Elevation(m)	Number	

Figure 3.12 Design of the Floor Table

The function of the Floor table is to allow the user to input the general information of a project in the beginning; the 3D modeling module is used to retrieve the data of a specific floor (its name, floor height, numbers of the standard floor and the bottom elevation of the floor) and make sure the objects in the drawing stand for correct components in the real world. Once this step is finished, the newly created floor layers of the project will appear in the layer menu. We can say that the Floor table is actually the base of 3D modeling.

3.3.1.2 Column table

Field Name	Data Type	Description
id	Text	
perimeter(mm)	Number	
cs area(m2)	Number	
height(mm)	Number	
subtractb(m2)	Number	
subtracts(m2)	Number	
subtractw(m2)	Number	
formwork area(m2)	Number	
v(m3)	Number	
Floor	Text	
elevation(mm)	Number	
componenttype	Text	
property1	Text	
property2	Text	

Figure 3.13 Design of the Column Table

As shown in Figure 3.13, the fields specified in the Column Table are as follows:

[id] -- This is the handle for an object in AutoCAD and represents a unique alphanumeric tag in a database. In AutoCAD, an object ID and a unique handle are the two ways of referencing an object. A handle is persistent in a drawing for the lifetime of the object. In general, we usually use a handle unless we plan to work with certain ObjectARX functions that require an object ID. This field can act as a primary key in the database and is efficient when used for queries.

[perimeter] – The perimeter of a column

[cs area] – The cross section area of a column

[height] – The height of a column. The default value is defined as the height of a specific floor. For example, if the column is on the second floor, the height of the column equals the height of the second floor, which is calculated by measuring the distance from the bottom elevation of the second floor to the bottom elevation of the third floor.

[subtractb] – This field is designed to list the amount of overlapped area between a column and the beams intersected with the column. As shown in Figure 3.14, there are two beams (Beam1 and Beam2) that intersect with Column1. So when we calculate the form work area of Column1, since the intersected area will not be contacted with the form, as a result, these parts should be subtracted from the amount of the side-area of Column1.

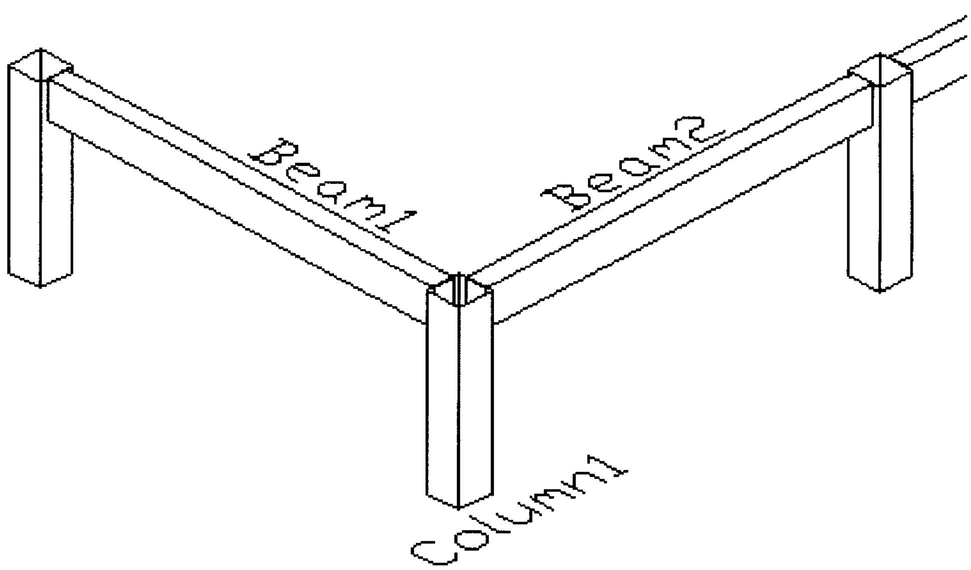


Figure 3.14 Space Relations Between Components

Similarly, the next field **[subtracts]** is designed to store the data of the overlapped area between the column and the intersected slabs; and **[subtractw]** is designed to store the data of the overlapped area between the column and the intersected concrete walls.

[formwork area] The formwork area of a column.

[v] The volume of a column.

[Floor] The floor name on which a column locates.

[elevation] The bottom elevation of a column.

[componenttype] The data in this field indicates the type of a component. In the Column table, it should be "column".

[property1] [property2] The two fields are reserved for user defined properties.

3.3.1.3 Beam table

Field Name	Data Type	Description
id	Text	
width(mm)	Number	
height(mm)	Number	
length(mm)	Number	
subtract(m2)	Number	
formwork area(m2)	Number	
v(m3)	Number	
Floor	Text	
elevation(mm)	Number	
componenttype	Text	

Figure 3.15 Design of the Beam Table

[id] The handle for an object in AutoCAD.

[width] The width of a beam's cross section.

[height] The height of a beam's cross section.

[length] The length of a beam.

[subtract] This field is designed to list the amount of overlapped area between a beam and the slab that is connected to the beam. The area should be deducted from the side area of the beam.

[formwork area] The formwork area of a beam.

[v] The volume of a beam.

[Floor] The name of the floor on which a beam is located.

[elevation] The bottom elevation of a beam.

[componenttype] The data in this field indicates the type of a component. In the Beam table, it should be "beam".

3.3.1.4 Slab table

Field Name	Data Type	Description
id	Text	
perimeter(mm)	Number	
thickness(mm)	Number	
area(m2)	Number	
formwork area(m2)	Number	
volume(m3)	Number	
Floor	Text	
elevation(mm)	Number	
componenttype	Text	
property1	Text	
property2	Text	

Figure 3.16 Design of the Slab Table

[id] The handle for an object in the AutoCAD.

[Perimeter] The perimeter of a specific slab.

[thickness] The thickness of a slab.

[area] The area of a slab.

[subtract] This field is designed to list the amount of overlapped area between a beam and the slab which is connected to the beam. The area should be deducted from the side area of the beam.

[formwork area] The formwork area of a slab.

[volume] The volume of a slab.

[Floor] The floor name on which a slab locates.

[elevation] The bottom elevation of a slab.

[componenttype] The type of a component. In the Slab table, it should be "slab".

[property1] [property2] User defined properties.

3.3.1.5 ConcreteWall table

Field Name	Data Type	Description
id	Text	
length(mm)	Number	
height(mm)	Number	
thickness(mm)	Number	
subtract door(m2)	Number	
subtract window(m2)	Number	
face area (m2)(single side)	Number	
formwork area(m2)	Number	
volume(m3)	Number	
Floor	Text	
elevation(mm)	Number	
componenttype	Text	
property1	Text	
property2	Text	

Figure 3.17 Design of the ConcreteWall Table

[id] The handle for an object in AutoCAD.

[length] The length of a specific concrete wall.

[height] The height of the wall.

[thickness] The thickness the wall.

[subtract door] The door area if there doors are present in the wall. When we calculate the form work of the concrete wall, the amount of the area should be deducted.

[subtract window] The window area of the wall.

[face area (single side)] The single side area of the wall.

[formwork area] The formwork area of the wall.

[volume] The volume of the wall.

[Floor] The name of the floor on which the wall is located.

[elevation] The bottom elevation of a concrete wall.

[componenttype] The type of a component.

[property1] [property2] User defined properties.

3.3.1.6 The NonConcreteWall table

Field Name	Data Type	Description
id	Text	
length(mm)	Number	
height(mm)	Number	
thickness(mm)	Number	
subtract door(m2)	Number	
subtract window(m2)	Number	
face area (m2)(single side)	Number	
formwork area(m2)	Number	
volume(m3)	Number	
Floor	Text	
elevation(mm)	Number	
componenttype	Text	
property1	Text	
property2	Text	

Figure 3.18 Design of the NonConcreteWall Table

This table is the same as the ConcreteWall table except the value of the field "componenttype" is "nonconcretewall"

3.3.1.7 Door table

Field Name	Data Type	Description
id	Text	
width(mm)	Number	
height(mm)	Number	
area(m2)	Number	
frame length(m)	Number	
Floor	Text	
elevation(mm)	Number	
componenttype	Text	
Property1	Text	
Property2	Text	

Figure 3.19 Design of the Door Table

[**id**] The handle for an object in the AutoCAD.

[**width**] The width of a door.

[**height**] The height of the door.

[**area**] The area of the door.

[**frame length**] The frame length of the door which can be used to calculate the wood work.

[**Floor**] The floor name on which the door is located.

[**elevation**] The bottom elevation of the door.

[**componenttype**] The type of component.

[**property1**] [**property2**] User defined properties.

3.3.1.8 Window table

Field Name	Data Type	Description
id	Text	
width(mm)	Number	
height(mm)	Number	
area(m2)	Number	
frame length(m)	Number	
Floor	Text	
elevation(mm)	Number	
componenttype	Text	
Property1	Text	
Property2	Text	

Figure 3.20 Design of the Window Table

[id] The handle for an object in the AutoCAD.

[width] The width of a window.

[height] The height of window.

[area] The area of a window.

[frame length] The frame length of a window. It is different from a door's frame length because it includes the length of the four sides of a window while for a door it doesn't count the lower side of the door.

[Floor] The floor name on which a window locates.

[elevation] The bottom elevation of a window.

[componenttype] The type of a component.

[property1] [property2] User defined properties.

3.3.2 AutoCAD Drawing Preprocessing Module

Different designers use AutoCAD differently. Some like to use standard lines, others prefer polylines. Besides, blocks (and nested blocks) and external references are often employed in the drawings. There are no standards for which symbols to use for particular types of object. This increases the complexity of the drawing. Therefore, to generate a 3D model from the original AutoCAD drawings may also be a time consuming process. A compound object comprises more than one AutoCAD object. For example, a block is a compound object. There could be many types of AutoCAD objects in a block. To edit these objects, first we need to break the compound object into its component objects, and then we can edit the component objects.

In the proposed model, an AutoCAD drawing preprocessing module is designed to bridge the original AutoCAD drawing and further 3D modeling. After the preprocessing, all the drawing objects are in their basic types such as lines, arcs, etc. The preprocessing is necessary because it reduces the drawing objects' complexity and makes the 3D modeling process smoother and easier.

3.3.3 AutoCAD 3D Modeling Module

This module is designed to allow users to generate various 3D components based on the preprocessed AutoCAD drawing (2D) and the data retrieved from the Floor table. Each type of component is generated through a sub-module named after the type. The main interface of the module is shown in Figure 3.21.

.

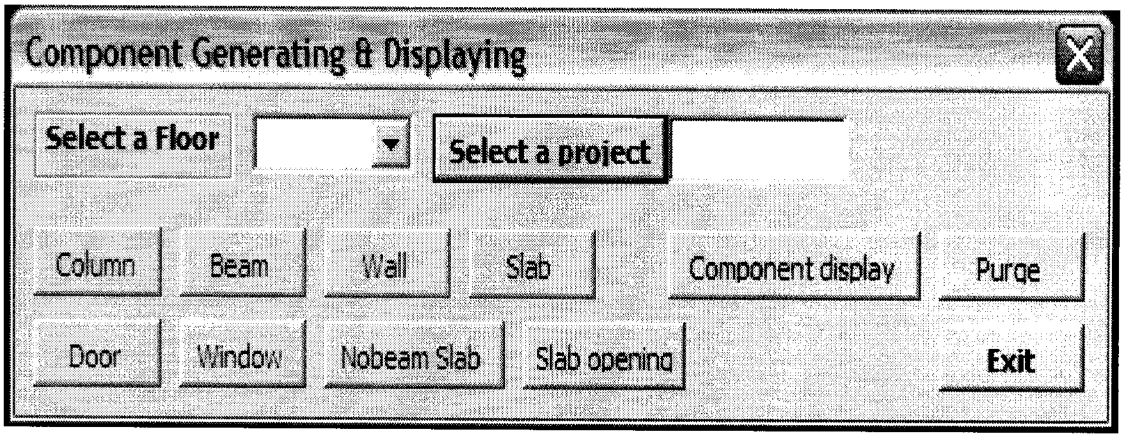


Figure 3.21 Main Interface of Component Generating

The 3D model is established floor by floor, as in ordinary construction procedures. The components on a particular floor include columns, walls, doors and windows of the floor and beams, slabs above the floor. Thus, if we are generating the 3D components of the second floor, the floor plan and the structure plan of the second floor should be used to generate columns, walls, doors and windows. However, the floor plan and the structure plan of the third floor should be used to generate beams and slabs. Figure 3.22 illustrates this.

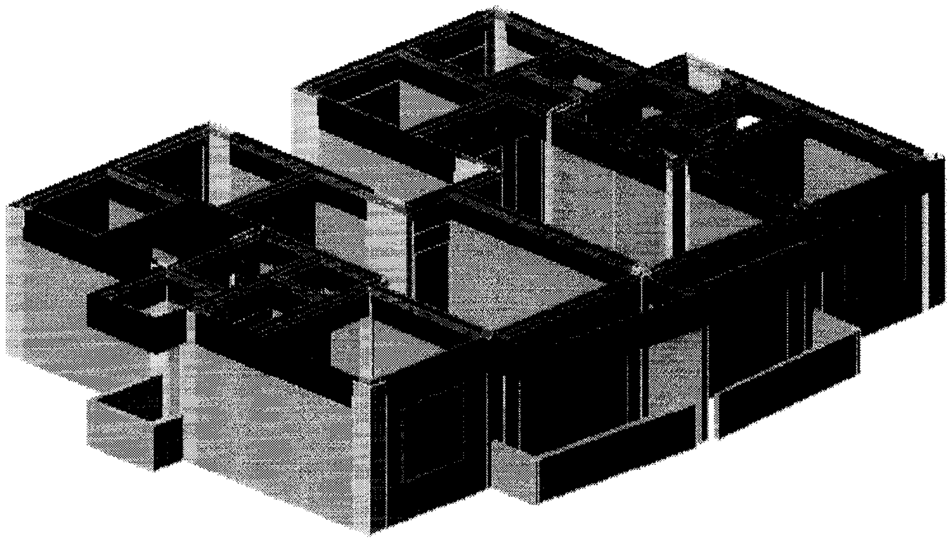


Figure 3.22 3D View of Floor Components

It should be mentioned that the 3D model does not just improve visualization. It is a smart model with a lot of non-graphical information attached to each component. The storage of non-graphical data inside the drawing file can have significant advantages. First, all of the graphical and non-graphical data are contained in a single file, which makes the data easier to manage. Second, since everything is stored in a single file, the objects and the non-graphical data associated with them are virtually inseparable.

3.3.4 Data Export Module

After the 3D model is generated, all of the graphical and non-graphical data are contained in the model. These data include the quantity take-off information we need to export to the external database. The Data Export Module is designed to accomplish this task.

Although AutoCAD combines its database connectivity feature in a single, comprehensive user interface called *dbConnect*, the *dbConnect* interface was designed to be very generic, providing basic data viewing and linking capabilities. With many real-world applications, it quickly falls short. To get AutoCAD and dbConnect to do the specific things you need, some level of customization is required (Mcfarlane, 2000).

From AutoCAD 2000, the database connectivity features of AutoCAD are based on Microsoft's OLE DB technology, which is one of the three primary technologies of the Microsoft Data Access Components (MDAC). MDAC provides data access that is independent of data stores, tools, and languages. It also offers a high-level, easy-to-use interface, and a low-level, high-performance interface with practically any data store available. We can use this flexibility to integrate diverse data stores and employ a wide variety of tools, applications, and platform services to create the right environment for various needs. These technologies provide the basic framework for general-purpose data access in Microsoft Windows operating systems.

Among the three primary technologies in MDAC, ActiveX Data Objects (ADO) is a high-level, easy-to-use interface with OLE DB. OLE DB is a low-level, high-performance interface to a variety of data stores. ADO and OLE DB both can work with relational (tabular) and non-relational (hierarchical or stream) data. Open Database Connectivity (ODBC) is another low-level, high-performance interface that is designed specifically for relational data stores. ADO provides a layer of abstraction between the client, middle-tier applications and the low-level OLE DB interfaces. ADO uses a small set of Automation objects to provide a simple and efficient interface to OLE DB. This interface makes ADO the perfect choice for developers in higher level languages – such as Visual Basic and VBScript – who want to access data without having to learn the intricacies of COM and OLE DB. In the Data Export Module, ADO is employed to connect and communicate with external databases. The connections between ADO, OLE DB, ODBC, and various data sources are shown in Figure 3.23.

The application in Figure 3.23 is the data consumer that uses ADO as the interface to access the data source. ADO itself makes use of OLE DB technology (contained in OLE DB providers), which enables ADO to connect to various data sources such as Microsoft Access, SQL Sever, etc. Commands and data are exchanged between the application (data consumer) and the database using ADO and the OLE DB provider, making ADO and its underlying OLE DB technology the link between an application and a data source.

An OLE DB provider for a certain data source provides an OLE DB interface with the data source. Using the OLE DB provider enables ADO to access a data source. A database connection through OLE DB usually offers the best performance for an ADO connection. OLE DB providers are available for the most important data sources, such as Microsoft Access, SQL Sever, and Oracle. From Figure 3.23, we can notice that using OLE DB providers has one less layer than using an ODBC driver. That makes the connection more direct and more efficient because of less used overhead. In this research, Microsoft Jet 4.0 is used to provide the direct interface between ADO and the external Microsoft Access database.

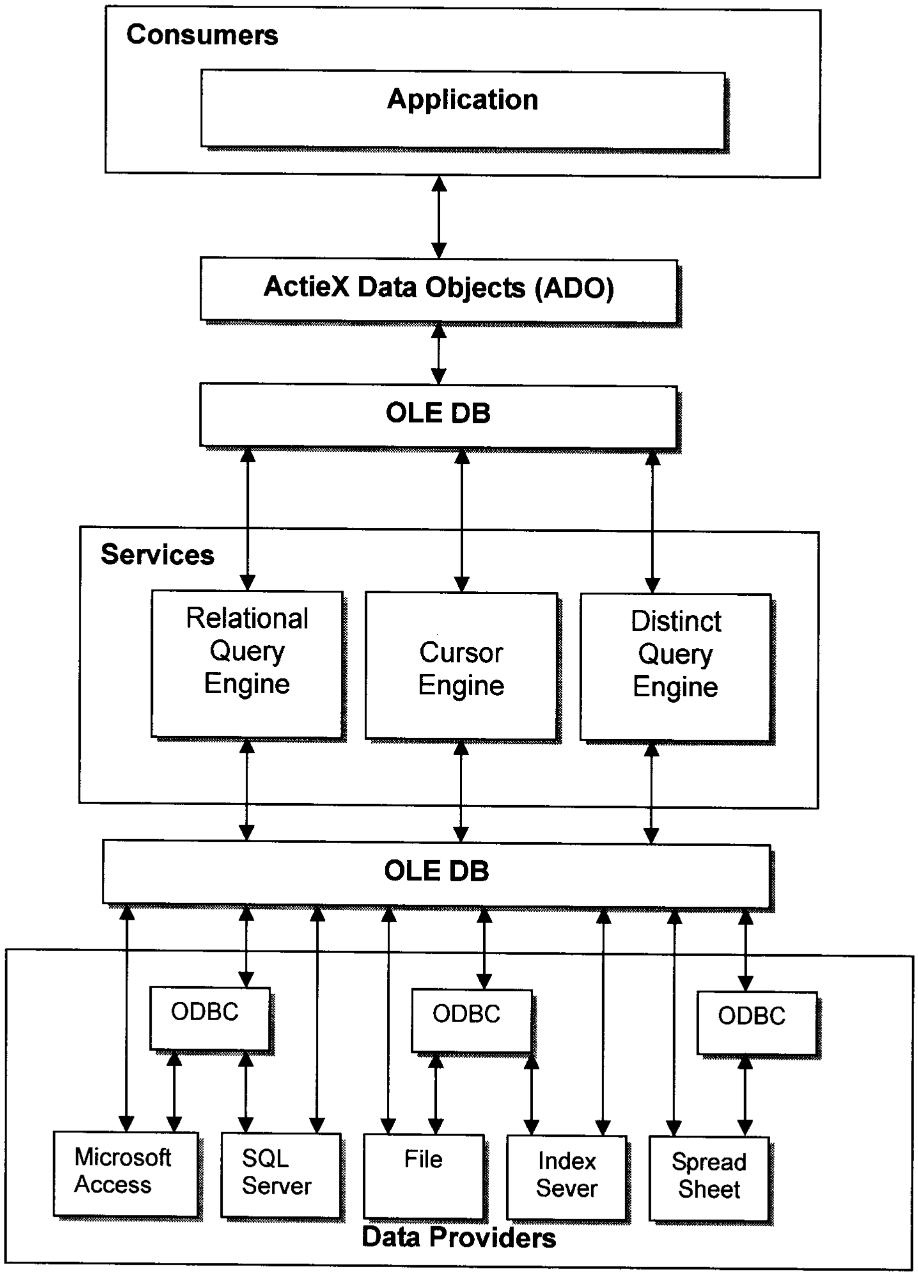


Figure 3.23 Connections between ADO, OLE DB, ODBC, Access

By running the Data Export Module, the tables created at the very beginning of the proposed model will be populated with corresponding component information. During this process, the space relationships among various components will be analyzed first, after which the final results are exported.

3.4 Summary

The proposed model of an AutoCAD-based quantity take-off has been presented in this chapter. The model consists of several major modules, namely: the Project database module, the Project electronic drawing preprocessing module, the 3D AutoCAD modeling module and the Data export module. AutoCAD 2004 was chosen as the platform of the proposed model, and the embedded VBA in AutoCAD is employed as the primary programming environment. ActiveX Data Objects (ADO) are used to connect and communicate with the external database. The following chapter will explain the implementation of the proposed model.

CHAPTER 4

IMPLEMENTATION OF THE PROPOSED MODEL

4.1 Introduction

The implementation stage of the proposed quantity take-off model is presented in this chapter. It is rooted in the previous stages of this research and is the final stage of the model development. As the model is implemented with the AutoCAD embedded VBA, the modules composed in the model are Global AutoCAD VBA projects stored in separate files with the extension "dvb" (Drawing Visual Basic). Since VBA projects are saved in a separate file, they can be used throughout a session.

4.2 System Description

The Quantity Take-off Model is developed to facilitate the estimators' retrieval of detailed quantity information from the AutoCAD generated construction drawings (dwg files). All components of the system are integrated using Visual Basic and ActiveX technology.

- System platform: AutoCAD 2004
- Types of applicable structures: Concrete buildings.
- Programming environment: AutoCAD 2004 built-in VBA.
- Database management system: Microsoft Office Access 2003.
- Operating system: Microsoft Windows XP.
- Unit system of measurement: Metric unit system.

The following paragraphs illustrate the implementation of each component of the system.

4.3 The Customized Menu

Since the system is run in AutoCAD 2004, the customized menu group designed for the use of quantity take-off should be embedded into AutoCAD 2004. Through a customized AutoLisp command "QT" (Figure 4.1), the customized menu named "Quantity" will be inserted into the original AutoCAD menu group (Figures 4.2, 4.3).

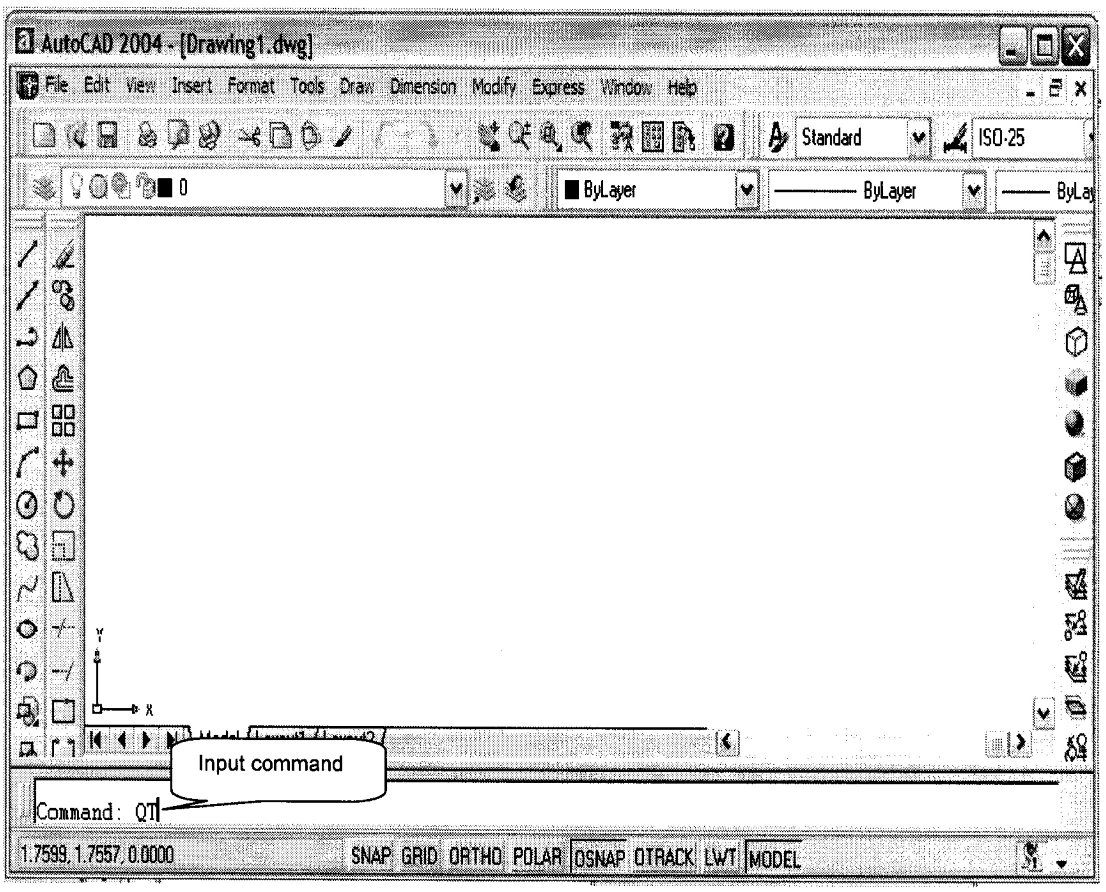


Figure 4.1 Input AutoLisp command "QT"

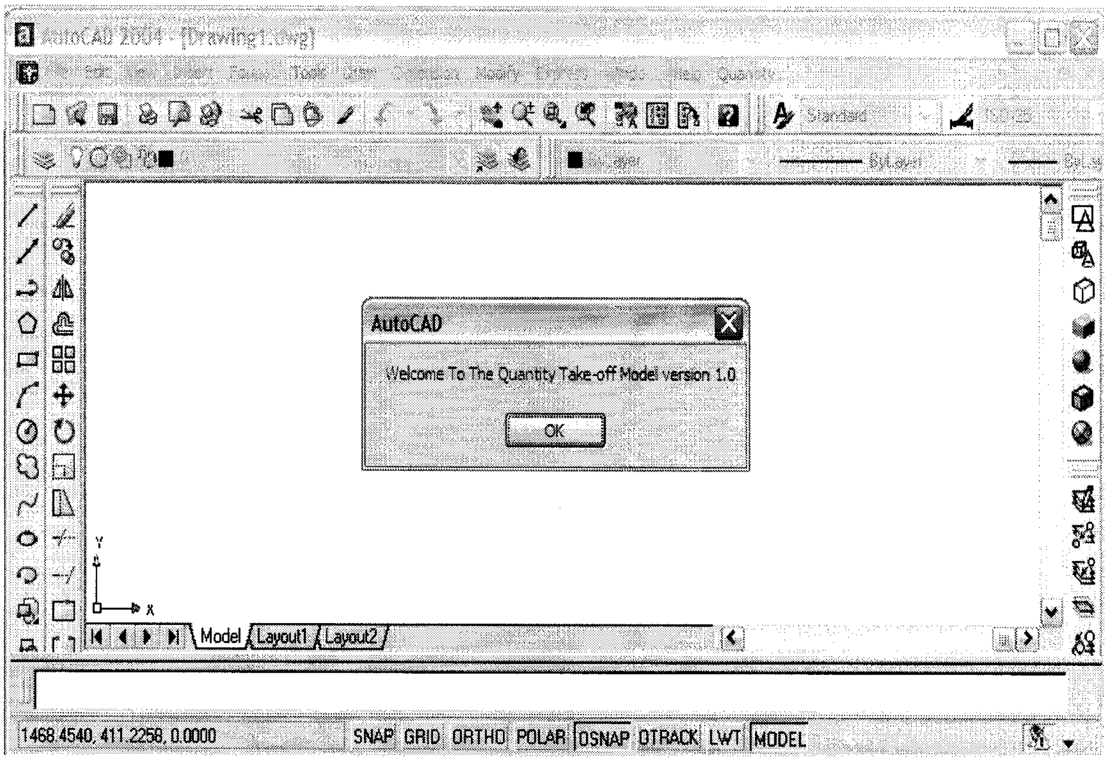


Figure 4.2 System Loading

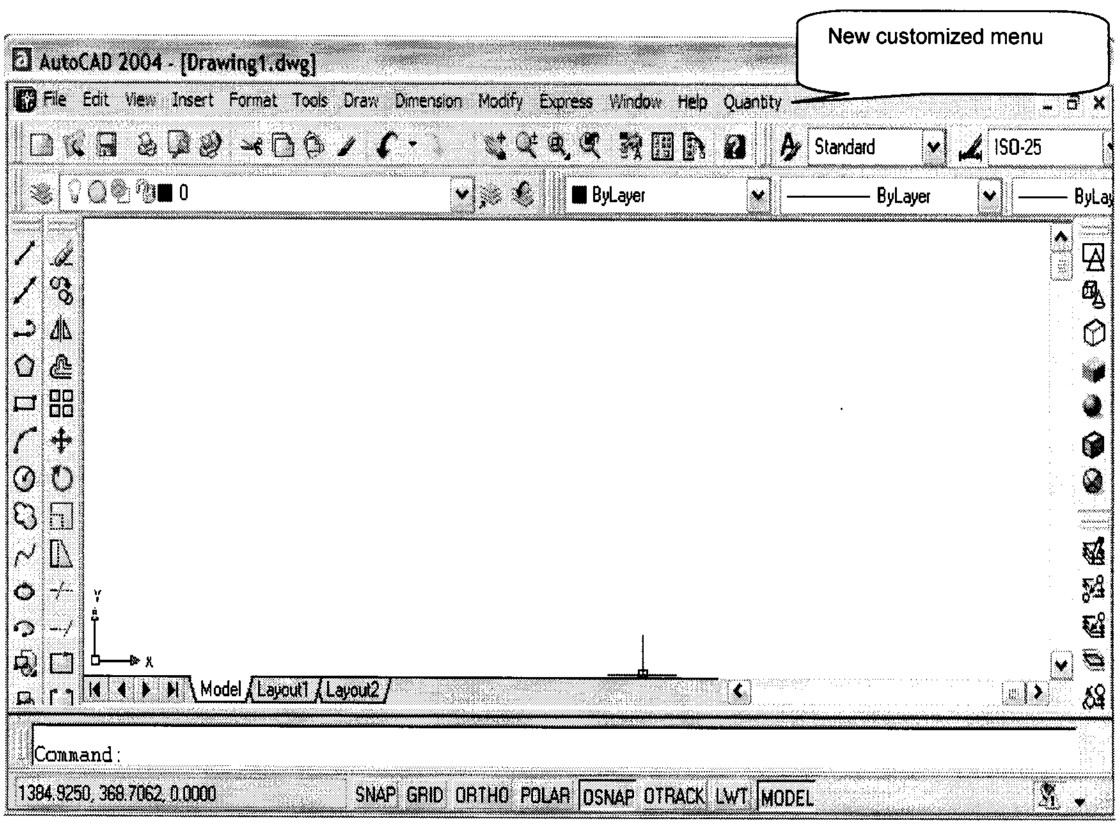


Figure 4.3 The GUI after Customized Menu Loaded

4.4 Establishing a New Project

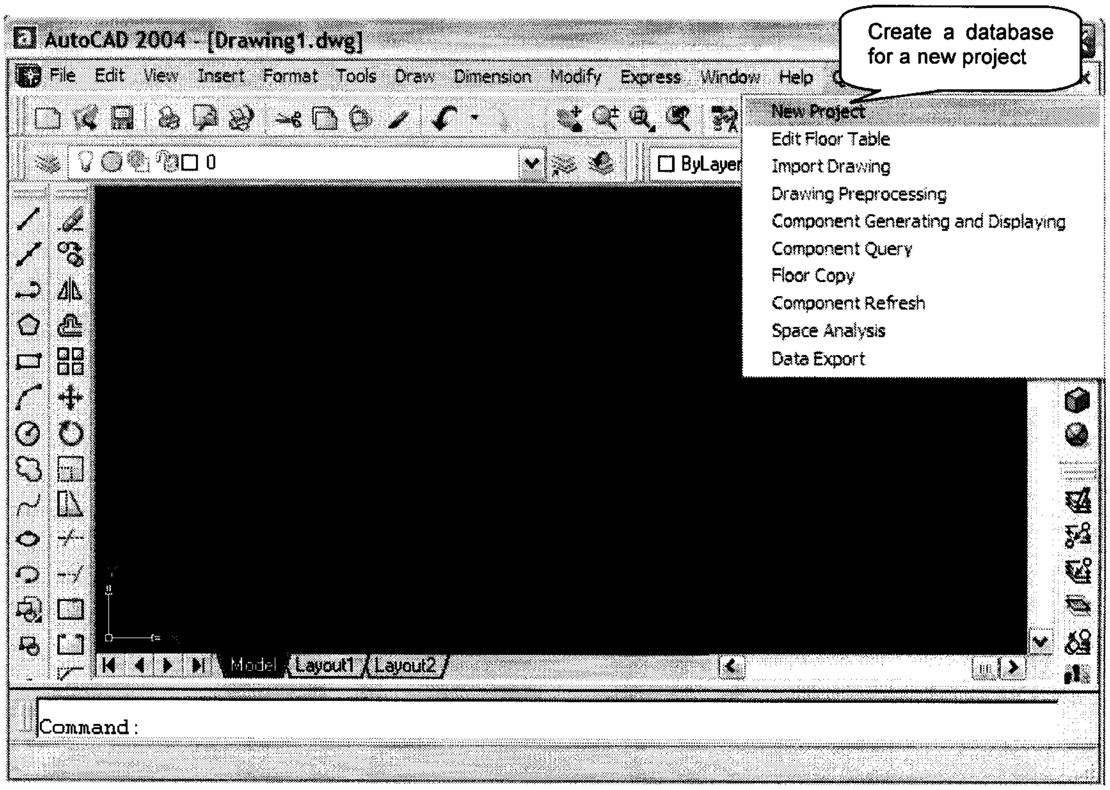


Figure 4.4 New Project Menu Item

Once the user clicks the "Quantity" menu, the menu items will be pulled down and displayed. There are nine menu items in the "Quantity" menu, and each one is a VBA project that consists of a set of forms and modules (subroutines and functions). The next step is to click the menu item "New Project," which will launch the procedure of creating a new database.

In chapter 3, it was mentioned that ActiveX Data Objects (ADO) would be used to communicate between AutoCAD and an external database. ADO is designed to provide developers with a powerful, logical object model for programmatically accessing, editing, and updating a wide variety of data sources through OLE DB system interfaces. The most common function of ADO is to query a table or tables in a relational database, retrieve and display the results in an application, and perhaps allow users to make and save changes to the data. However, if users want to manage databases by creating tables or adding columns, it will be easier to employ Microsoft ActiveX Data Objects Extensions for Data Definition Language and Security (ADOX). ADOX is an extension of the ADO objects and programming model. ADOX includes objects for schema creation and modification, as well as security. As a companion library to the core ADO objects, ADOX offers additional objects for creating, modifying, and deleting schema objects, such as tables and procedures. It also includes security objects to maintain users and groups and to grant and revoke permissions on objects. To use ADOX with VBA, a reference must be established to the ADOX type library. The description of the ADOX library is "Microsoft ADO Ext. for DDL and Security" (Figure 4.5).

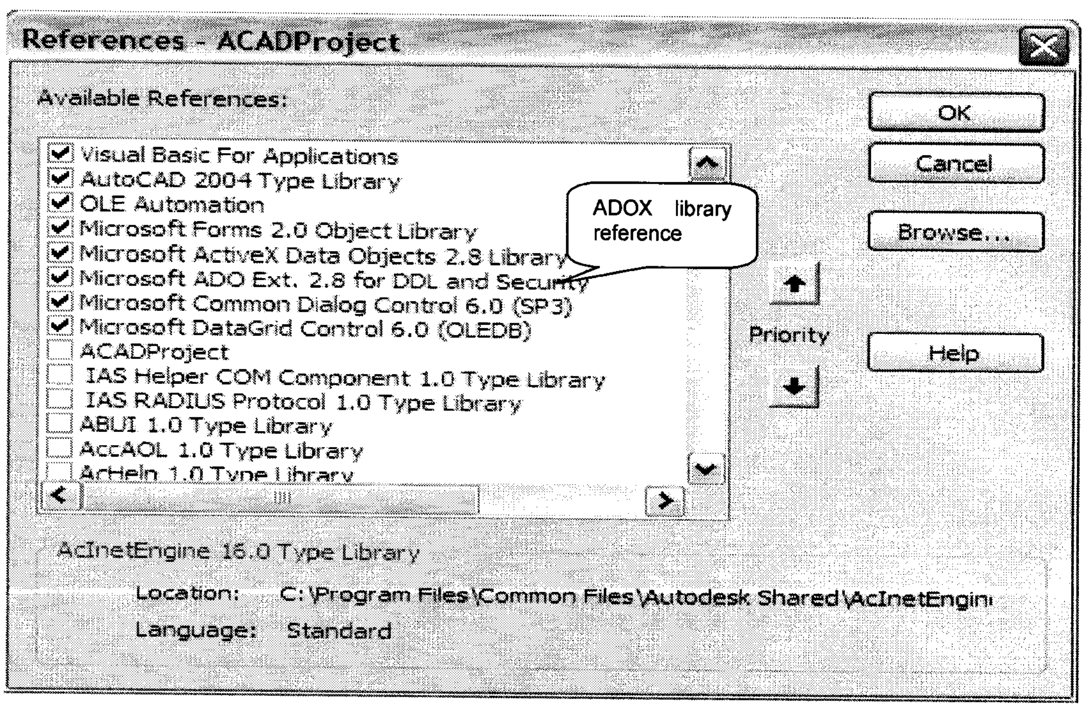


Figure 4.5 Reference to the ADOX type Library

The sample coding of applying ADOX to the database application is shown in

Figure 4.5a.

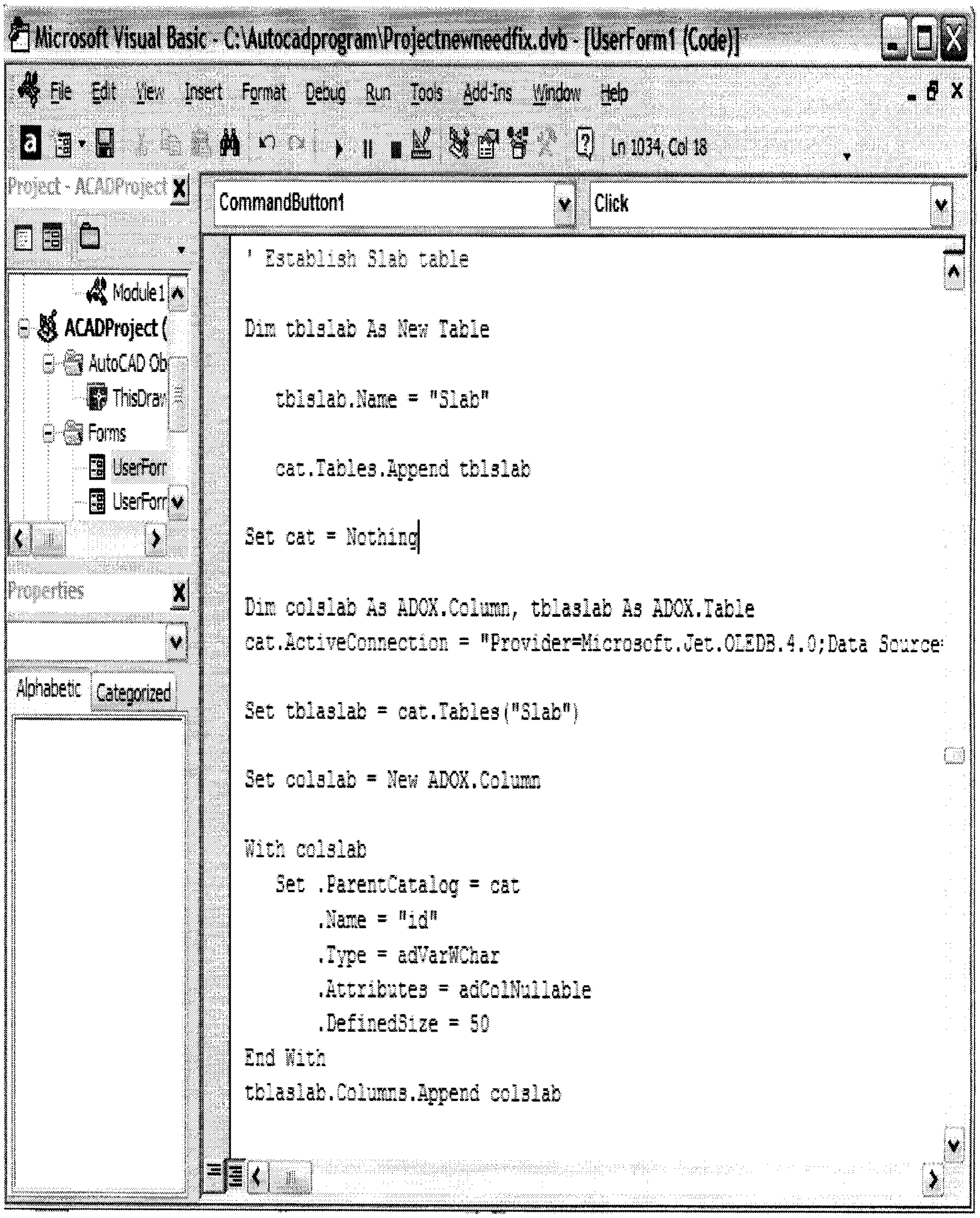


Figure 4.5a Sample Application of ADOX in Coding

The interface of creating a new database is shown in Figure 4.6.

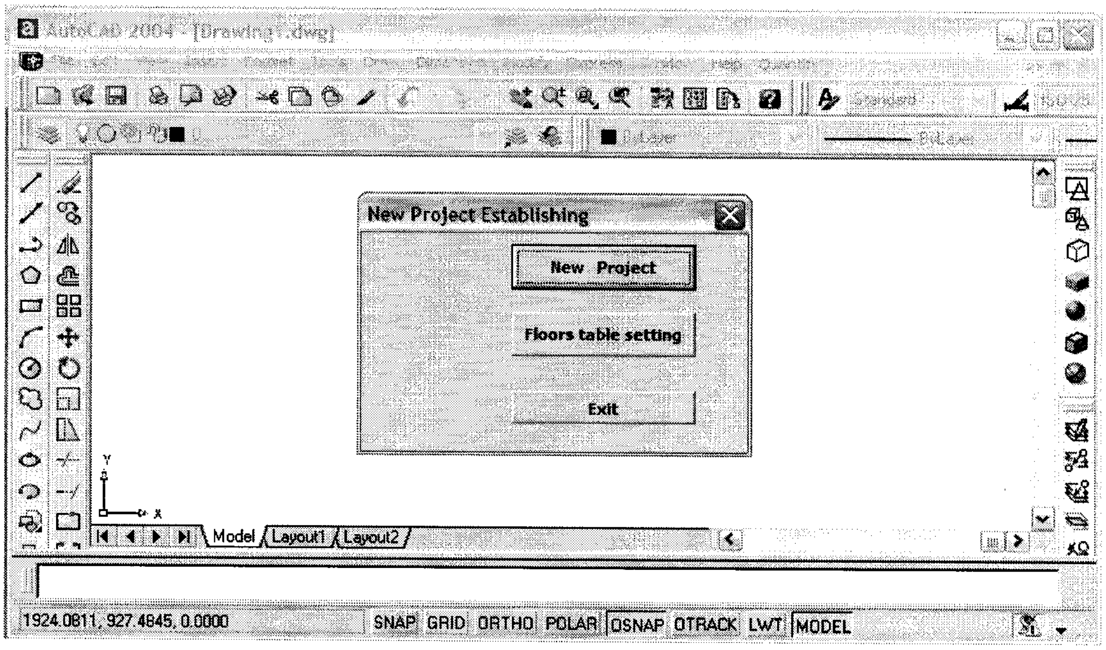


Figure 4.6 Establishing a New Project

When the user presses the "New Project" button, a dialog box will appear to prompt the user to choose the directory and input the name of the new project (see Figure 4.7).

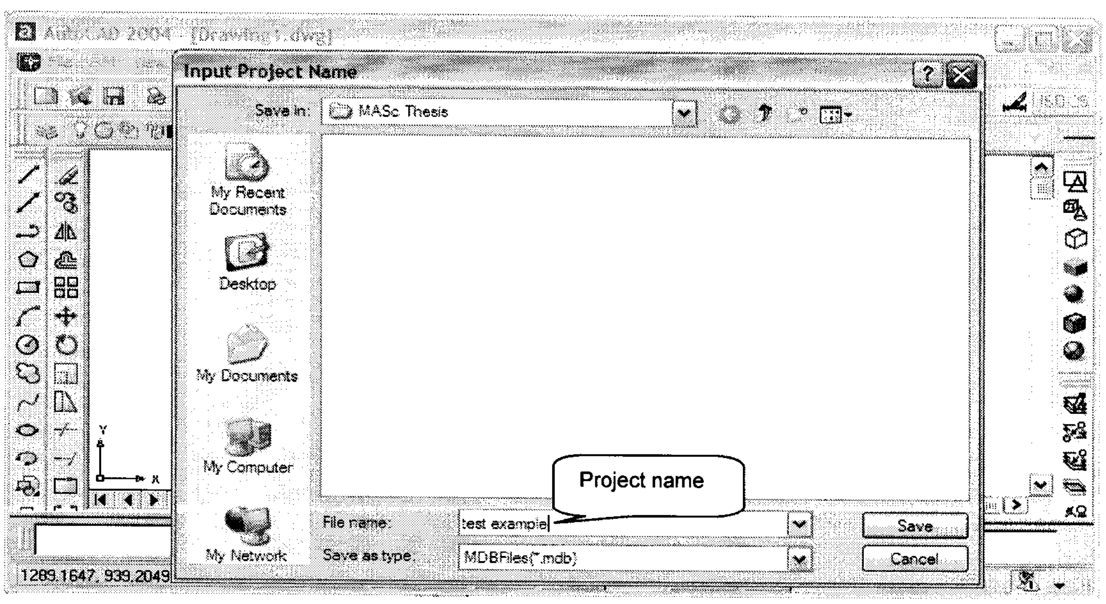


Figure 4.7 Dialog Box for Inputting Project Name

After the user inputs the project name and presses the "save" button, this process is finished, and the interface will return to the one in Figure 4.6 to allow the user to set the Floor table. The user should provide some necessary floor information to facilitate the following 3D modeling. To set the Floor table, the user first needs to press the "Floor table setting" button, and the "Floor Table" form will appear (Figure 4.8).

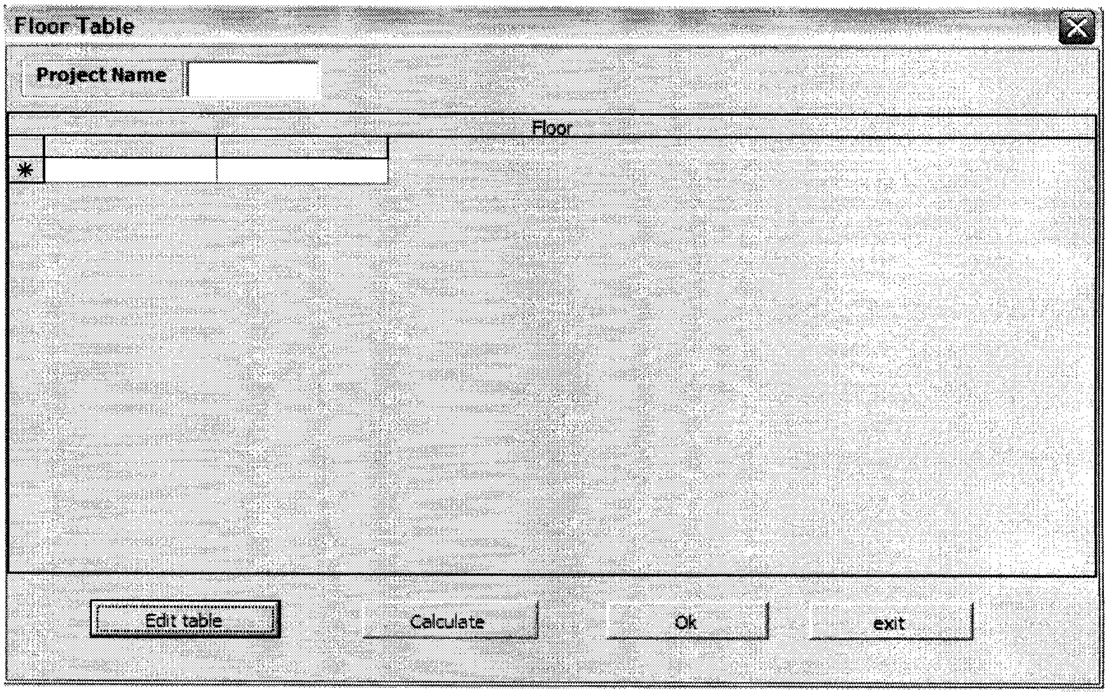


Figure 4.8 The "Floor Table" Form

When the user hits "Edit table," another dialog box (see Figure 4.9) will prompt the user to select a project (database) in which the Floor table is going to be set. After the project selection (Figure 4.10), a message box (Figure 4.11) will appear to let the user confirm that the correct project has been selected. The full path and the file name can be seen in the message box. Now the project name appears in the text box above the floor table and the default value of the floor table comes into view.

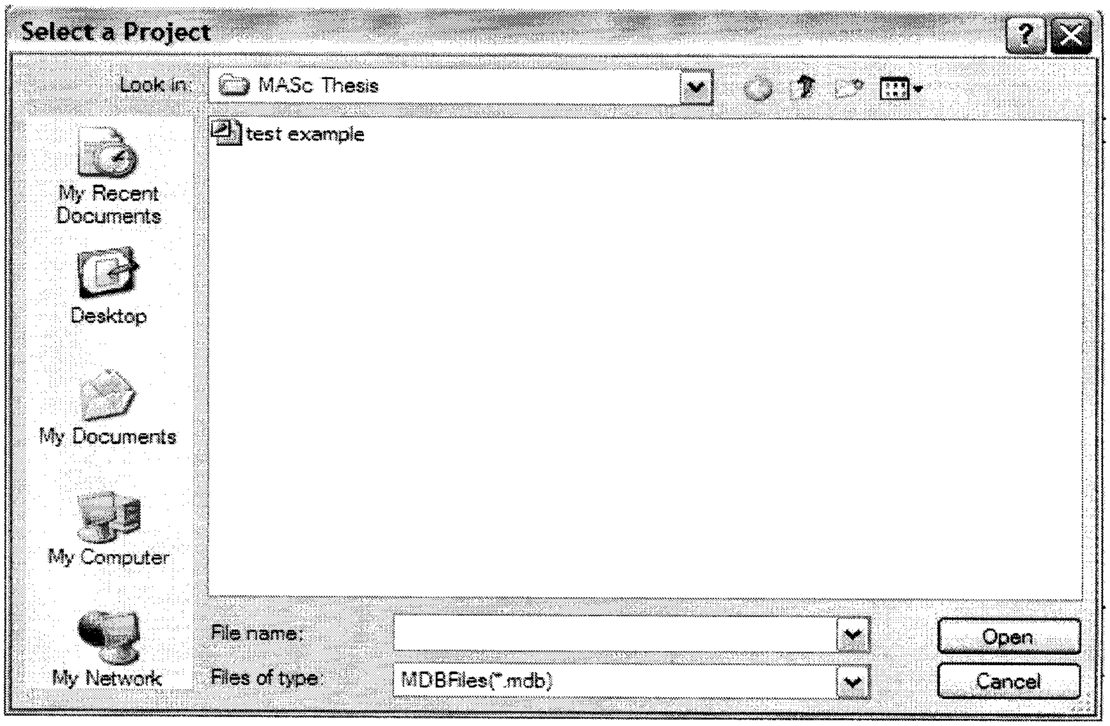


Figure 4.9 Before Project Selection

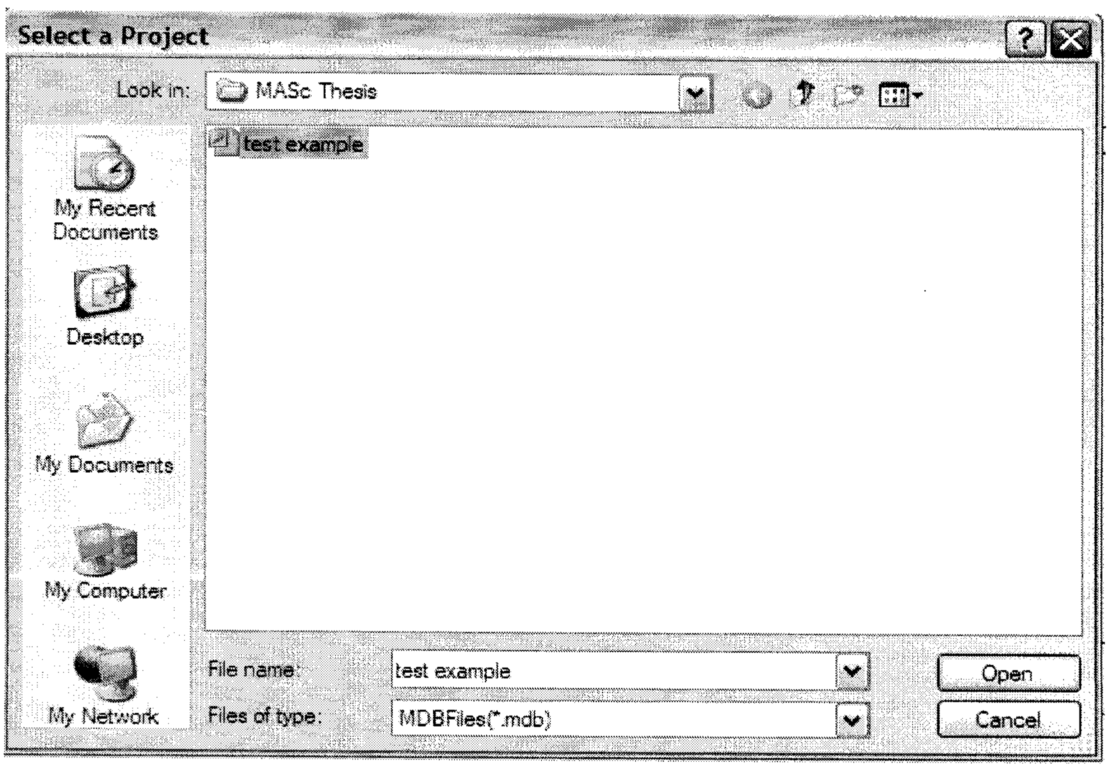


Figure 4.10 After Project Selection

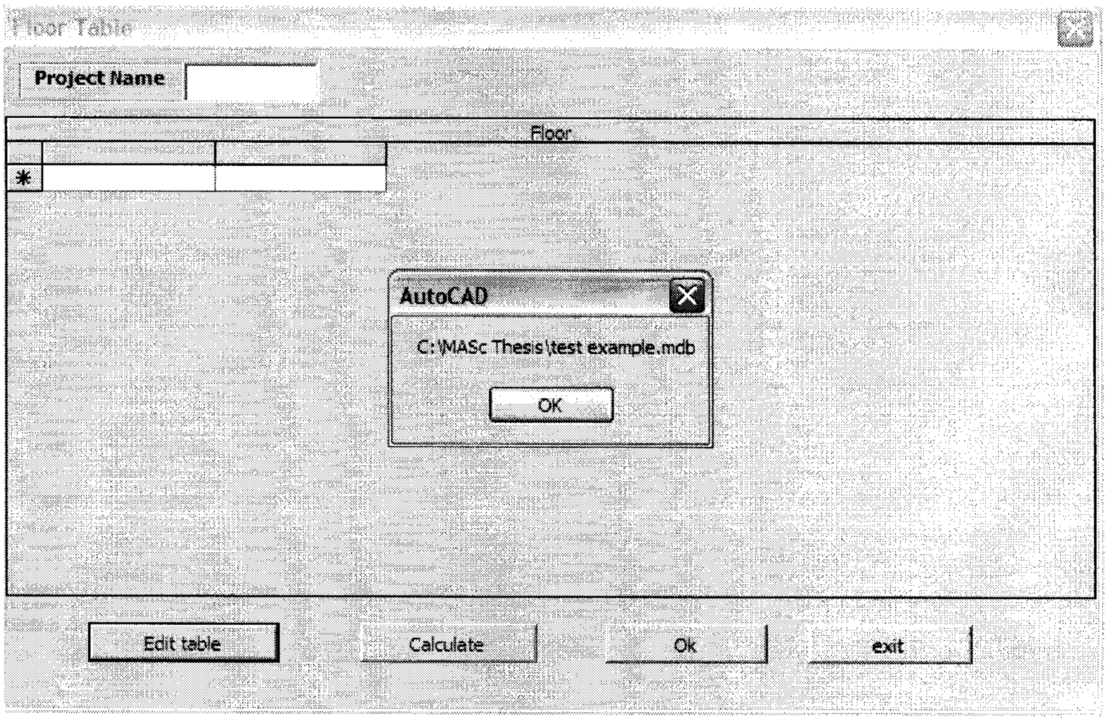


Figure 4.11 Message Box

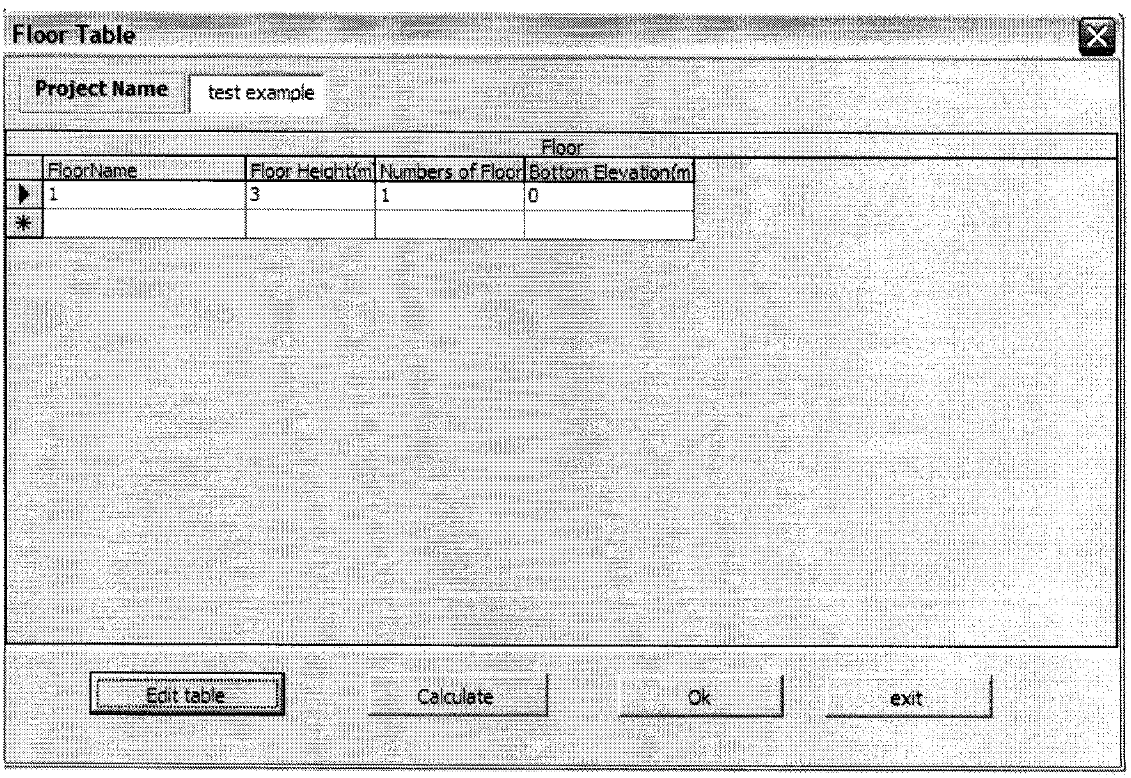


Figure 4.12 Project Name

There are four columns in the floor table. The user needs only to fill in the four columns of the first row. As for the following rows, only the first three columns in each row require input. As well, the order of the floors must be ascending and consecutive (Figure 4.13).

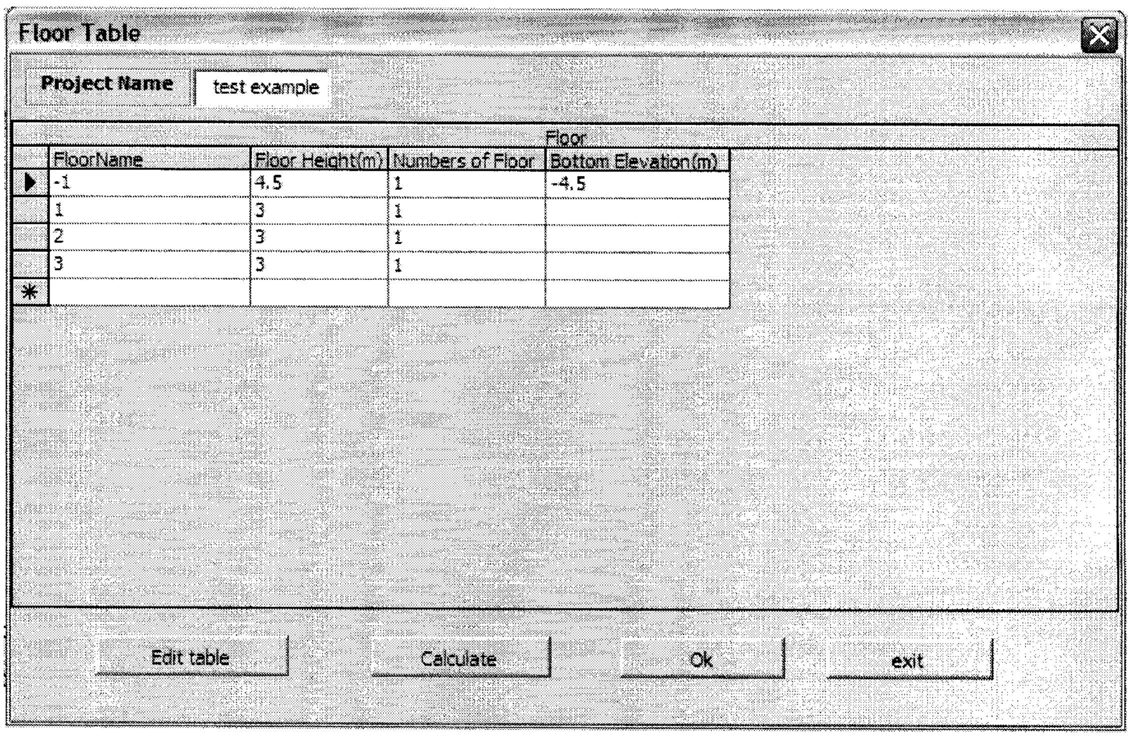


Figure 4.13 Floor Table Data Inputting

When the inputting is finished, click the "Calculate" button, and the value of the fourth column will be filled automatically (Figure 4.14). Now the Floor table is ready for use and we can move on to the next step – 3D modeling. If the user finds that there are errors in the floor table, he still has a chance to modify it. In the "Quantity" menu, below the "New Project" item, there is now another item: "Edit Floor Table" (Figure 4.15); through this menu item, the user can reenter the Floor table, edit and modify it.

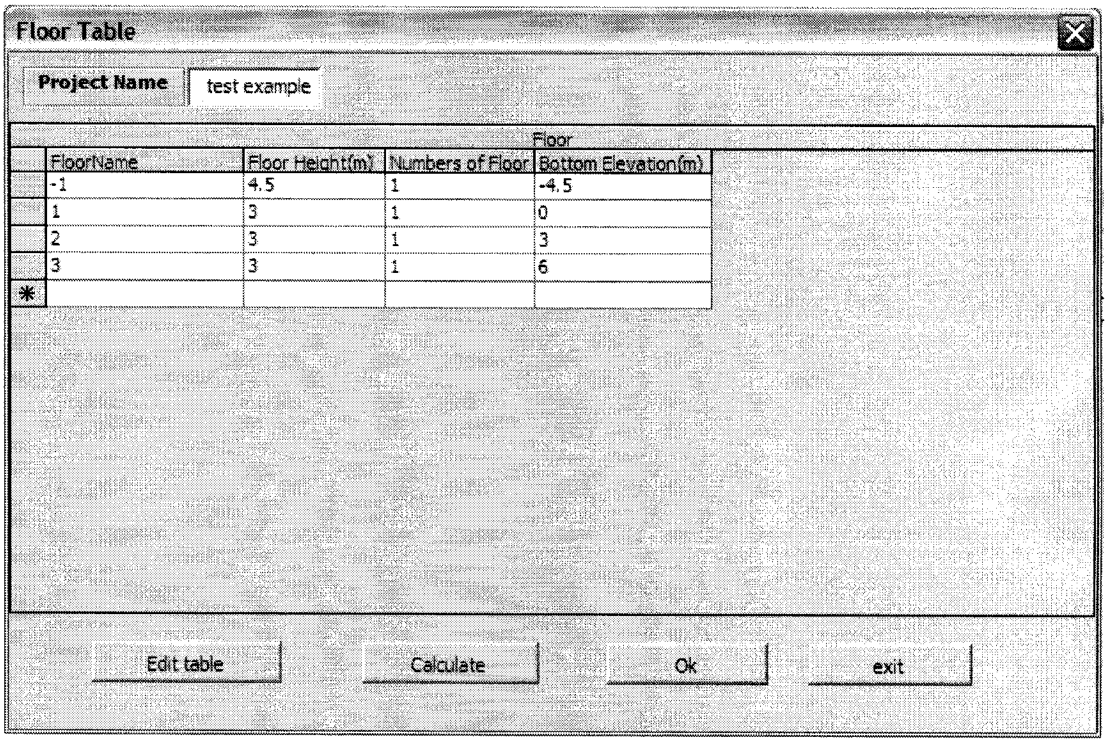


Figure 4.14 Automatic Column Filling

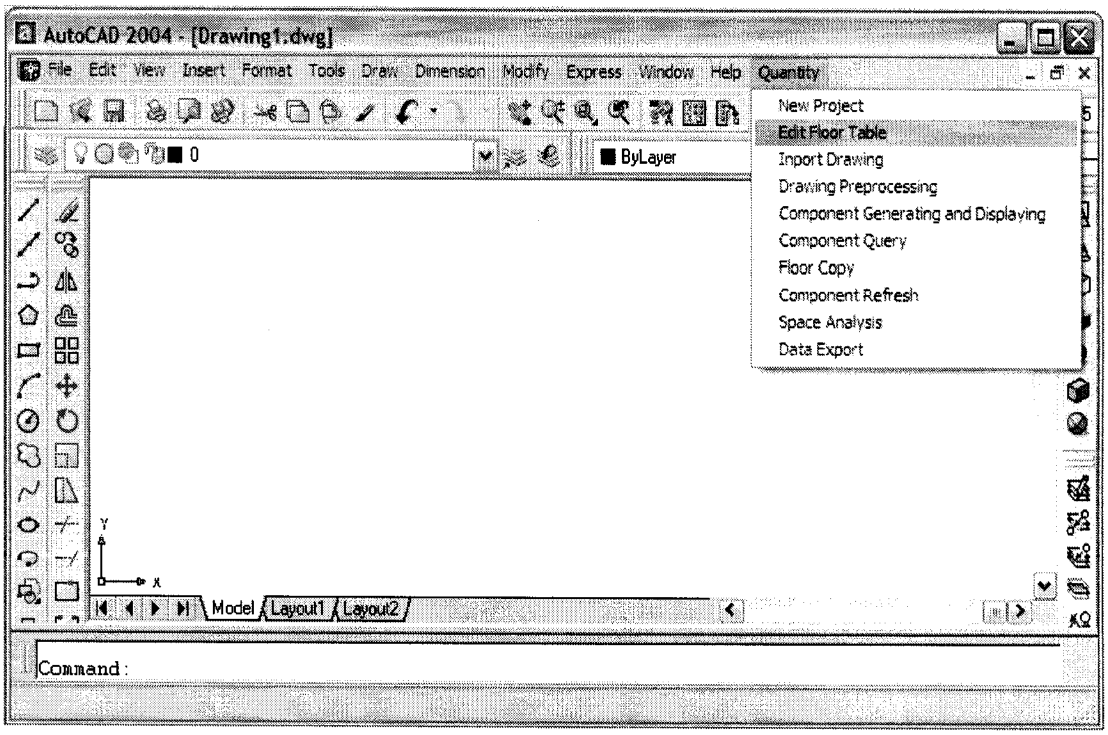


Figure 4.15 Edit Floor Table

4.5 Drawing Preprocessing

The AutoCAD drawings used for 3D modeling are in "dwg" format. This type of drawings can have many types of AutoCAD objects such as lines, polylines, arcs, blocks, external references. However, the more type of objects that the drawings have, the more complicated the process of 3D modeling will be. So, the goal is to make the type of AutoCAD objects as few as possible. If the drawings we are using for modeling have external references, we should bind the external references to the drawings first. The drawings will then be imported for preprocessing (Figure 4.16, Figure 4.17, Figure 4.18).

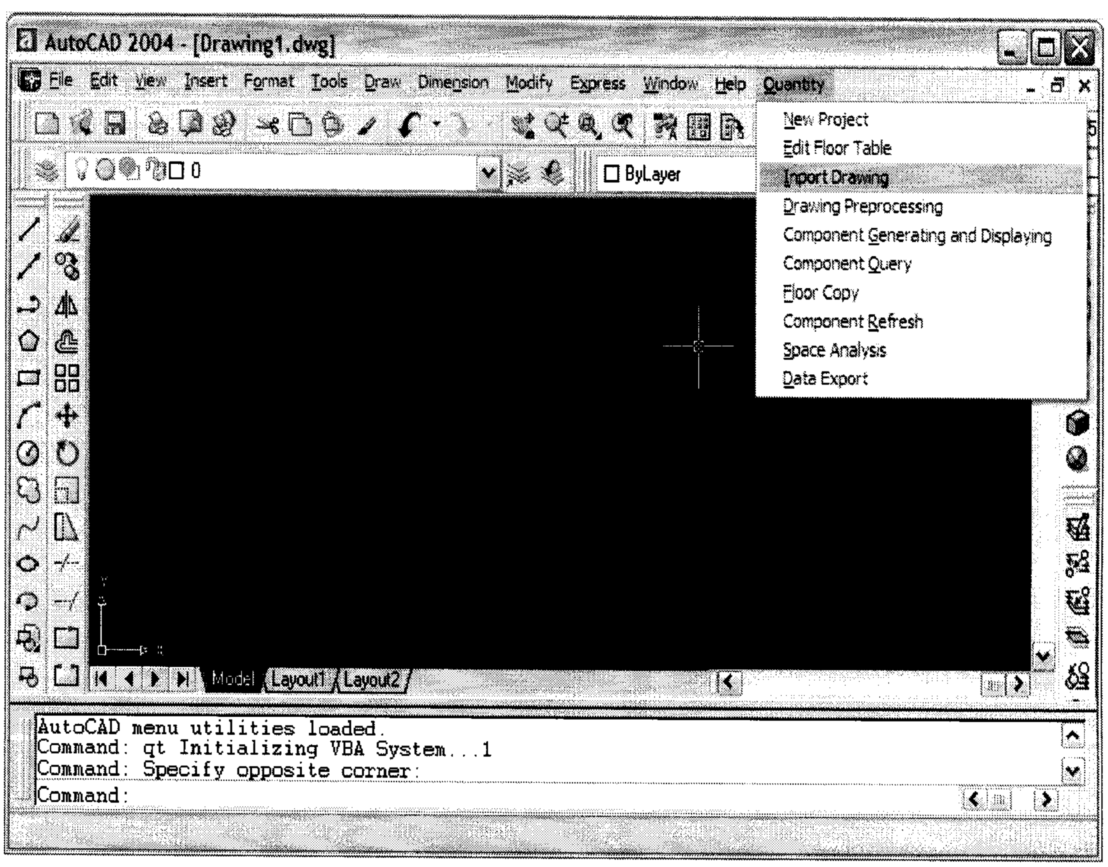


Figure 4.16 Import Drawing Menu Item

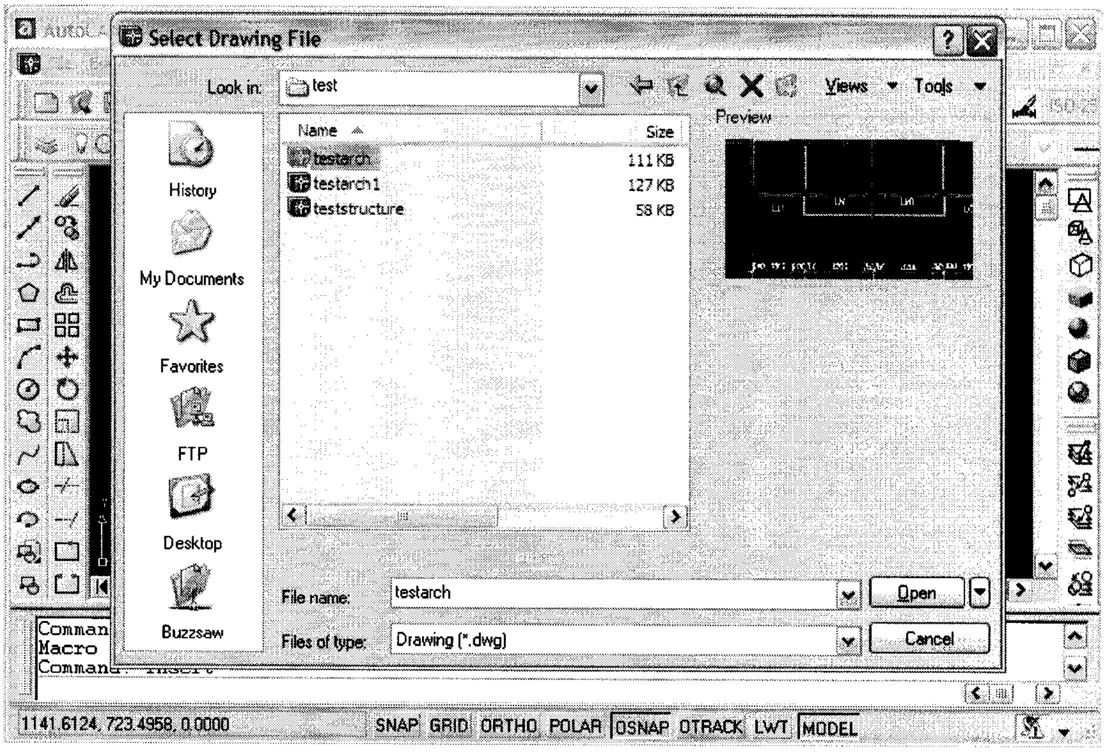


Figure 4.17 Select Drawing File

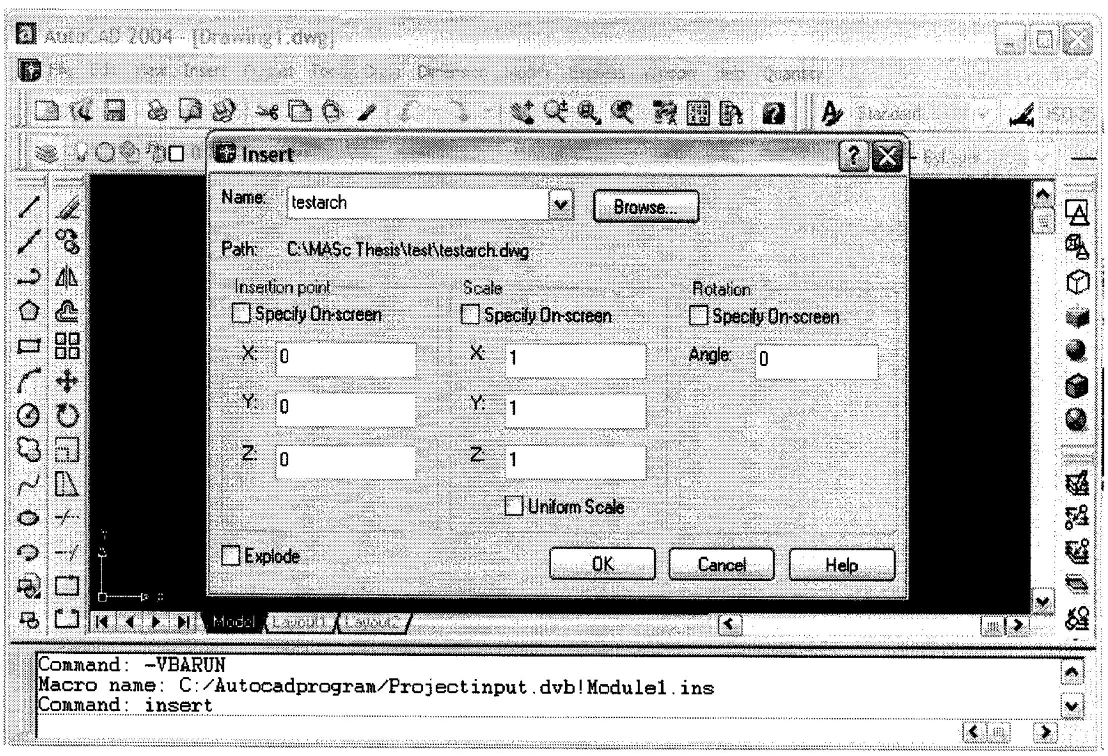


Figure 4.18 Specify the Inserting Mode

After importing, the interface will resemble that shown in Figure 4.19. Since the drawing file is imported as a block, the original drawing file will be left untouched and could be used over again in this way. We can edit the imported drawing, using 'Drawing Preprocessing' menu item, to make it more convenient for 3D modeling,

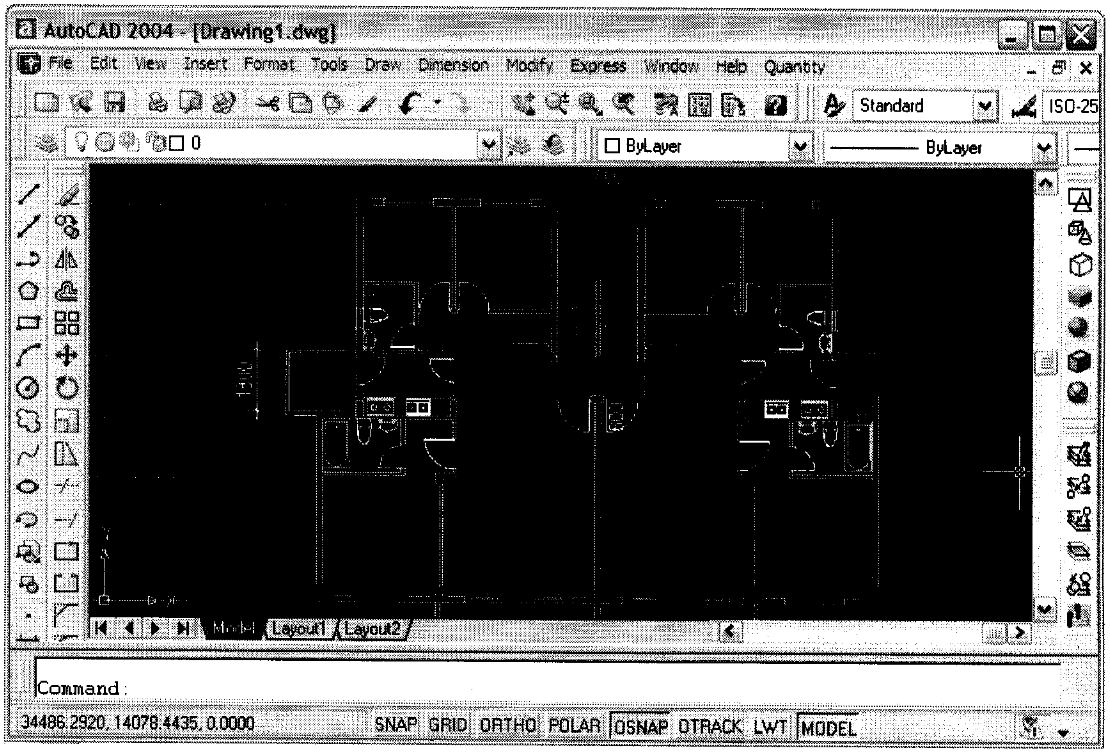


Figure 4.19 The Imported Drawing

In the Drawing Preprocessing process, all the compound AutoCAD objects in the present drawing will be broken down into basic AutoCAD objects, mostly lines, circles and arcs. This step will make further 3D modeling easier to manage because there are fewer types of objects to deal with. This also means that job efficiency can be improved. The difference between the non-processed and processed drawing can be observed through the properties window in Figure 4.20 and Figure 4.21.

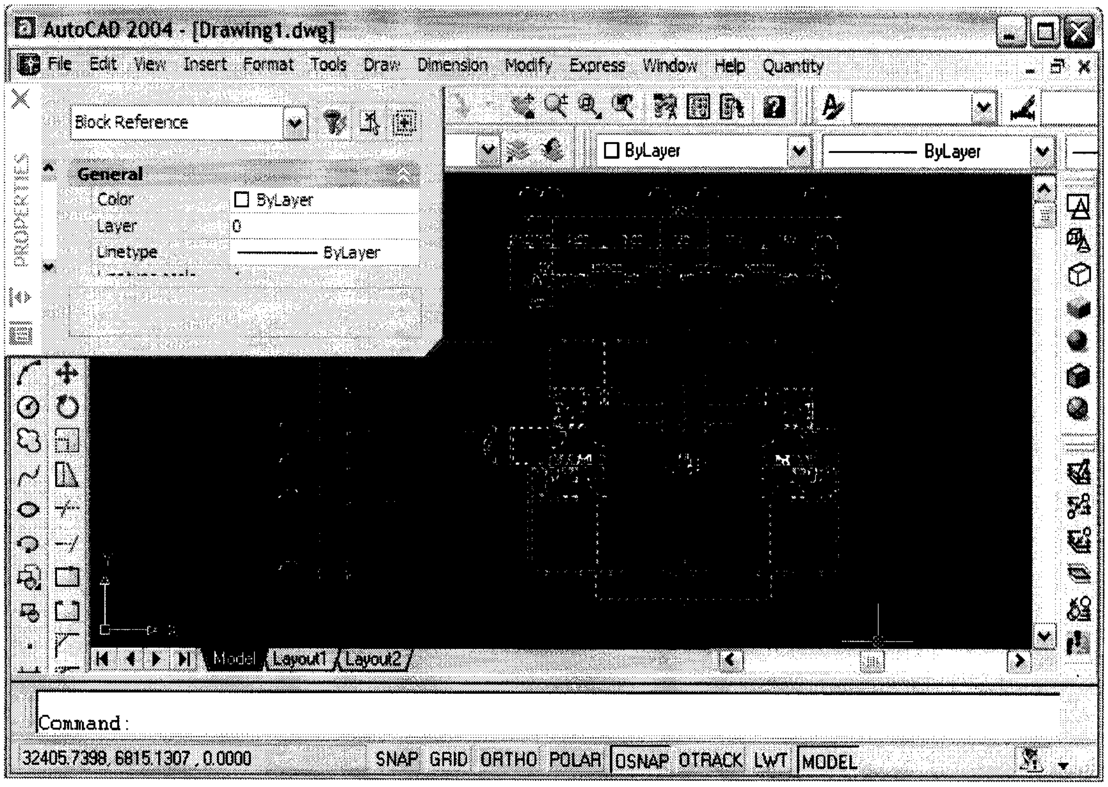


Figure 4.20 Drawing Before Preprocessing

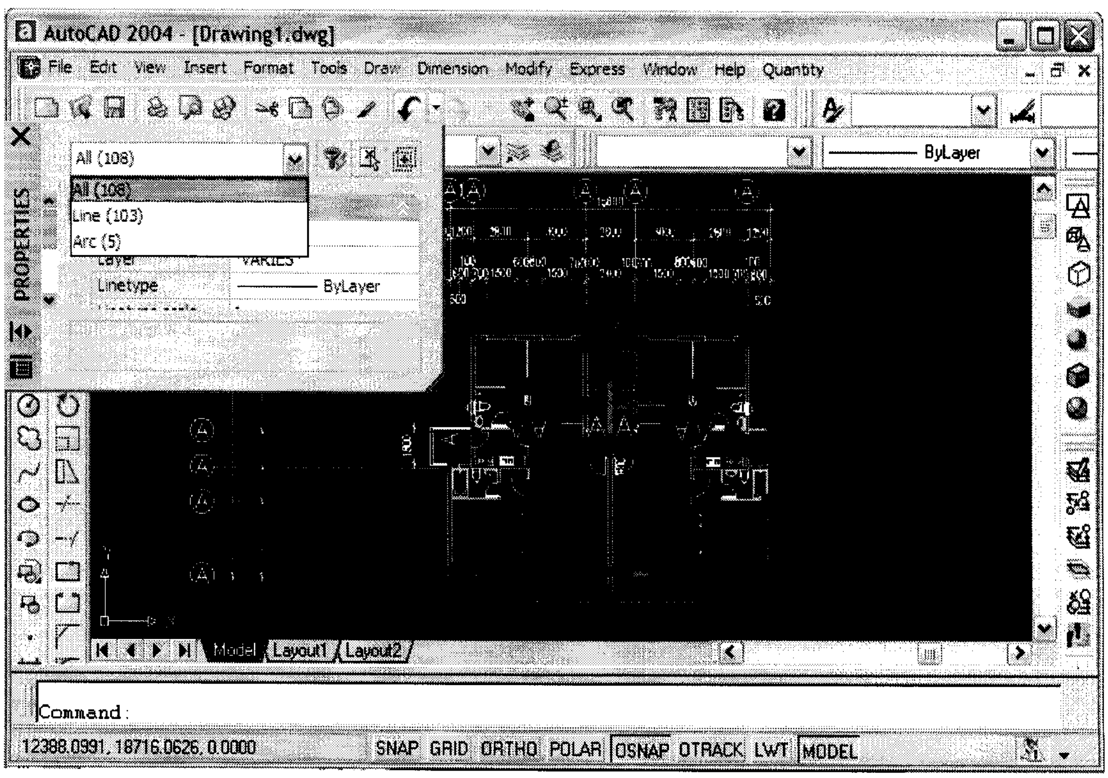


Figure 4.21 Drawing After Preprocessing

4.6 3D Modeling

After the foregoing preparation work, it is time to begin another important task: 3D Modeling. The 3D model will be established floor by floor based on the processed architectural plan, structural plan and the information contained in the Floor table. All the components on a particular floor are created on their corresponding layer in AutoCAD. When the Floor table is established, the floors' corresponding layers are also created in AutoCAD (Figure 4.22).

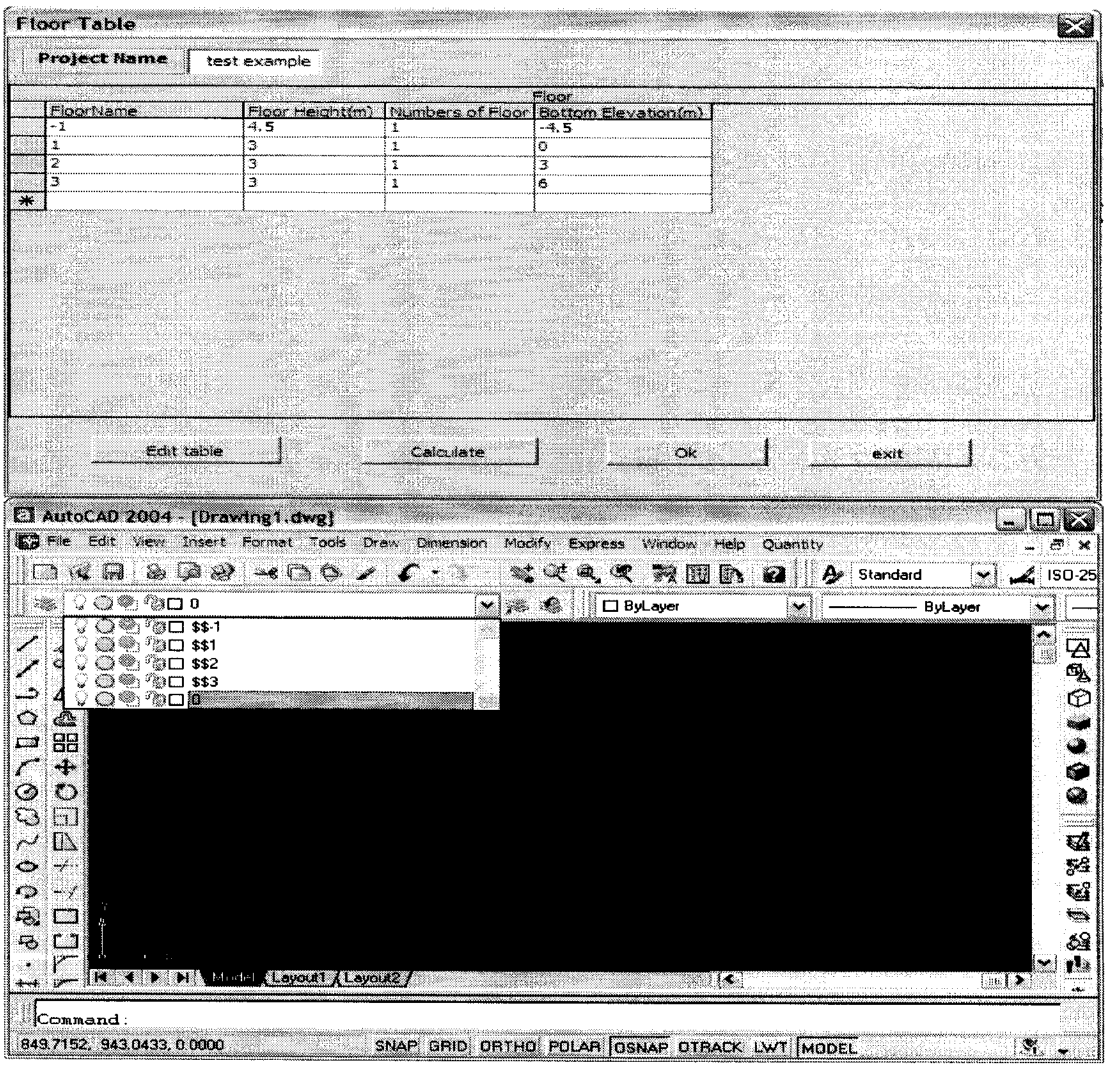


Figure 4.22 Floors in Floor Table and Corresponding Layers in AutoCAD

The relationship between the floors in the Floor Table and corresponding layers in

AutoCAD is illustrated in Table 4.1.

Table 4.1 Floors in the Floor Table and Corresponding Layers in AutoCAD

Floors in Floor Table	Corresponding Layers in AutoCAD
-1	$$-1
1	$$1
2	$$2
3	$$3

The module will allow the user to obtain the 2D information from the architectural plan and the structural plan, and the third dimension information from the established Floor table. For each floor, it is recommended that the user generate the components following the order as shown in Figure 4.23:

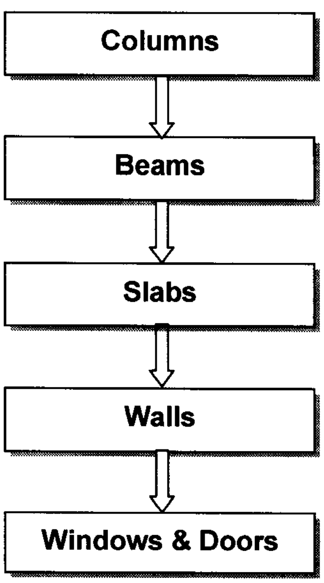


Figure 4.23 Modeling Order

This is almost the same as the ordinary order of construction. First the columns, then the beams are built between columns or as cantilever beams. Slabs are usually built within the boundaries of beams and columns. Windows and doors belong to a specific wall.

Since the construction drawings are designed and drafted by professionals, the designers will normally put similar types of objects on the same layer. For example, the columns, walls, beams, and other types of objects are assigned on separate layers. The modeling process benefits a lot from these arrangements. Before generating one type of component, the user can isolate the graphical objects on one layer, which stand for one kind of component, from other graphical objects. By doing so, it will be easier for the user to identify the graphical objects that need to be generated into 3D components (Figure 4.24, Figure 4.25).

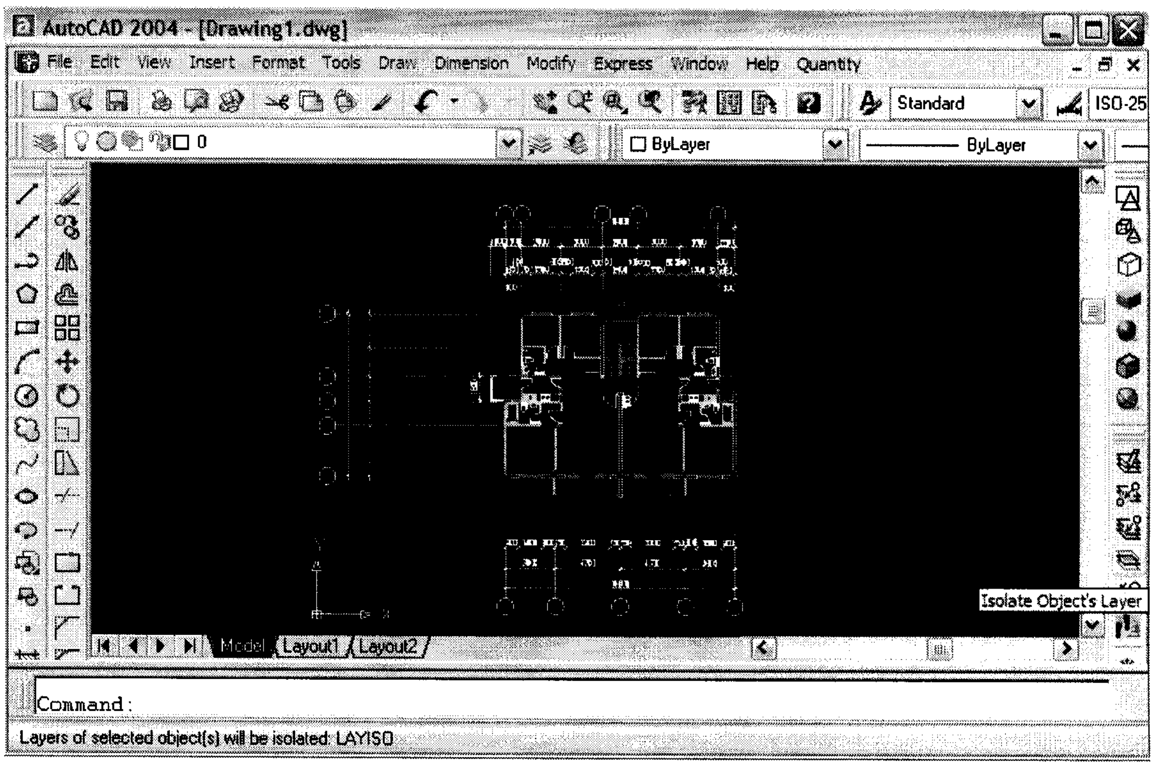


Figure 4.24 Interface Before Isolating Object's Layer

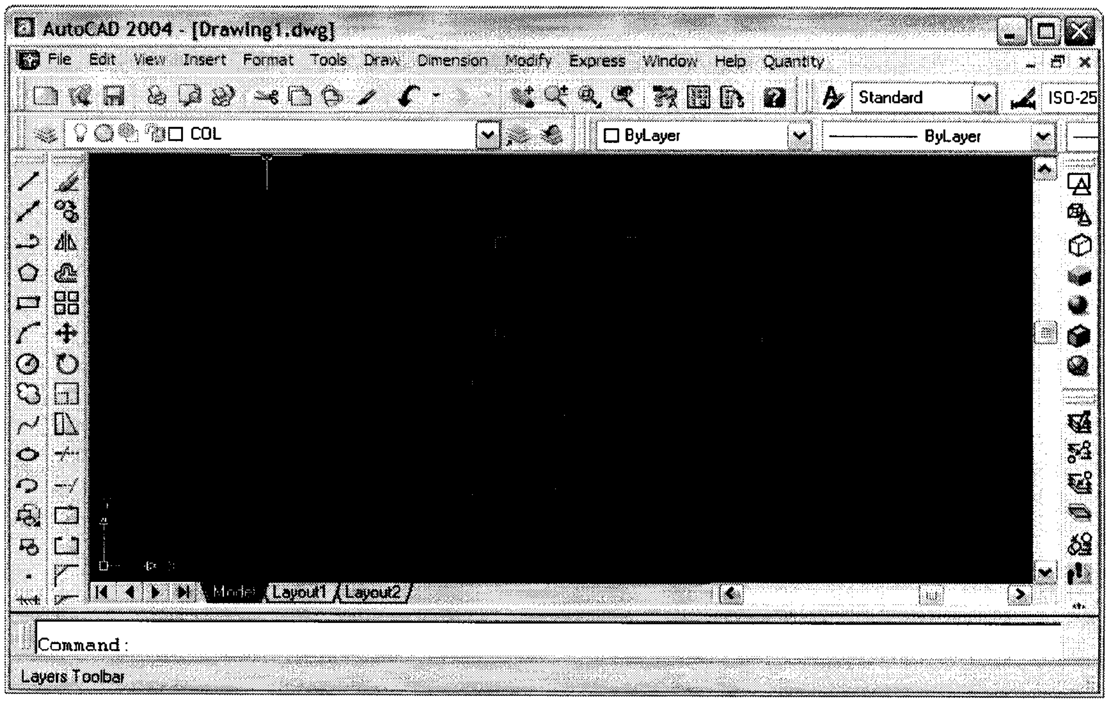


Figure 4.25 Interface After Isolating Object's Layer

By clicking on the menu item "Component Generating and Displaying," the user will enter into the process of 3D modeling (Figure 4.26, Figure 4.27).

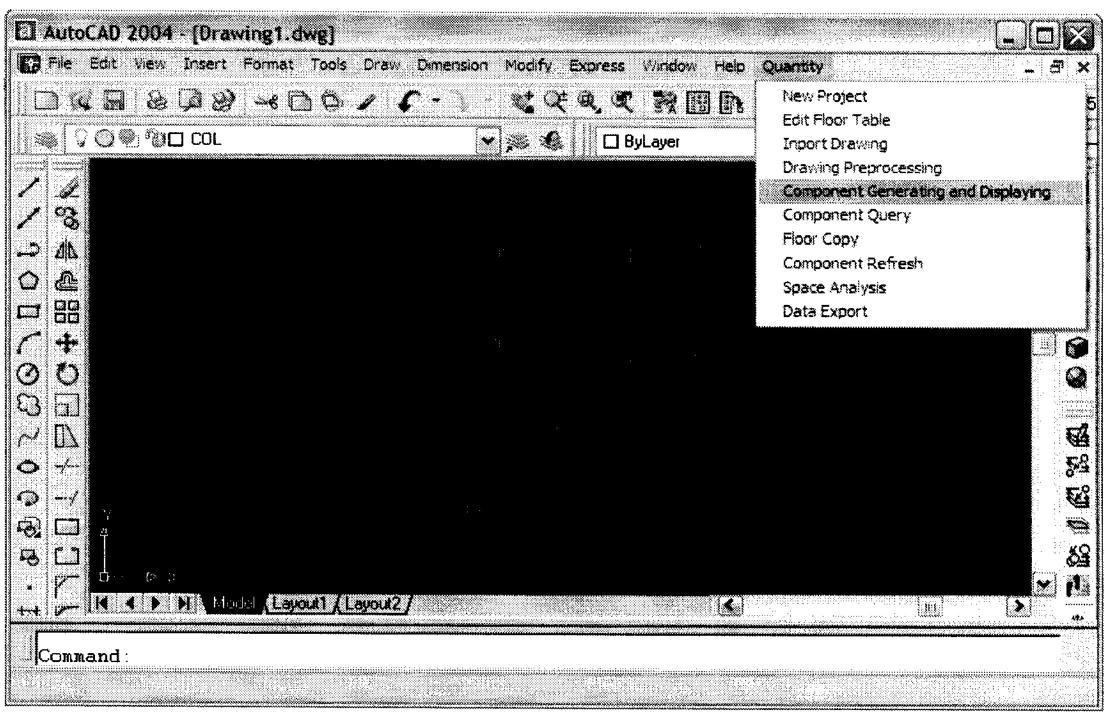


Figure 4.26 Component Generating and Displaying Menu Item

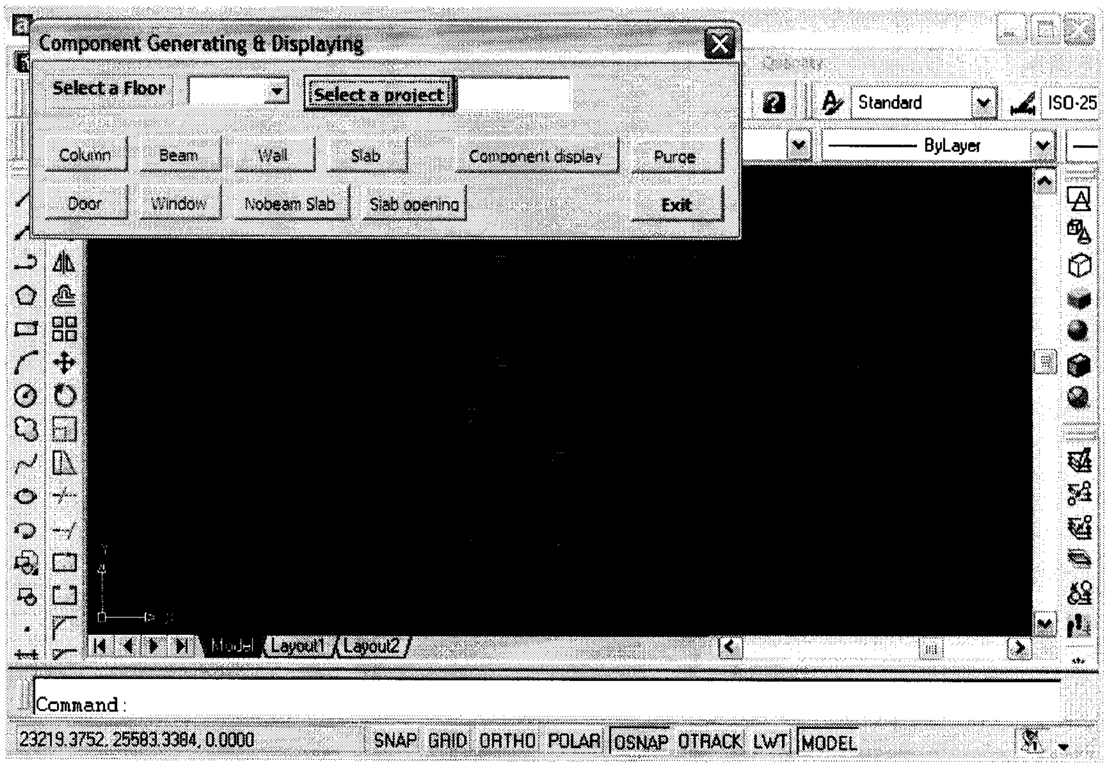


Figure 4.27 The Main Form of 3D modeling

As we mentioned in the previous paragraphs, the 3D modeling is performed floor by floor and the floor information is to be obtained from a specific project database, so the first step of 3D modeling is to select a project and a particular floor. When the user hits the button "Select a project", a dialog box appears and prompts the user to select the targeted project from the available projects under a specified directory (see Figures 4.28 and 4.29).Once finished, the name of the project will appear in the text box beside the selection button (Figure 4.30).

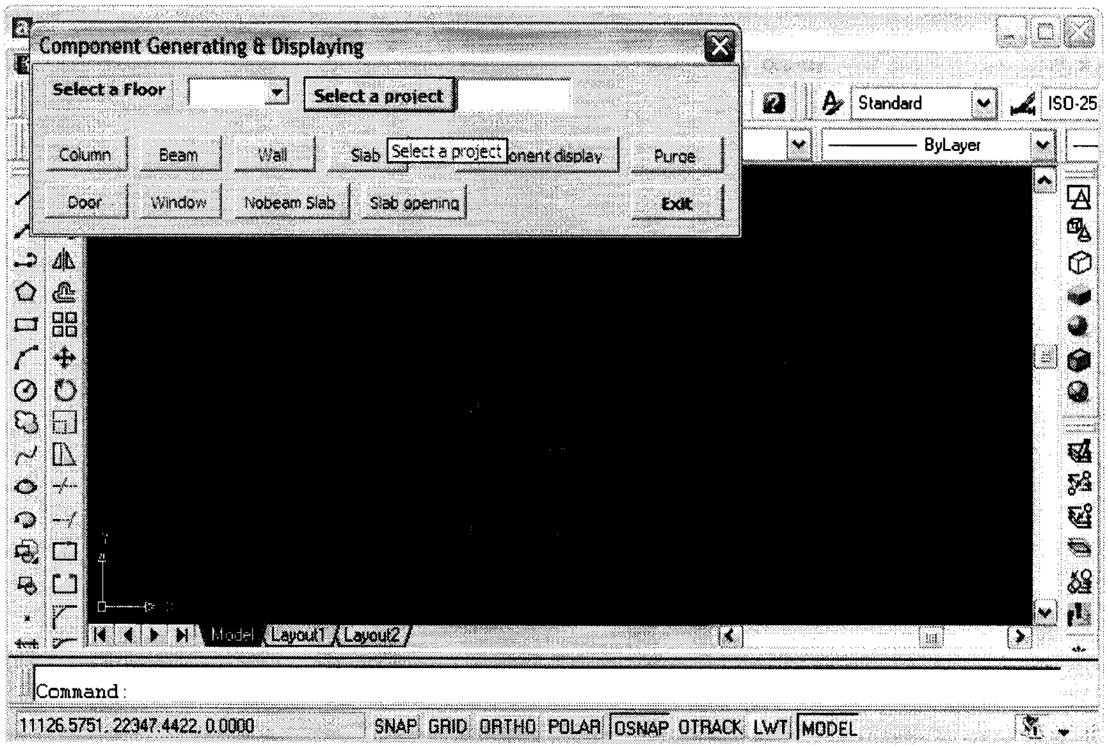


Figure 4.28 Select a Project

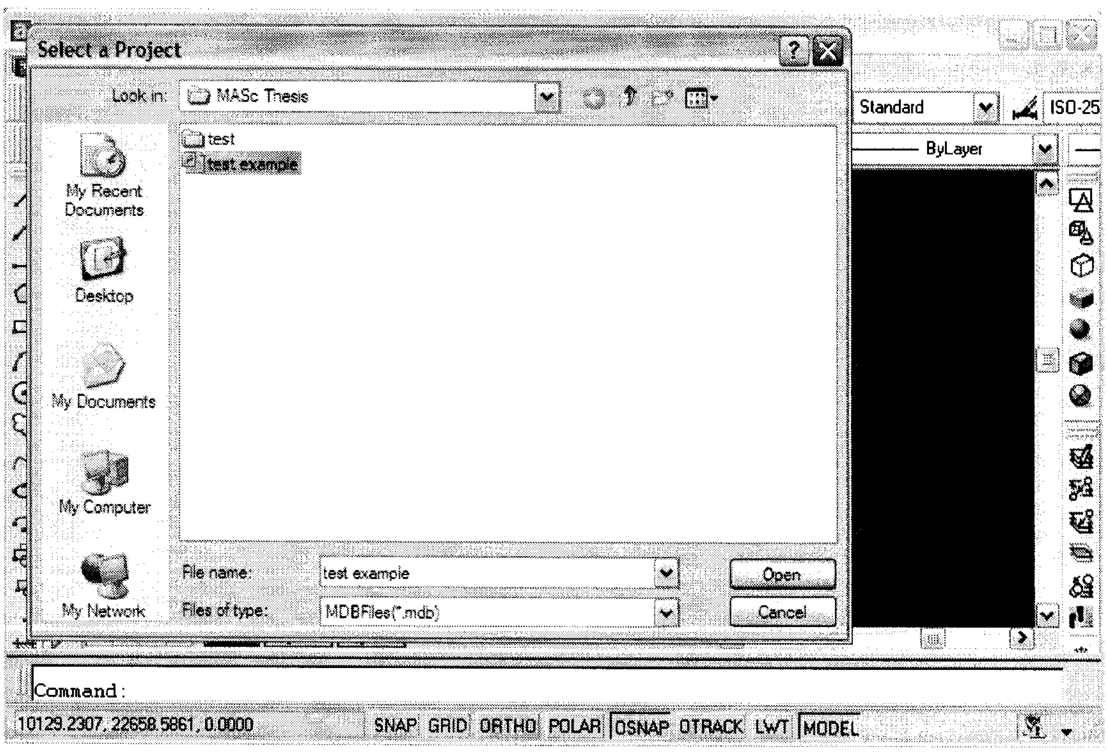


Figure 4.29 Dialog Box of Selecting a Project

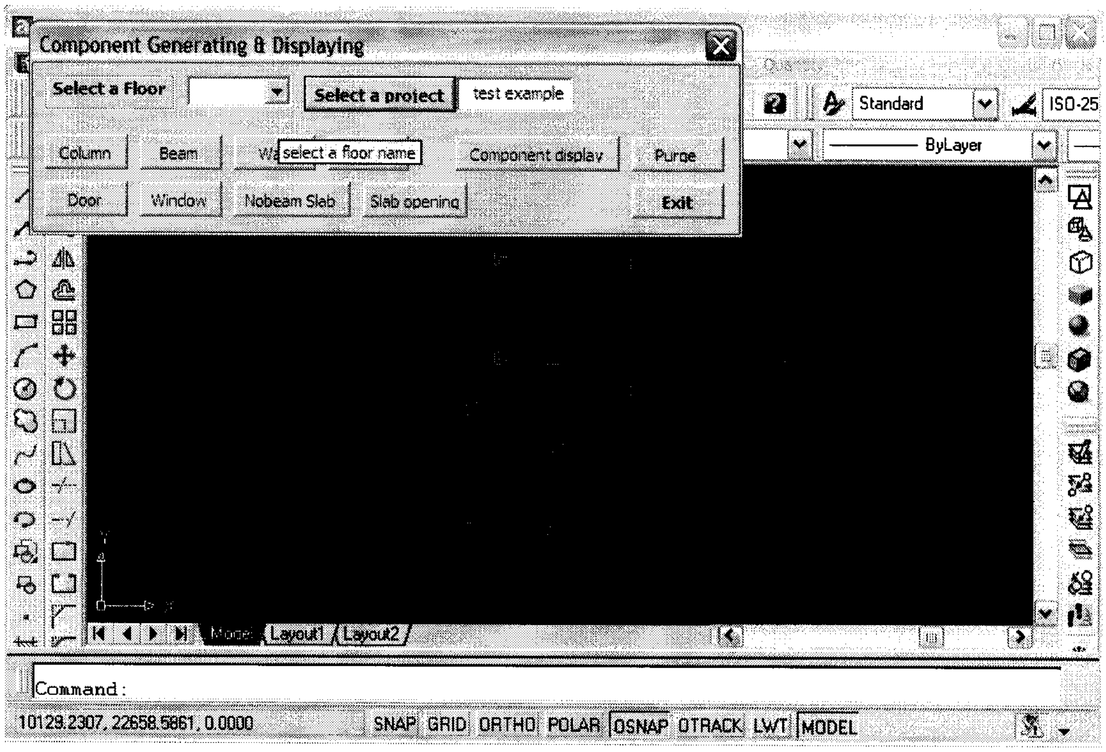


Figure 4.30 Selected Project Name

Now, the user is allowed to go further, selecting the targeted floor name from the list box in the main form. All the floor names of the selected project are listed in the box (Figure 4.31). Once again, after the selection is made, the selected floor name appears in the list box (Figure 4.32). By this time, the environment of modeling on a certain floor has been already set. Under this setting, only the components that will be generated on the targeted floor and the processed drawing will appear on the screen. This is very important because the user can focus on the modeling of components on the selected floor without interfering with the components on the other floors.

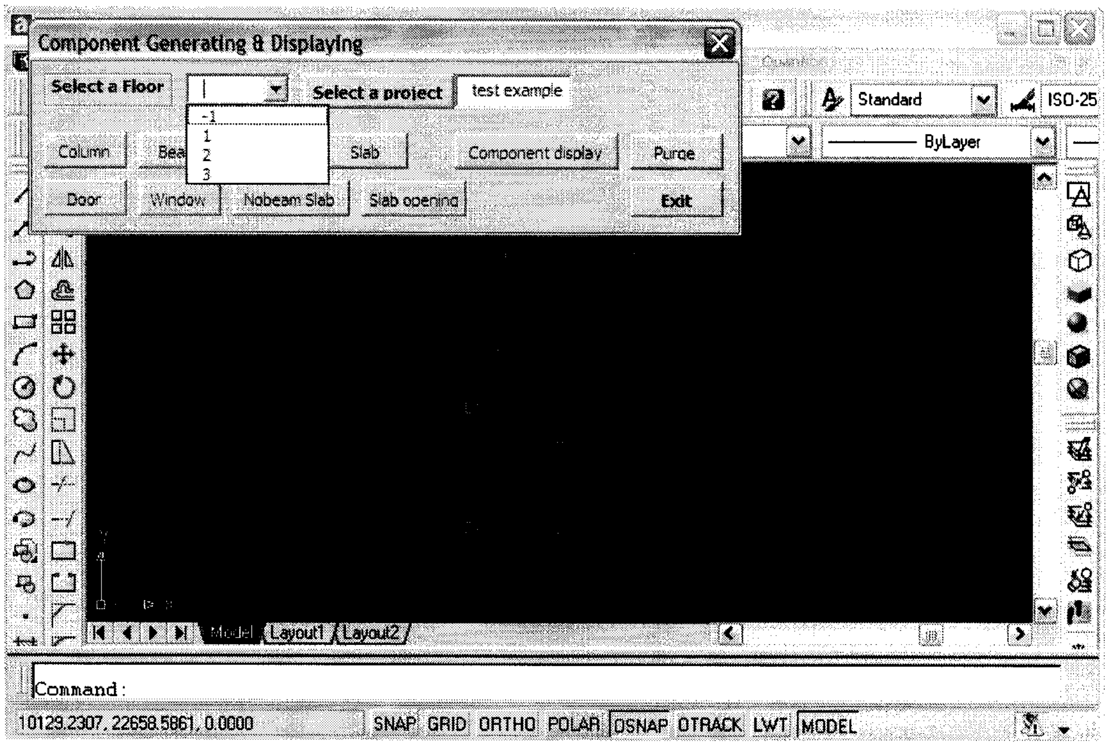


Figure 4.31 Floors Names for Selecting

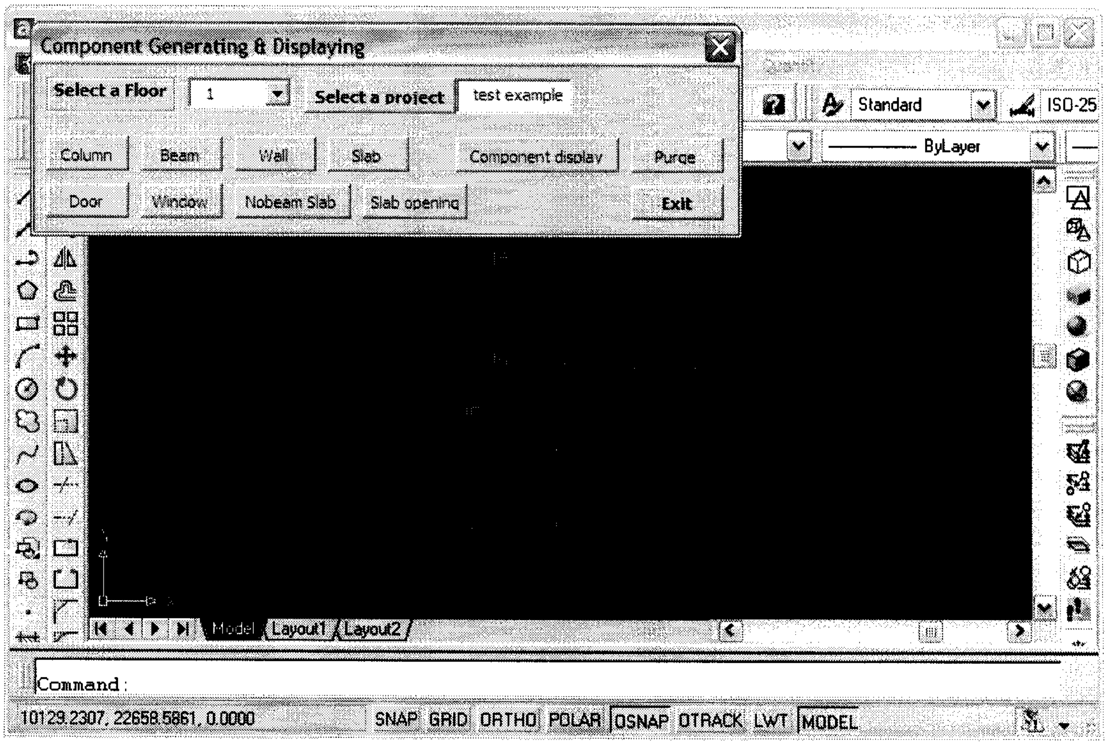


Figure 4.32 Targeted Floor Name

4.6.1 Column modeling

According to the recommended order of modeling, the user should begin with the columns (Figure 4.33). Four methods are designed to facilitate column modeling.

Method 1: If the columns were assigned on the same layer, the user only needs to click "Select an Object", selecting one graphical object on the layer (Figure 4.34), and then the "Create" button; normally, all the columns (3D) on the targeted floor will be generated automatically. This can be observed in Figure 4.35. The colour of successfully generated columns is yellow. Figures 4.36 through 4.38 show the 3D views of the generated columns.

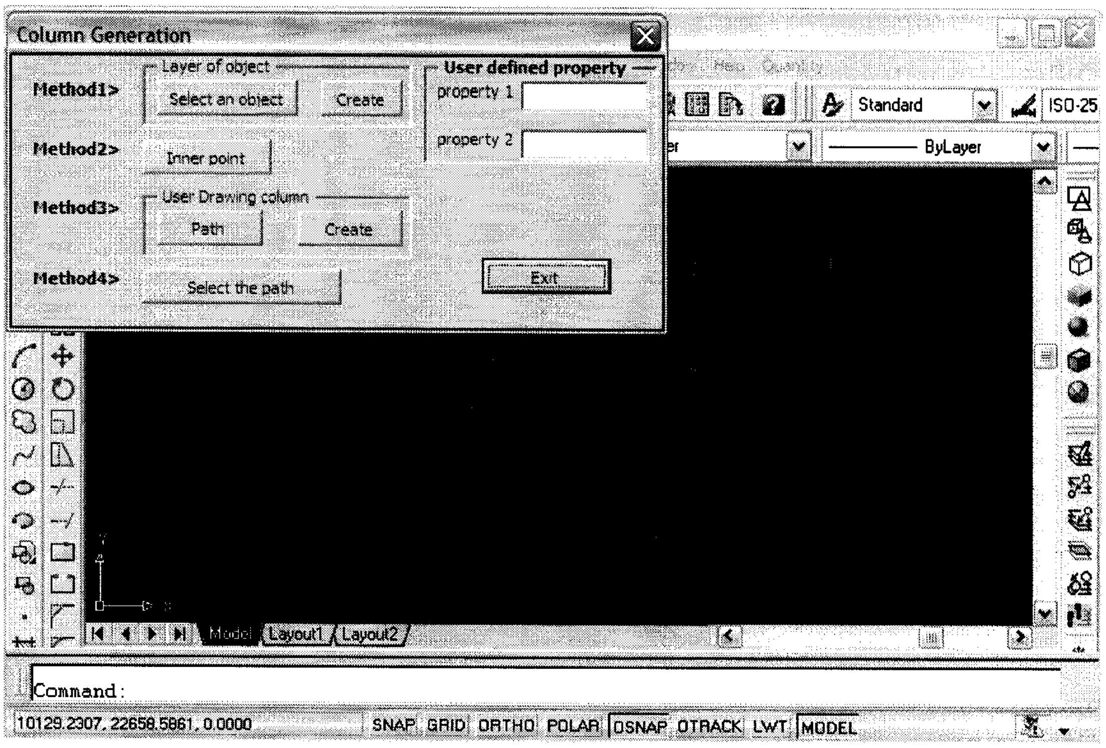


Figure 4.33 The Interface of Column Modeling

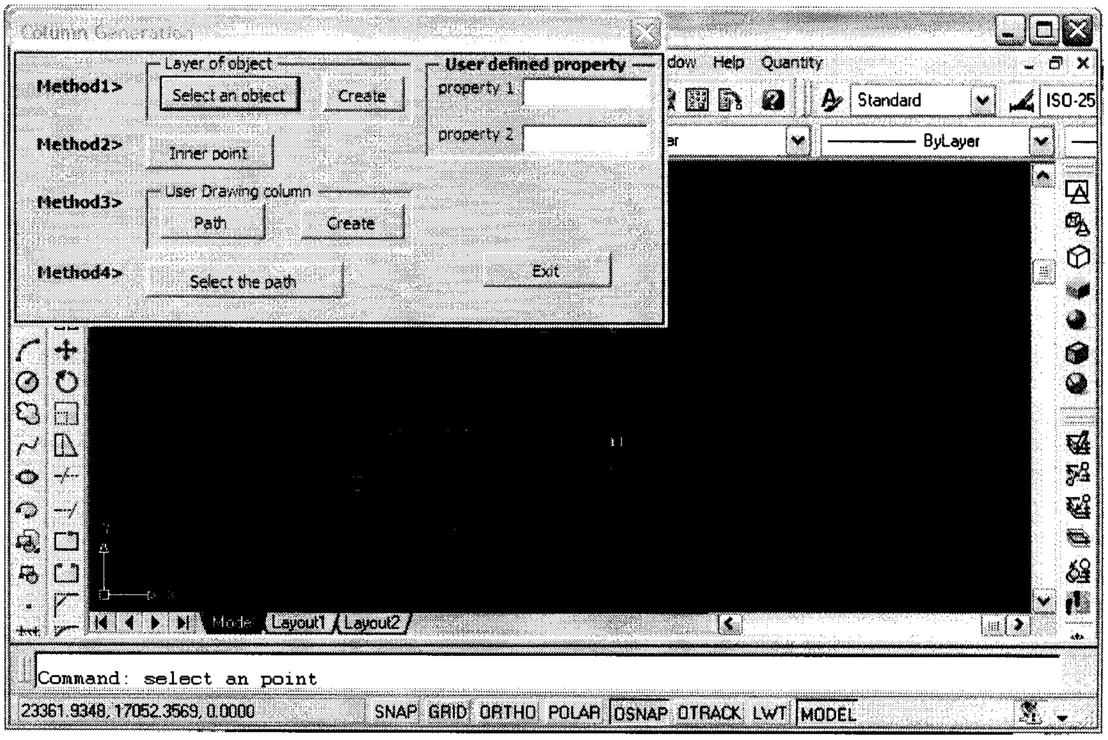


Figure 4.34 Selecting an Object on the Layer of Column

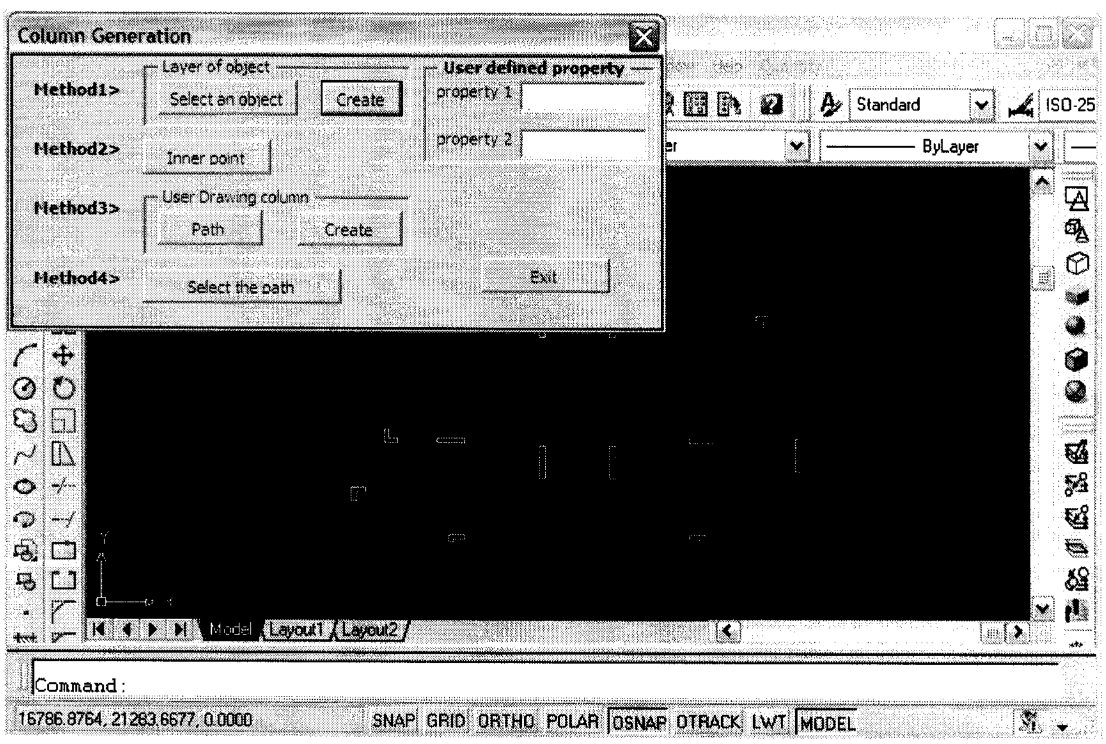


Figure 4.35 The Generated Columns

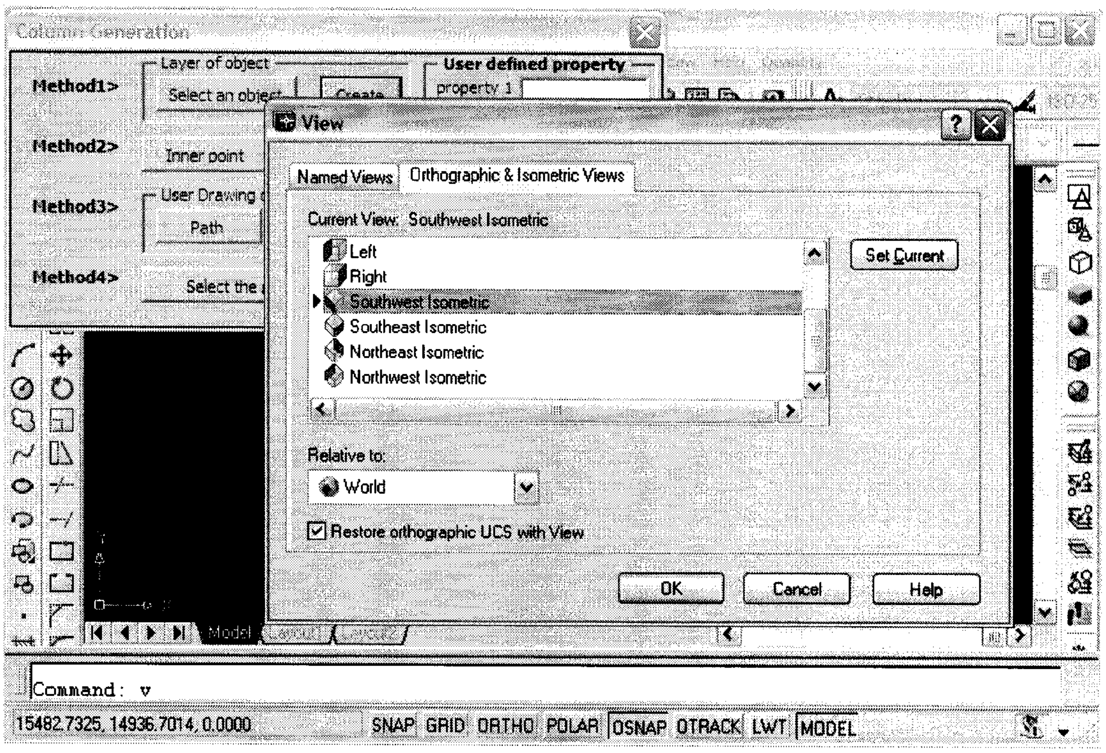


Figure 4.36 3D View Setting

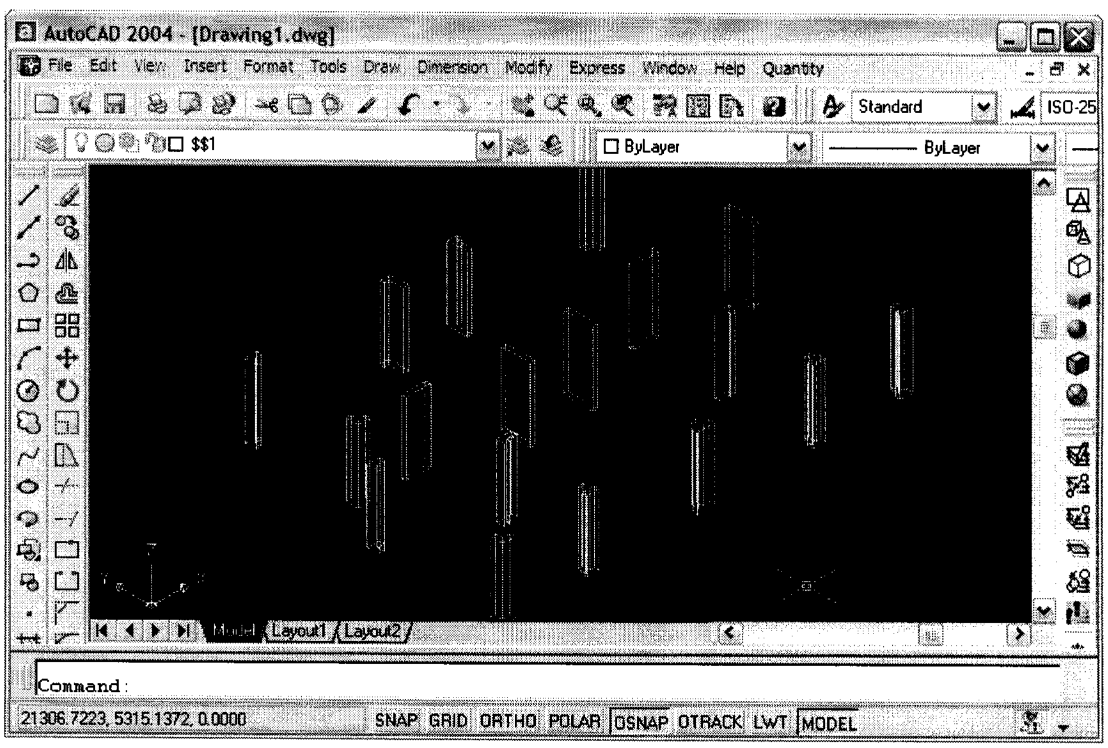


Figure 4.37 3D Wireframe

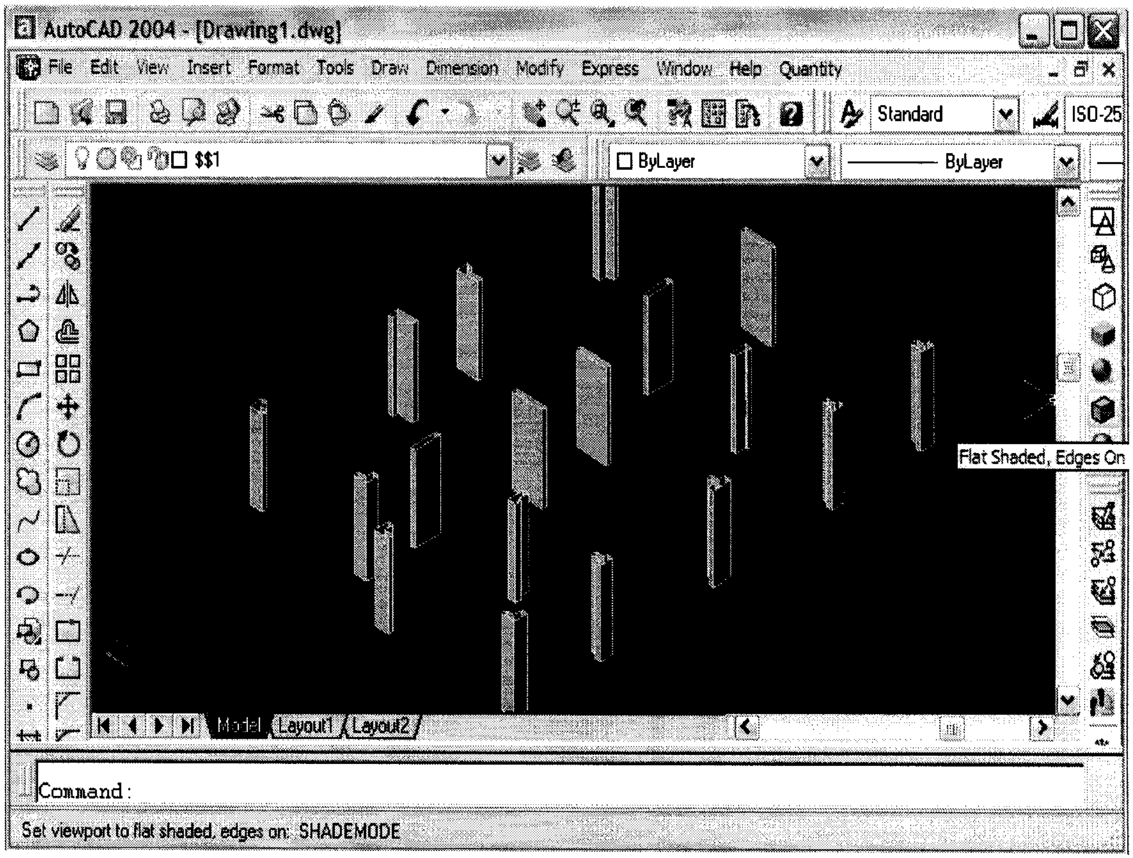


Figure 4.38 Flat Shaded, Edges On 3D View

Method 2: Because a normal column in the drawing should be a closed area, the second method is designed particularly to be applied in such scenarios. By just hitting the "Inner point" button, and then clicking in the closed area (Figure 4.39), a 3D column object will be generated. Similarly to Method 1, successfully generated columns also should turn yellow (Figure 4.40). Figure 4.41 shows some other examples using Method 2.

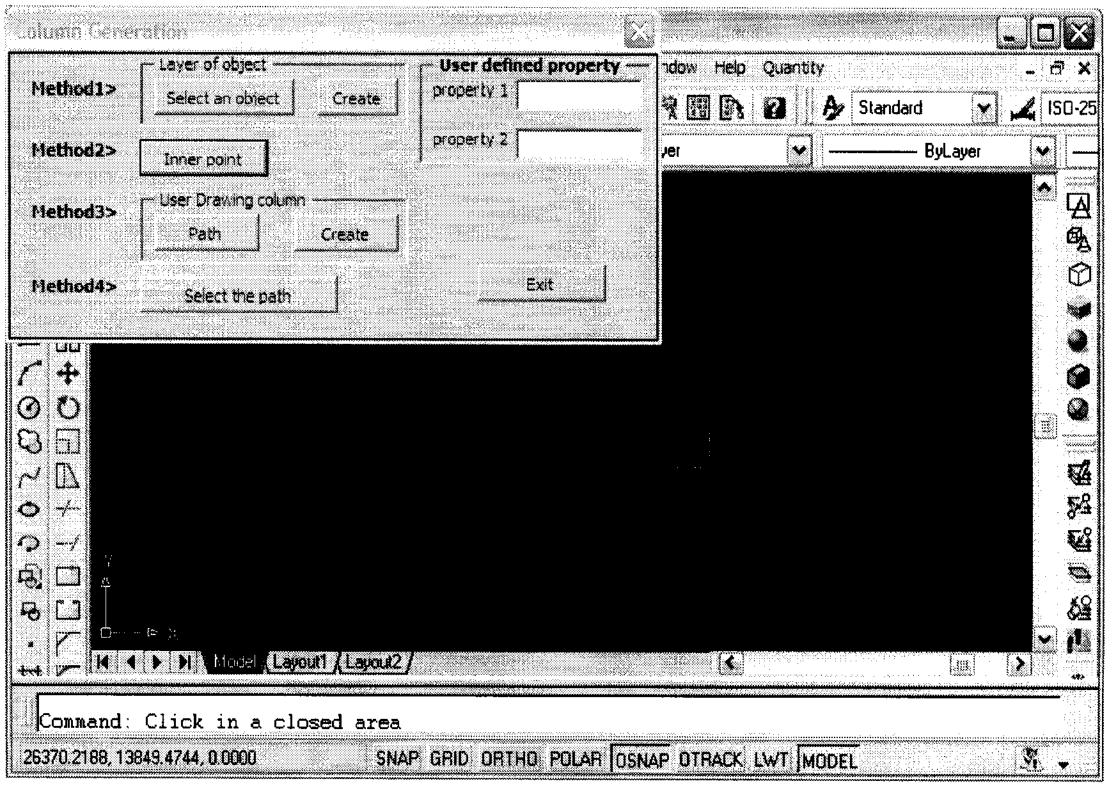


Figure 4.39 Clicking in a Closed Area

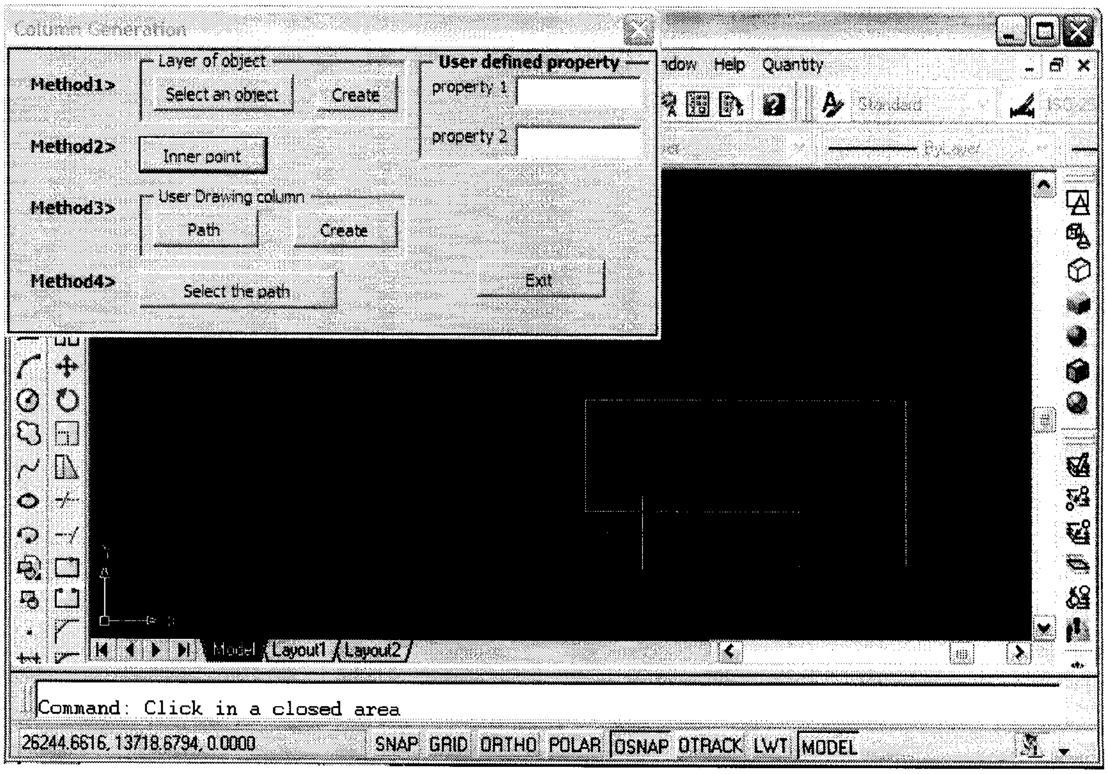


Figure 4.40 Generated Column (Method 2)

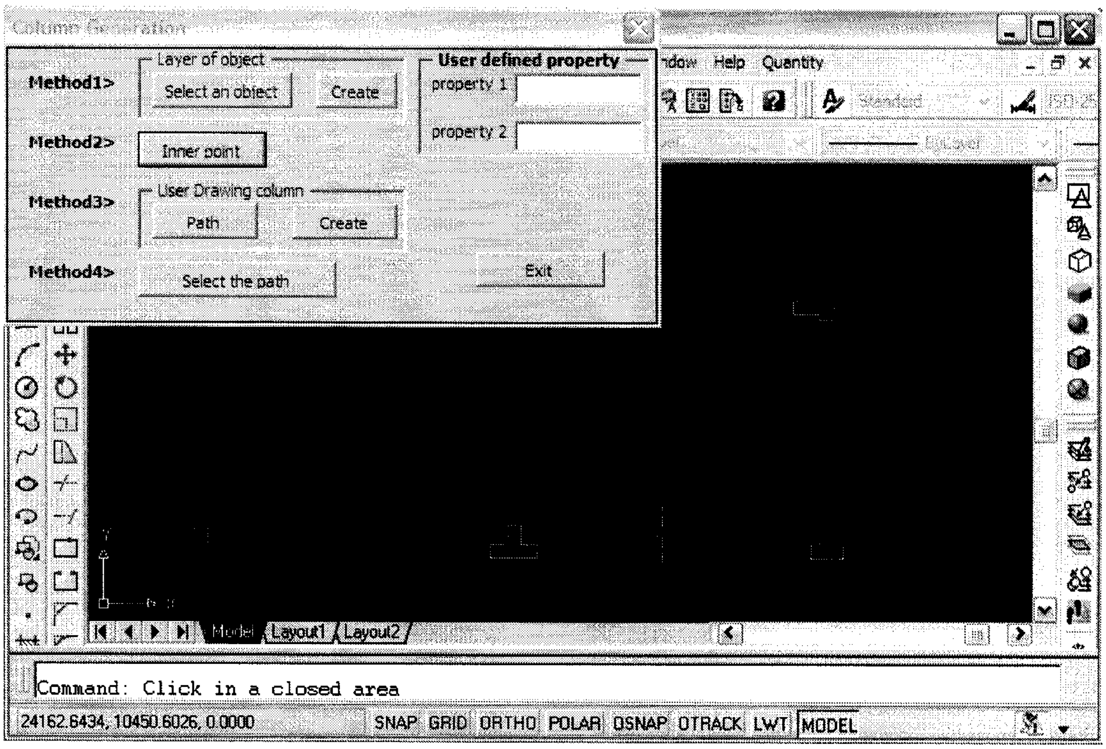


Figure 4.41 Generated Columns (Method 2)

Method 3 and 4: Sometimes the previous two methods can not successfully generate columns, and the problem is usually that the column lines are not closed. To deal with this situation, a self-drawing method is provided. First, the user needs to draw a 2D path (outline) of the column and then click the "Create" button to generate the 3D column. In this way, the user can generate one column at a time (Figures 4.42 and 4.43). Or the user can draw several paths one time, and click the "Select the path" button to select the paths continuously (Figures 4.44 and 4.45).

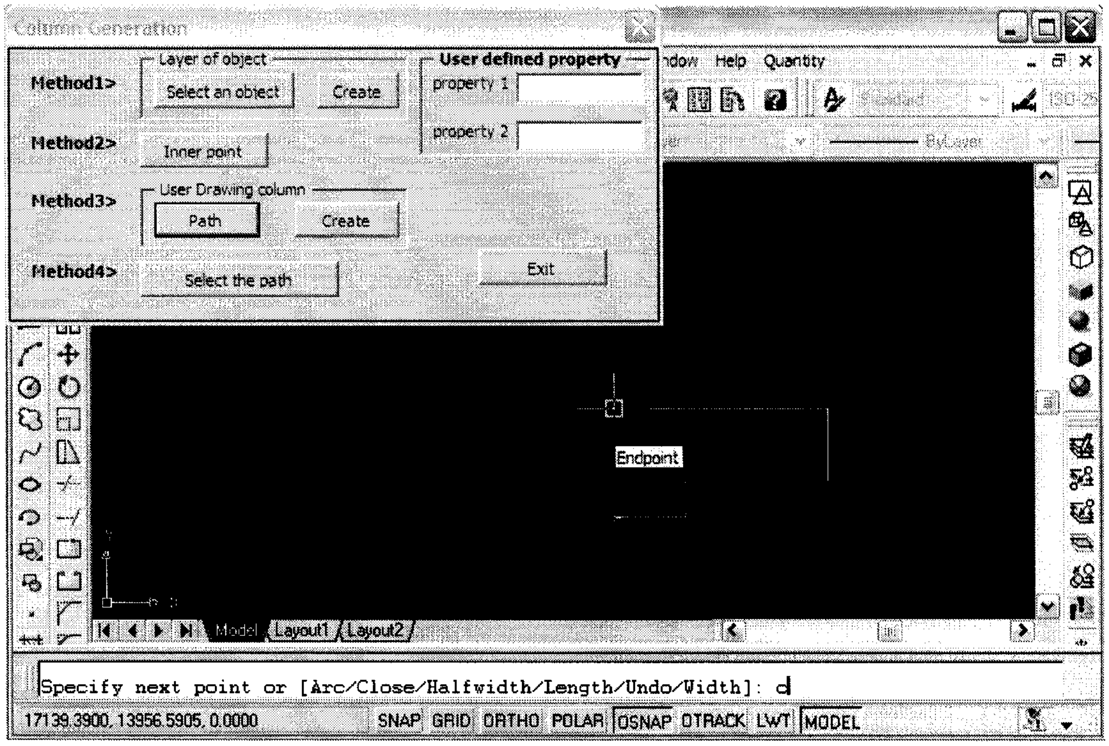


Figure 4.42 User Drawing Path

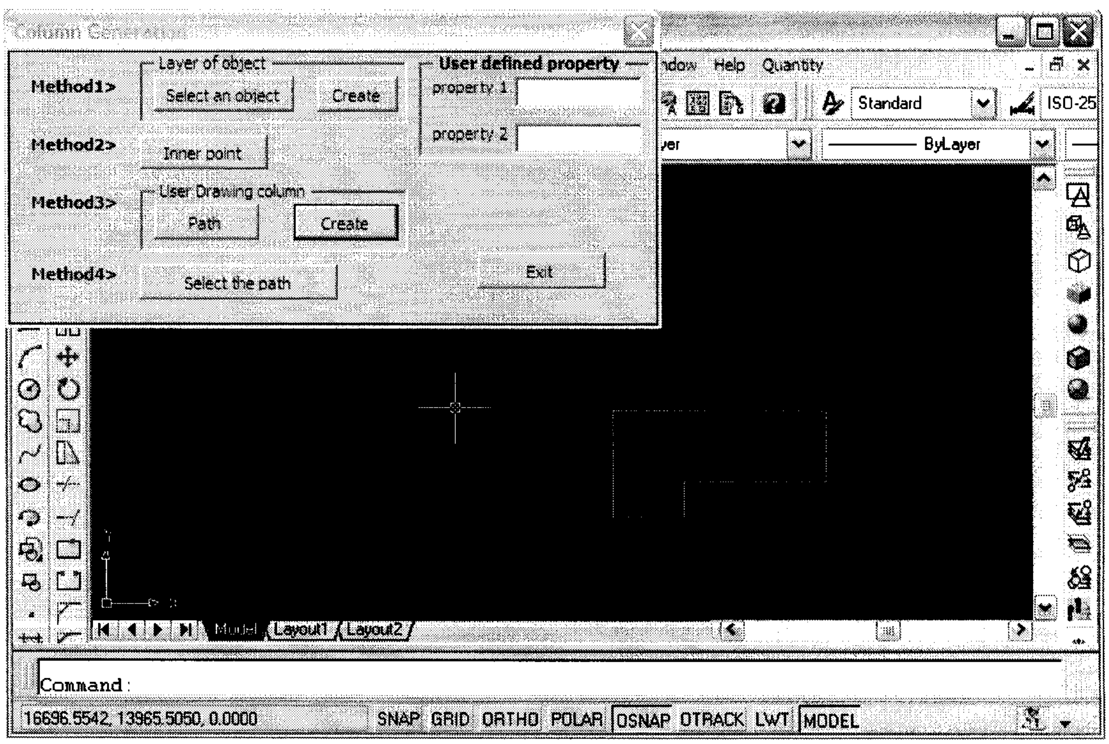


Figure 4.43 After Hitting "Create" button

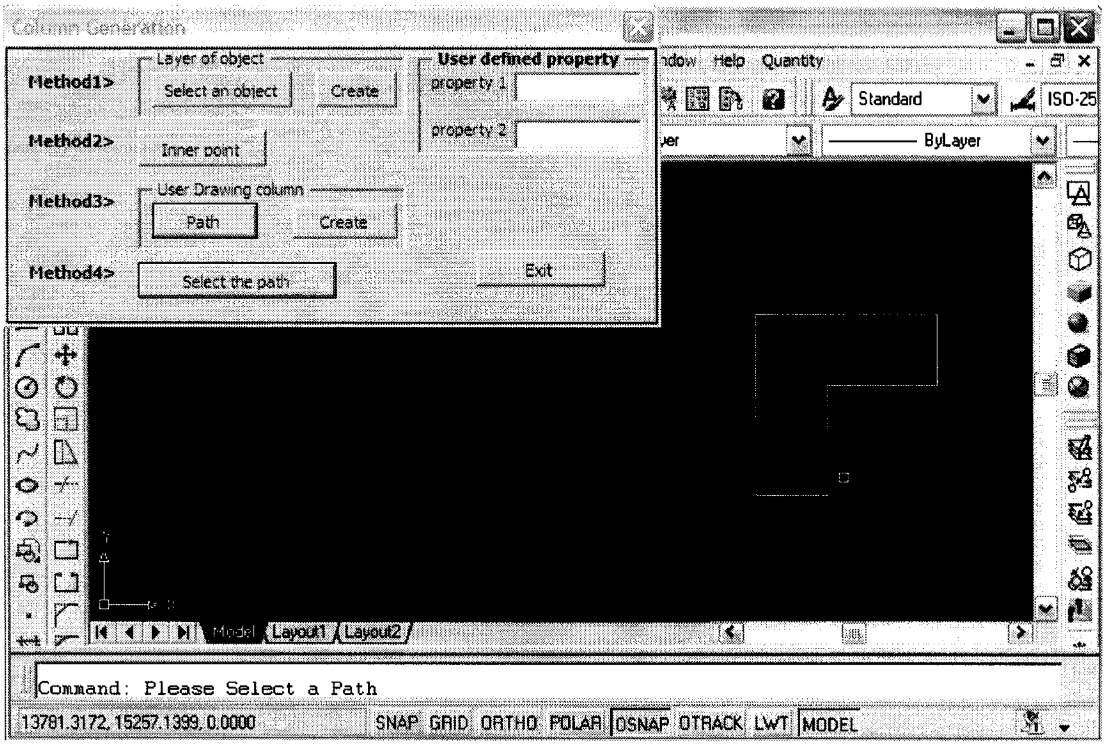


Figure 4.44 Before Selecting the Path

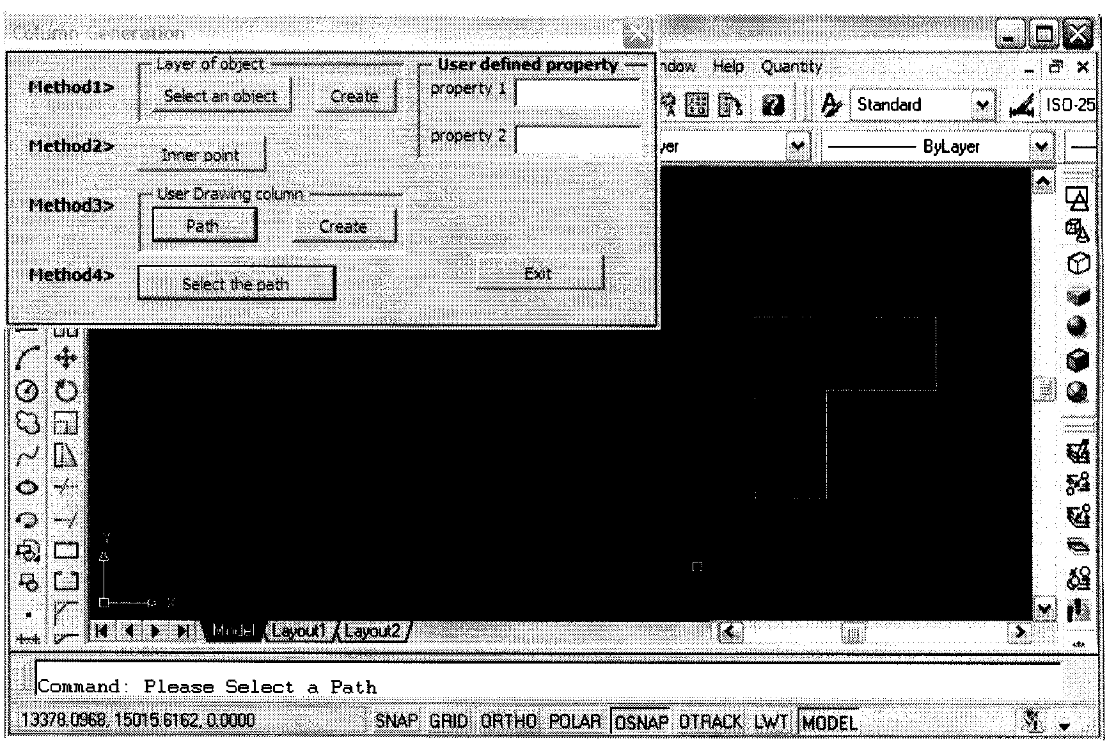


Figure 4.45 After Selecting the Path

For the user's convenience, a very useful and handy tool is provided that the user can use for an instant component information query. In the "Quantity" menu, there is a menu item which is called "Component Query" (Figure 4.46). After selecting this item, the program then will prompt the user to choose a component on the screen (Figure 4.47). Once a valid component object is selected, a form which contains the information of the selected component appears and in the meantime, the selected component is highlighted on screen (Figure 4.48). In this form, the 'primary' information is displayed because it has not taken into account its spatial relationship with other types of components. Detailed information such as formwork and deductions of overlapped areas will be processed in Data Export module. In the final database, all the information of a particular component can be found.

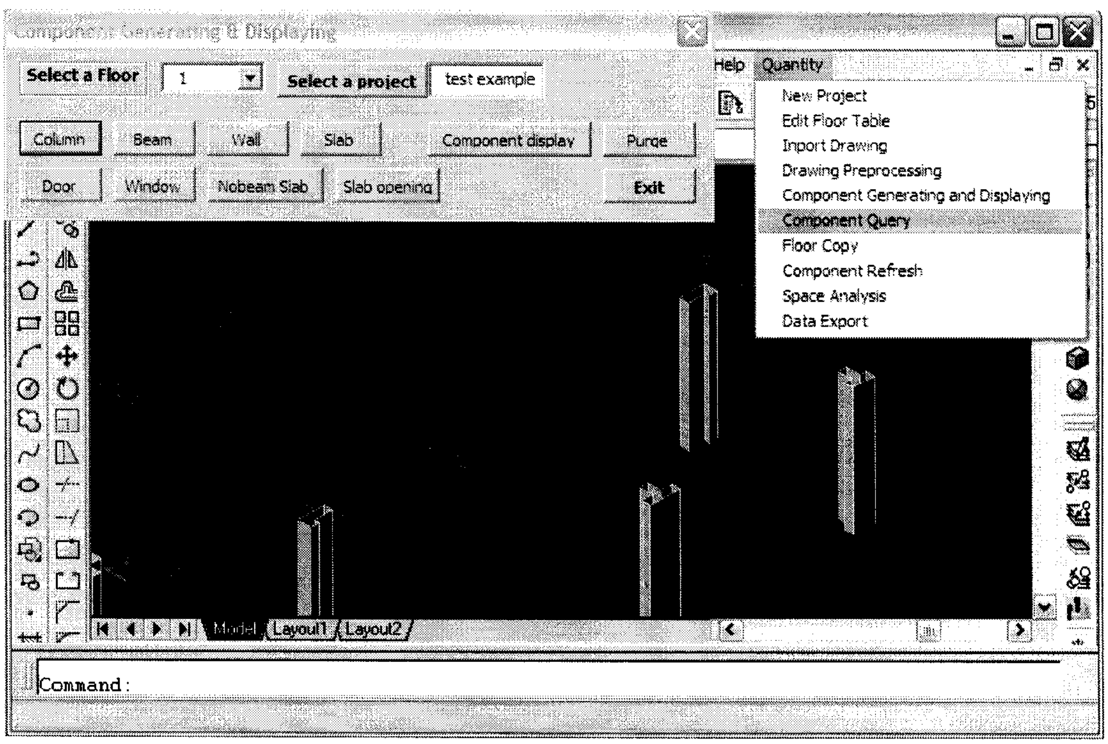


Figure 4.46 Component Query Menu Item

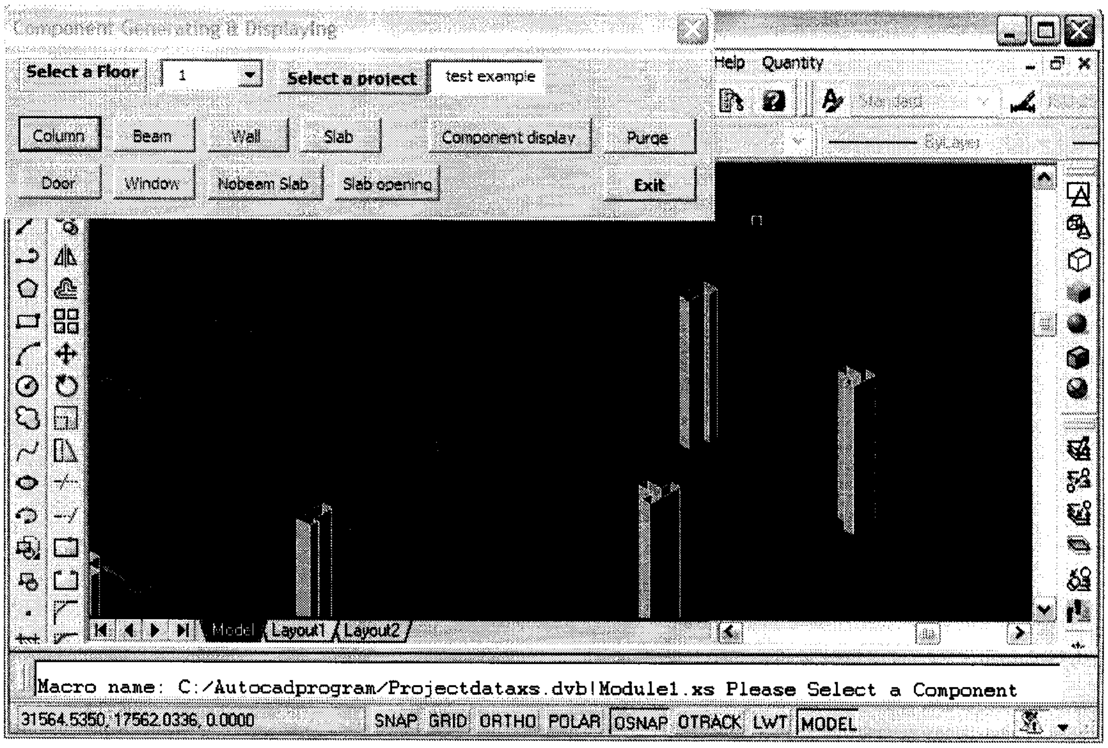


Figure 4.47 Select a Component for Query

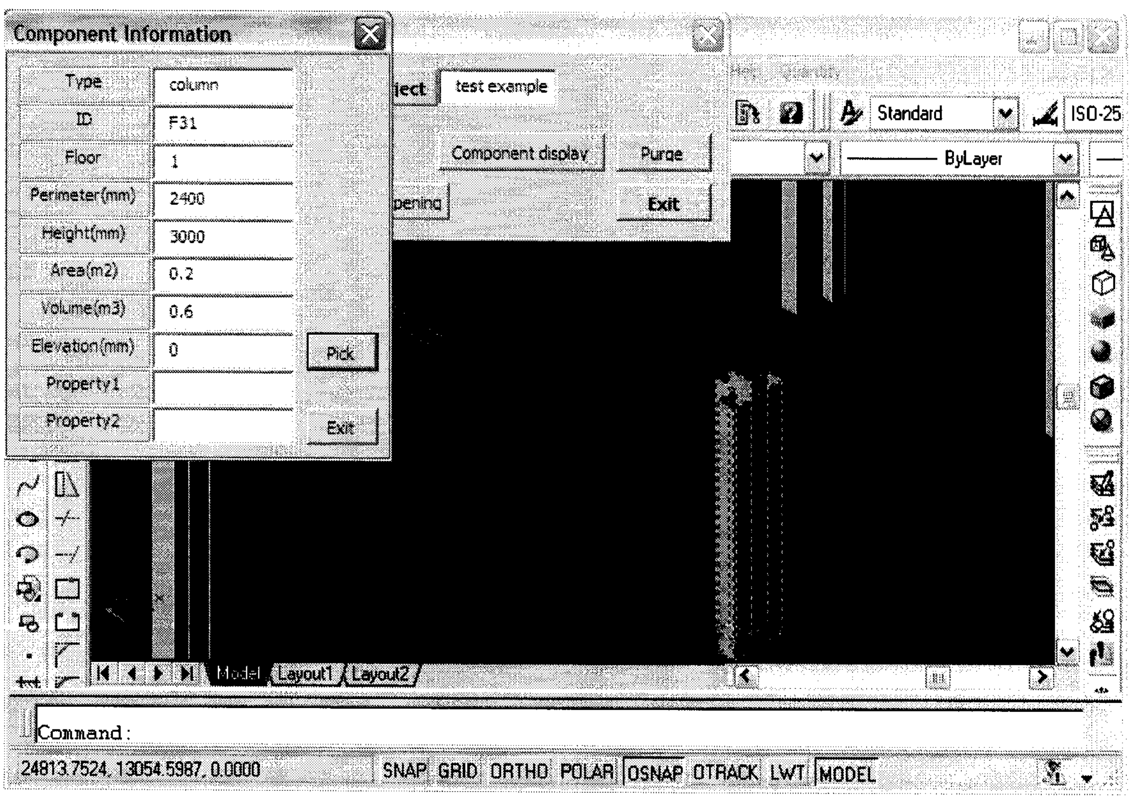


Figure 4.48 The Highlighted Component and its Information

4.6.2 Beam modeling

To continue the modeling process, the structural drawing is needed for beam modeling. Again, the user needs to import the drawing using the "Import Drawing" menu item (Figure 4.49). Once the targeted structure file is selected (Figure 4.50), the file will be inserted into the present drawing (Figure 4.51). Sometimes the inserted drawing does not overlap onto the previously generated components drawing precisely (Figure 4.52); this can be easily solved by calibrating the two drawing using an intersection point on the axis or a certain point of the structure. Here, we used an endpoint of a column for calibration (Figure 4.53). Figure 4.54 gives us another view of 3D after calibrating.

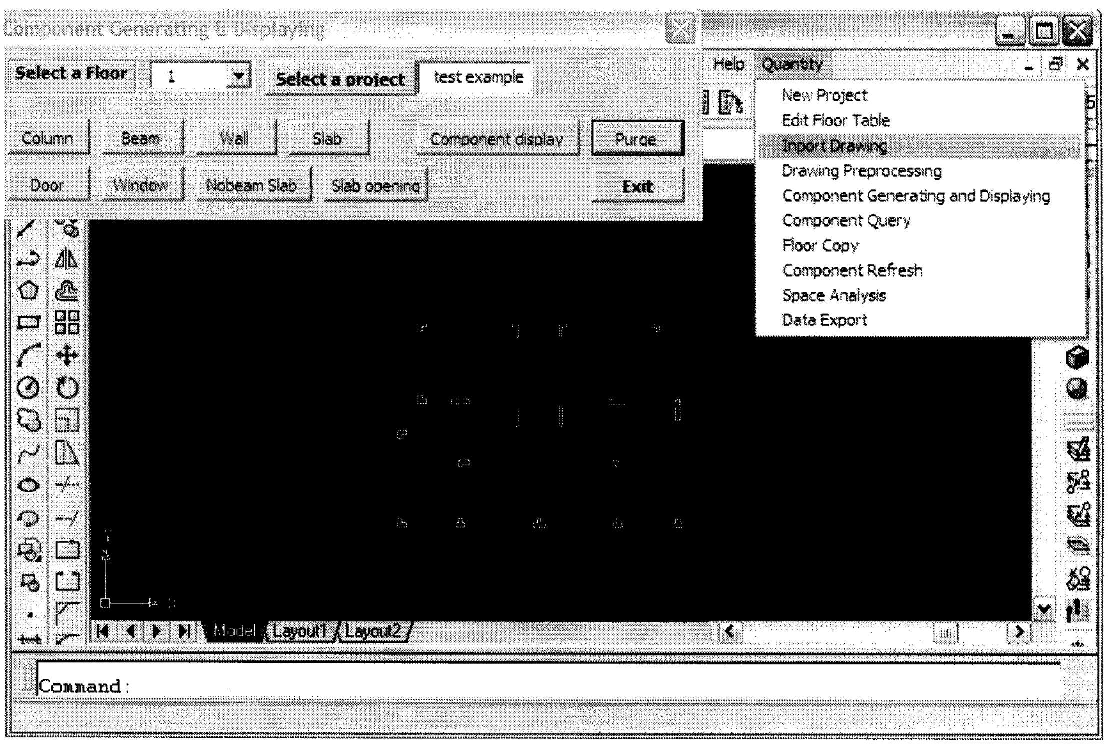


Figure 4.49 Import Structure Drawing

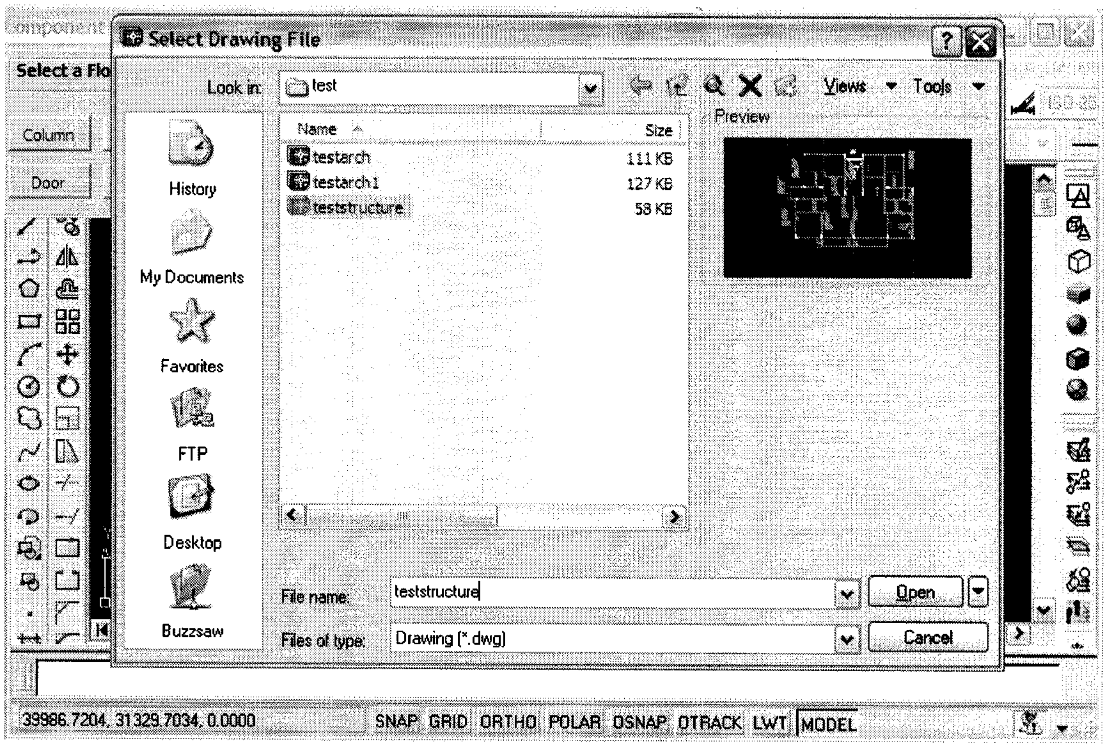


Figure 4.50 Dialog Box for Selecting a Structure Drawing

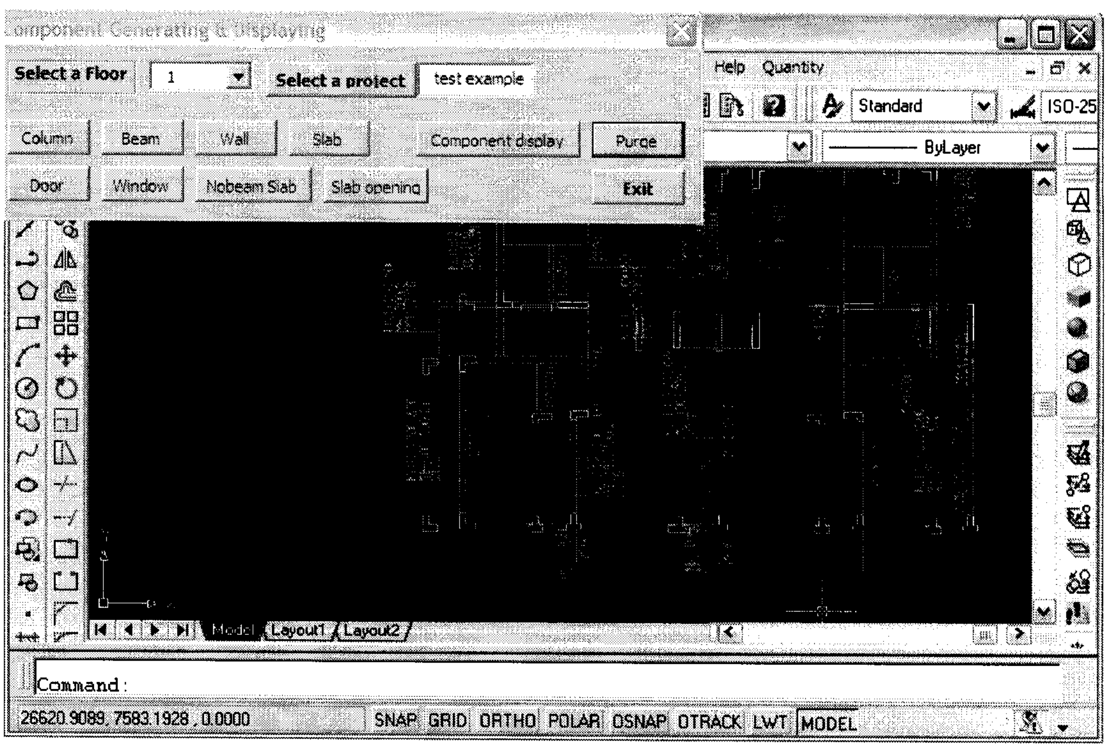


Figure 4.51 Imported Structure Drawing

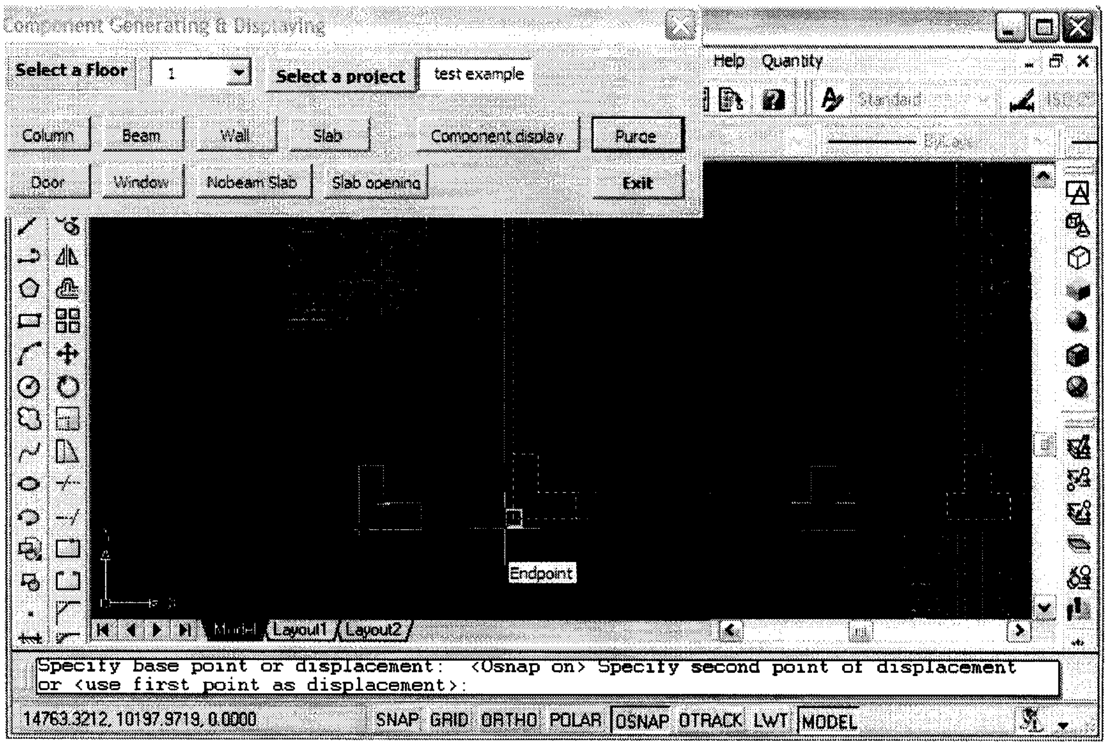


Figure 4.52 Before Calibrating

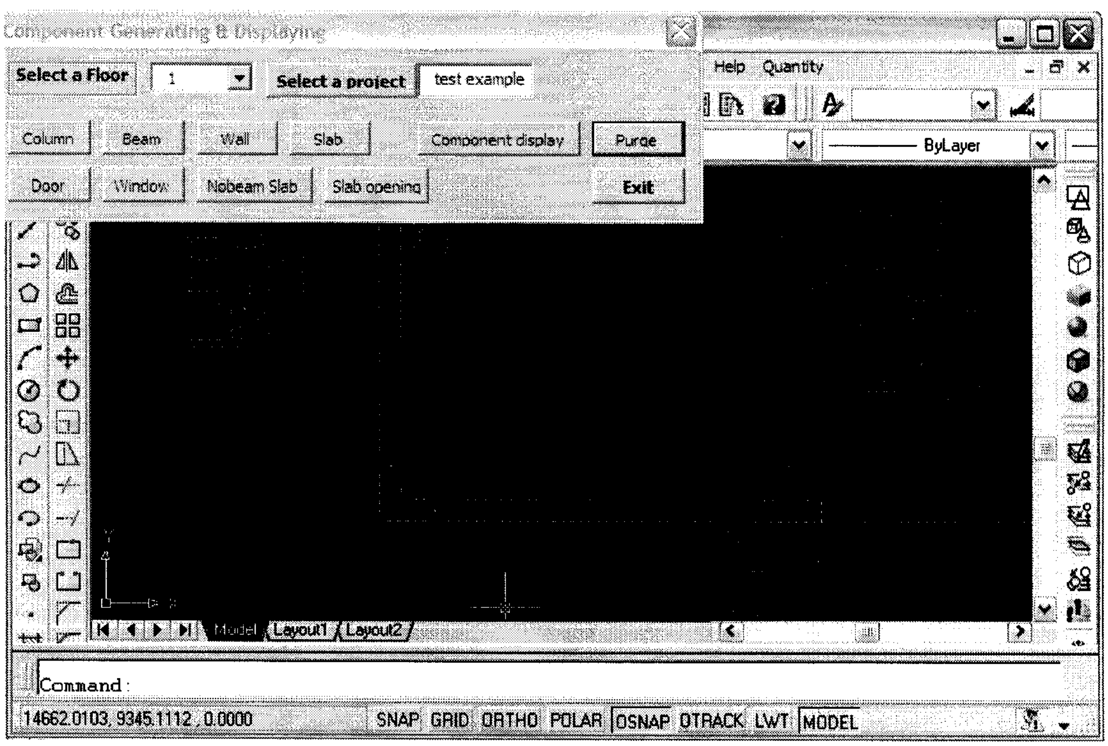


Figure 4.53 After Calibrating

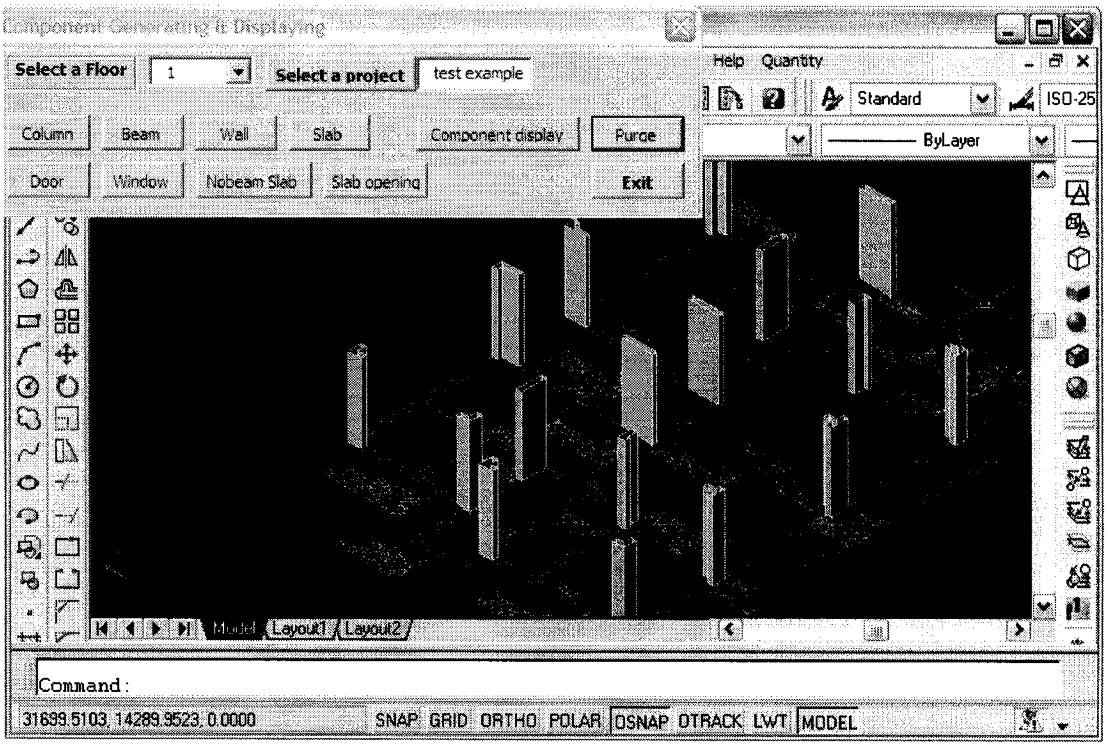


Figure 4.54 3D View After Calibrating

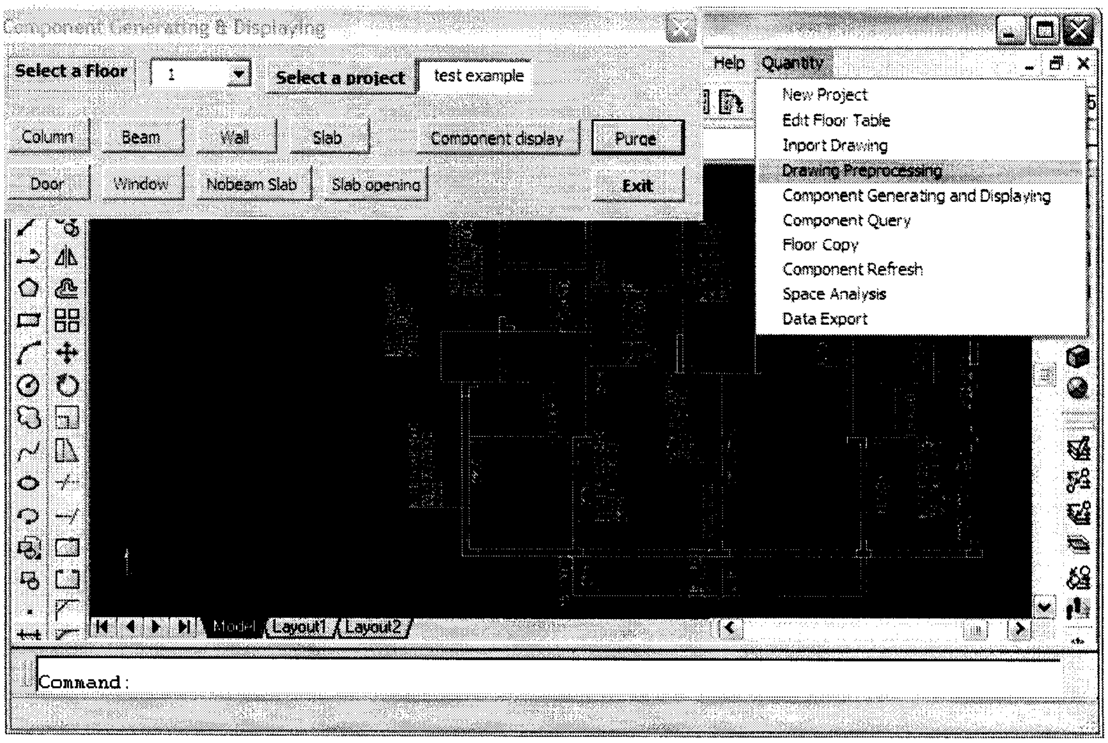


Figure 4.55 Preprocessing Structural Drawing

The interface of beam modeling is shown in Figure 4.56. Three methods are provided for the modeling.

Method 1: This method requires the user to input the height (mm) of the beam which is going to be modeled; the width is not needed (Figure 4.57). When the "Pick a beam line" button is clicked, the user is prompt to select one line of a particular beam (Figure 4.58). Because a beam consists of two lines, the program will search the other line of the beam and retrieve the width of the beam automatically; and then finish the modeling. The selected beam should then turn red (Figures 4.59 and 4.60). This method for beam modeling is fast, but there is one requirement: the physical width of the beam in the original drawing must be equal to its dimension in the drawing.

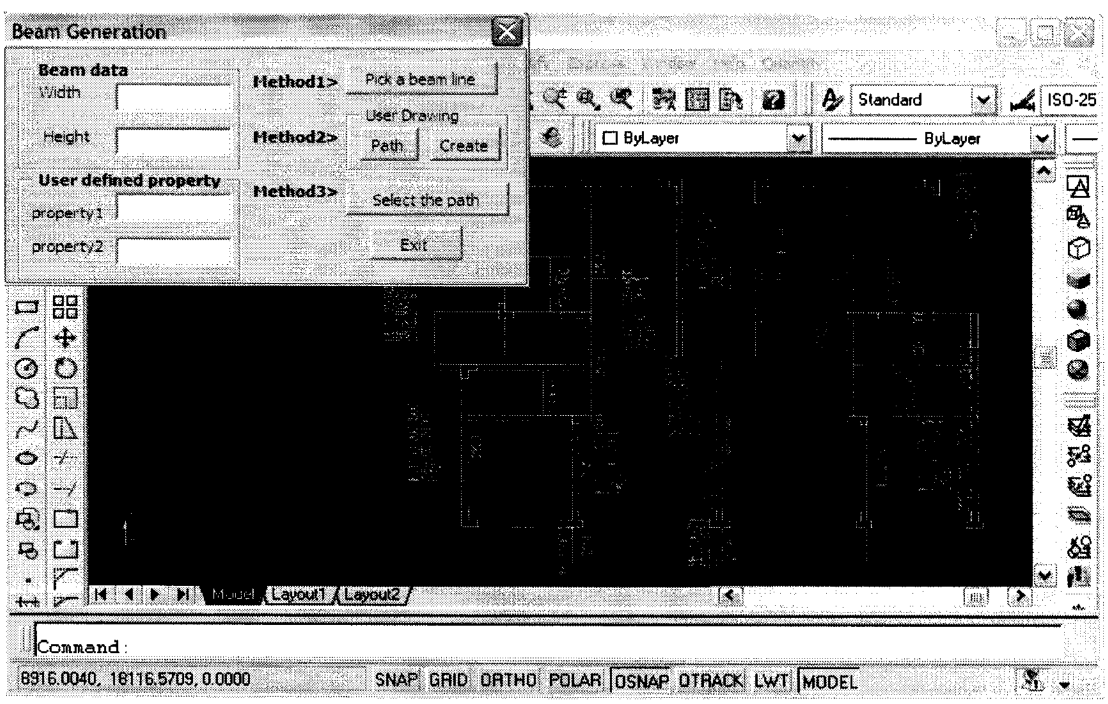


Figure 4.56 The Interface of Beam Modeling

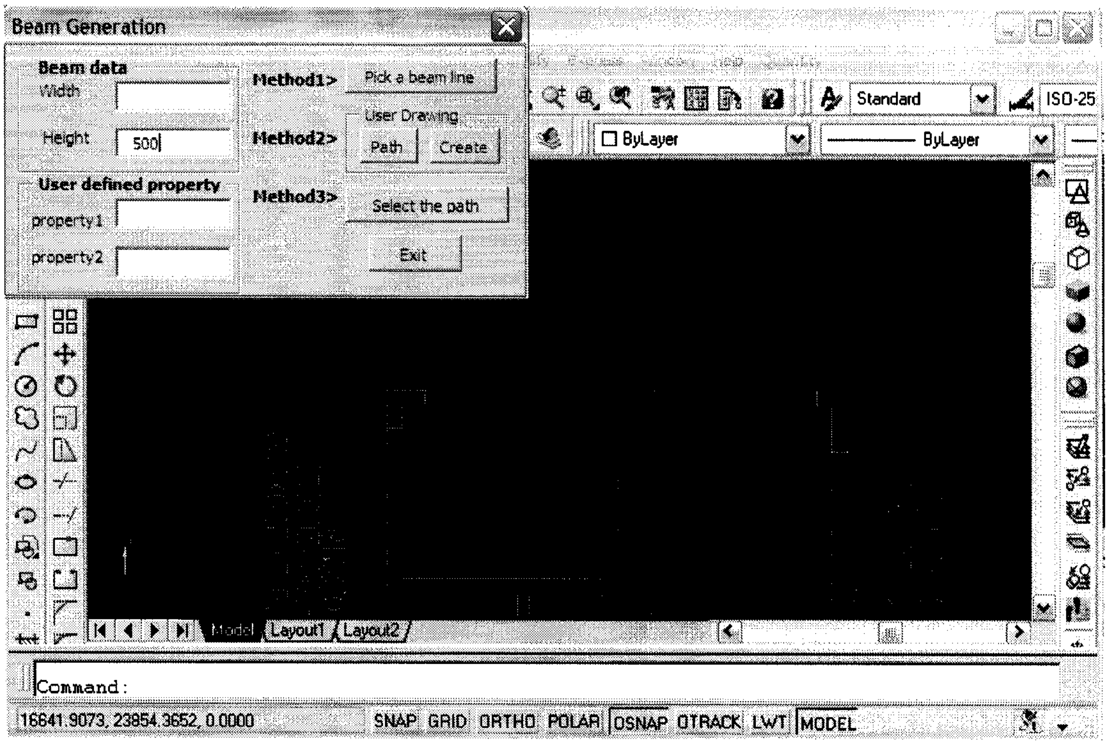


Figure 4.57 Input Beam Height

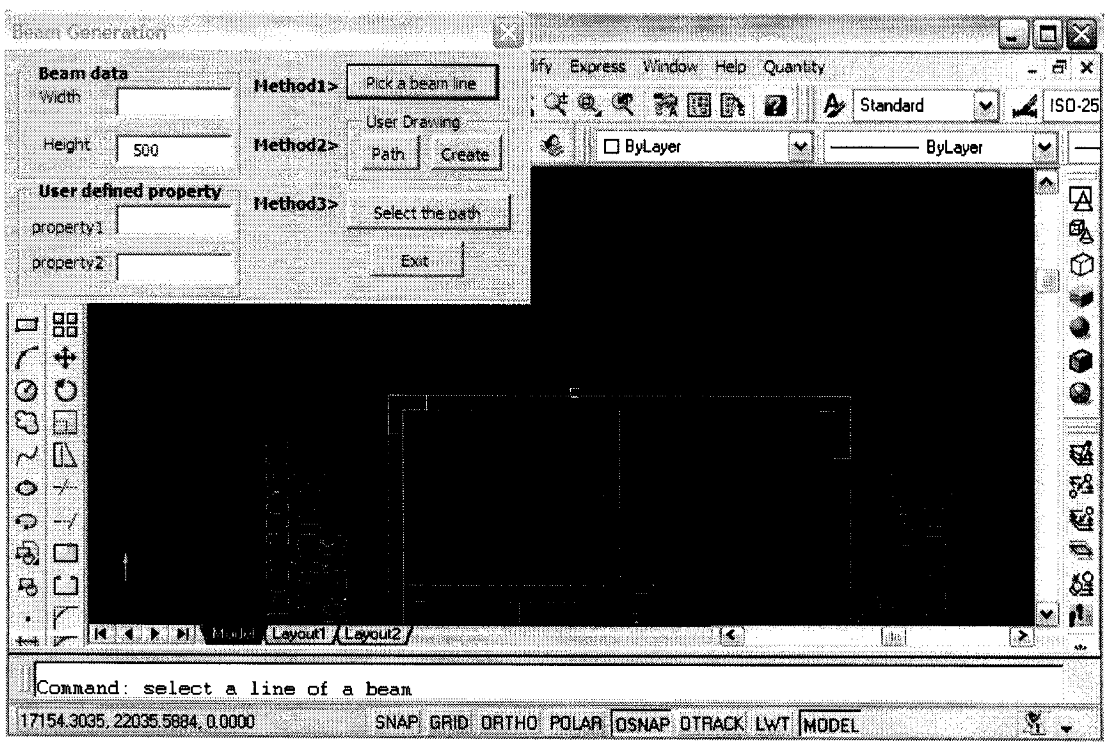


Figure 4.58 Selecting One of the Two Beam Lines

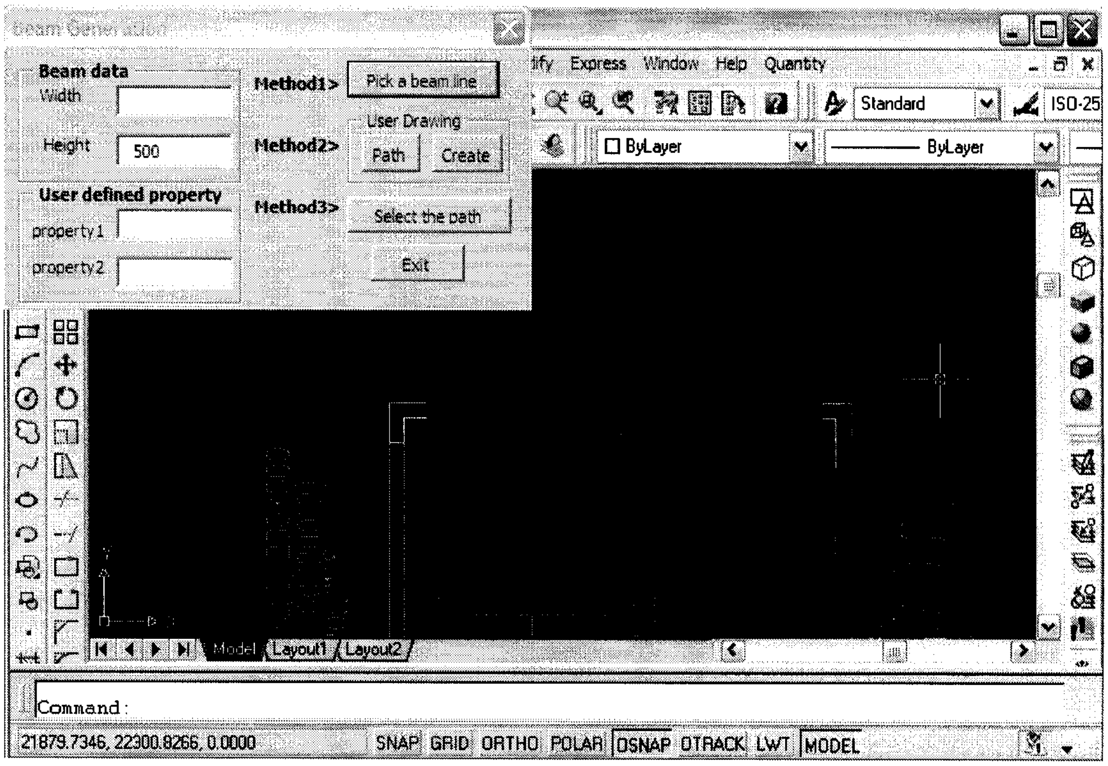


Figure 4.59 The Generated Beam

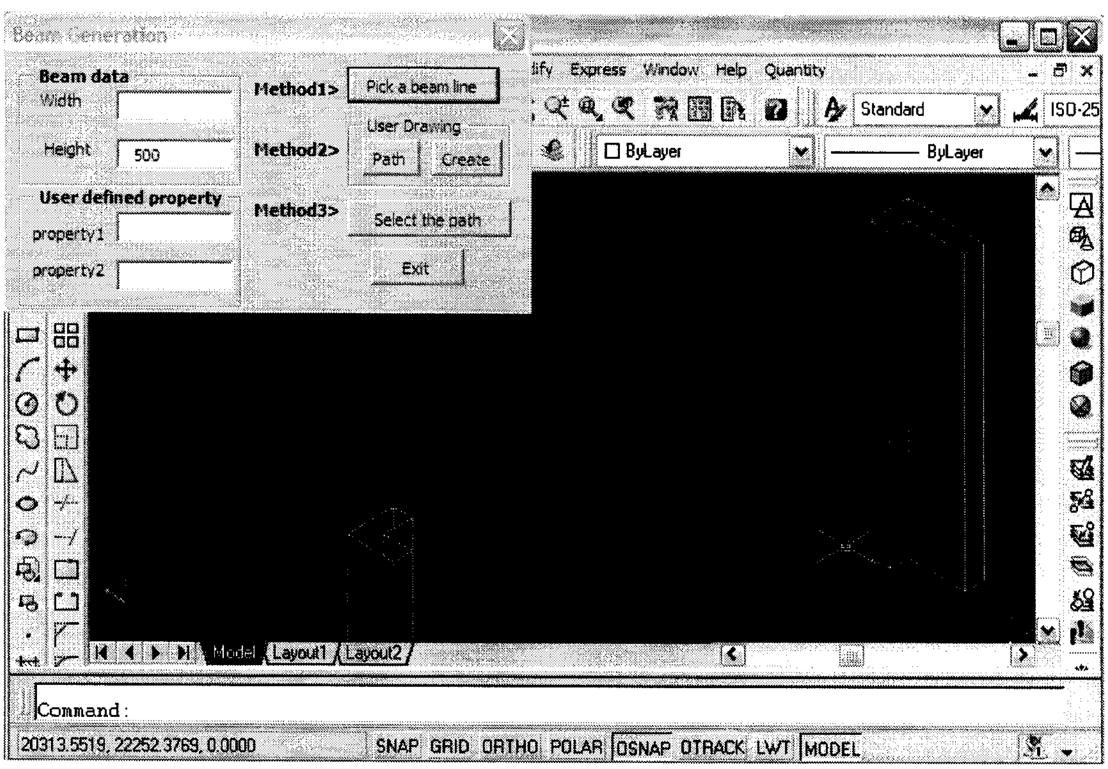


Figure 4.60 The 3D View of Generated Beam

Method 2: This method requires that the user input the width and height of the targeted beam according to the imported structure drawing and click the "Path" button, which draws the path of the beam (Figure 4.61). Finally, the user clicks the "Create" button to finish the modeling (Figures 4.62 and 4.63).

Method 3: If some beams' share a cross section (width and height), the user can draw the paths of these beams first (Figure 4.64), and then select the paths one by one to finish the modeling of these beams (see Figures 4.65 - 4.67).

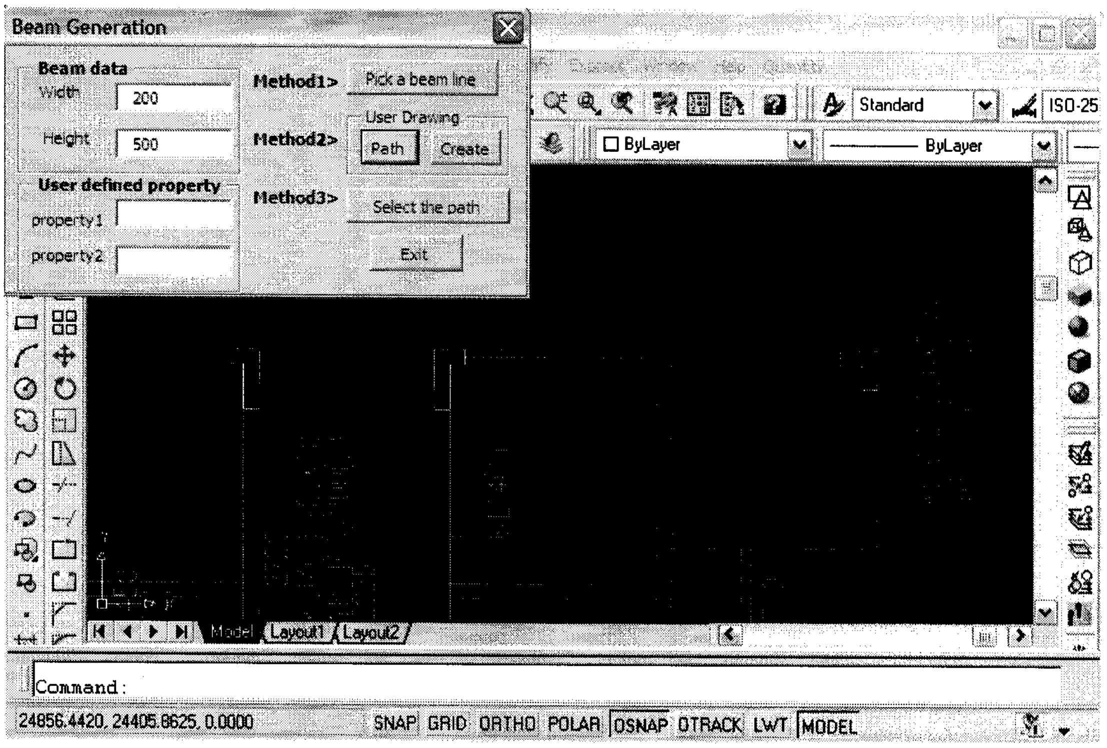


Figure 4.61 Beam Data Input & Path Drawing

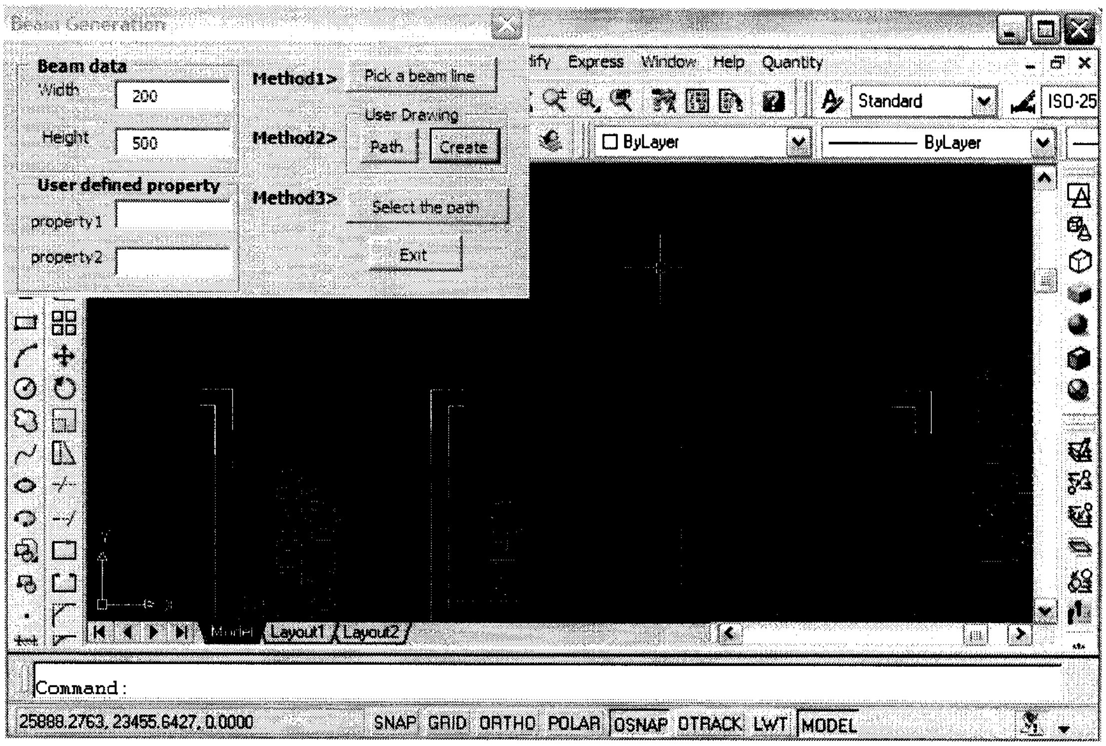


Figure 4.62 Result of Method 2

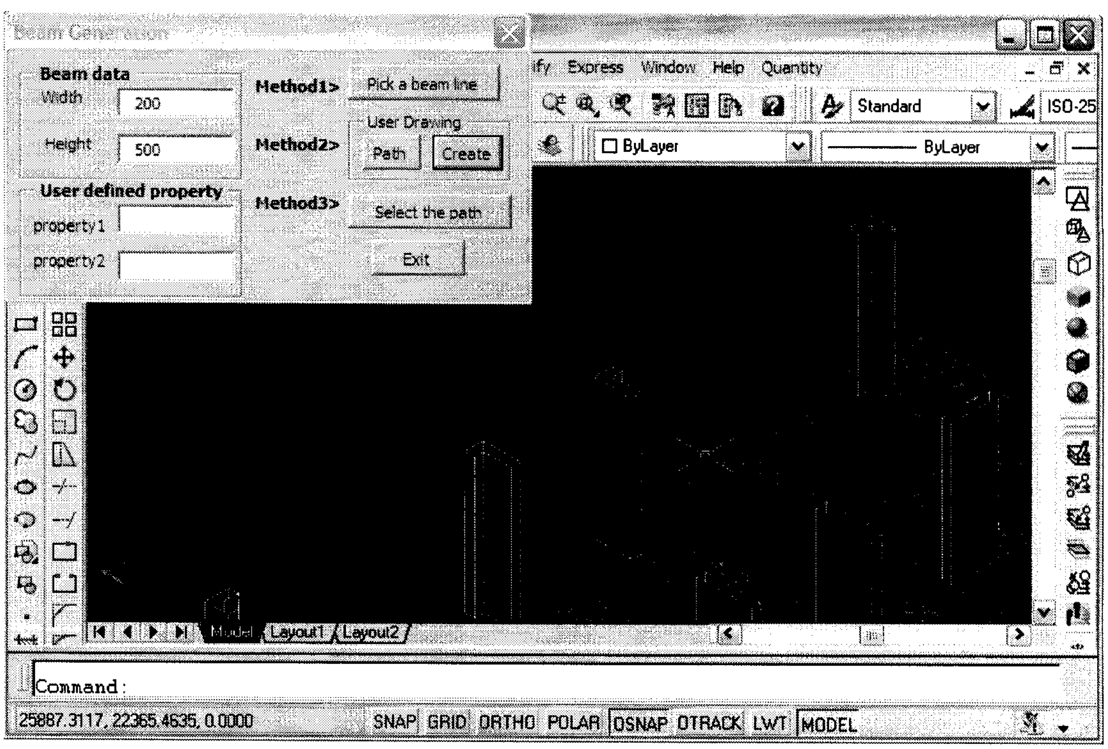


Figure 4.63 Result of Method 2 (3D)

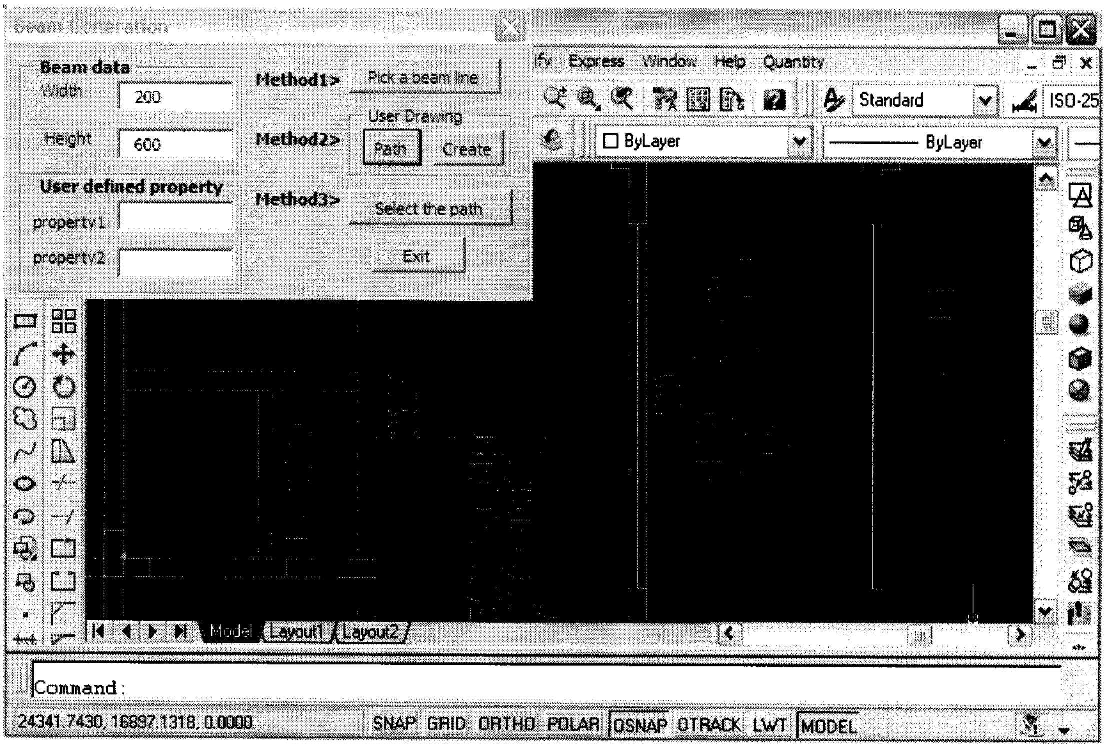


Figure 4.64 Drawing Two Paths

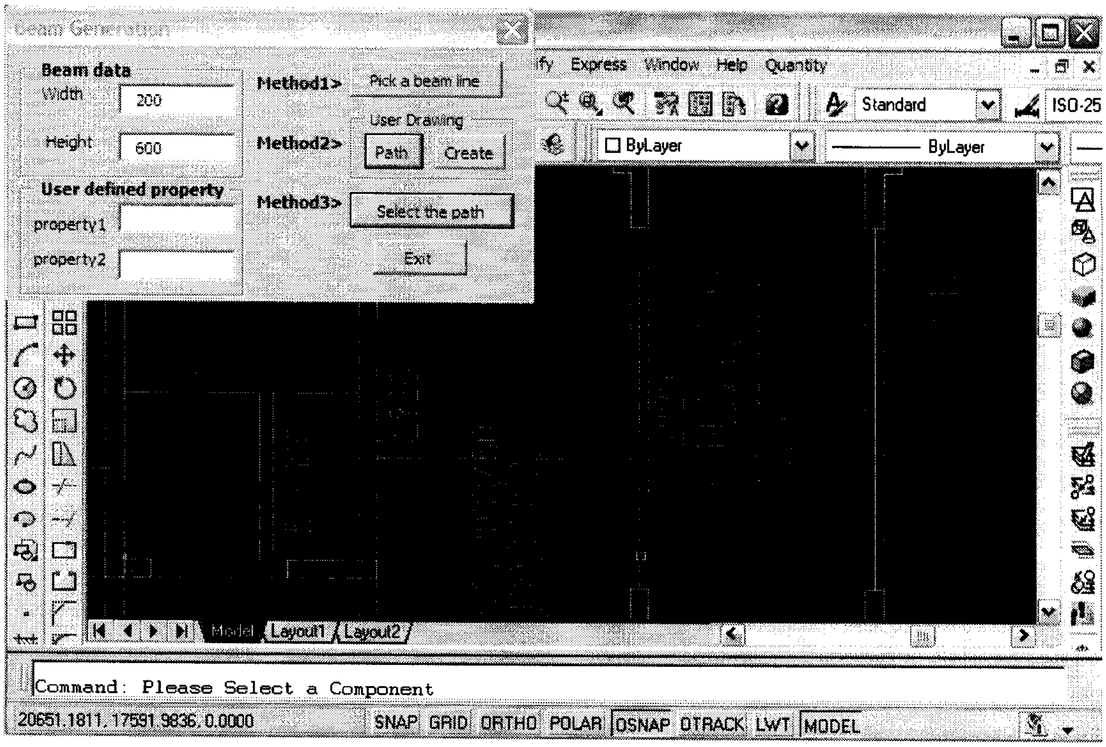


Figure 4.65 Selecting Two Paths

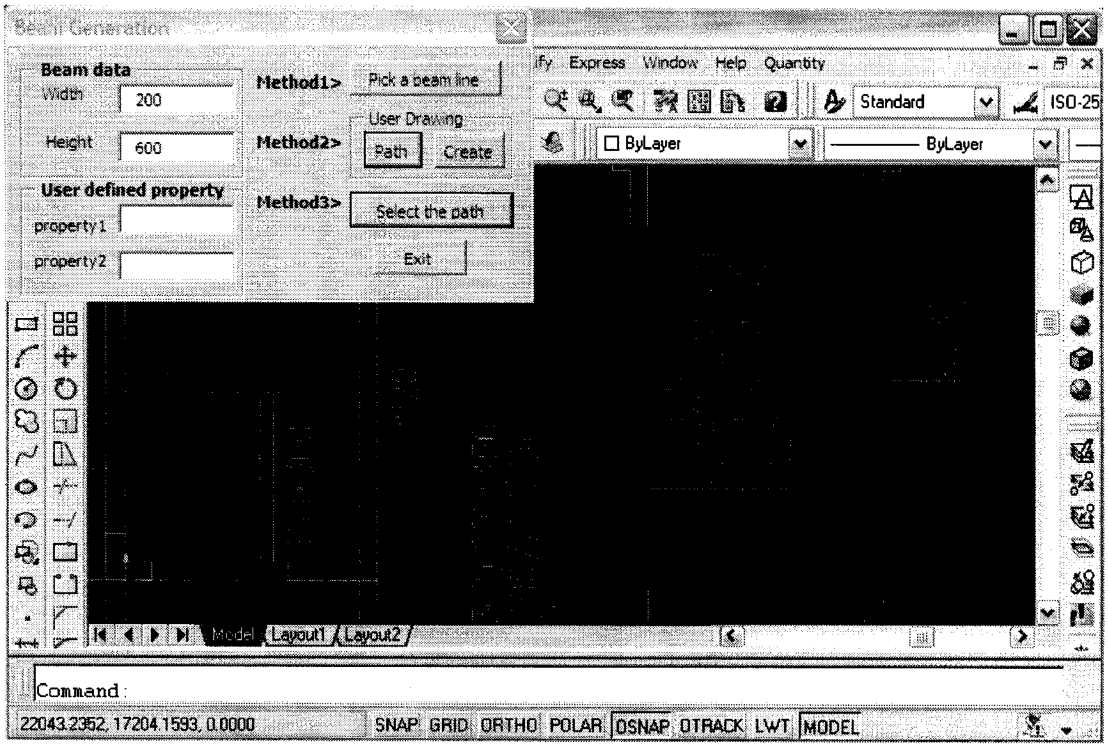


Figure 4.66 Result of Method 3

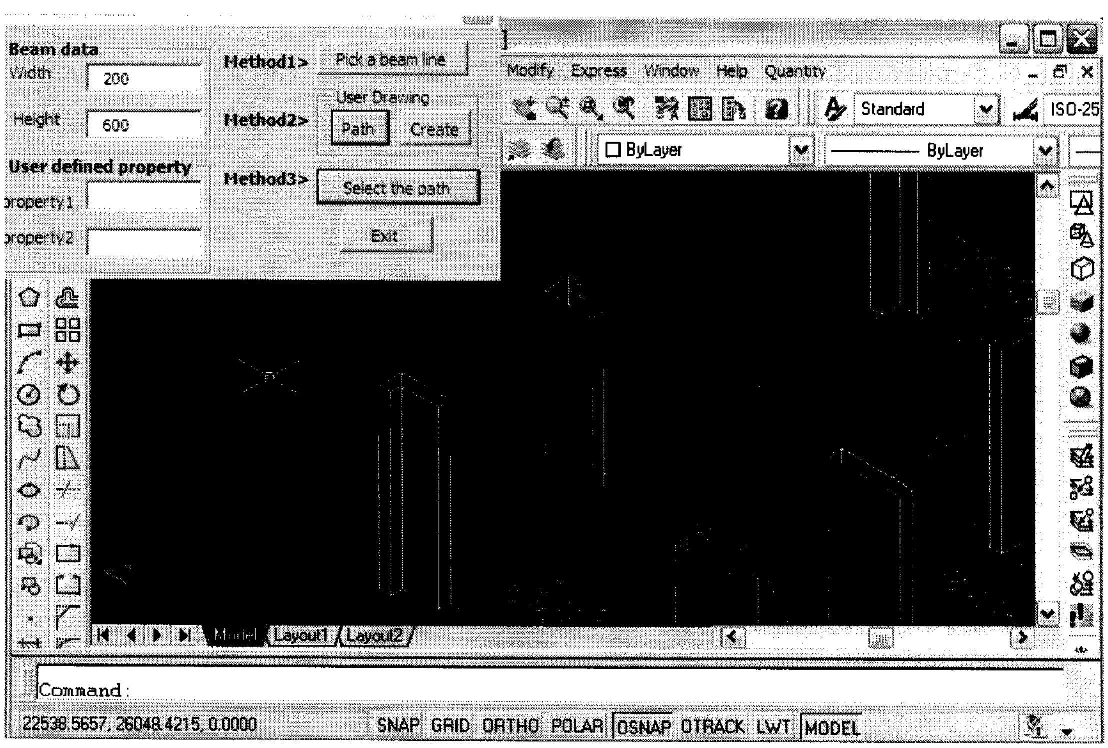


Figure 4.67 Result of Method 3 (3D)

To query the information of a certain beam, the menu item "Component Query" can be used (Figure 4.68).

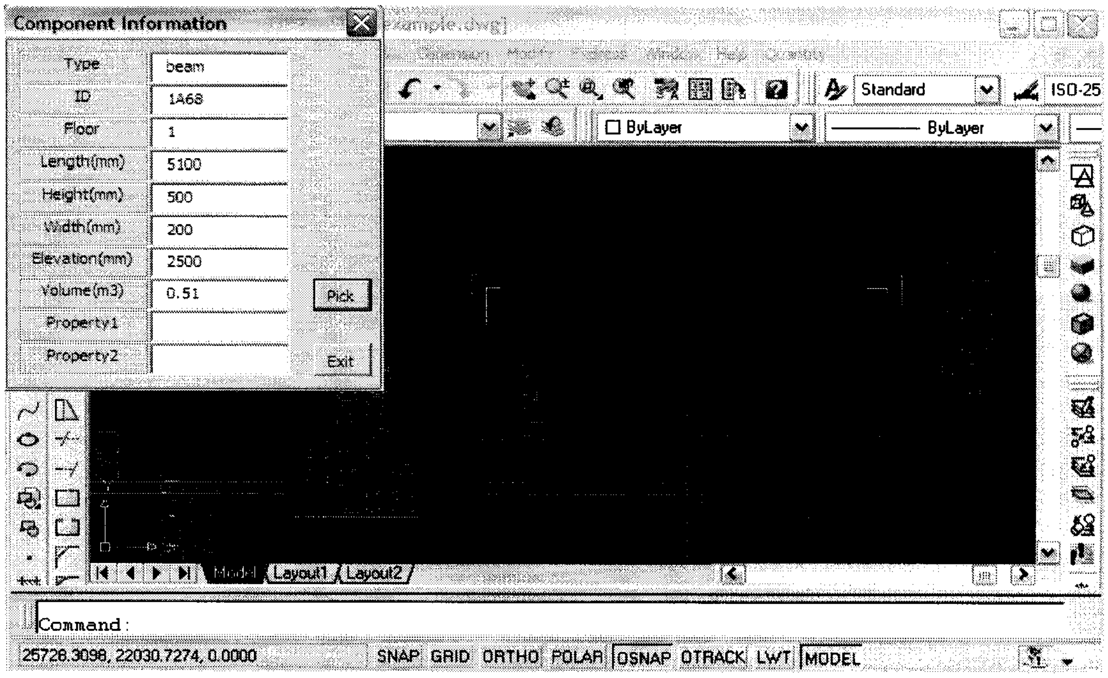


Figure 4.68 Beam Query

Figure 4.69 3D View of Columns and Beams

4.6.3 Slab Modeling

After the modeling of columns and beams, the user might feel that the screen is confusing because there are so many objects compiled on the screen (Figure 4.70). Thus, a cleaning tool is included to purge the imported drawing objects that are not going to be used any more. By clicking the "Purge" button, a neater view will be presented to the user – only the generated 3D model objects are left on screen (Figure 4.71).

There are three methods provided for slab modeling. Before modeling, the slab thickness input is required (Figure 4.72).

Method 1: Since the slabs are normally built within the boundaries of columns and beams, this method makes full use of the boundaries. When the user clicks the "Pick in a closed area" button, all the boundaries of beams appear in cyan (Figure 4.73). Each click in a closed area will create one slab (Figure 4.74).

Method 2 and 3: Figures 4.75 through Figure 4.79 illustrate the other two methods provided for slab modeling. Similar to column modeling, the path should be a closed area.

As shown in Figures 4.80 and 4.81, a query is also available for slabs.

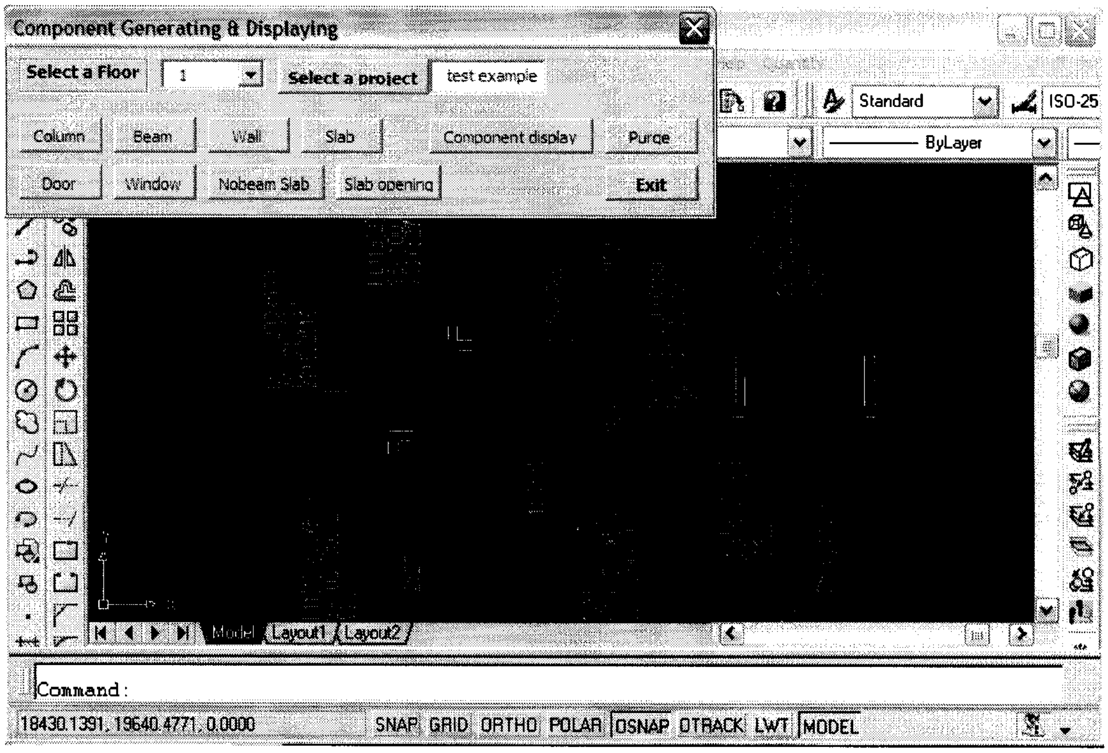


Figure 4.70 Screen Shot before Purging

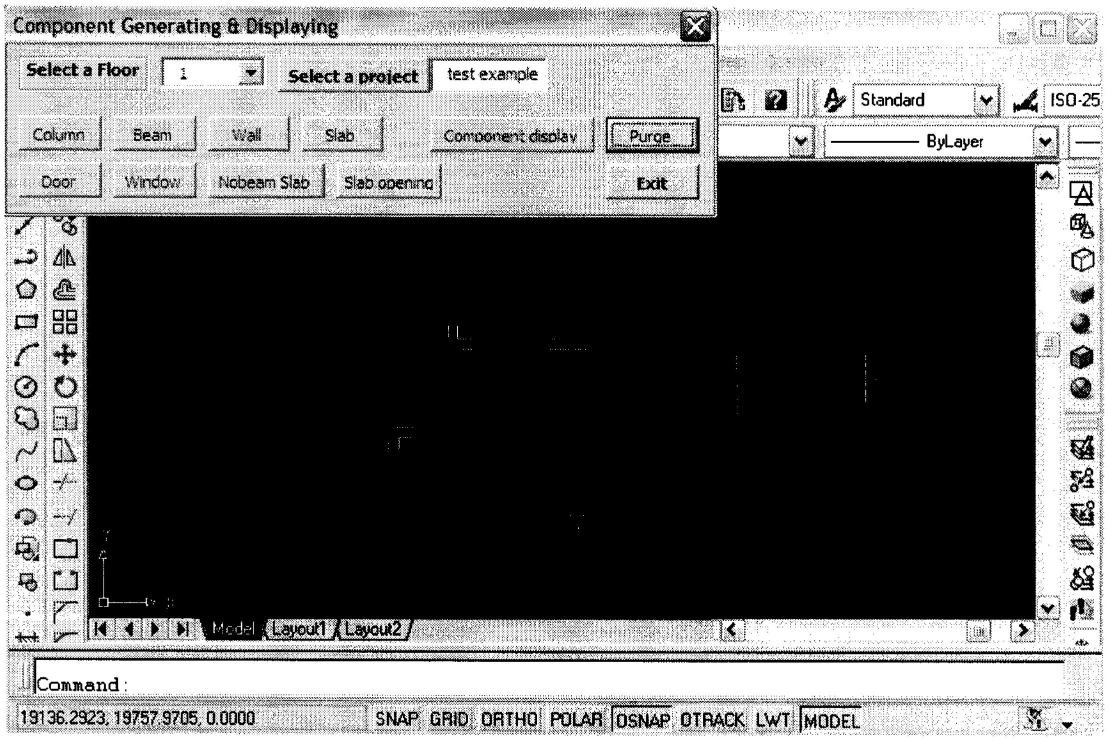


Figure 4.71 Screen Shot after Purging

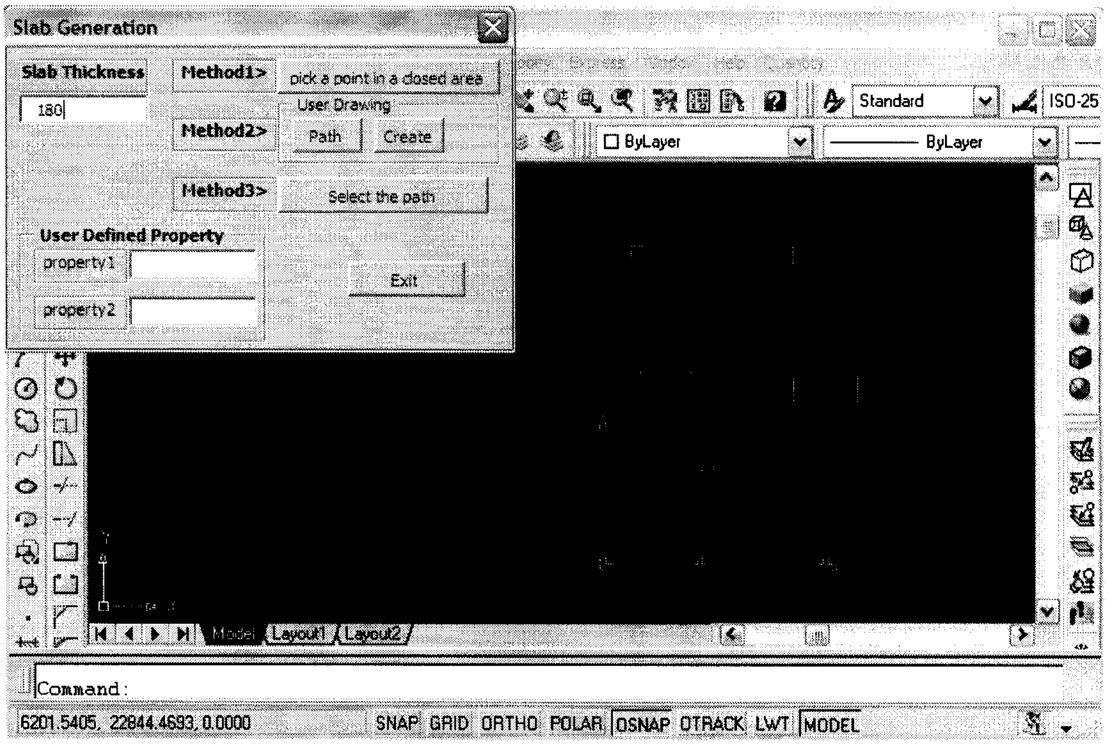


Figure 4.72 Slab Modeling Interface

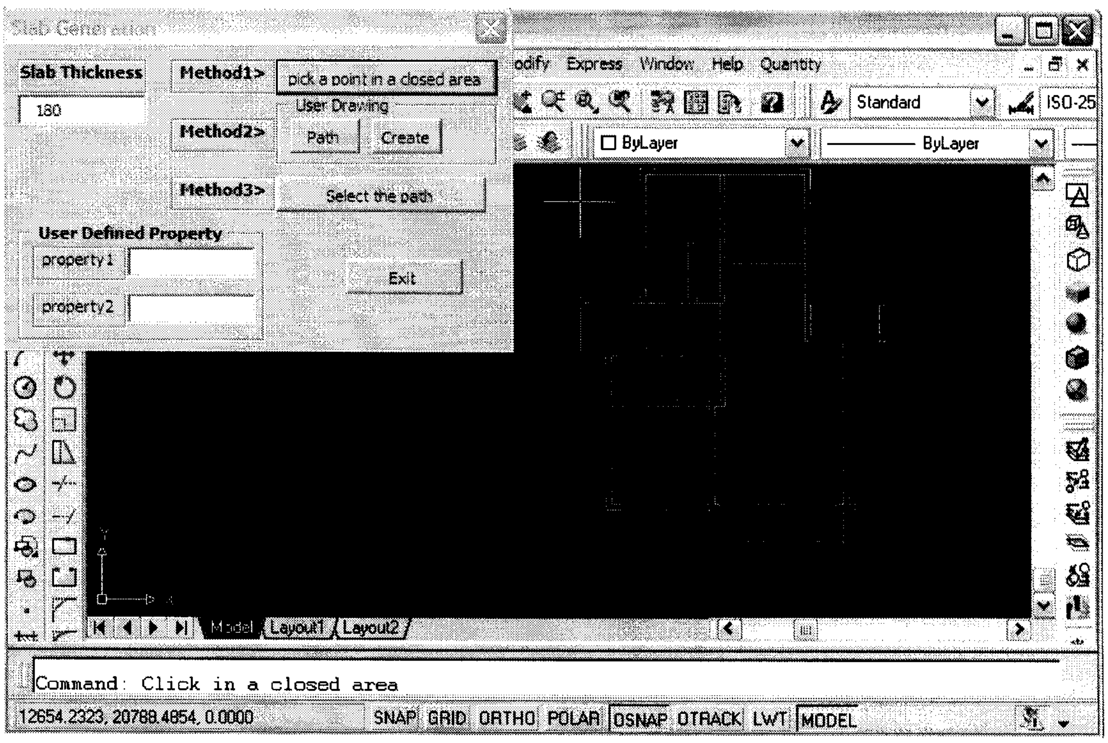


Figure 4.73 Displaying the Beam Boundaries

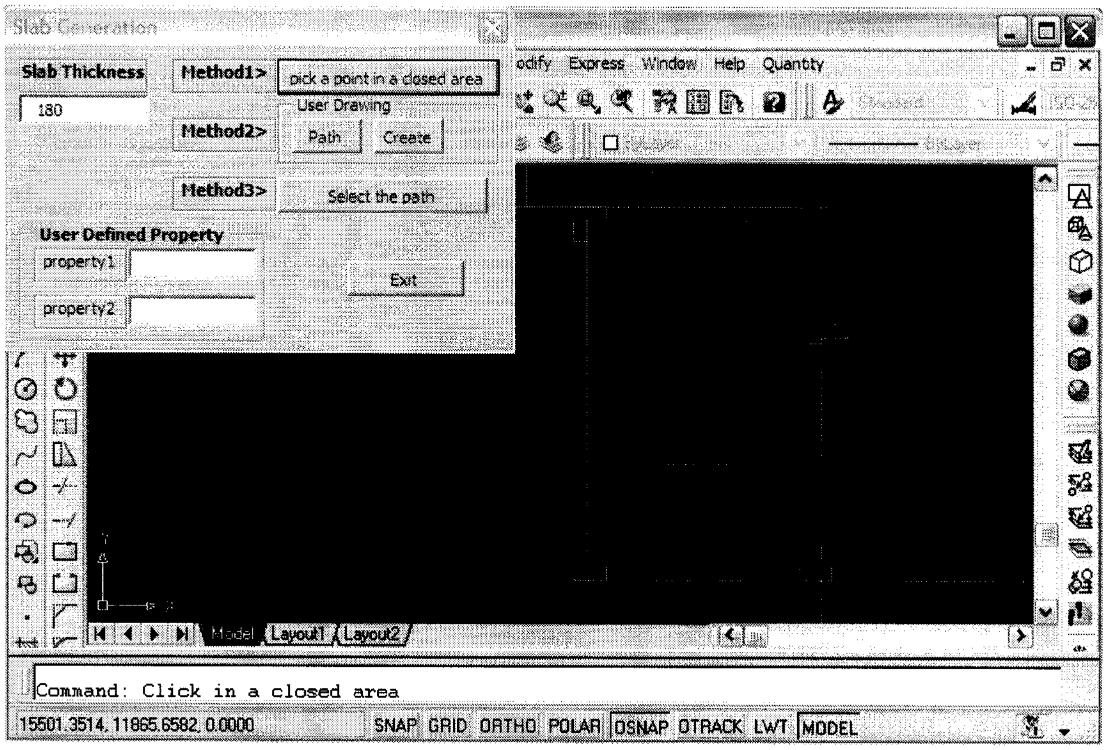


Figure 4.74 Clicking in a Closed Area

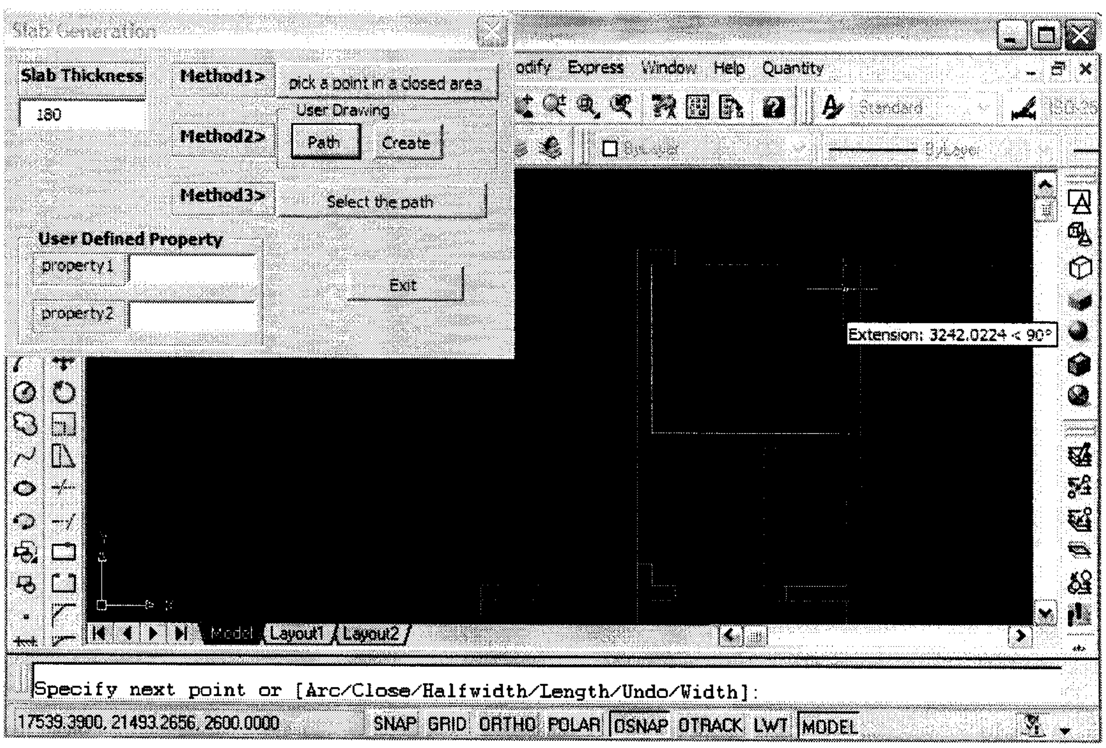


Figure 4.75 Draw the Slab Path (Method 2)

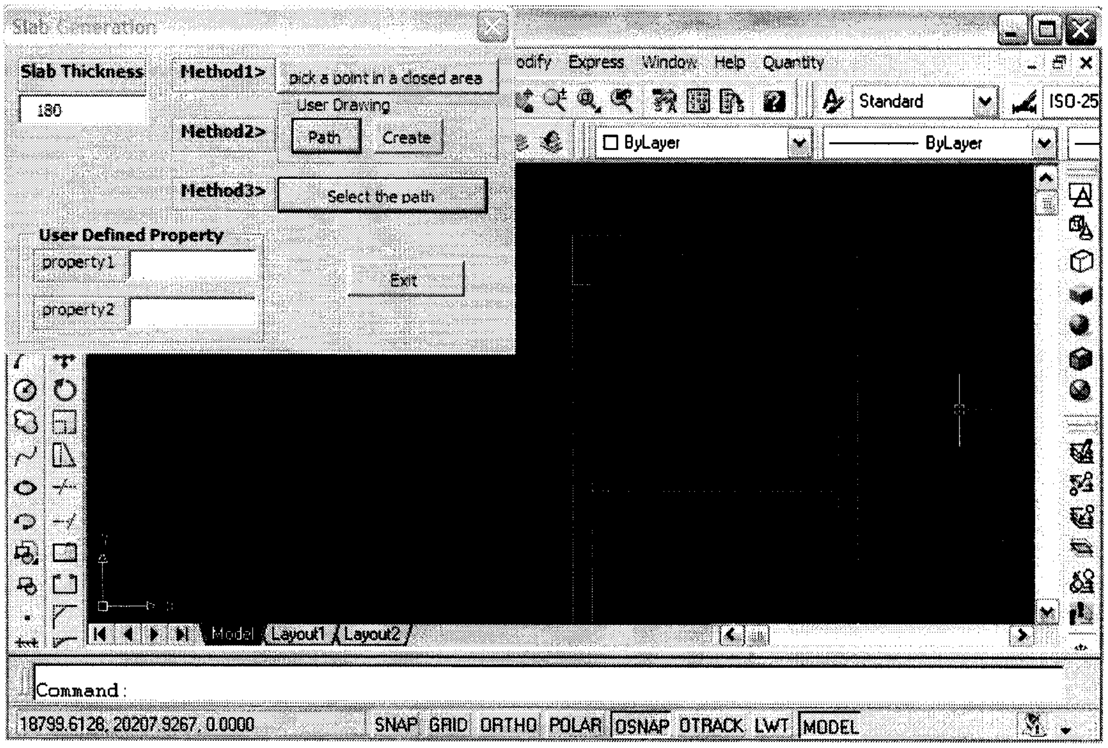


Figure 4.76 Method 2 Result

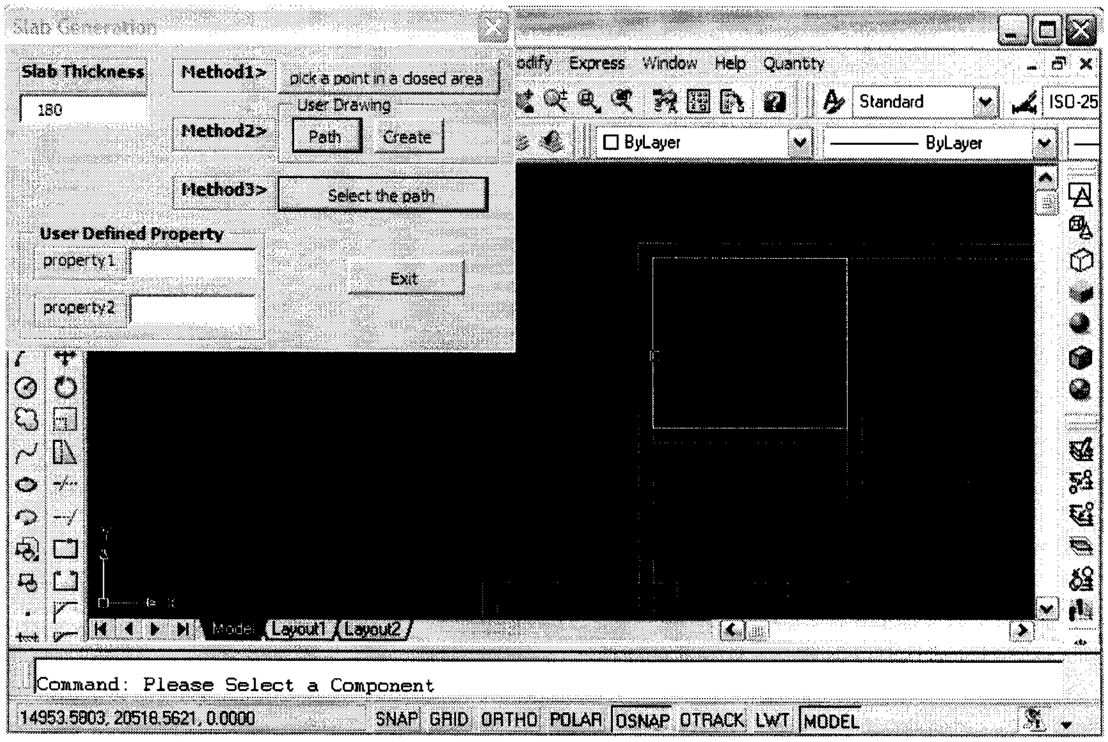


Figure 4.77 Selecting the Path (Method 3)

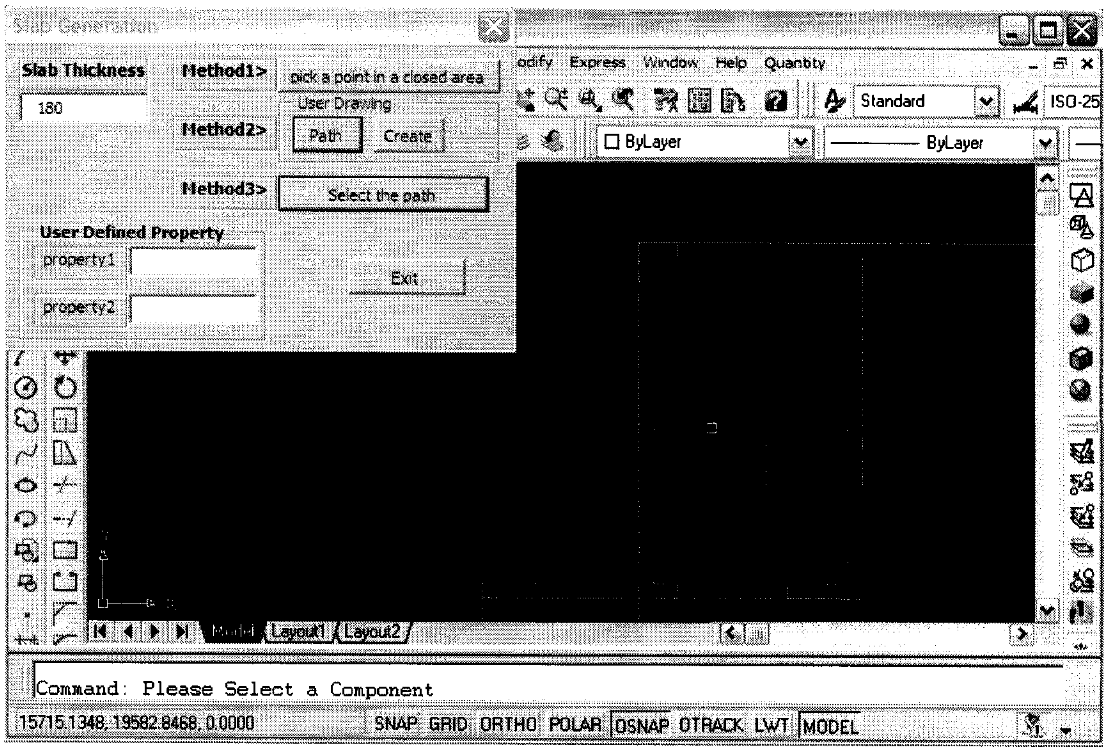


Figure 4.78 Method 3 Result

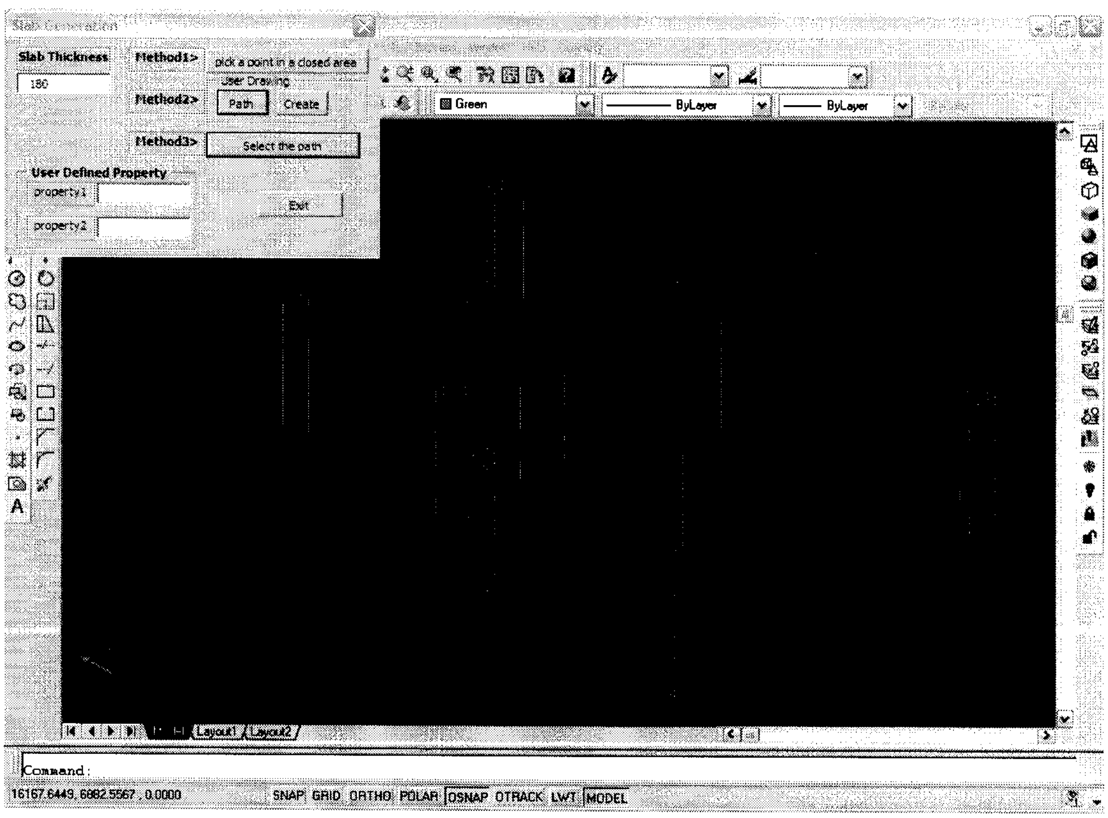


Figure 4.79 Highlighted Slab Model

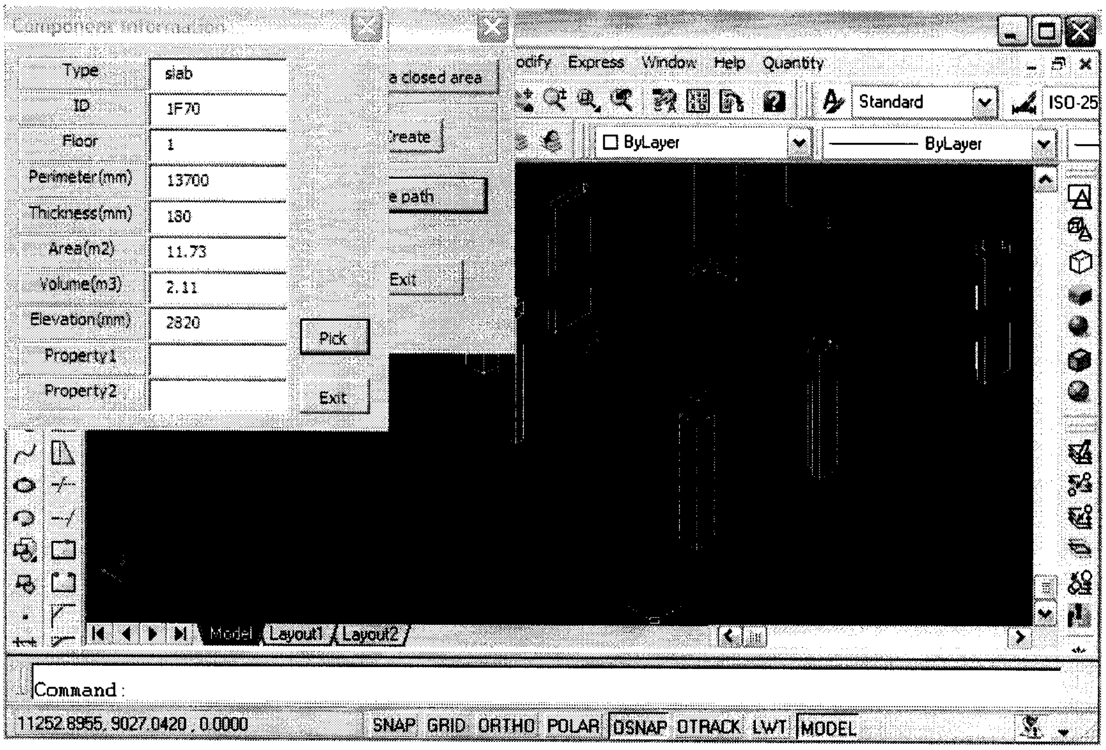


Figure 4.80 Slab Query (3D)

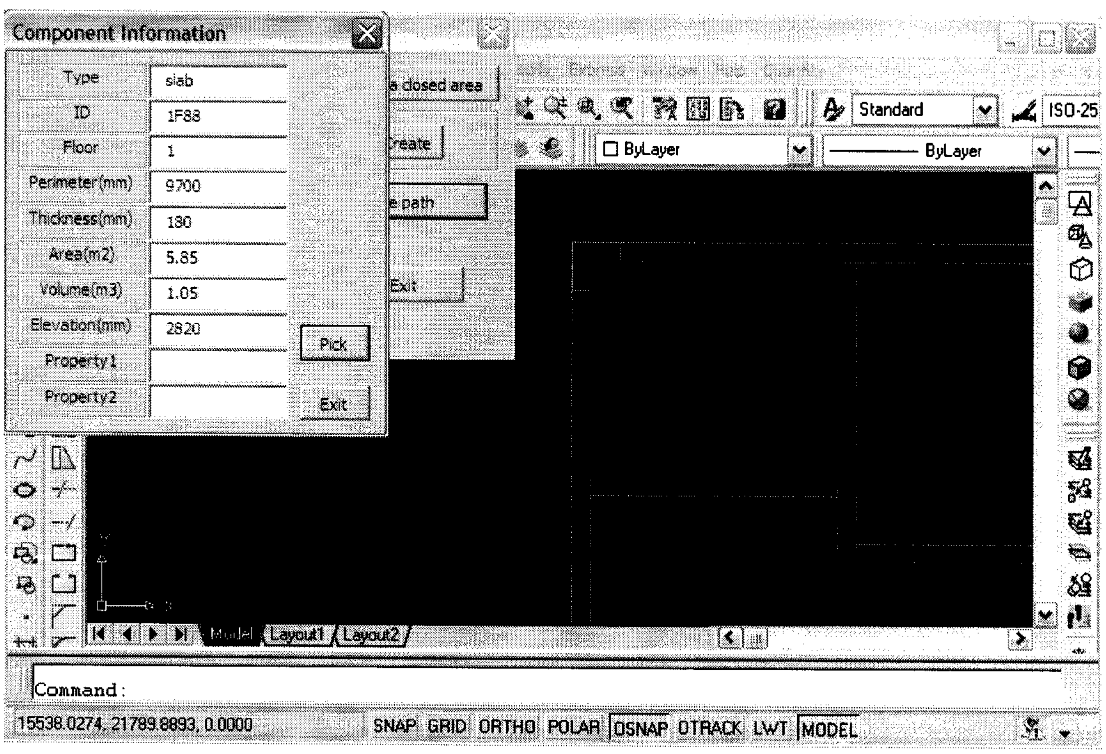


Figure 4.81 Slab Query (2D)

4.6.4 Wall modeling

The interface of wall modeling is shown in Figure 4.82. The import of the architectural floor plan is needed for the wall modeling. Because there are so many objects in it, the AutoCAD 2004's embedded command "Layoff" can be utilized to turn off some unnecessary layers. Only leave the wall object and its related dimensions in the imported drawing (Figure 4.83, Figure 4.84).

On the other hand, the generated components such as beams and slabs may also be visual obstacles for the modeling of walls; the ideal state is just leaving the columns on the screen and hiding the beams and slabs. Because the components on a given floor are created on the same layer in AutoCAD, it is not feasible to "turn off' the beams and slabs by using the command "Layoff". So a customized command is designed to solve the problem. As shown in Figure 4.85, in the form of "Component Generating & Display", there is command button "Component display", Once pressed, the interface of Figure 4.86 will appear. In this form, there is a text box indicating the name of the floor on which the components are modeled. The user can choose the type of component to be displayed by checking the check box beside the description of corresponding component. If the user checks only the "column", then only the columns on the selected floor will be displayed (Figure 4.87).

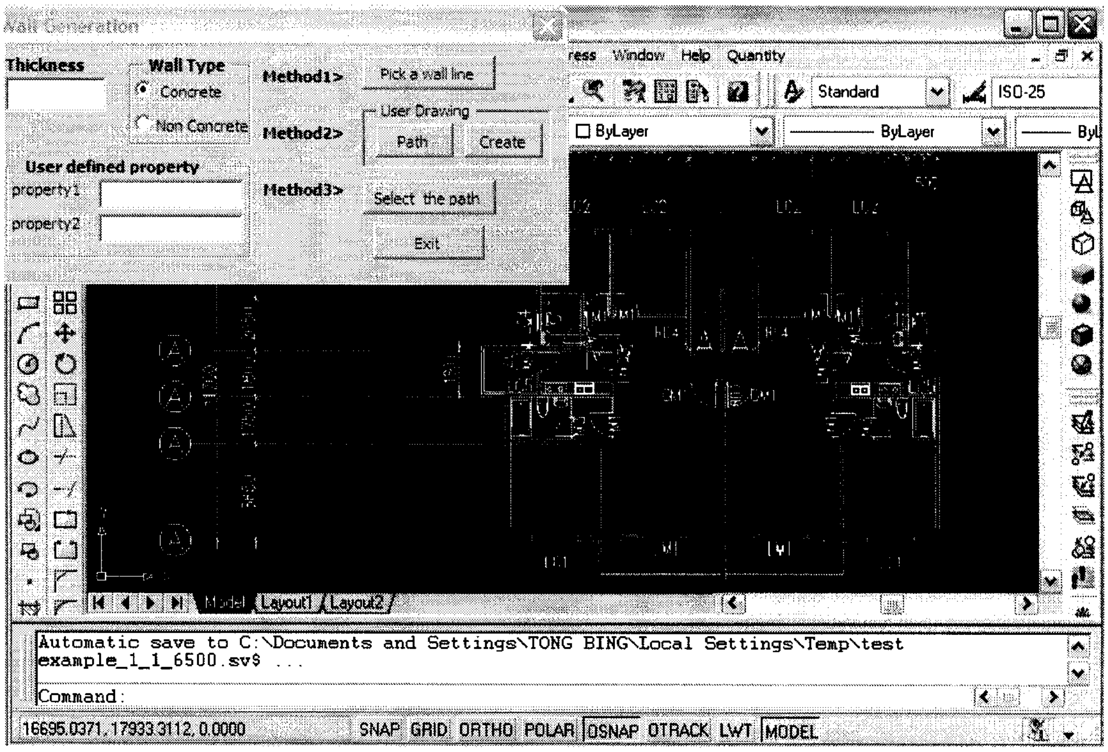


Figure 4.82 Main Form of Wall Modeling

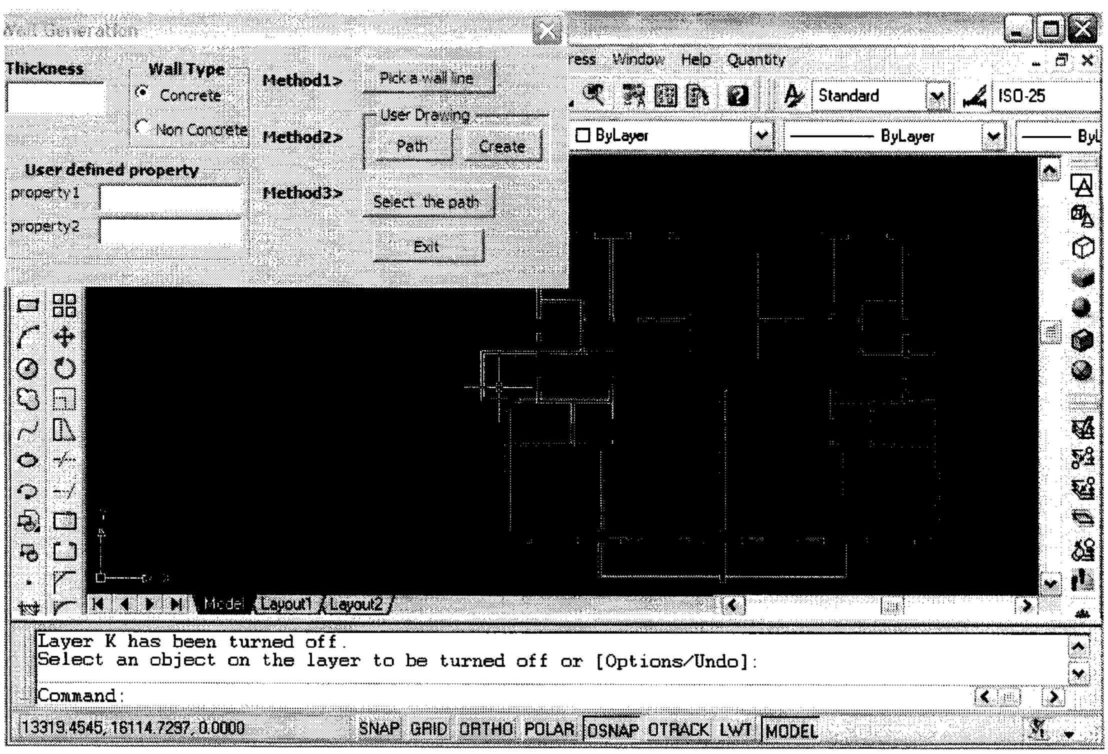


Figure 4.83 Turn Off the Unnecessary Layers

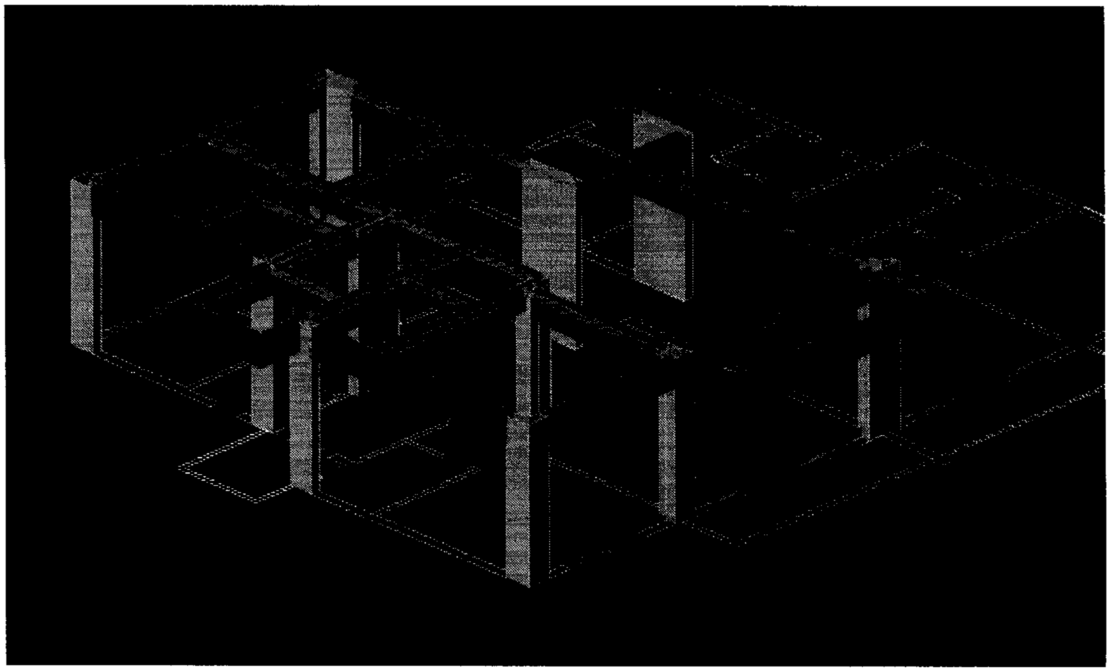


Figure 4.84 Turn Off the Unnecessary Layers (3D)

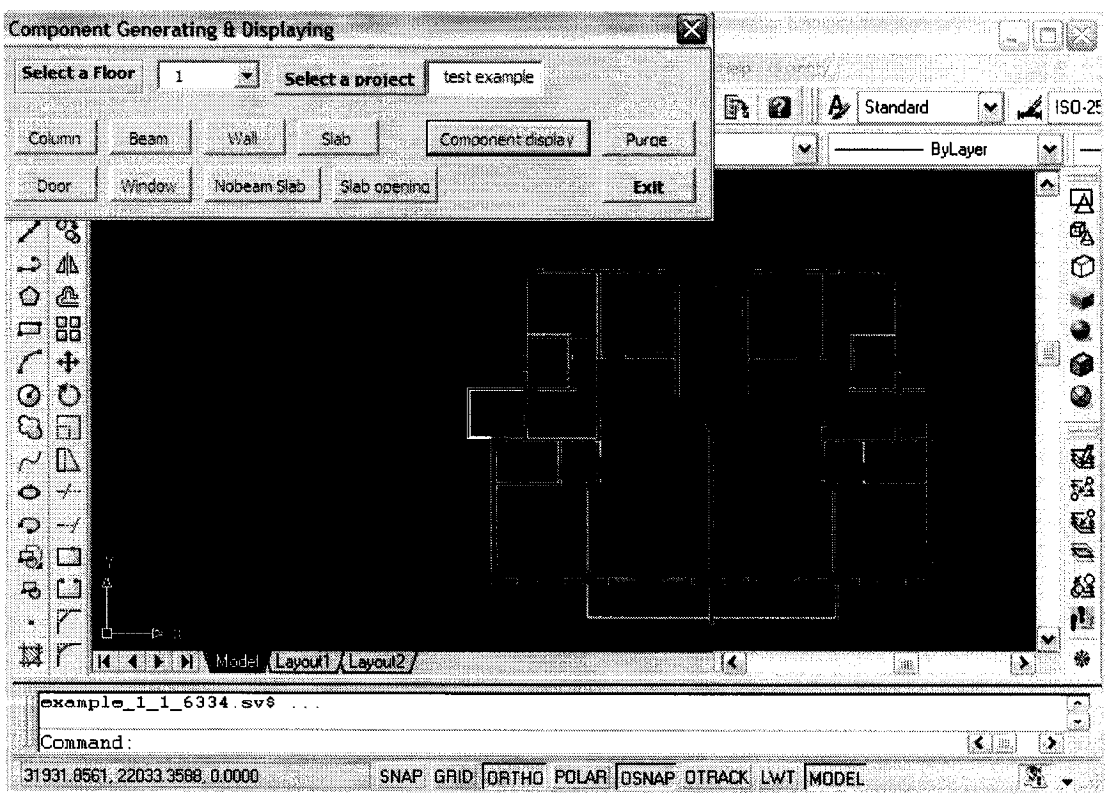


Figure 4.85 Component Display Command

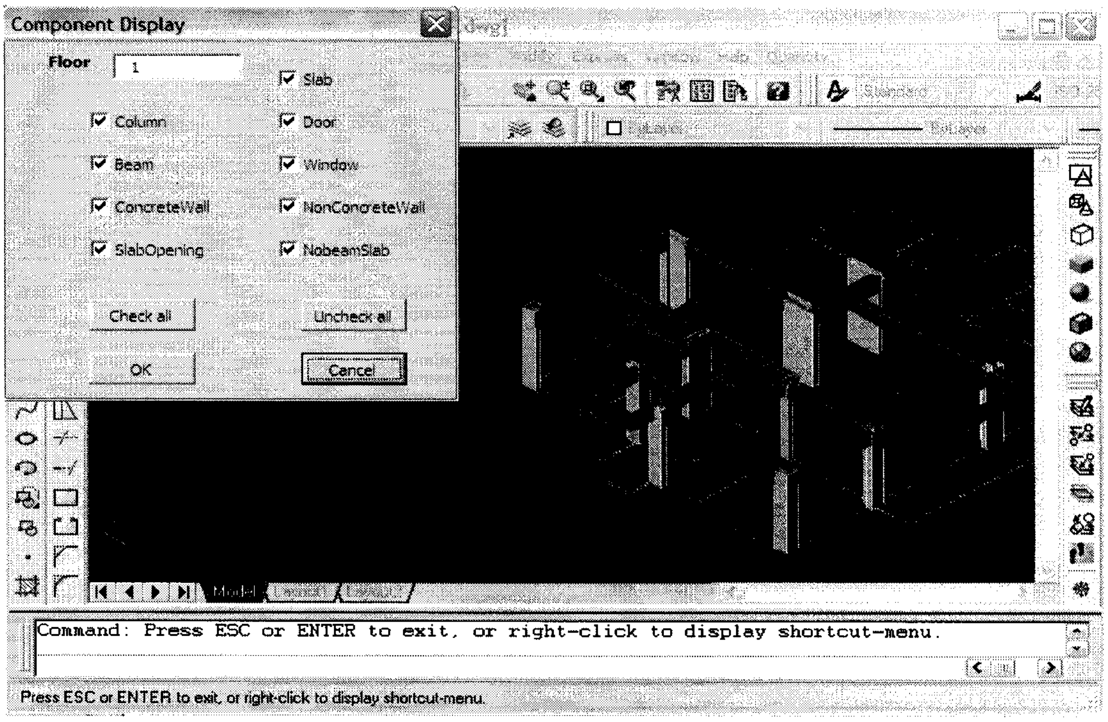


Figure 4.86 Selecting Component to Display

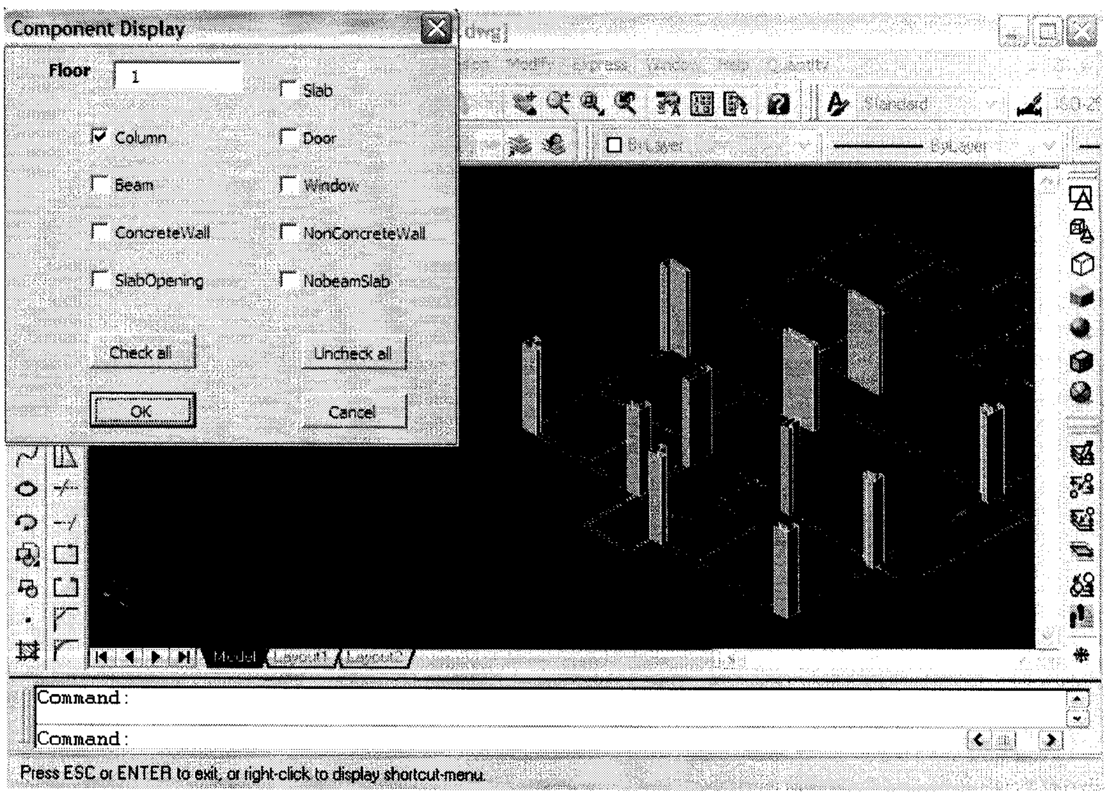


Figure 4.87 Selected Components

Three modeling methods are provided for wall modeling. For the convenience of data output and space relationship analysis, it is recommended that the walls be modeled in straight segments, not to create corners. And the default heights of the walls are set to be equal to the height of the floor on which the walls are going to be modeled. All of the three methods are similar to the ones of beam modeling.

Method 1: In this method, the thickness of the wall is not needed to be entered, but the user must choose the type of the wall. Two types of wall are defined: Concrete and Non Concrete (Figure 4.88). When modeling, the user should pick one of the two lines of which a wall consists, and the thickness will be measured and the wall modeling will be finished automatically. The color of the successfully generated Concrete wall is grey, and Non Concrete wall is light brown (Figure 4.89). If the user wants to define the wall with more detail, he can use the "User defined property". For example, he can enter "exterior" in the "property1" text box, "brick" in the "property2" text box. These properties will be exported to the external database and could be very useful for future's components filtering. For example, supposing the user wants to find all exterior walls that are constructed in brick, the user can easily do this using these two fields as filters in the database. The data could also be viewed through component query (see Figure 4.90).

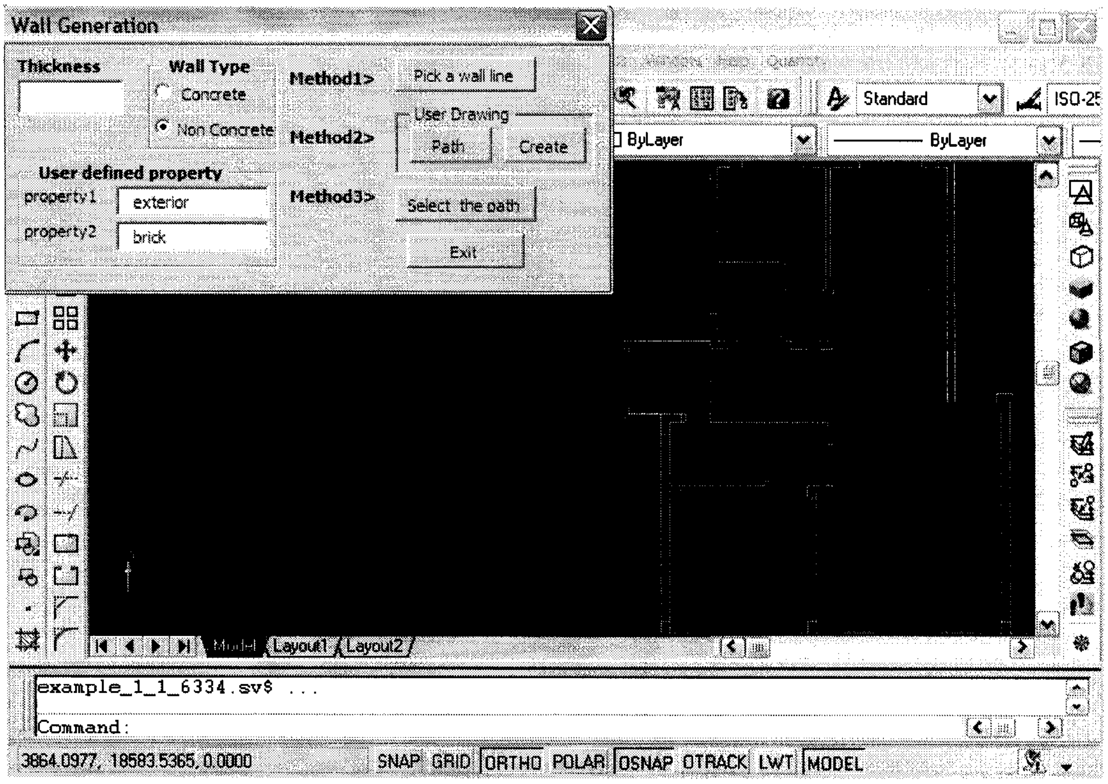


Figure 4.88 User Input

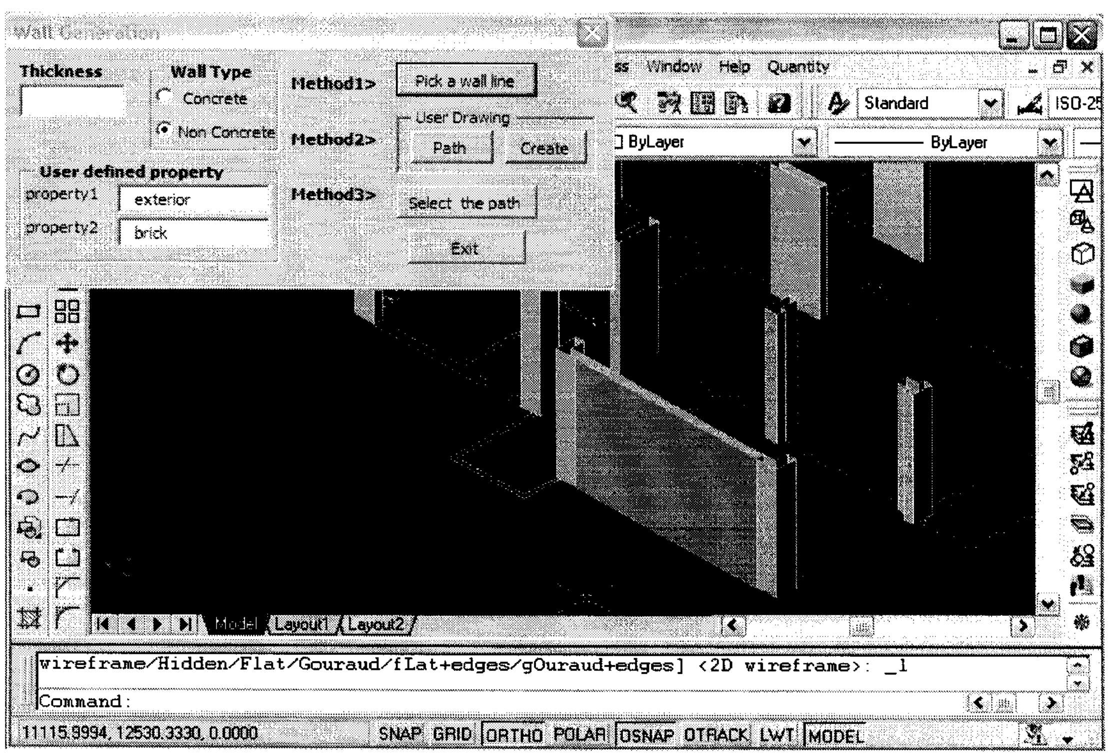


Figure 4.89 Result of Wall Modeling (Method1)

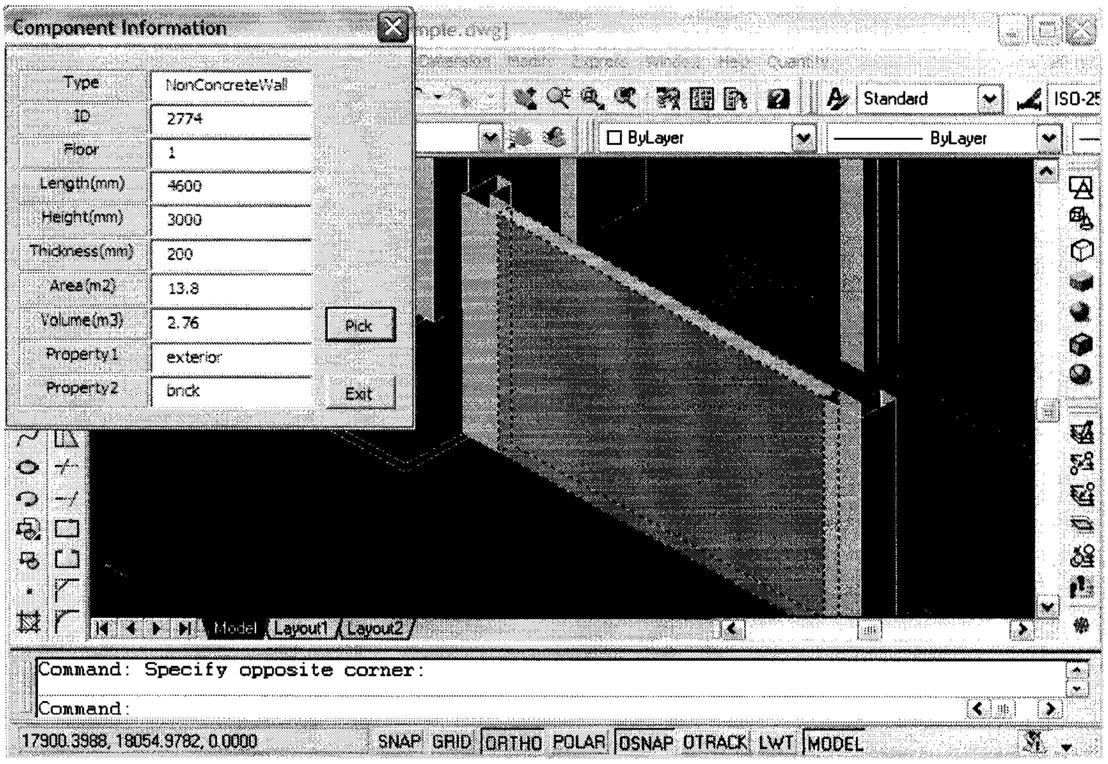


Figure 4.89 Query of Wall (Method1)

Methods and 3,: These two methods are almost the same as those of beam modeling. The differences are that the user needs to enter the thickness of the wall and select the wall type. They all require the user to draw the path of the wall, and then press “Create” after each drawing or select the paths after several path drawings. Figures 4.90 through 4.95 illustrate the process of employing these two methods for wall modeling.

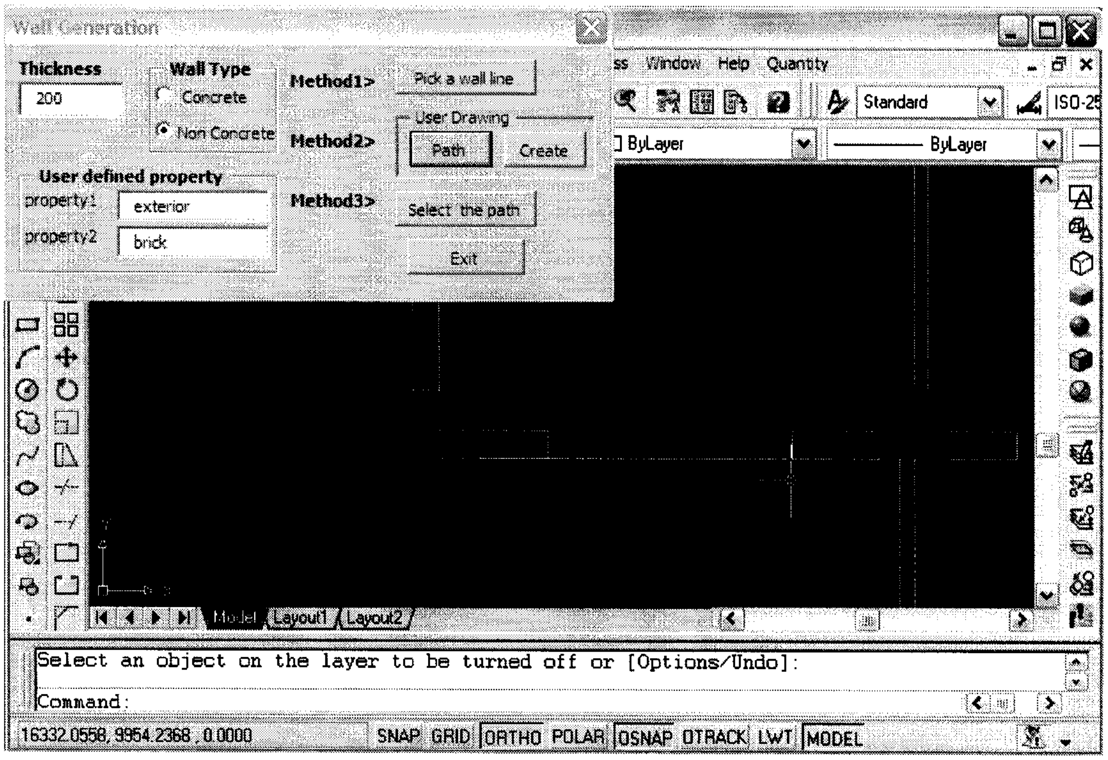


Figure 4.90 Wall Path Drawing (Method 2)

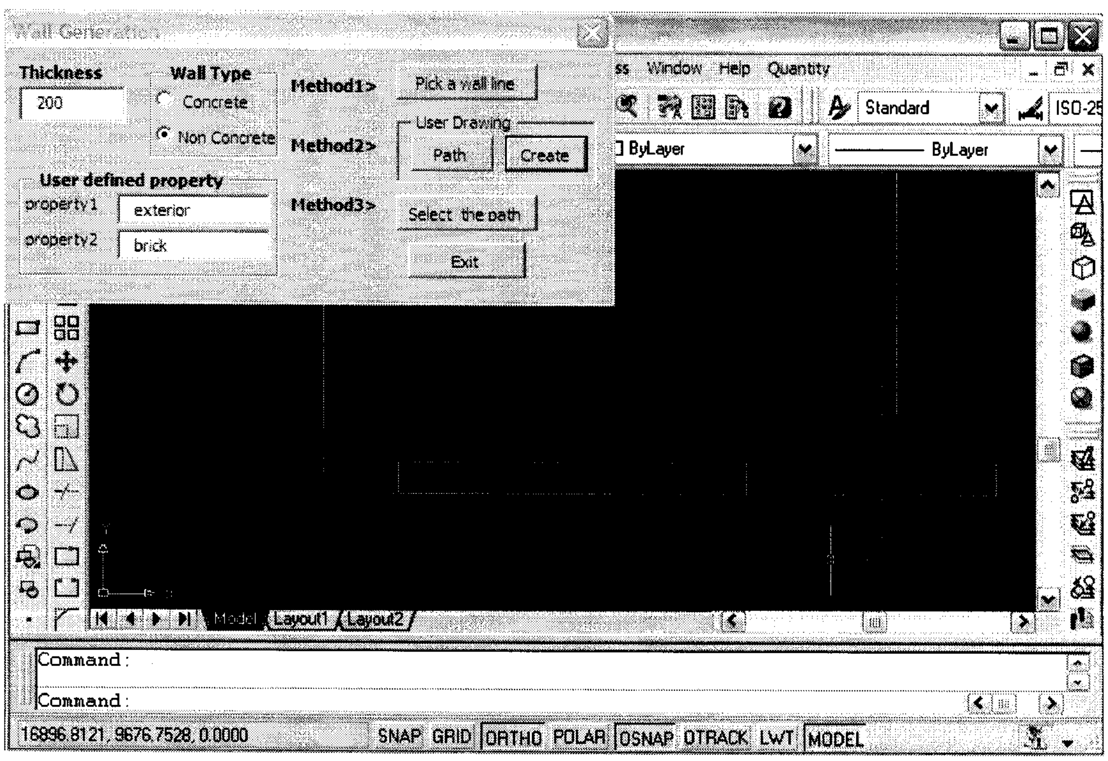


Figure 4.91 Wall Creating (Method 2)

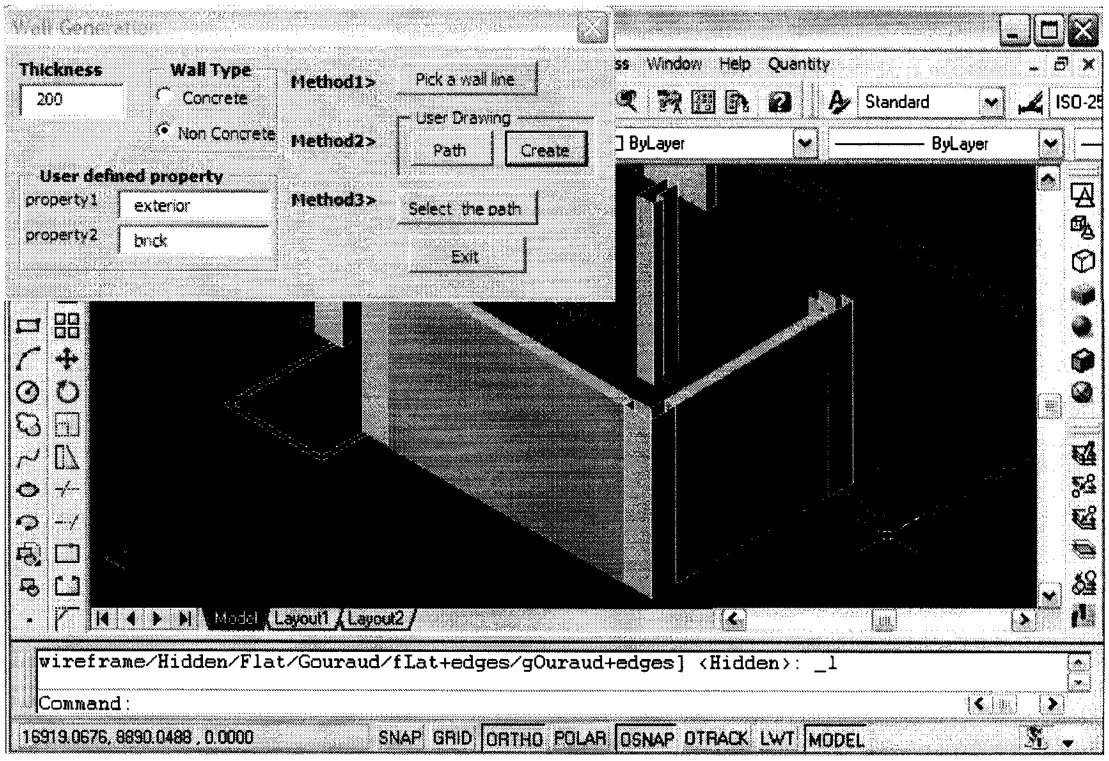


Figure 4.92 3D View of Wall (Method 2)

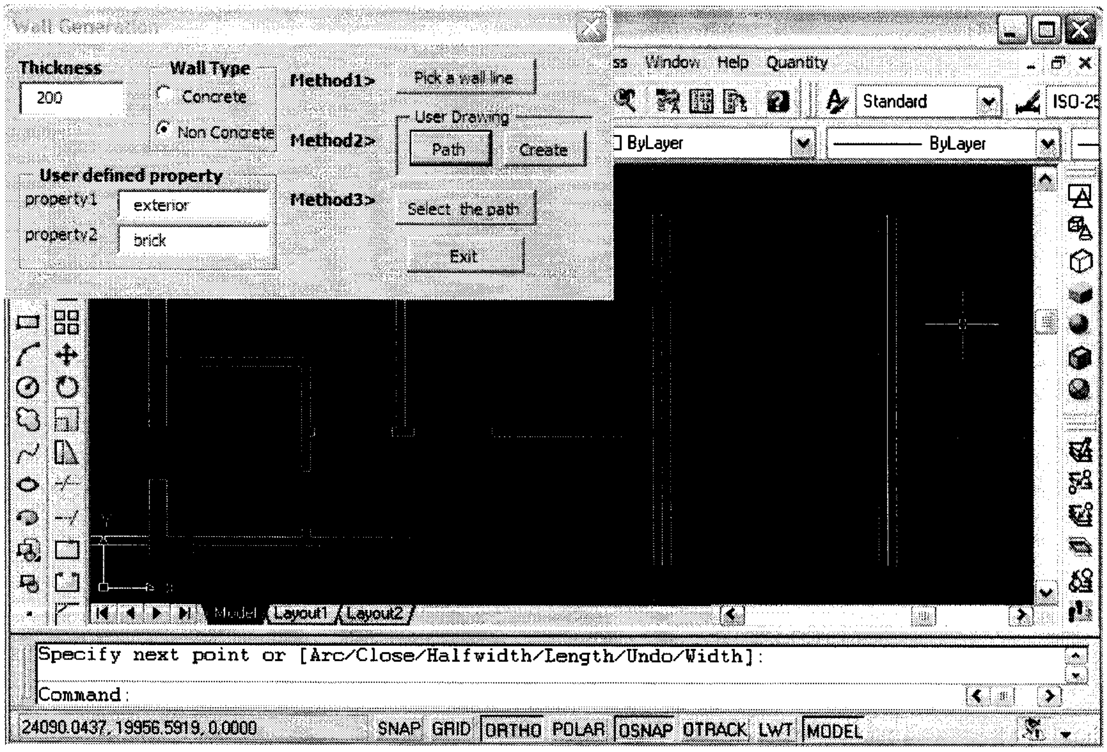


Figure 4.93 Before Path Selecting (Method 3)

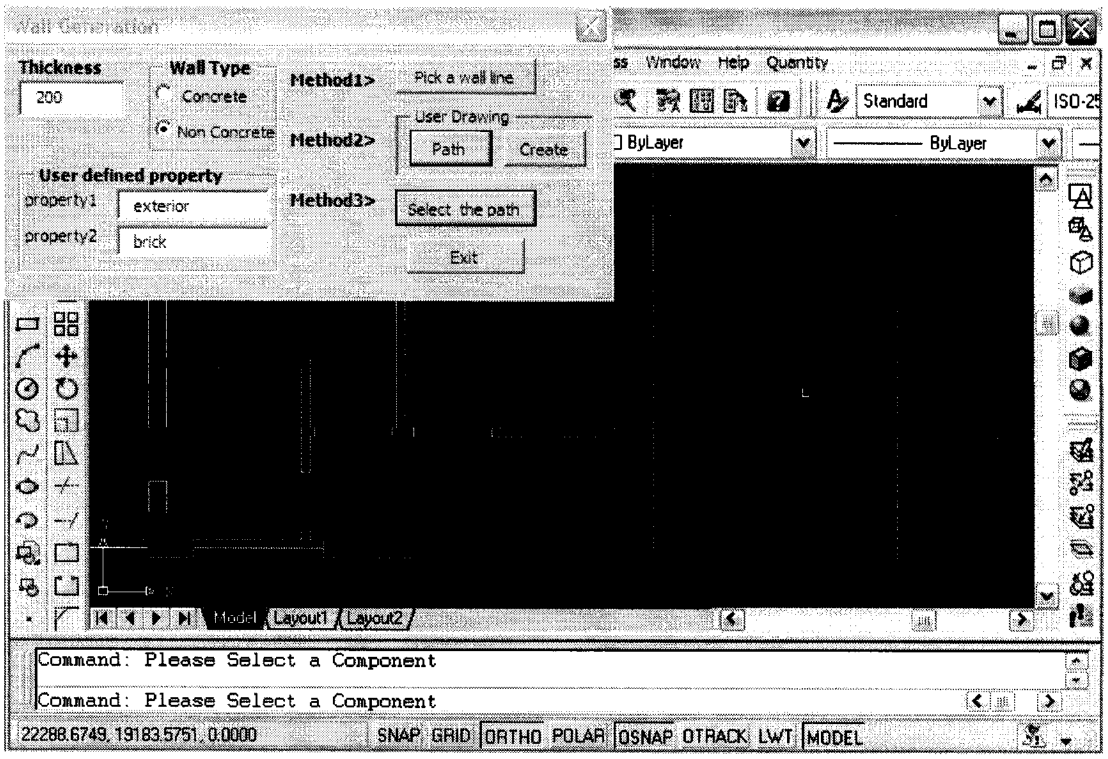


Figure 4.94 After Path Selecting (Method 3)

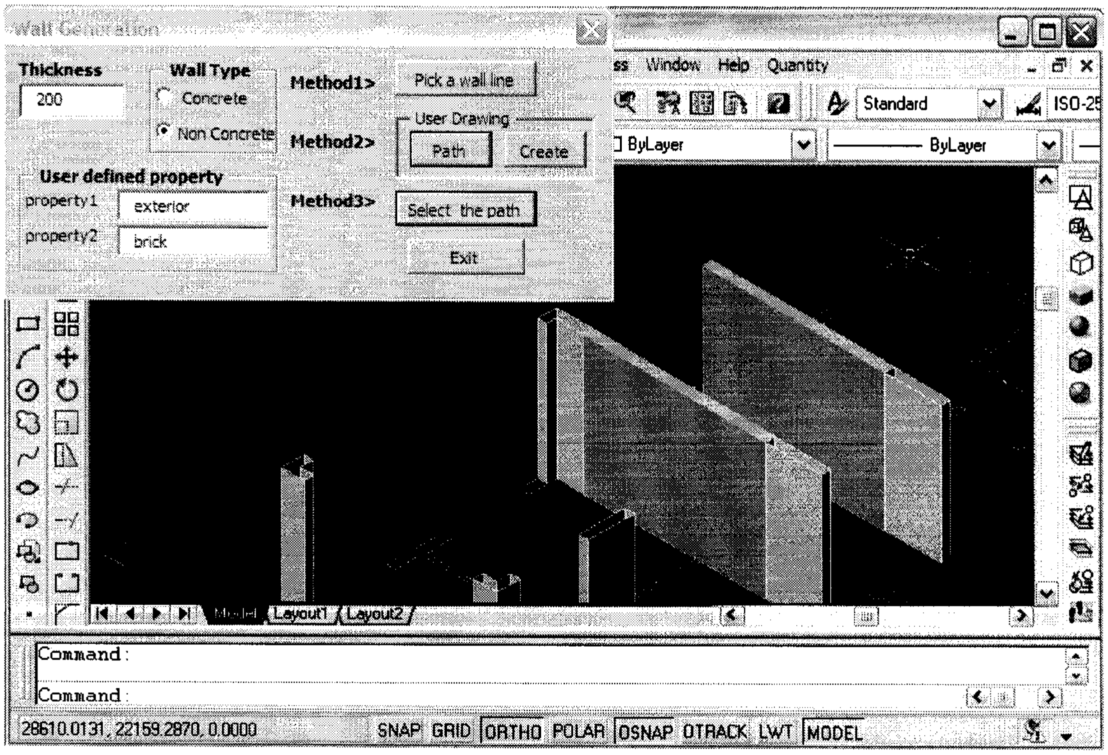


Figure 4.95 3D View of Wall (Method 3)

4.6.5 Door & Window Modeling

Compared to the previous components' modeling, the modeling of doors and windows is relatively easier, but it cannot be performed until the wall modeling has been finished. Only one method is provided: "Select a wallline". Because any door or window must belong to a wall, no matter what type the wall is. It could be concrete, brick, drywall, a curtain wall or something else.

Before modeling, the user should enter the width and the height of a door; the width, the height and the distance from the floor (mm) of a window. Then the user clicks the "Select a wallline" button, and the program will prompt the user to select a wall line which must be a modeled wall. The ideal position of the selecting point is the center of a door or a window; it doesn't need to be precise, though. The process is shown in Figures 4.96 through 4.101.

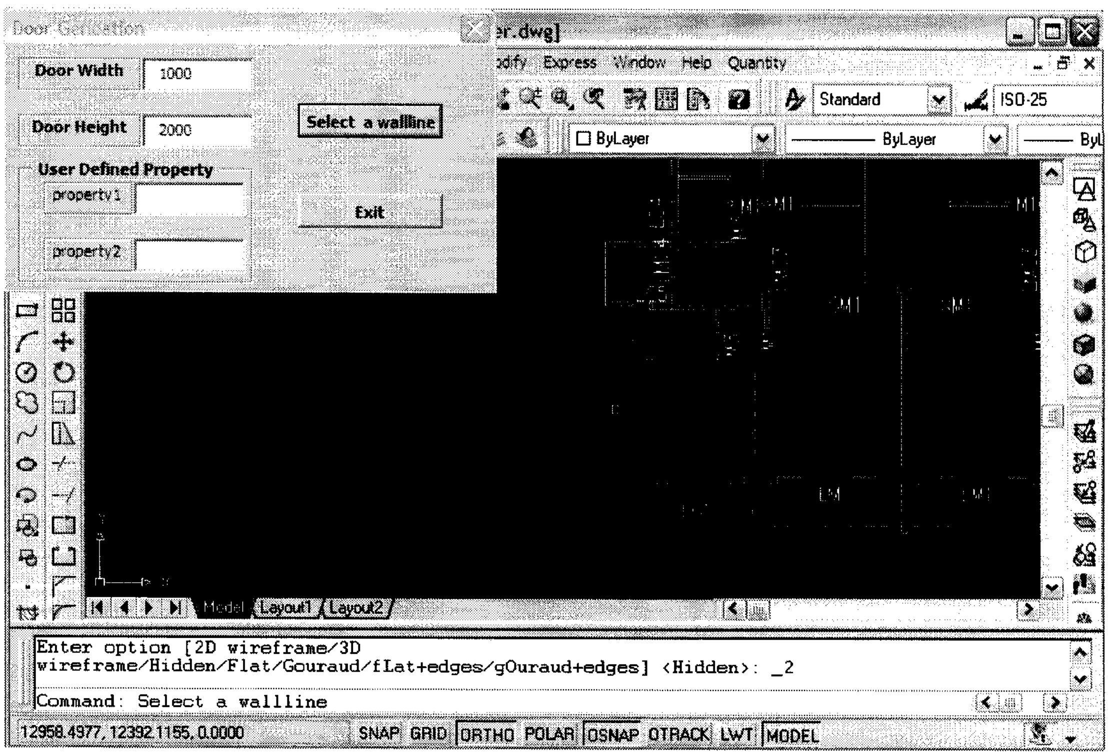


Figure 4.96 Prompt to Select a Wall Line (Door Modeling)

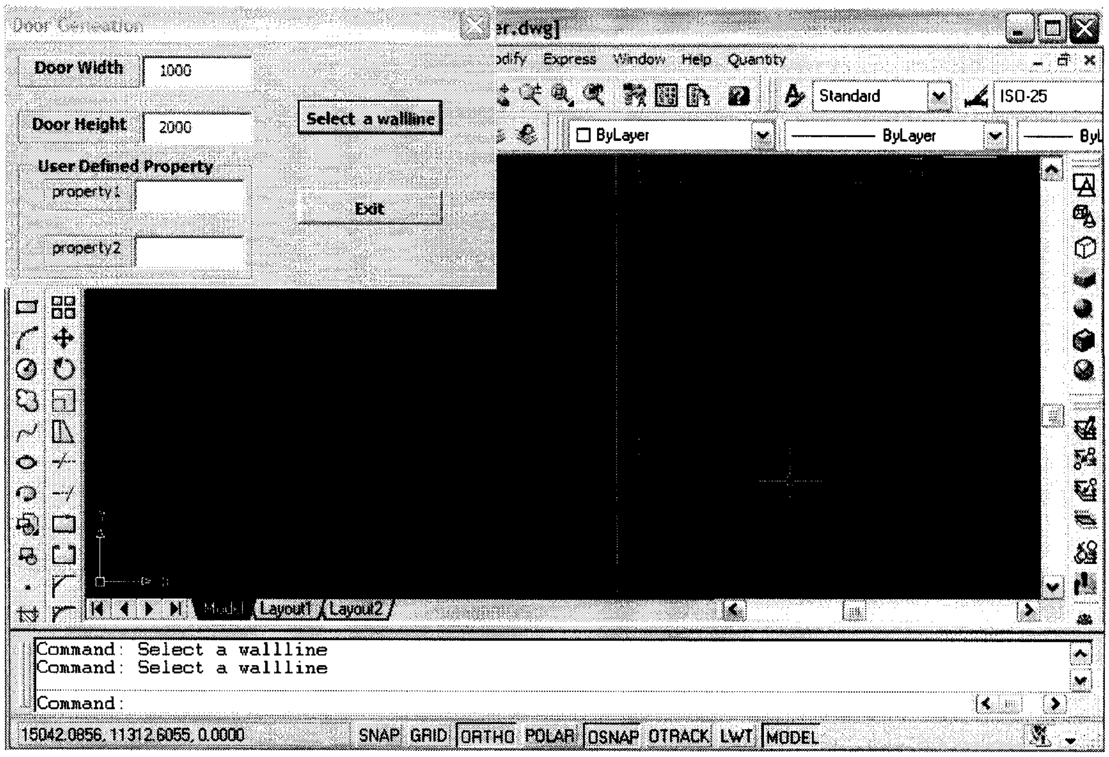


Figure 4.97 Generated Door

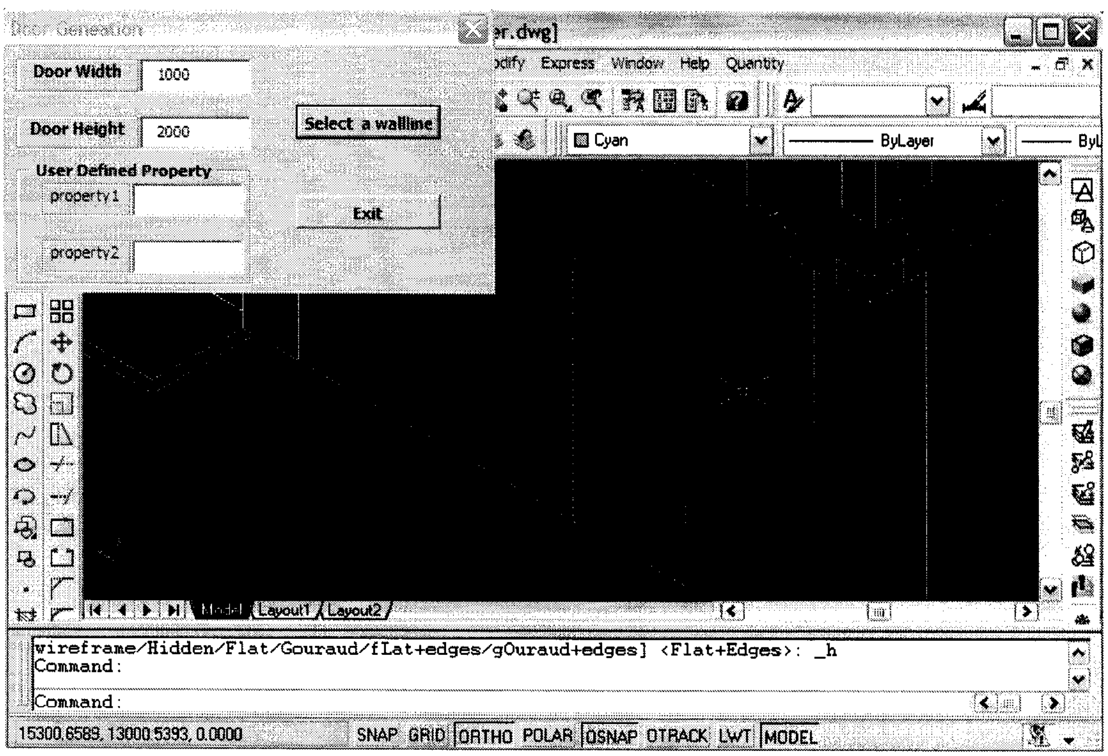


Figure 4.98 Generated Door (3D)

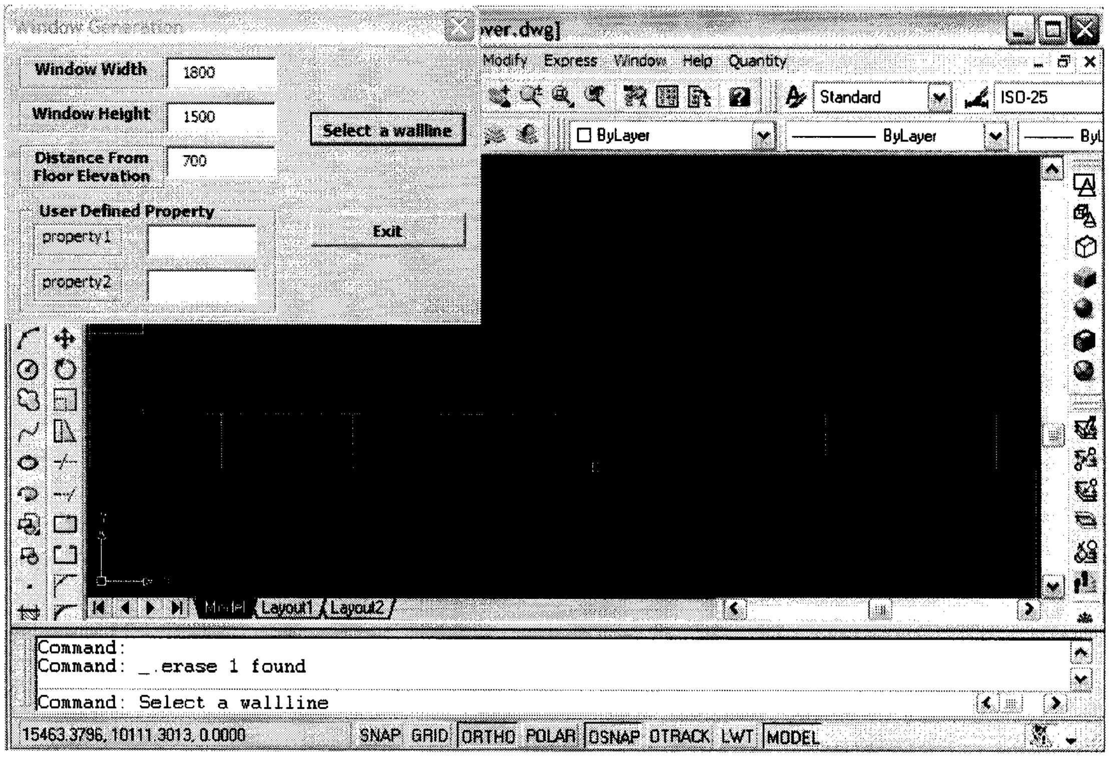


Figure 4.99 Prompt to Select a Wall Line (Window Modeling)

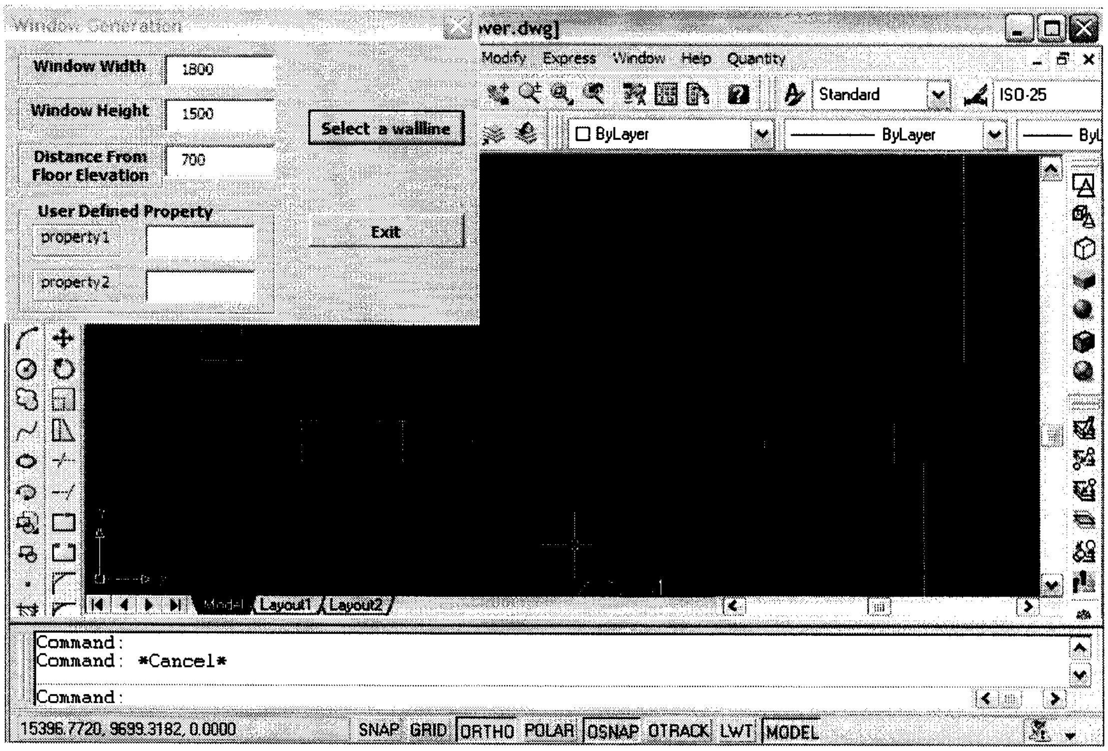


Figure 4.100 Generated Window

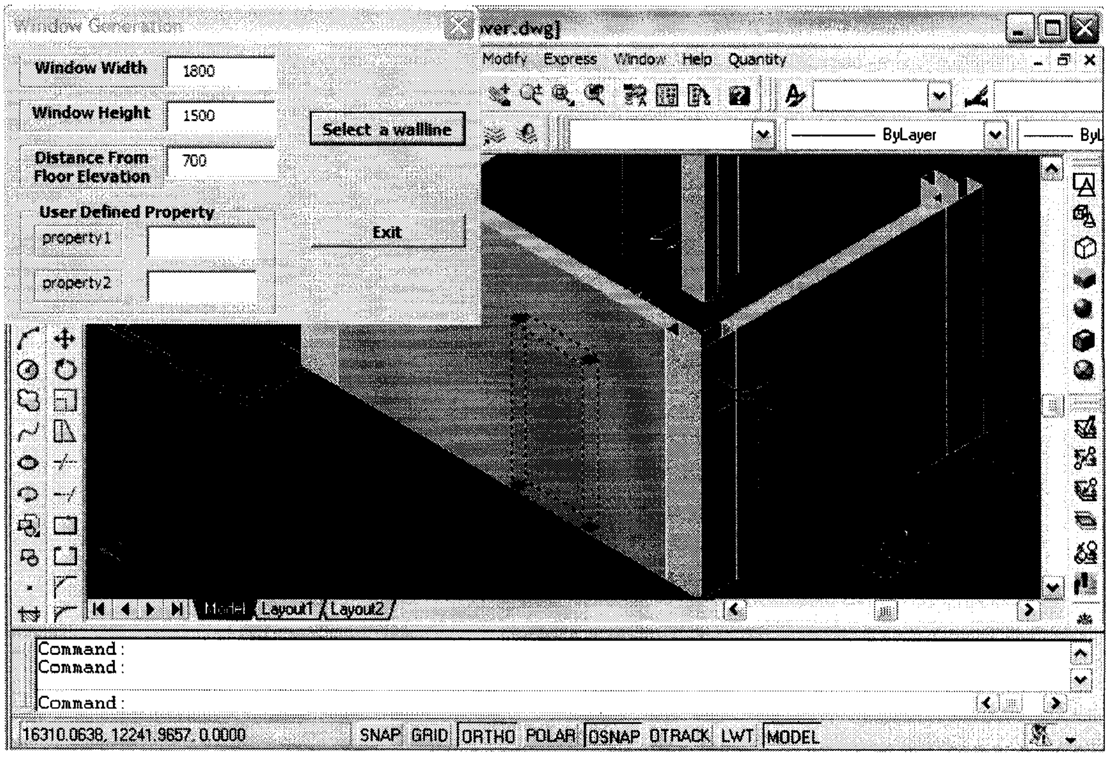


Figure 4.101 Generated Window (3D)

4.7 Data Export

This is the final process of the proposed model. As we mentioned in the previous chapters, ADO is the preferred method for AutoCAD to connect to and communicate with external databases. By using ADO objects, methods, and properties in AutoCAD 2004 VBA, the non- graphical information of each component in the completed 3D model is successfully exported to the external database. When exporting data to the database, the module also takes into account the spatial relationships among the components and their corresponding form work deduction. Figure 4.102 shows the sample code of the data export module.

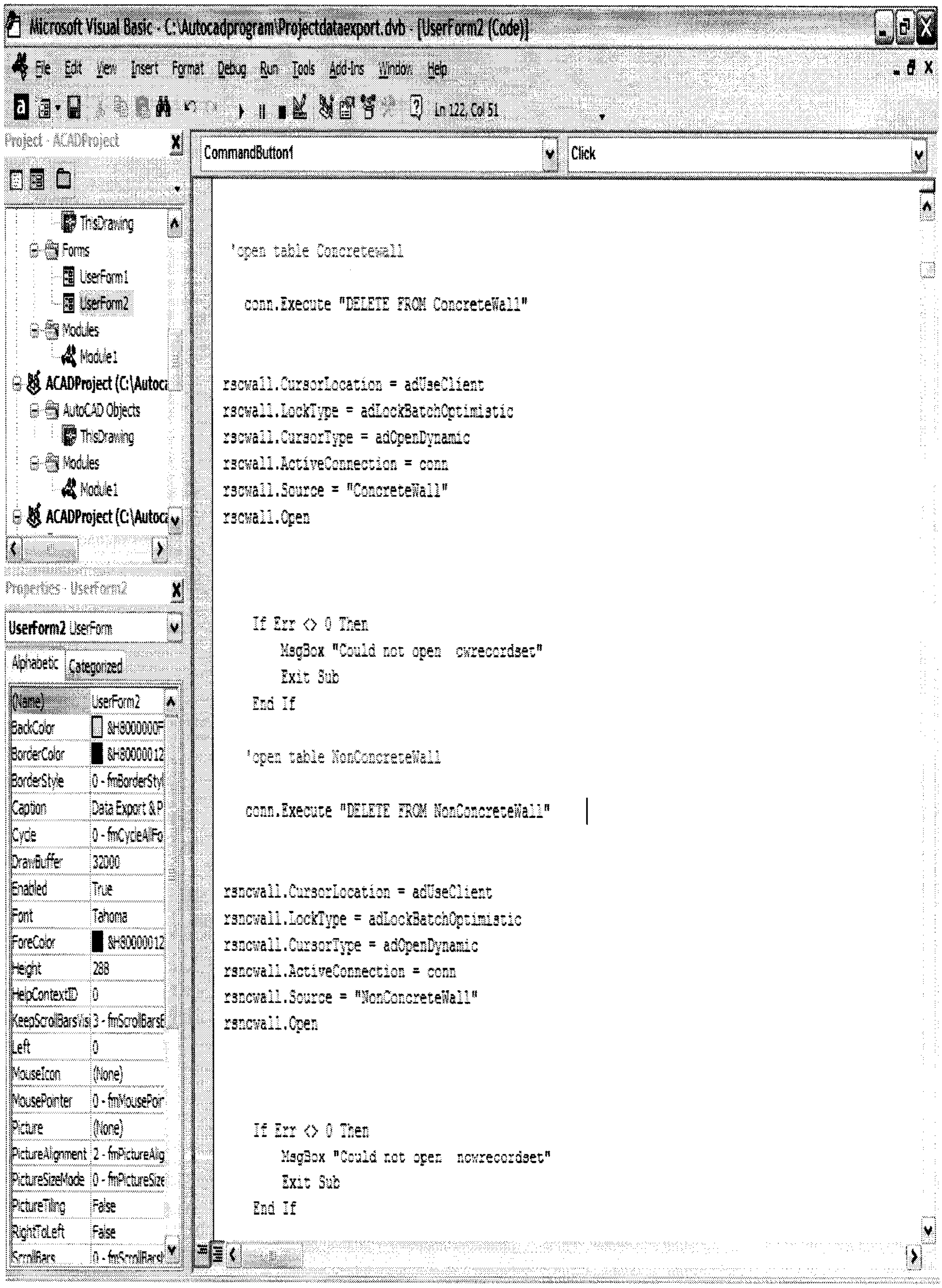


Figure 4.102 Sample Code for Data Exporting

The interface of data export is shown in Figure 4.103. The user only needs to select the targeted project database name from a dialog box (Figure 4.104), and then wait until the name of the project appears in the text box, signifying that the process of data exporting has finished. The user can have a preview of the populated database by selecting one type of components through the radio button. And the corresponding table in the external database will be show in the form. The amount of the records in a particular table will also be displayed in the form. Figures 4.105 through 4.111 show the samples of the tables preview.

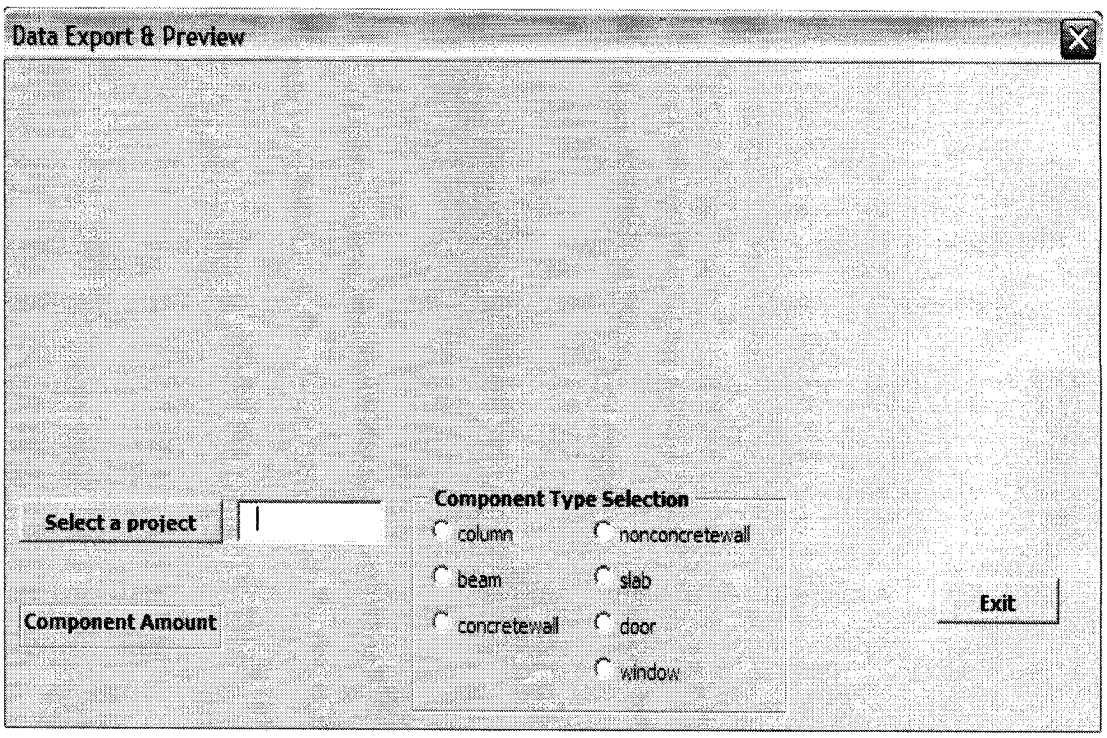


Figure 4.103 Data Export Interface

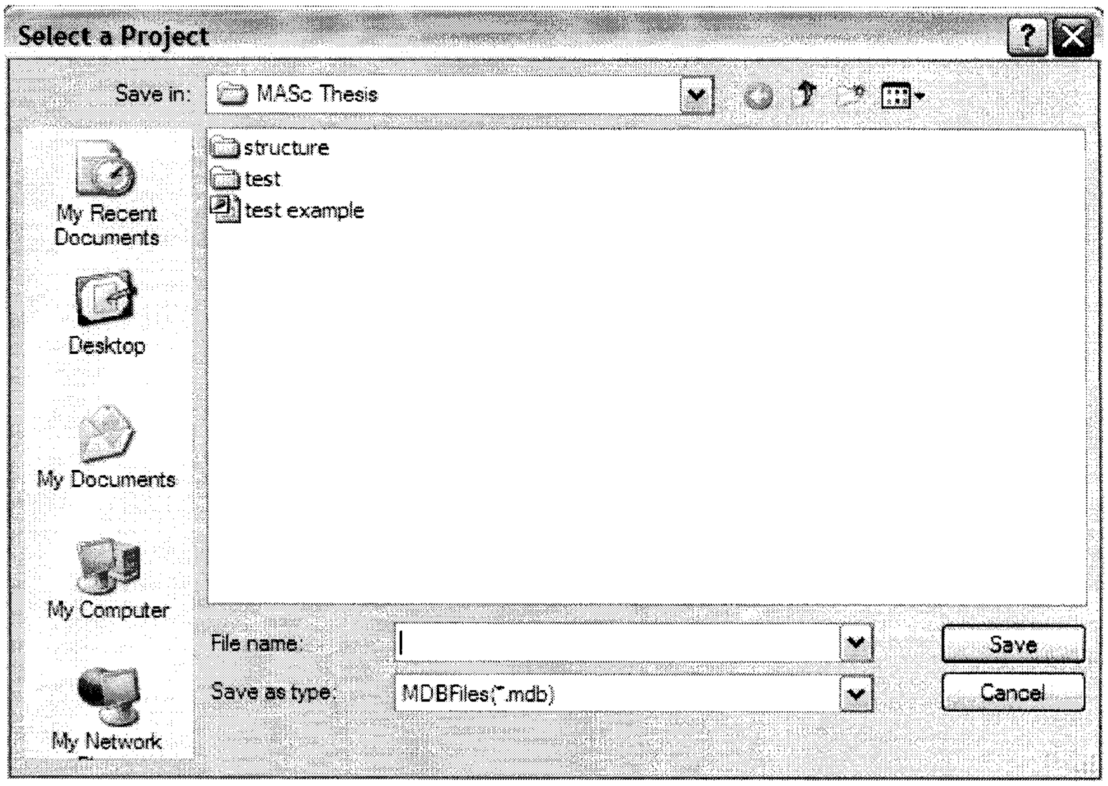


Figure 4.104 Dialog Box for Selecting a Project

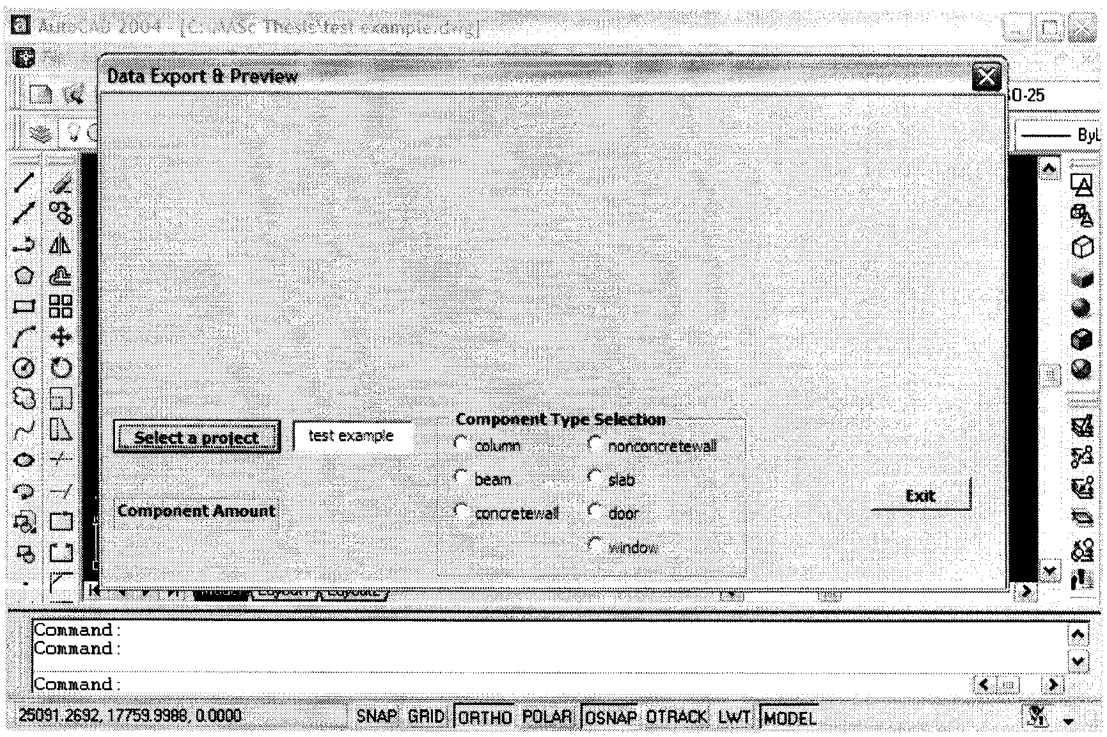


Figure 4.105 Data Exporting Finished

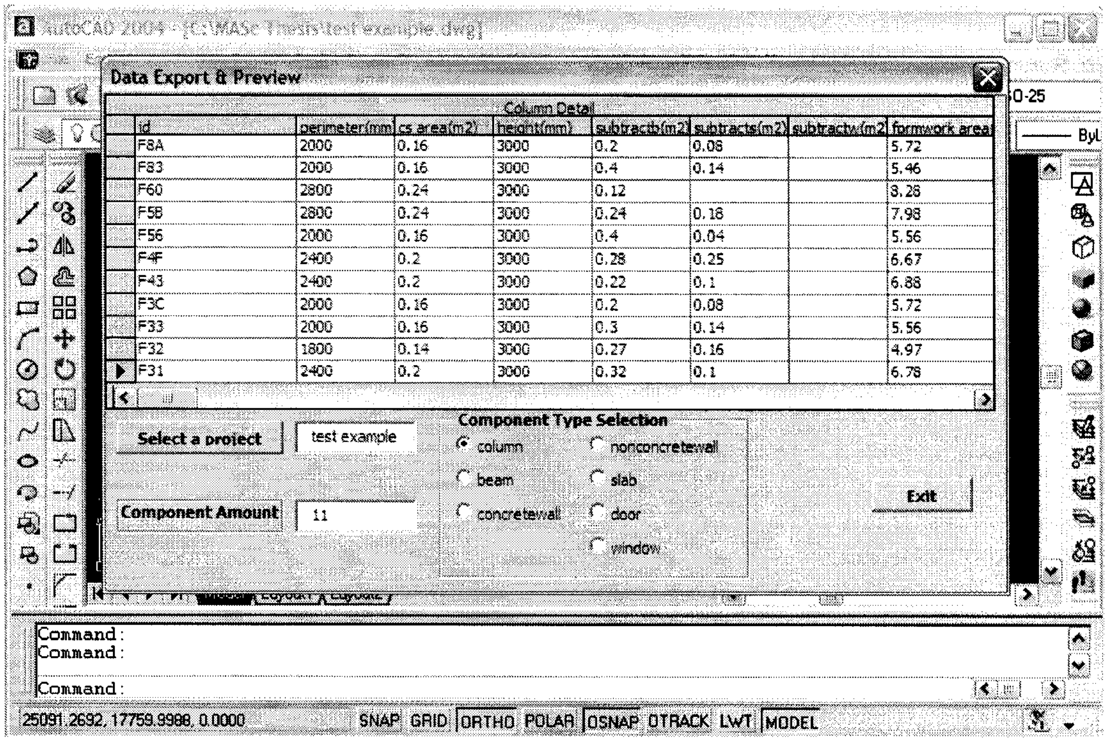


Figure 4.106 Column Table Preview

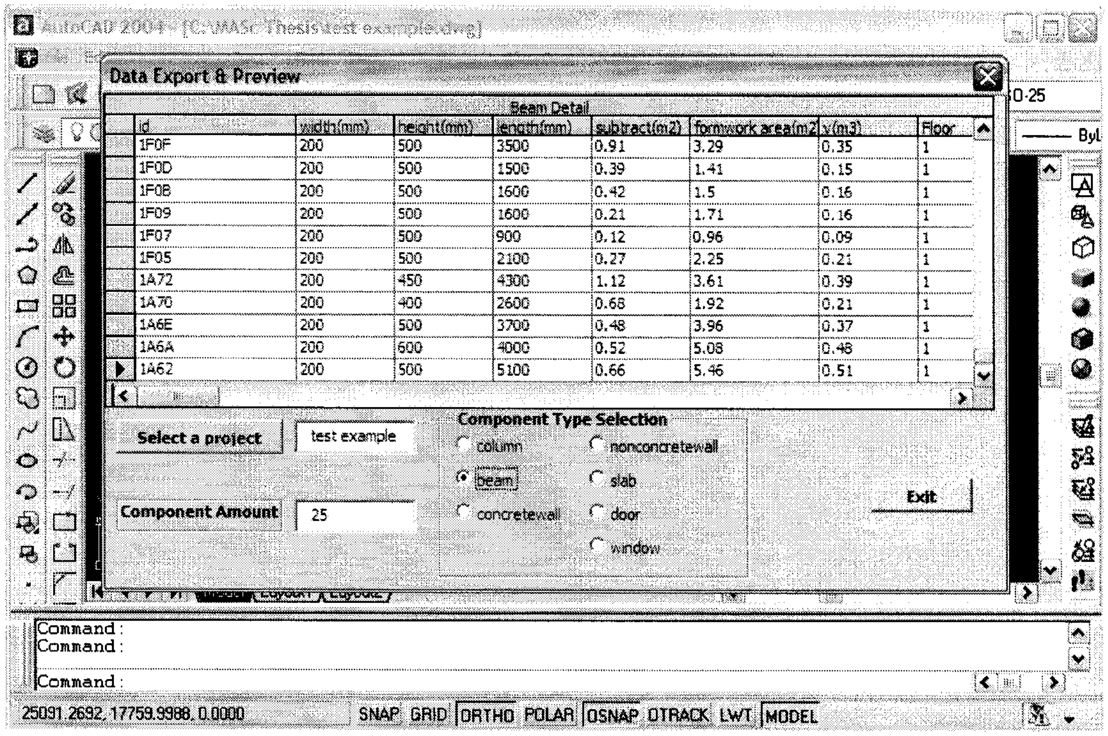


Figure 4.107 Beam Table Preview

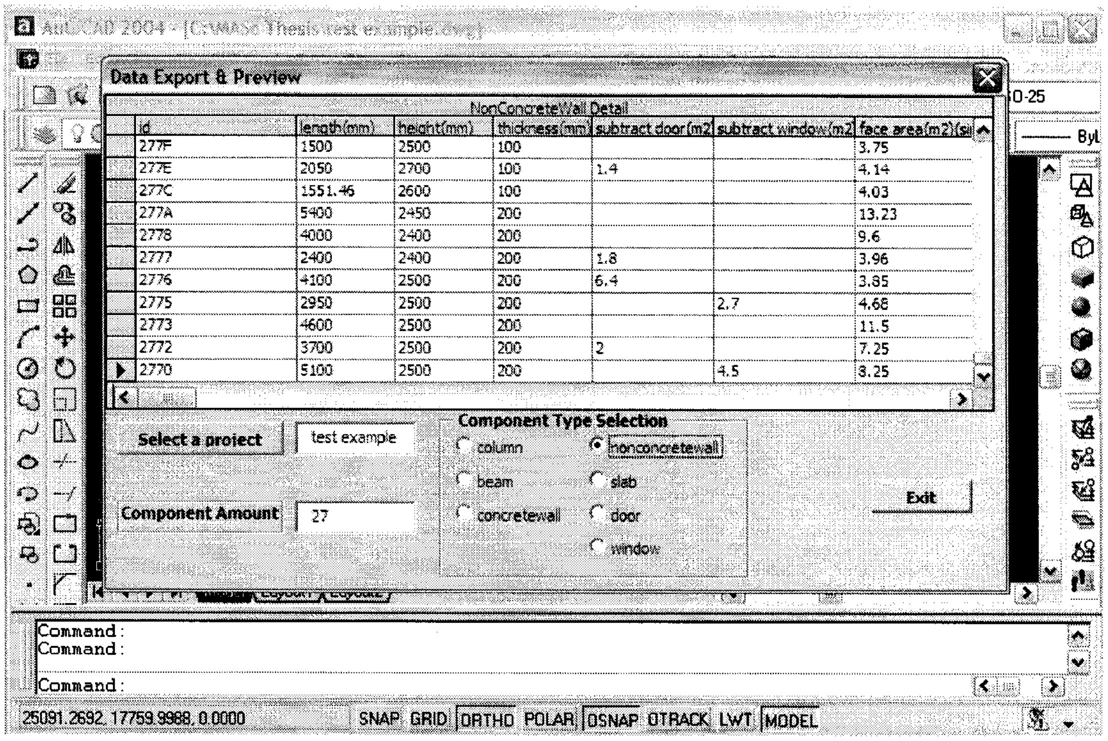


Figure 4.108 NonConcreteWall Table Preview

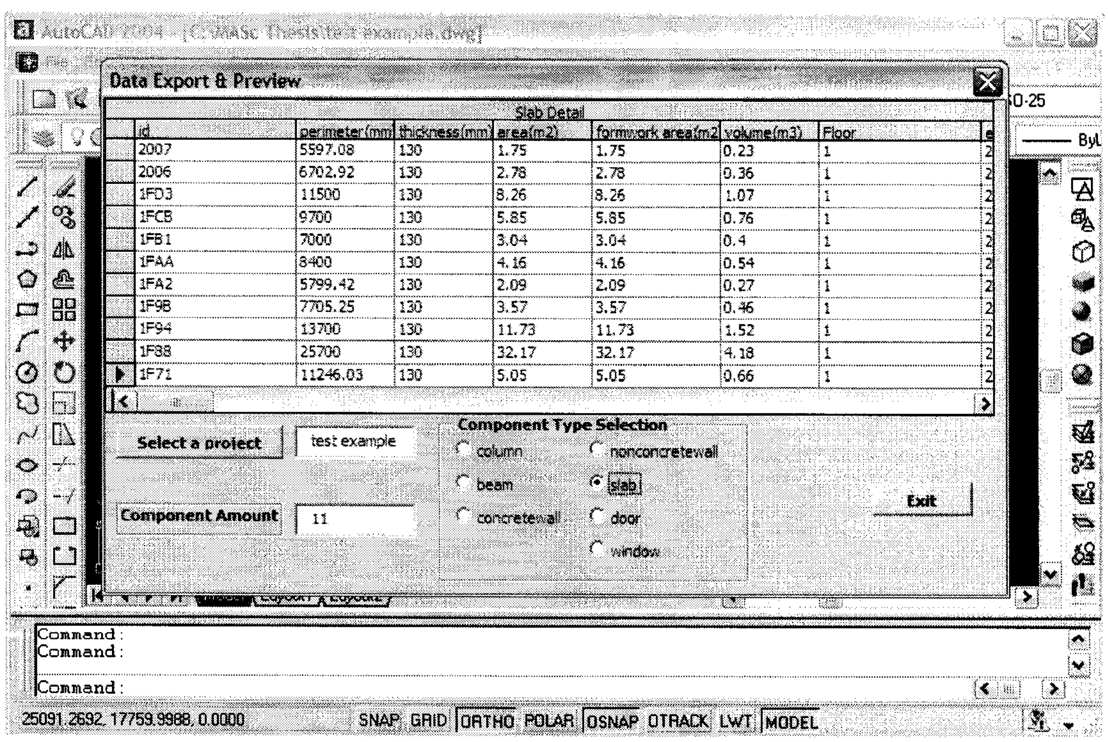


Figure 4.109 Slab Table Preview

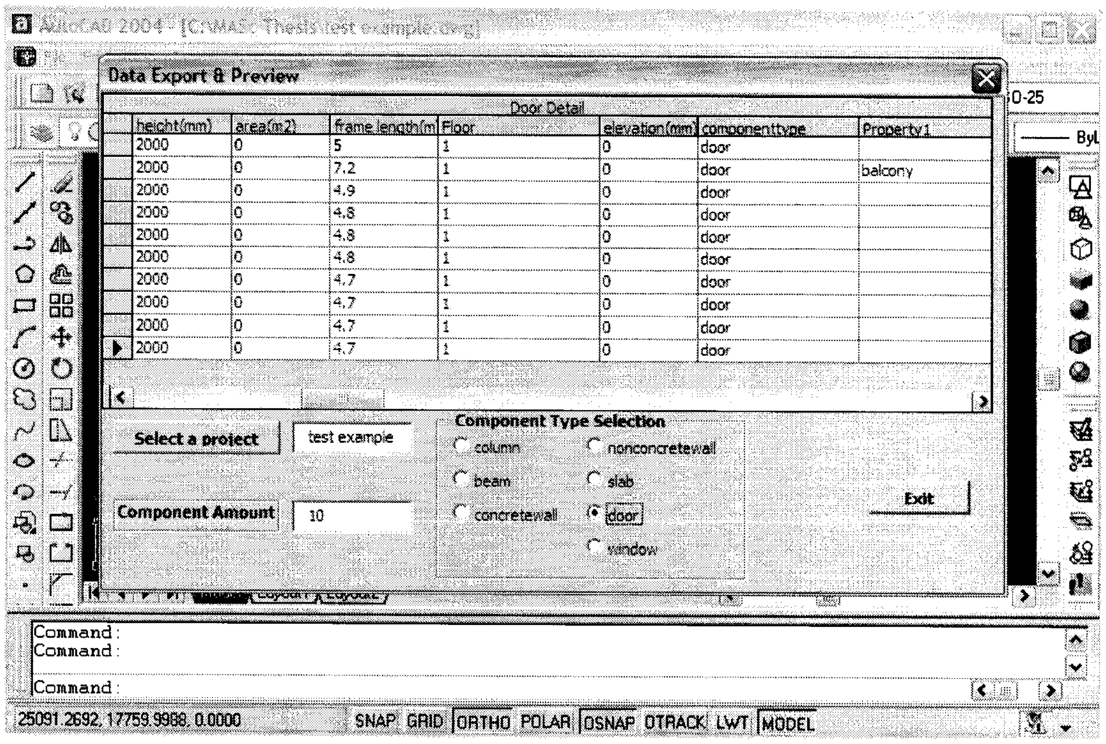


Figure 4.110 Door Table Preview

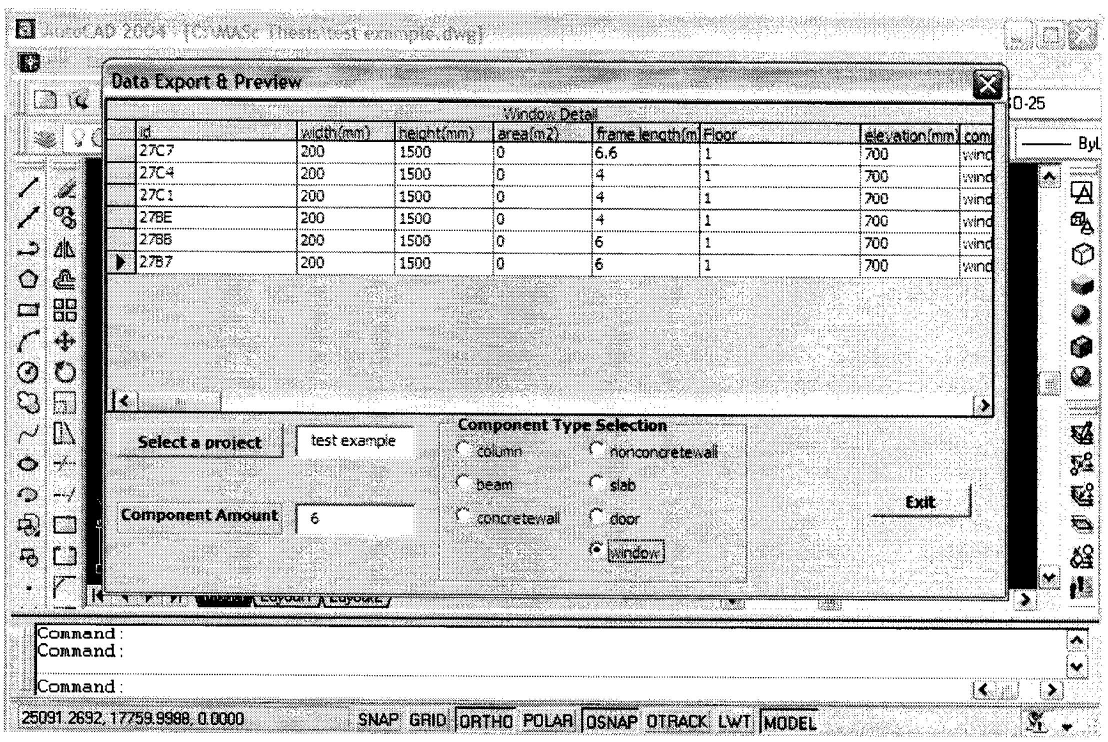


Figure 4.111 Window Table Preview

Besides the features illustrated in the forgoing paragraphs, the developed model provides a few useful auxiliary tools to facilitate the modeling process. One is "Floor Copy". The user could use this menu item to copy the model of one floor onto other floors which have the same plans and floor height as the existing floor (Figures 4.112 to 4.114).

Another practical tool is "Component Refresh." Whenever the user makes modifications of the established model, this menu item allows the user to update the model instantly. The updated information can be viewed through "Component Query". In Figure 4.115, the height of a column (highlighted) is stretched longer but without refreshing, the height information is not updated. However, in Figure 4.116, after refreshing, the height displayed is correct.

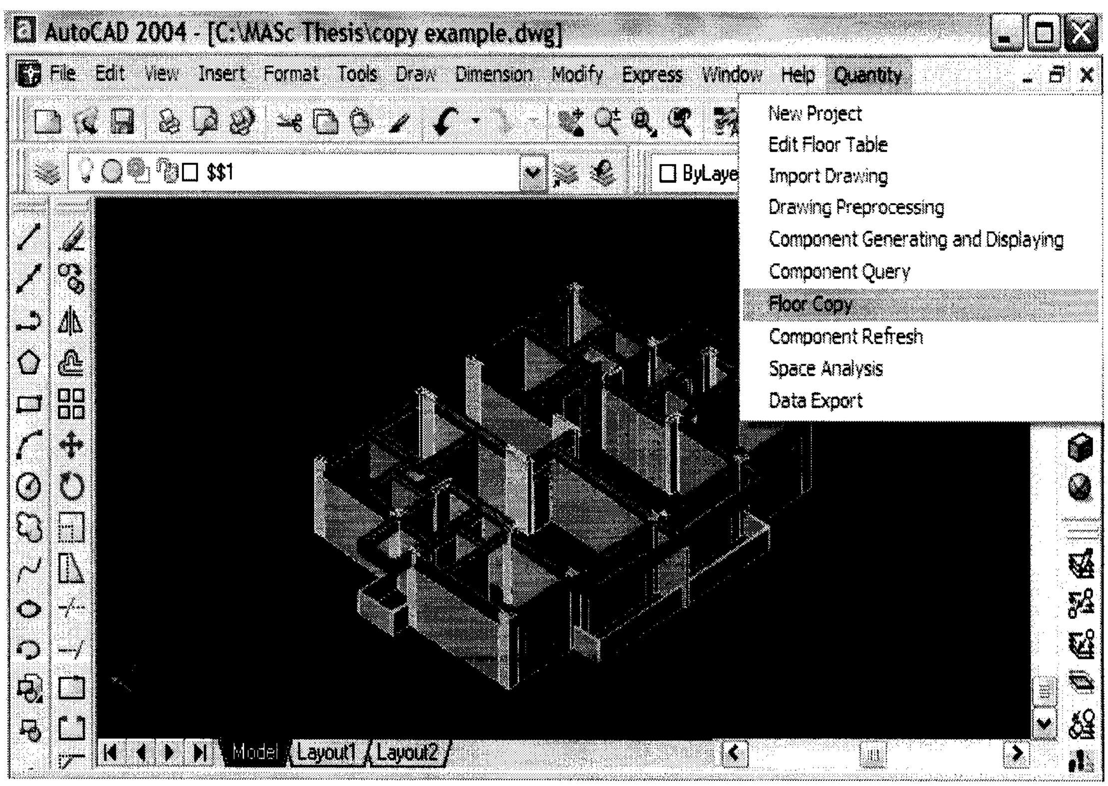


Figure 4.112 Select Floor Copy Menu Item

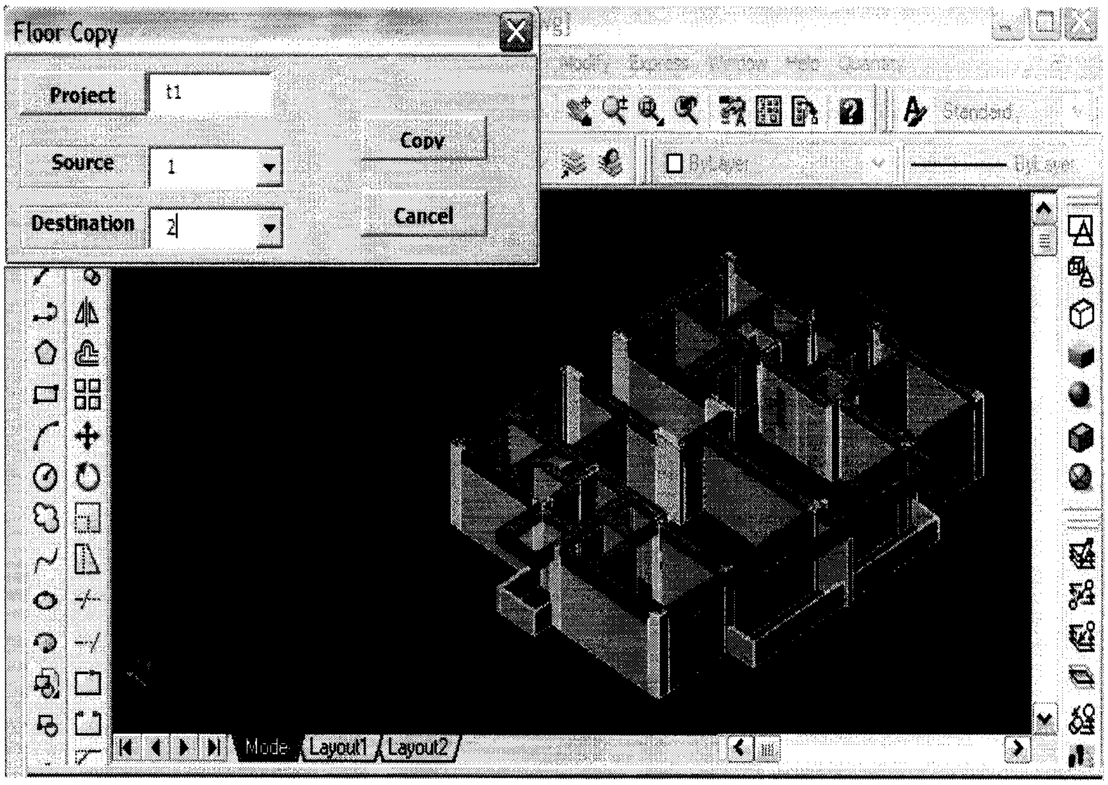


Figure 4.112 Select Project, Source Floor & Destination Floor and Hit Copy

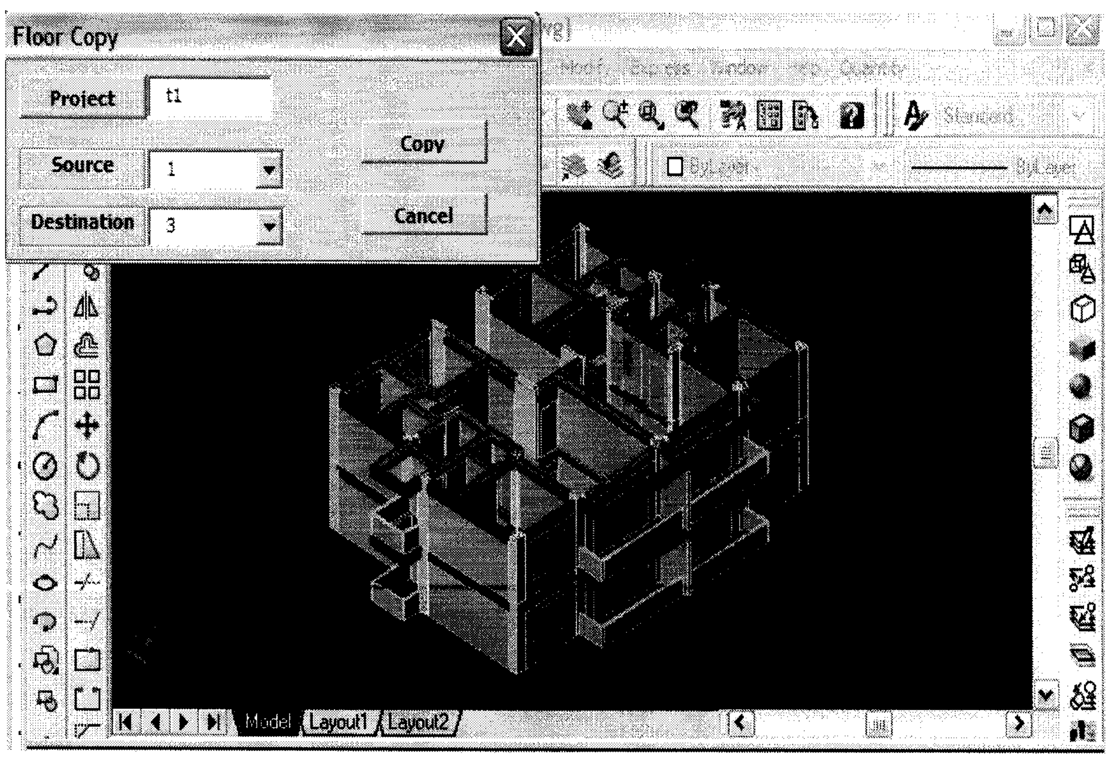


Figure 4.113 Select Destination Floor 3 and Hit Copy

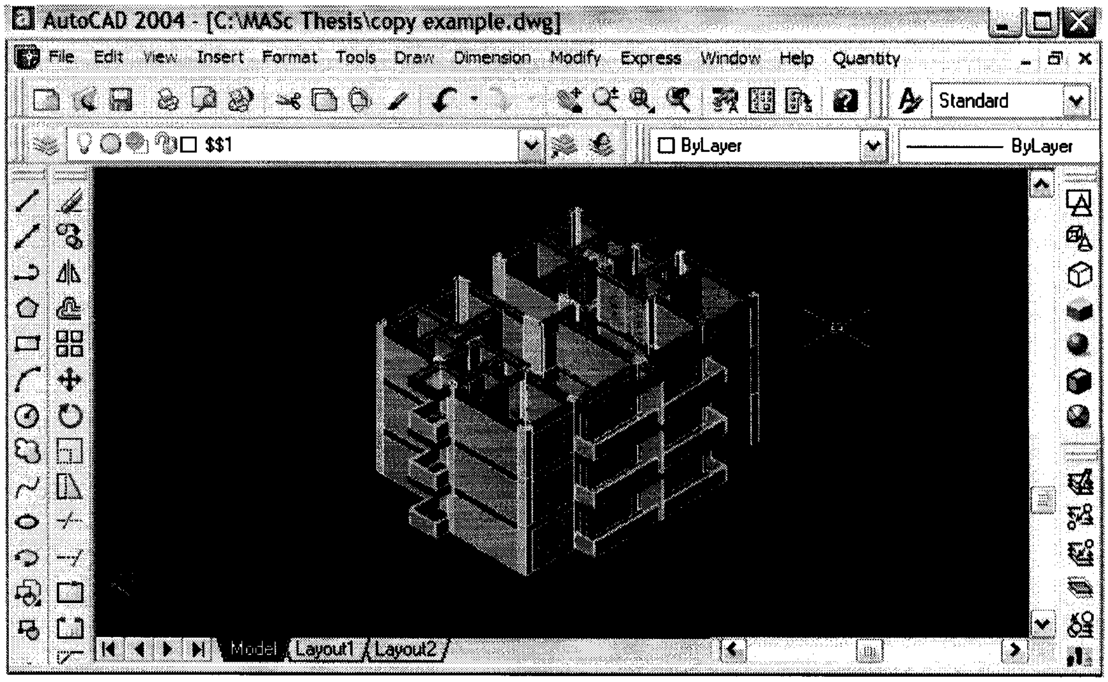


Figure 4.114 Result of Floor Copy

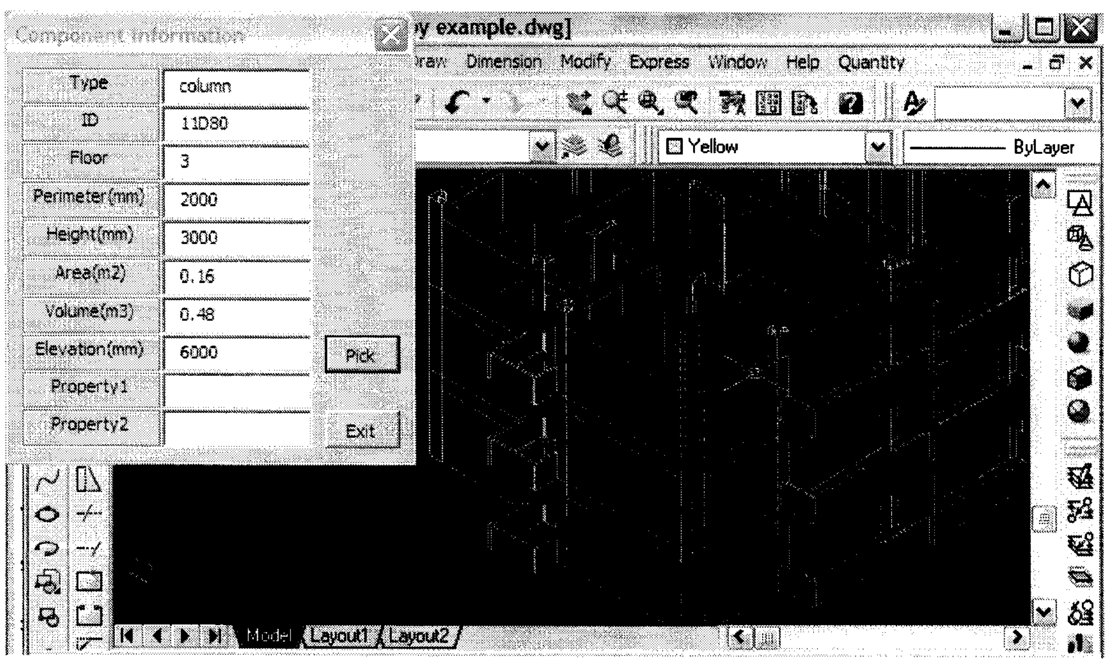


Figure 4.115 Information Before Component Refresh

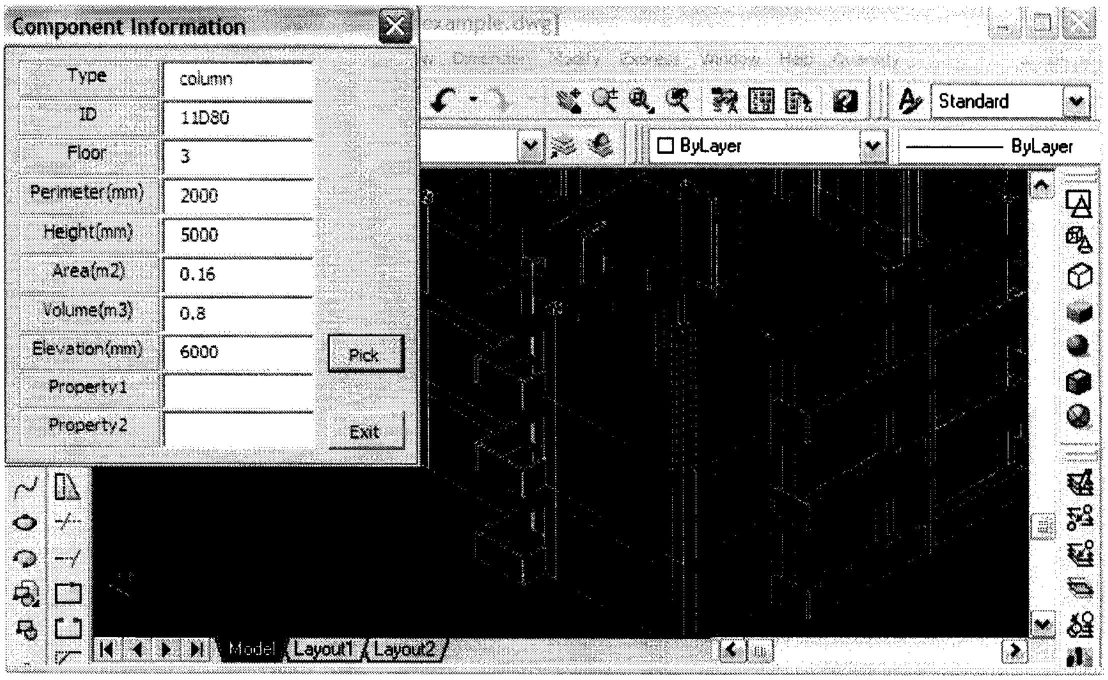


Figure 4.116 Information After Component Refresh

4.8 Summary

The implementation stage of the proposed model has been presented in this chapter. The model is implemented using VBA, ADO, and AutoLisp. There are four major components of the proposed model: the Project database module, the Project electronic drawing preprocessing module, the 3D AutoCAD modeling module and the Data export module. Each module is implemented as a global AutoCAD VBA Project. The interface of the model is user-friendly and the features it provides are powerful and practical.

CHAPTER 5

APPLICATION EXAMPLE

5.1 Introduction

This chapter presents an example to demonstrate the developed model in use. The example project is a small apartment unit drawn by the author. The apartment unit structure is a concrete framed two-way beam and slab structure. The exterior walls are made of brick. Two electronic drawings are used in the example: an architectural floor plan and a structural plan. A comparison is then performed between the results generated by the developed model and the results calculated manually to prove the validity of the model.

5.2 Quantity Take-Off Based On the Developed Model

The process of using the model to develop the quantity take-off is illustrated in the following figures.

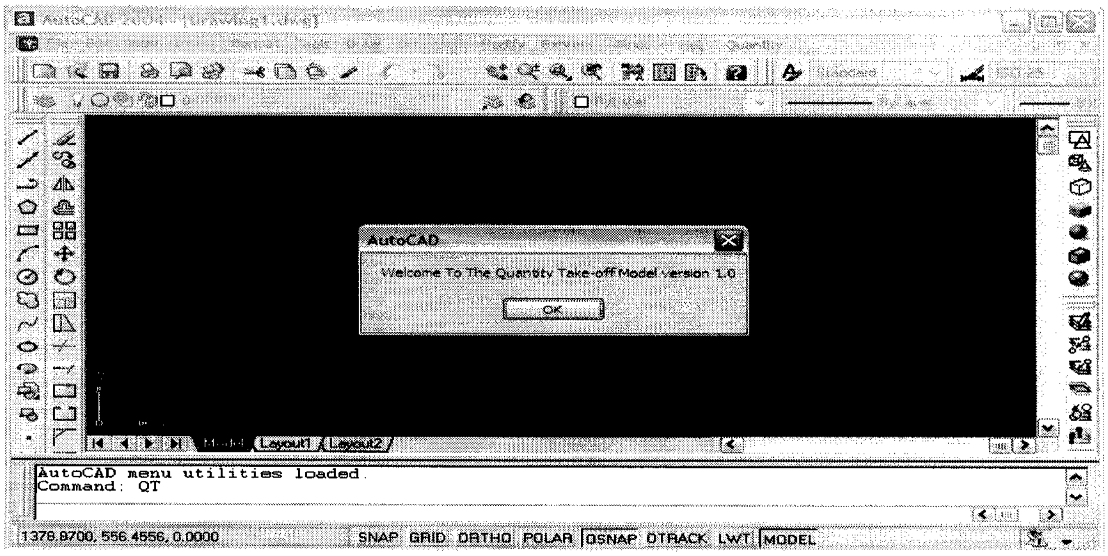


Figure 5.1 System Loading

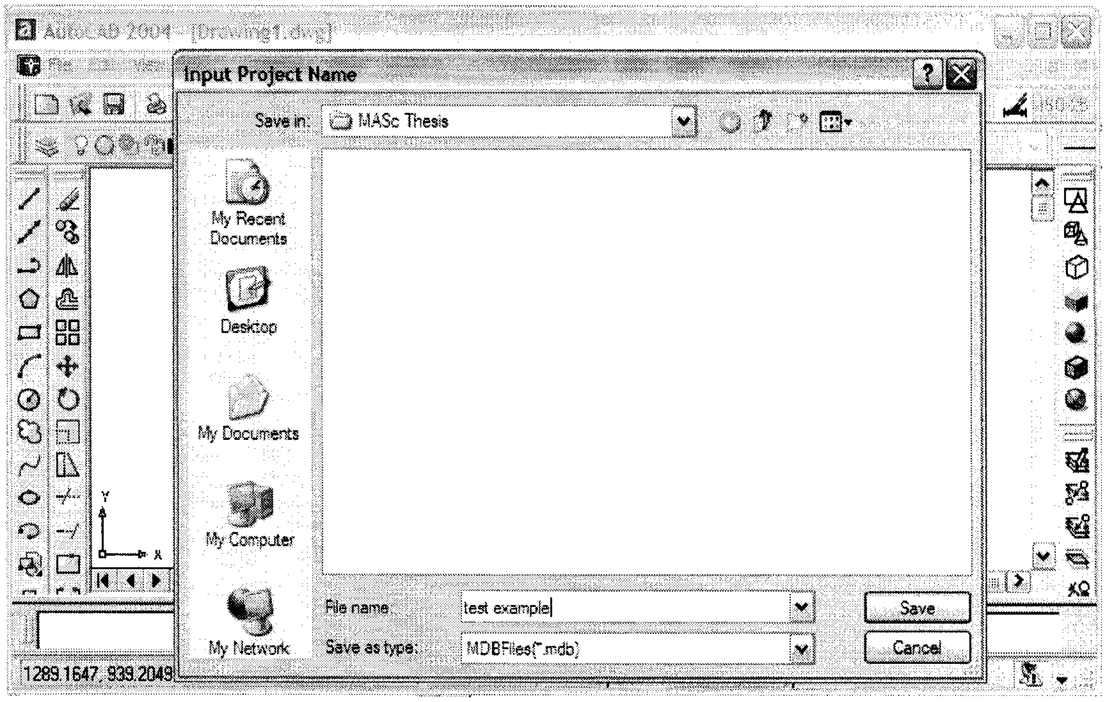


Figure 5.2 Entering the New Project Name

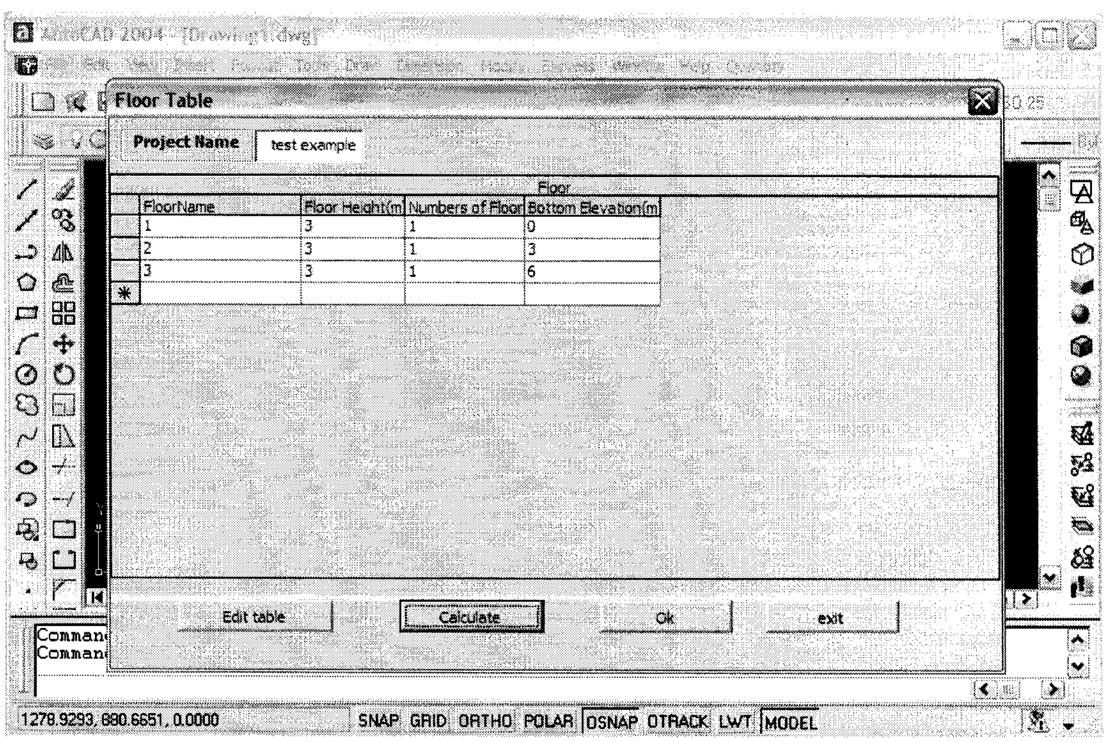


Figure 5.3 Floor Table Setting

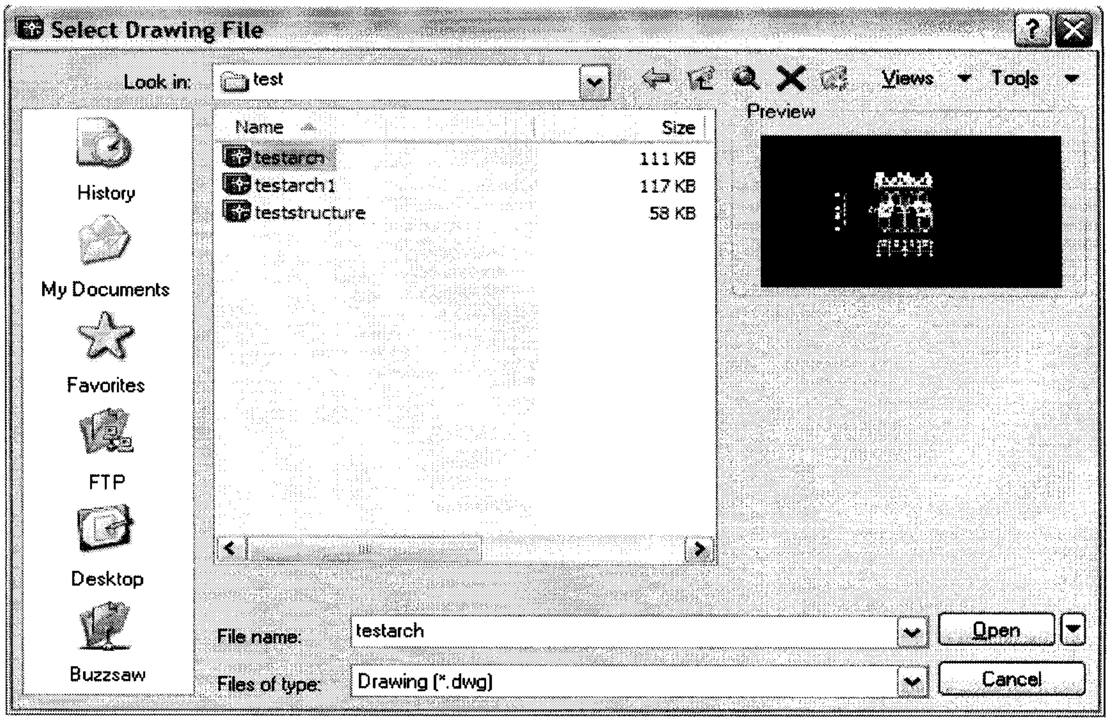


Figure 5.4 Import Drawing

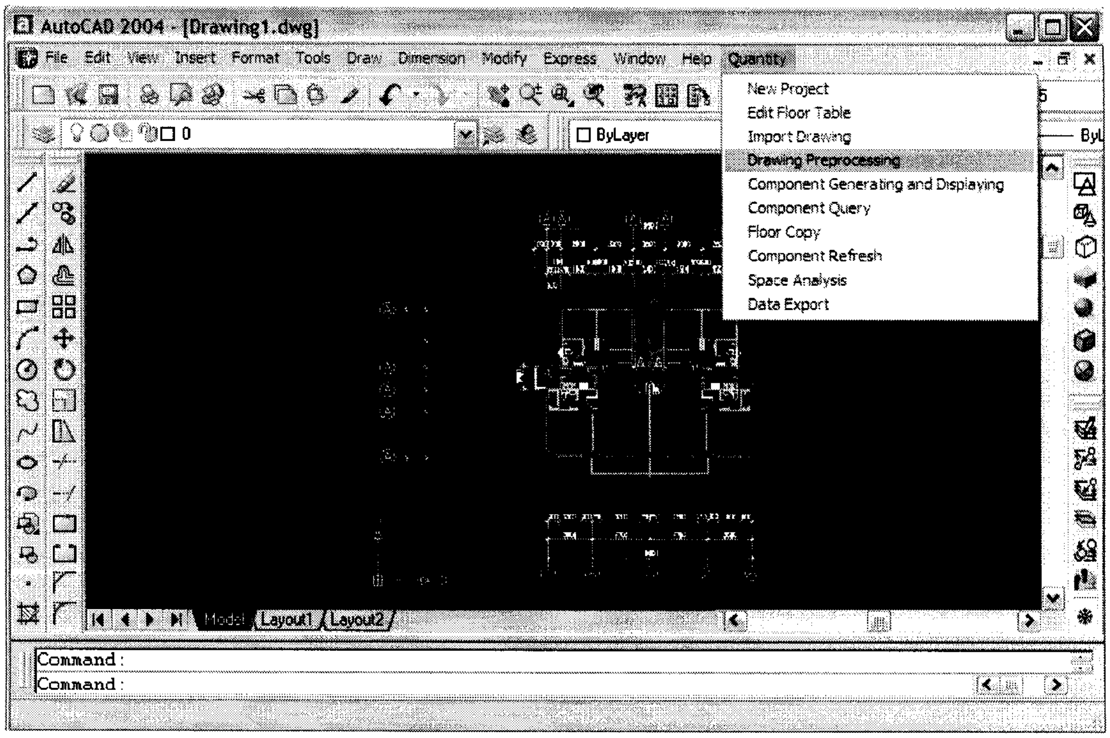


Figure 5.5 Drawing Preprocessing

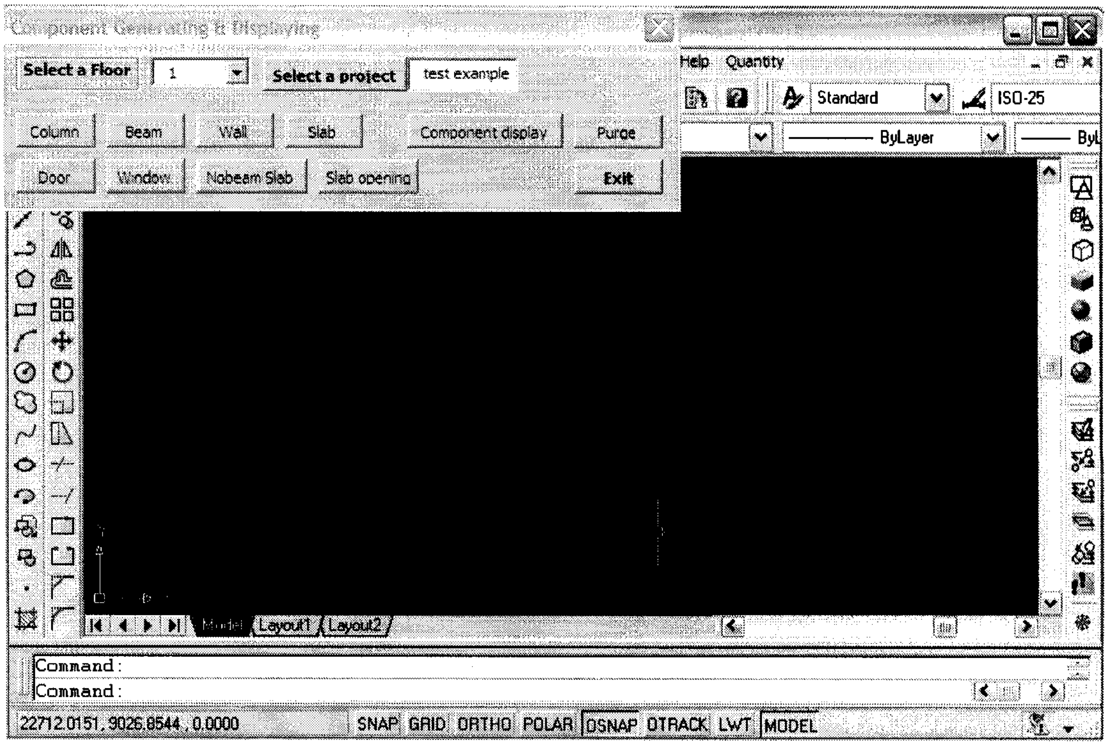


Figure 5.6 Preparing for Column Modeling

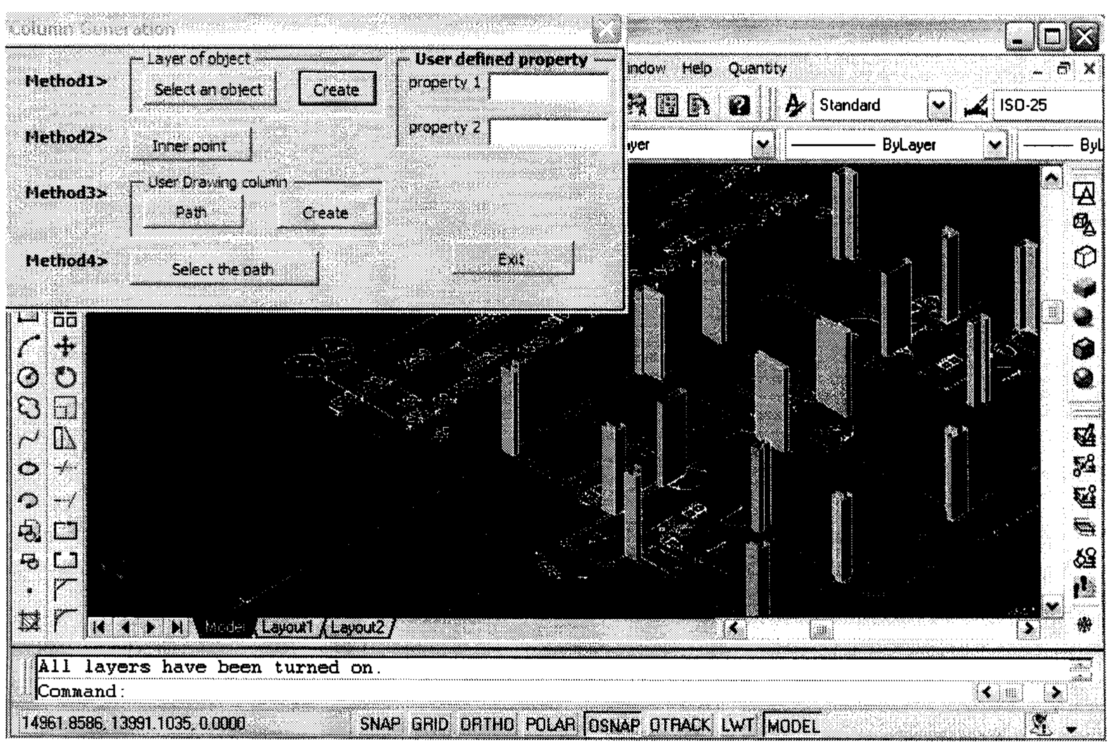


Figure 5.7 The Generated Columns

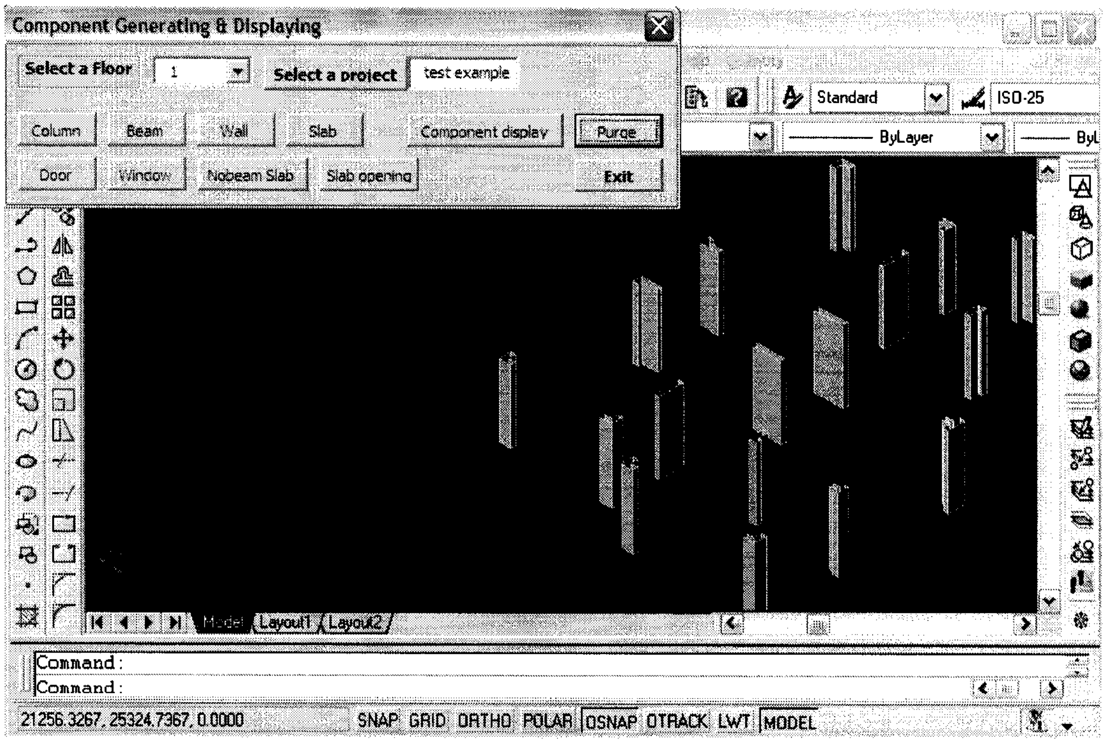


Figure 5.8 Purging the Unneeded Graphic Objects

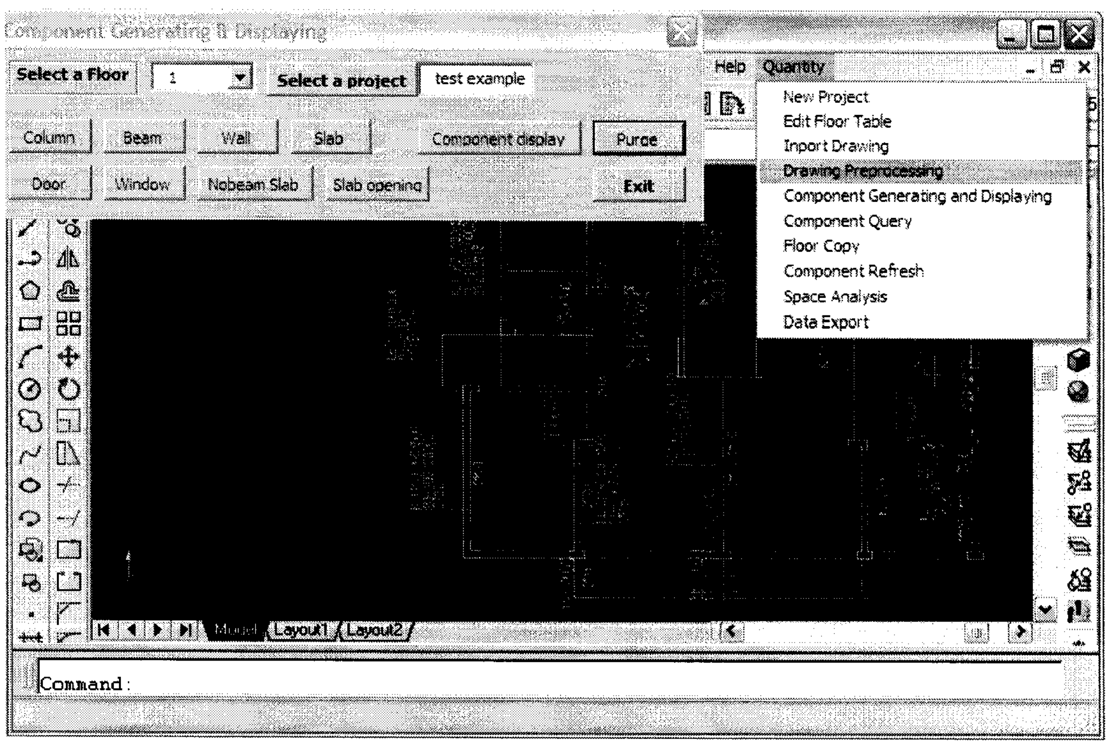


Figure 5.9 Importing Structure Plan & Preprocessing

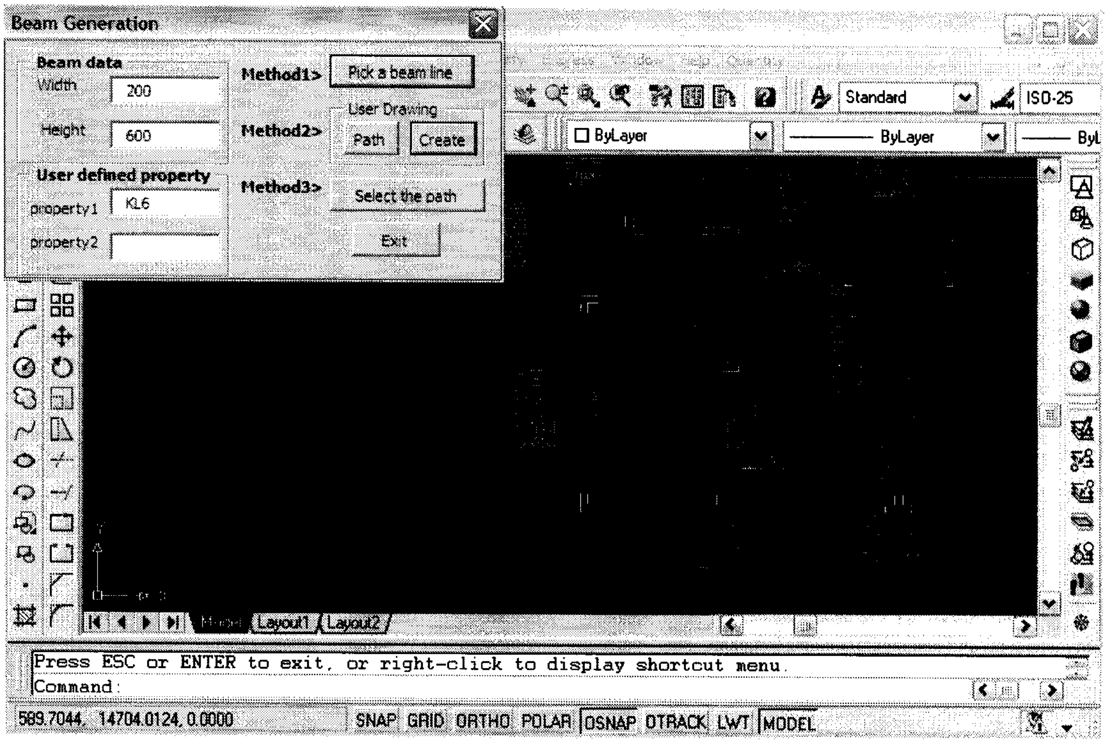


Figure 5.10 Beam Modeling

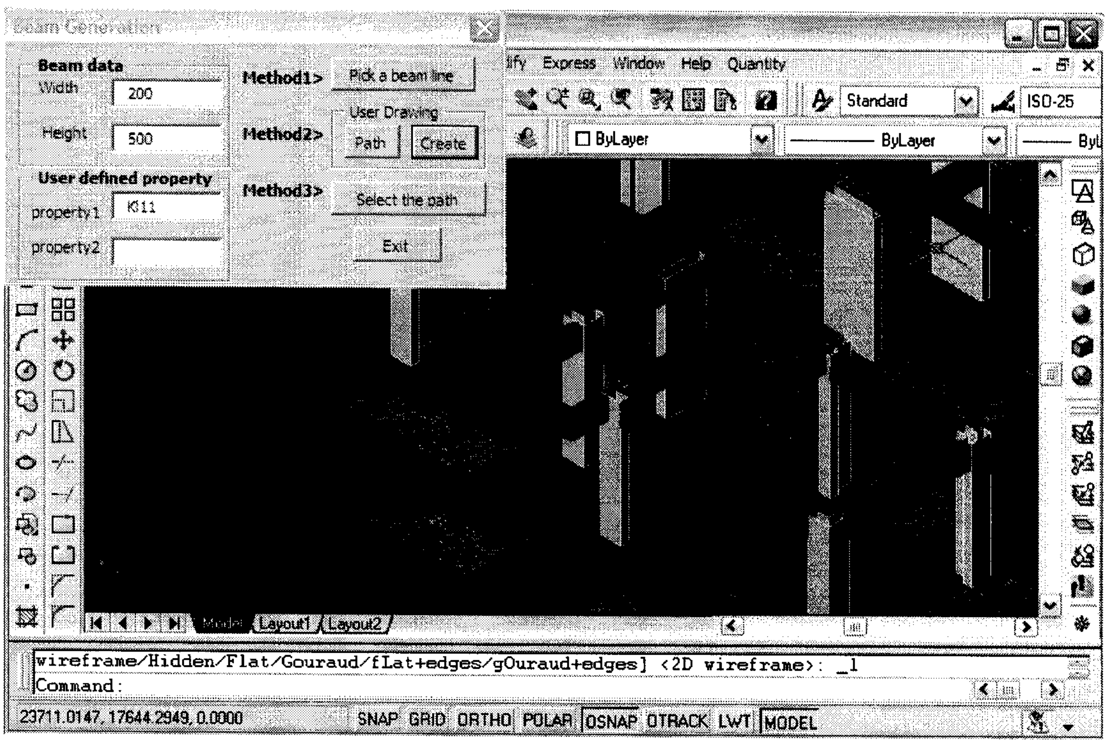


Figure 5.11 Beam Modeling (3D)

Since the apartment is a symmetrical structure, we just need to generate half of the whole model. The thickness of the slabs is assumed as 120mm.

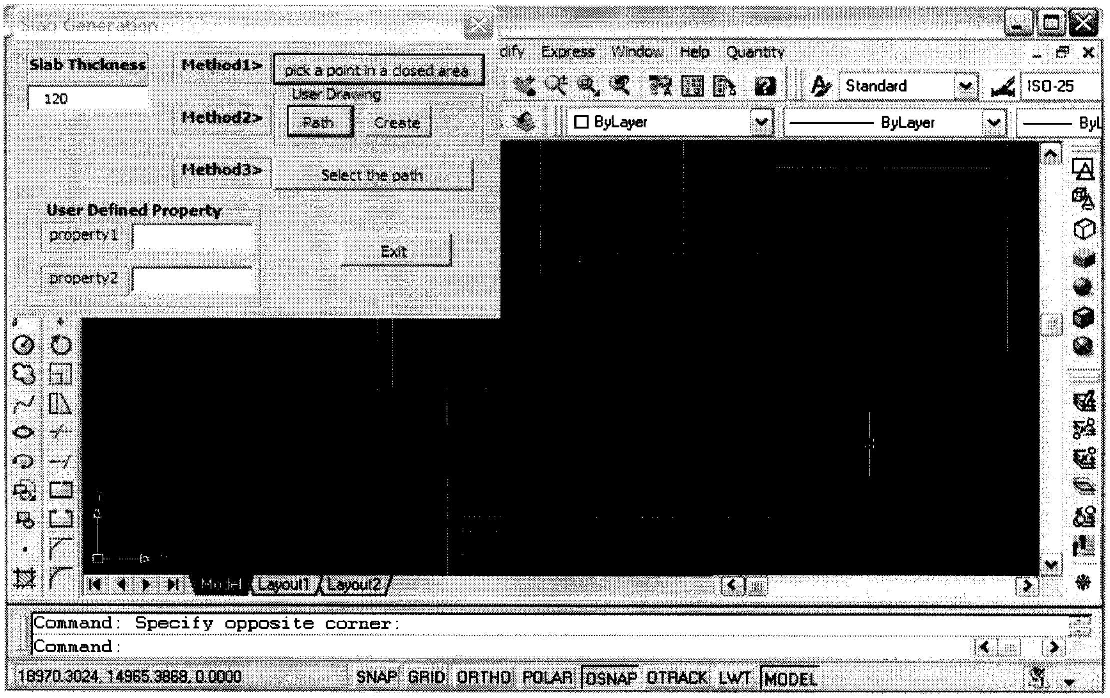


Figure 5.12 Slab Modeling

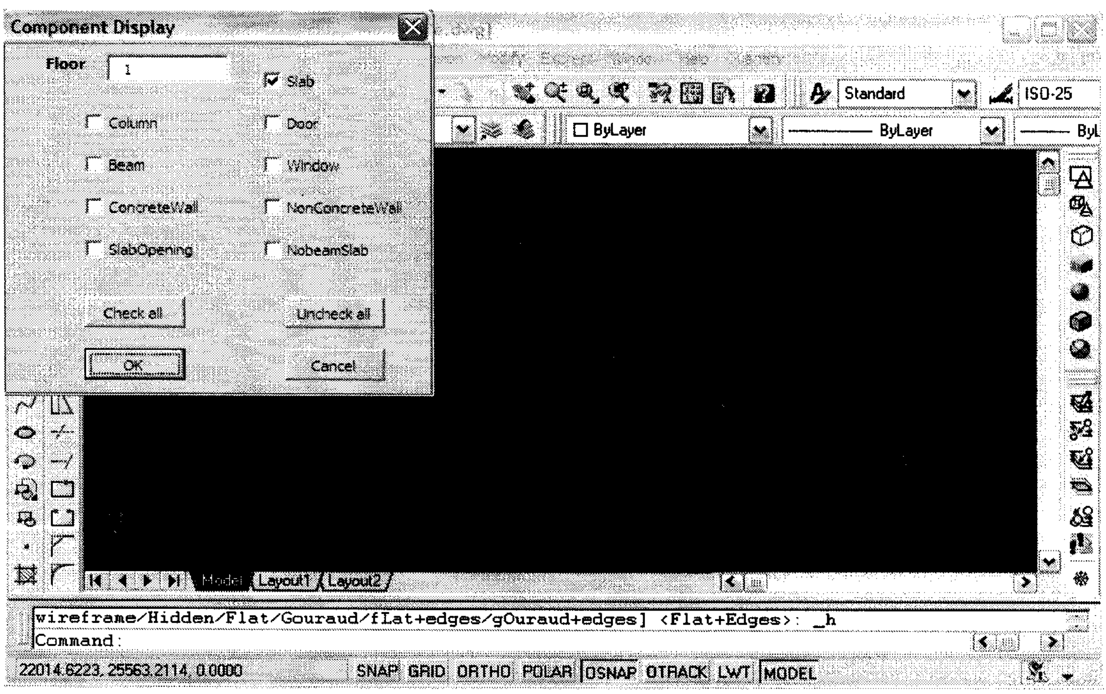


Figure 5.13 3D View of the Slabs

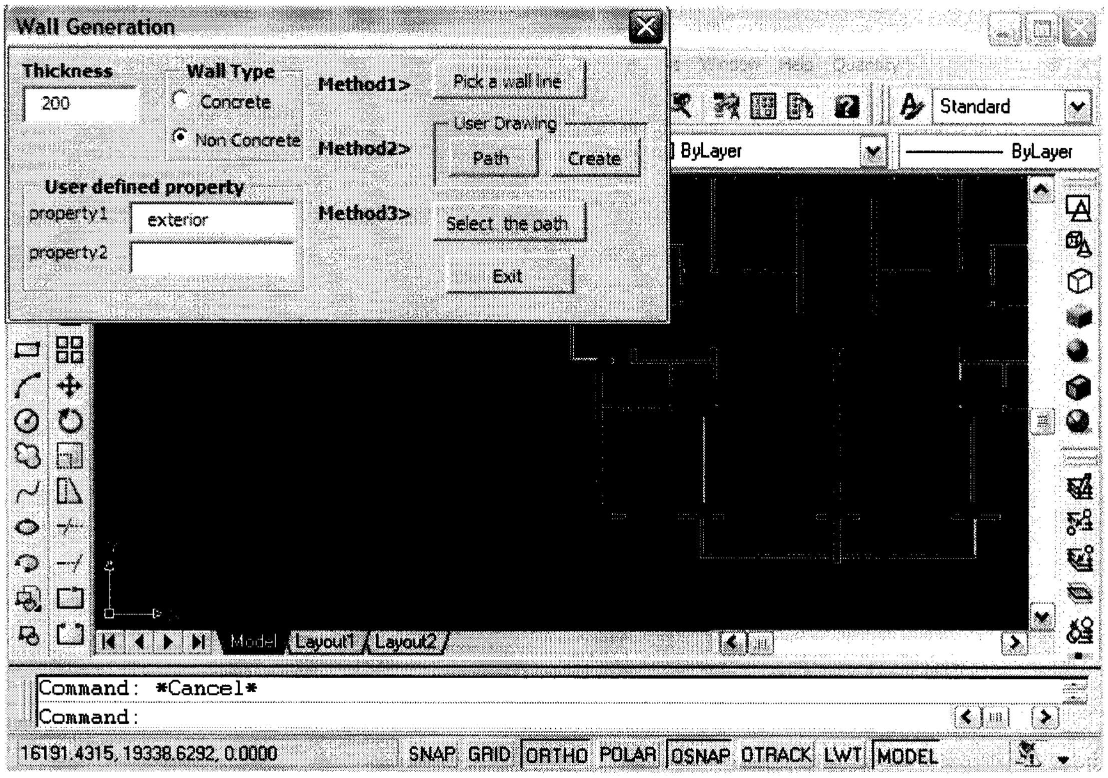


Figure 5.14 Preparing for Wall Modeling

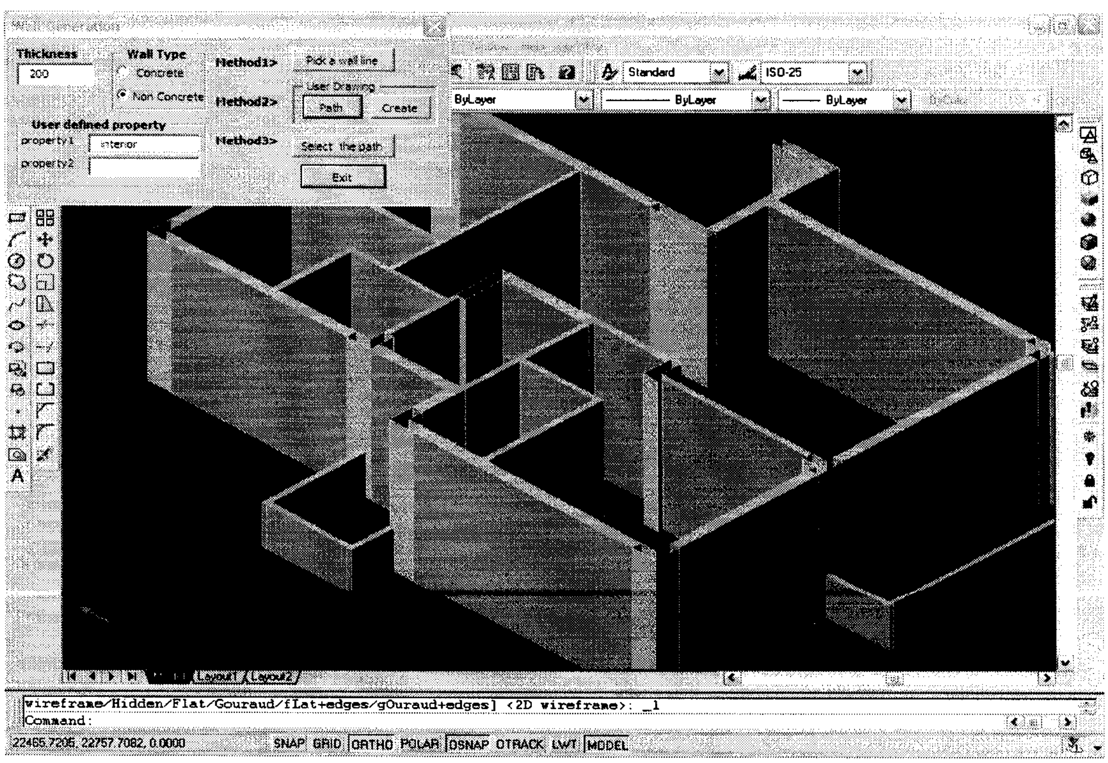


Figure 5.15 Finished Wall Modeling (3D)

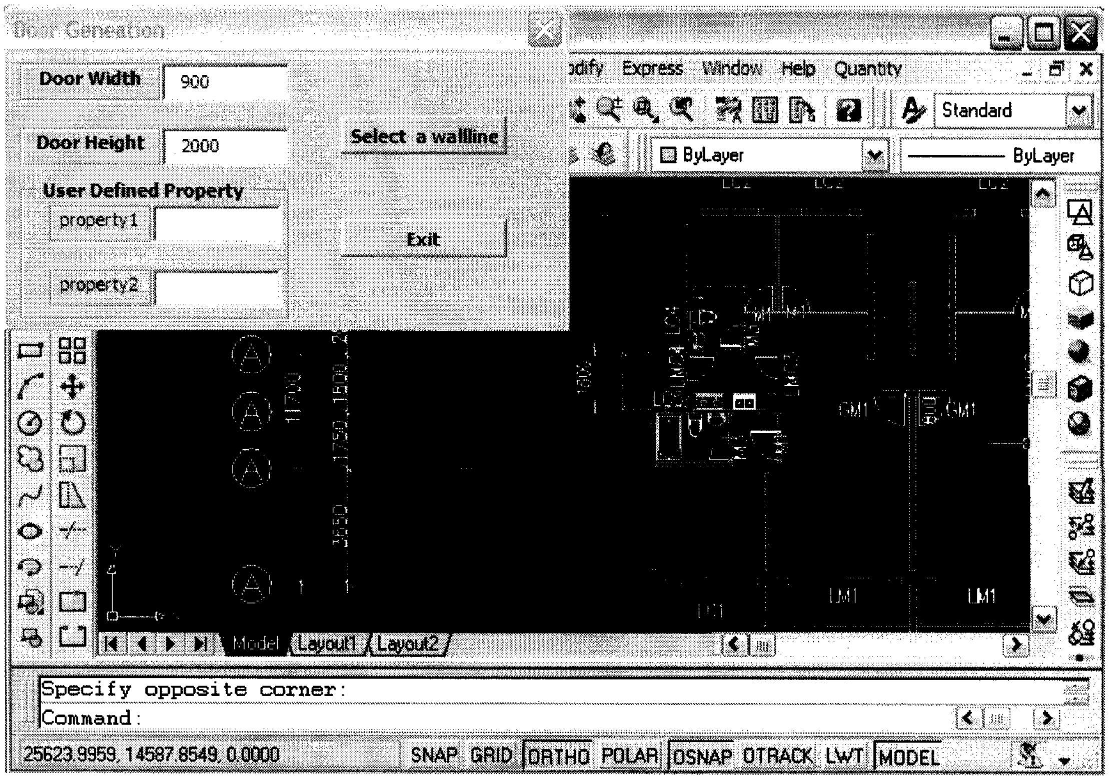


Figure 5.16 Preparing for Door Modeling

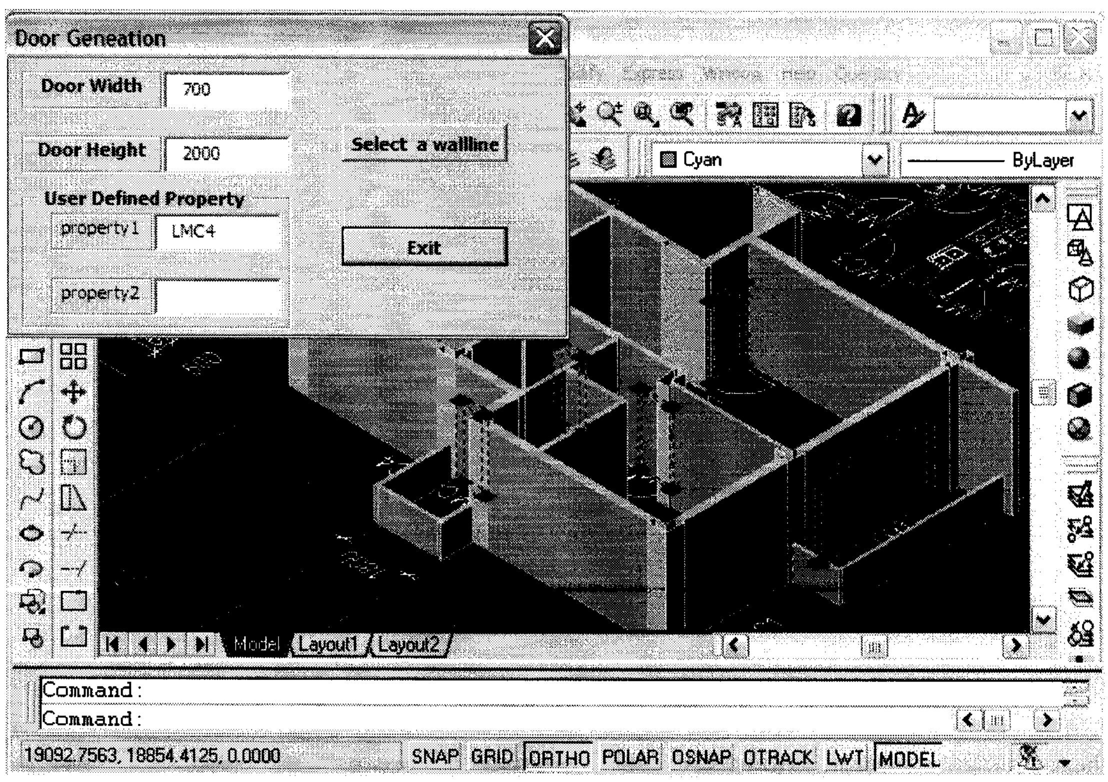


Figure 5.17 Finished Door Modeling (3D)

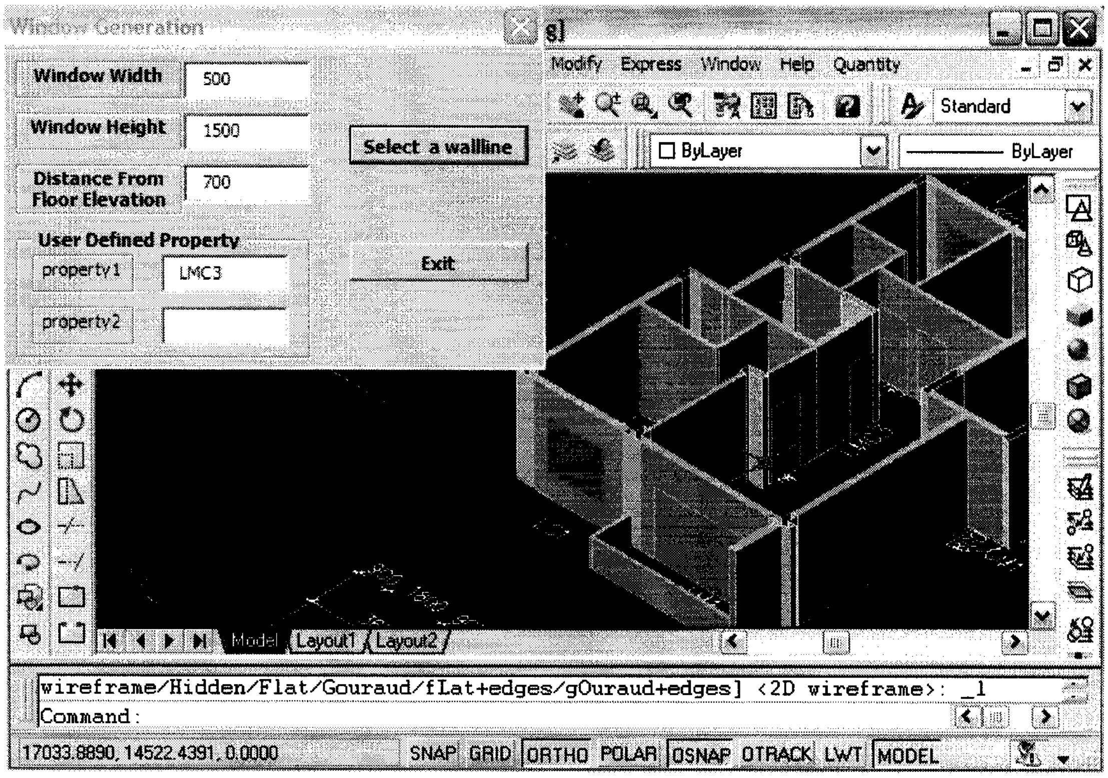


Figure 5.18 Finished Window Modeling (3D)

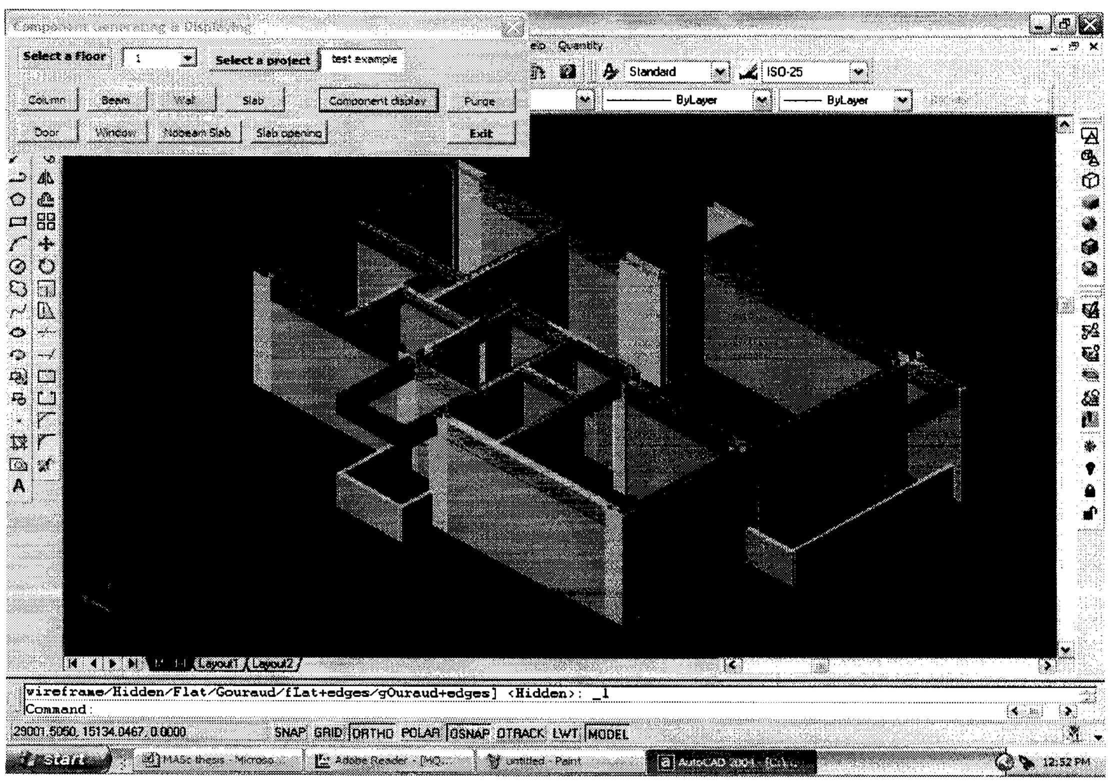


Figure 5.19 Project Model (3D)

It should be mentioned that the project model in Figure 5.18 is just a primary model. There are still some details which need to be refined. For instance, when modeling the walls, the height of the wall is assumed to be equal to the height of the floor, for the convenience of modeling. But in the real world, there are usually beams above the walls, so another module named "Space Analysis" is designed to fix the problem. By running this module, for a particular wall, if there is a beam above it, their spatial relationship will be analyzed and the height of the wall will be changed automatically. The module can also analyze the spatial relations between columns and beams to make sure their dimensions and locations are correct.

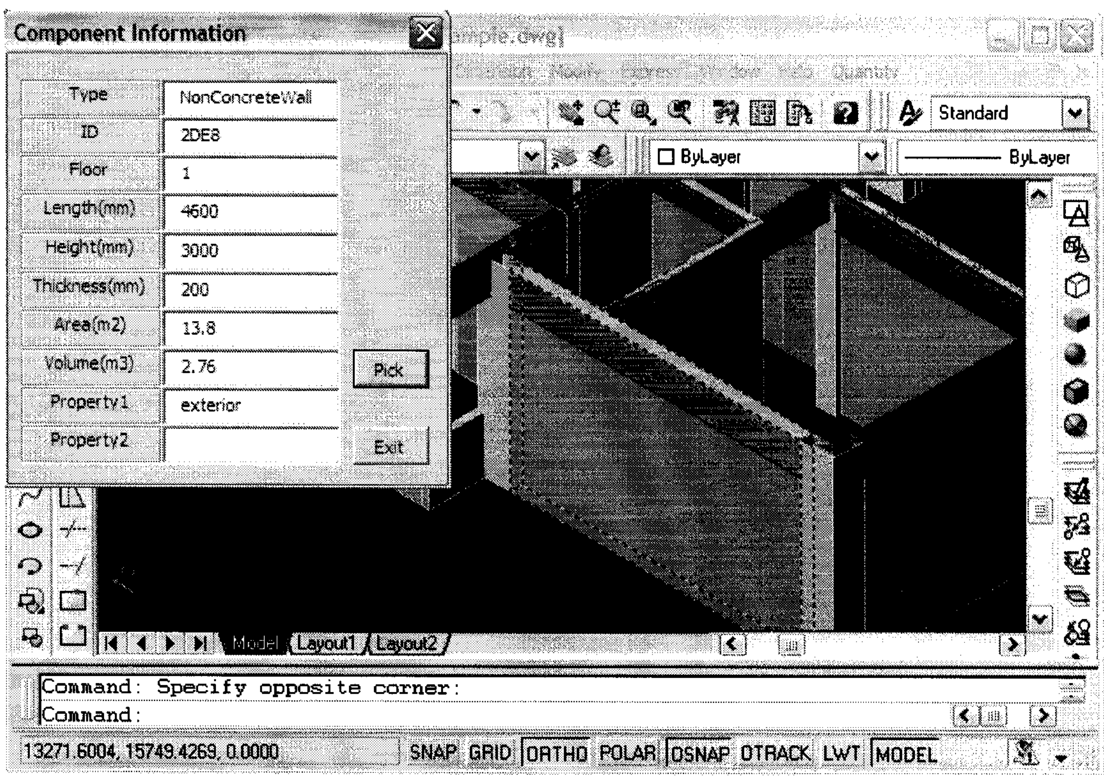


Figure 5.20 Before Space Analysis

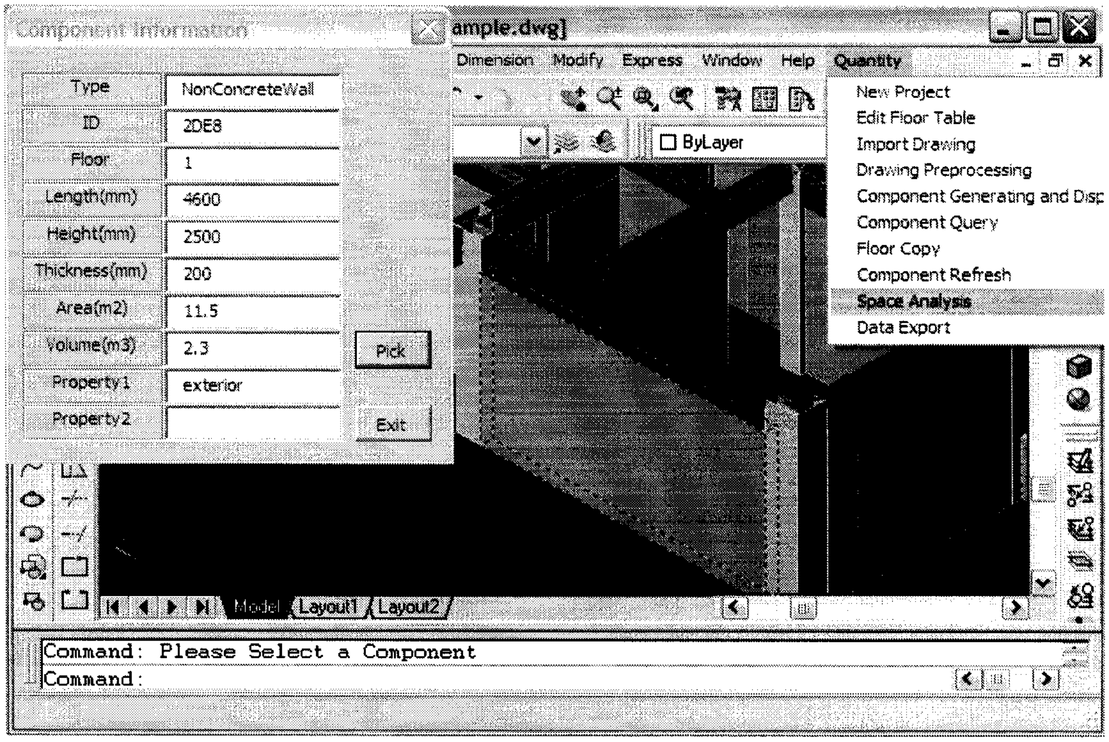


Figure 5.21 After Space Analysis

Figure 5.18 shows the component information before Space Analysis, the height of the wall (highlighted) is 3000mm which is the same as the floor height. However, after running the Space Analysis module, as we can see in Figure 5.19, the height of the wall has been changed to 2500mm. This is because the height of the beam above it has been deducted.

Now, we can export the quantity information of the components in the model to the project database. For the convenience of comparing the results with the manual calculated ones, only one unit of the apartment is modeled and its information is export to the external database. The following figures will continue the process.

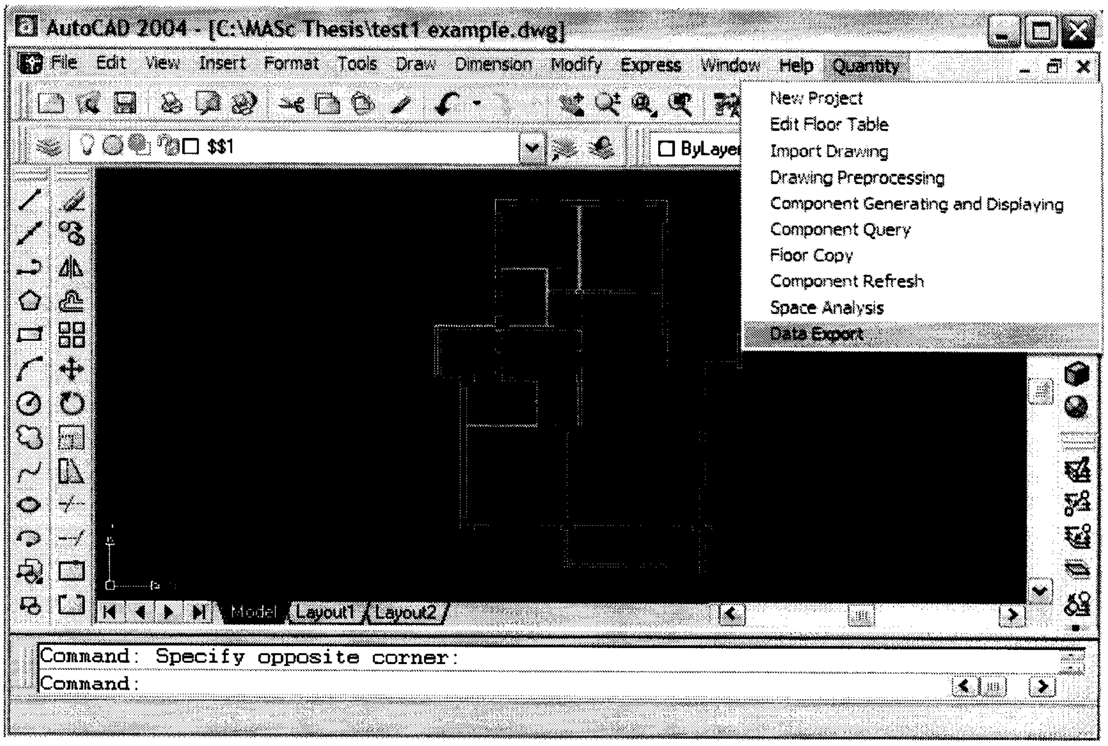


Figure 5.22 Preparing for Data Export

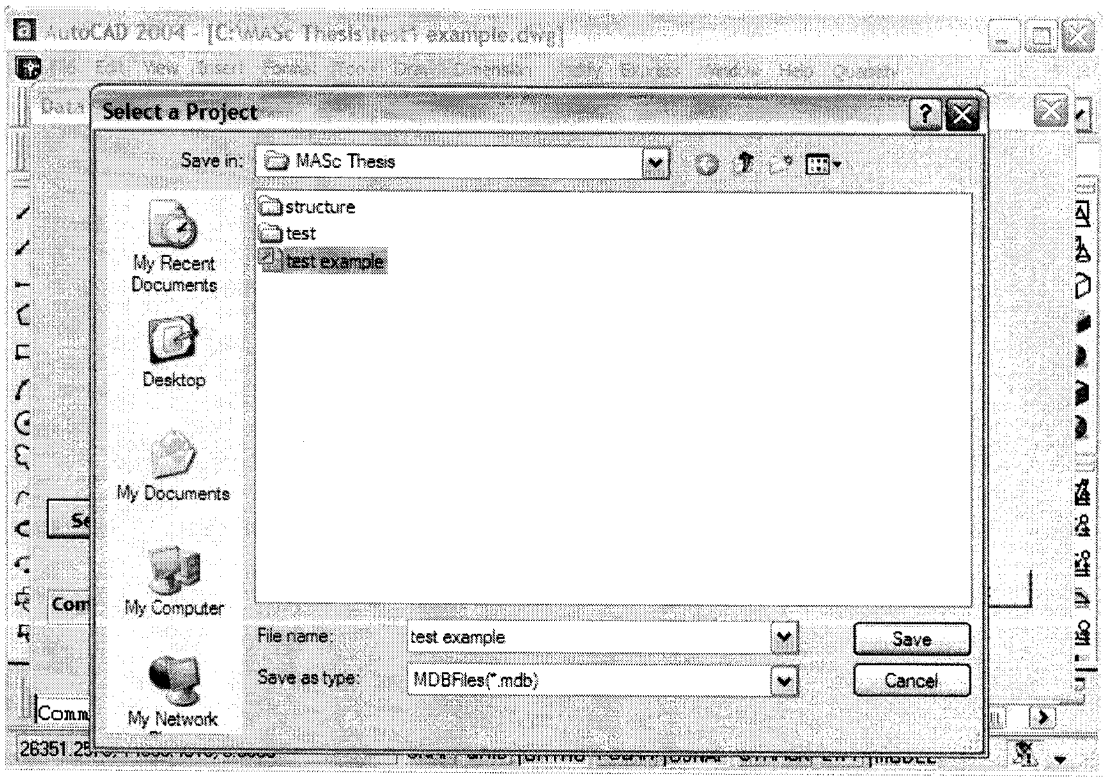
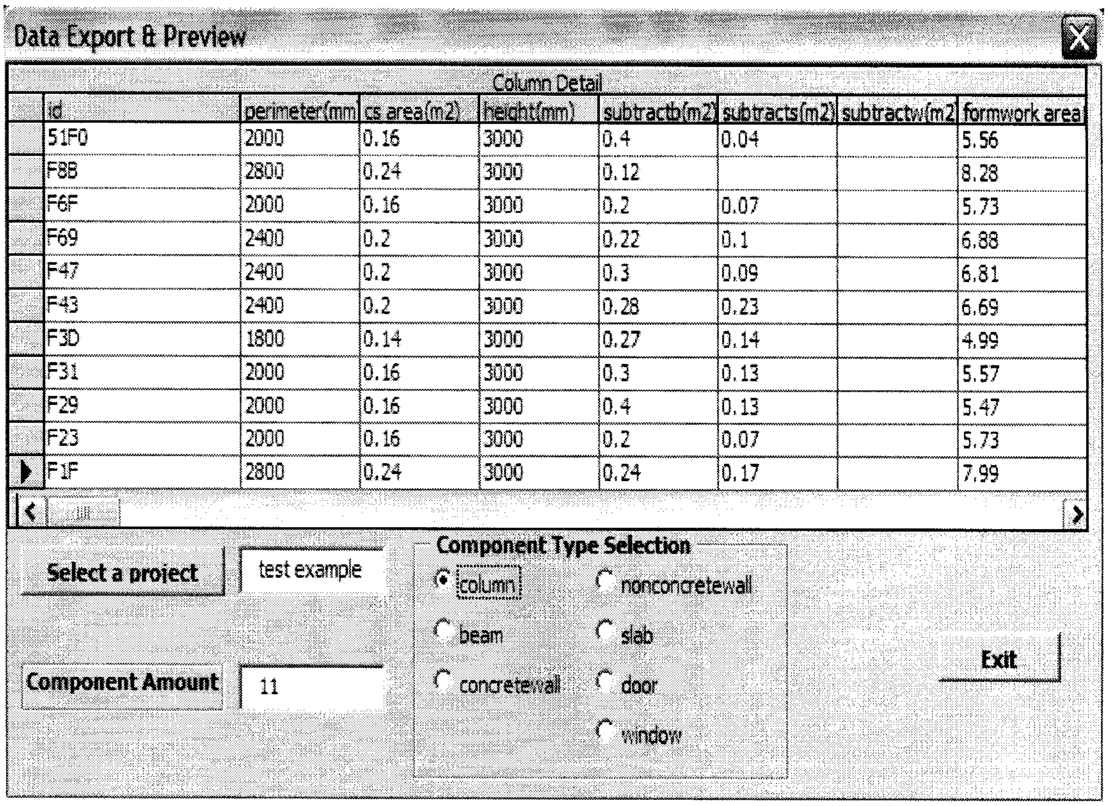
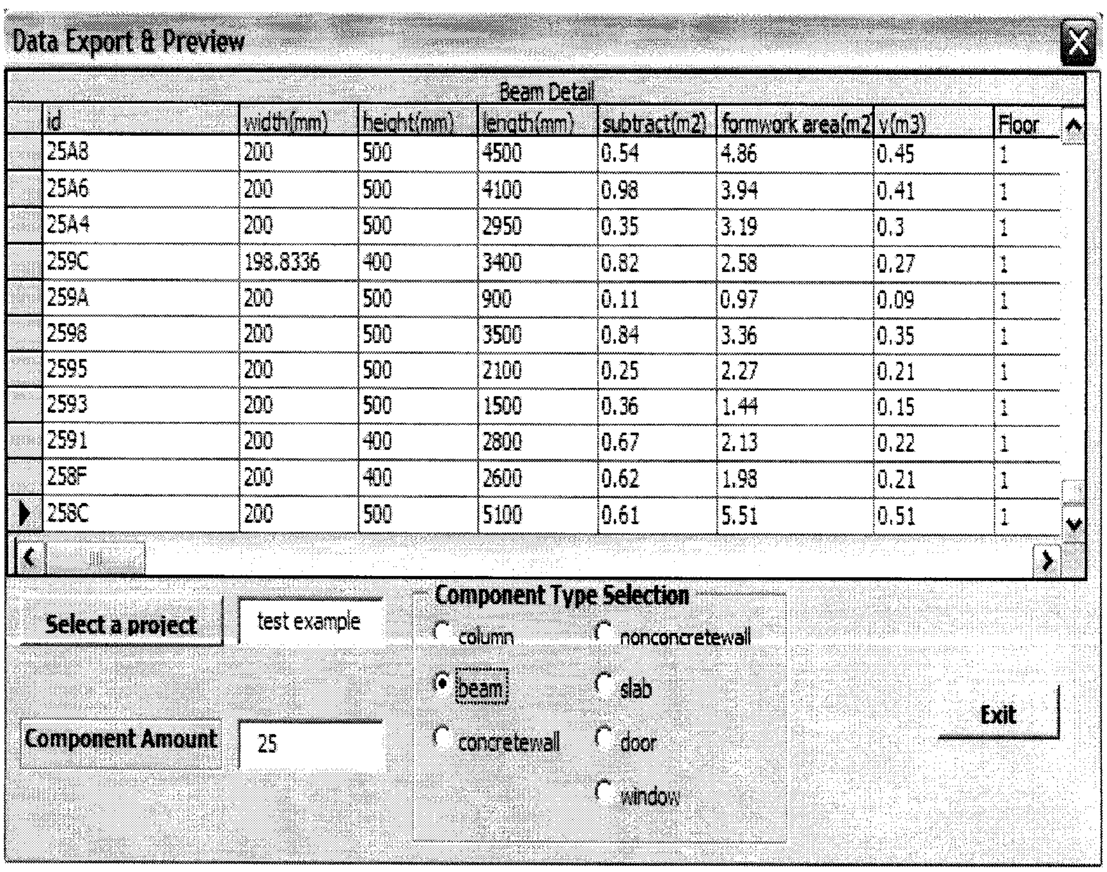


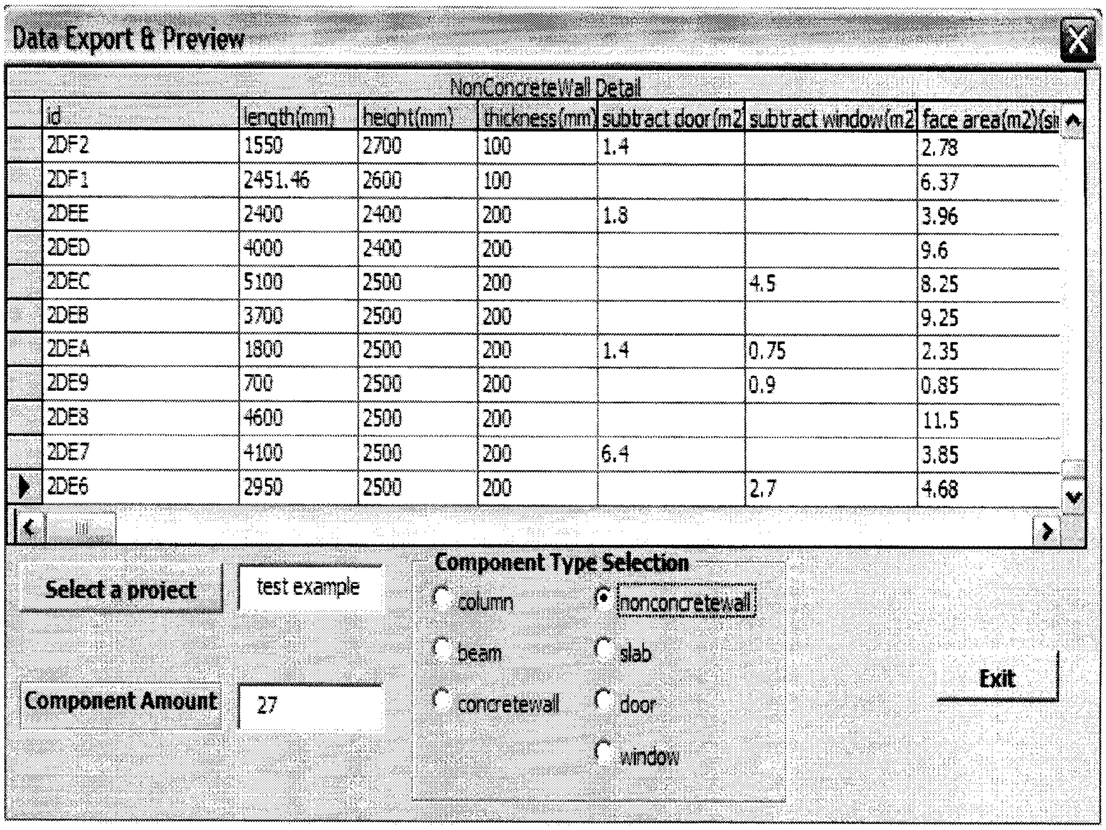
Figure 5.23 Selecting the Targeted Database



id	perimeter(mm	cs area(m2)	height(mm)	subtractb(m2)	subtracts(m2)	subtractw(m2	formwork area
51F0	2000	0.16	3000	0.4	0.04		5.56
F8B	2800	0.24	3000	0.12			8.28
F6F	2000	0.16	3000	0.2	0.07		5.73
F69	2400	0.2	3000	0.22	0.1		6.88
F47	2400	0.2	3000	0.3	0.09		6.81
F43	2400	0.2	3000	0.28	0.23		6.69
F3D	1800	0.14	3000	0.27	0.14		4.99
F31	2000	0.16	3000	0.3	0.13		5.57
F29	2000	0.16	3000	0.4	0.13		5.47
F23	2000	0.16	3000	0.2	0.07		5.73
F1F	2800	0.24	3000	0.24	0.17		7.99

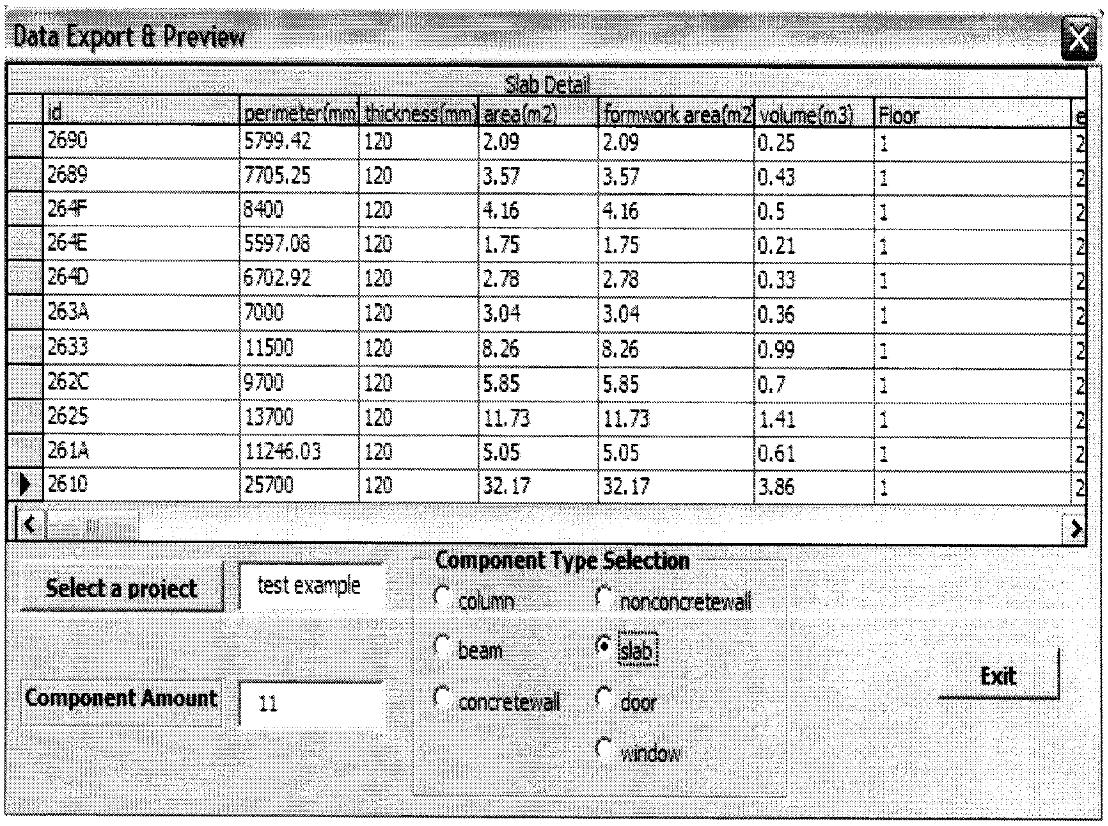


id	width(mm)	height(mm)	length(mm)	subtract(m2)	formwork area(m2	v(m3)	Floor
25A8	200	500	4500	0.54	4.86	0.45	1
25A6	200	500	4100	0.98	3.94	0.41	1
25A4	200	500	2950	0.35	3.19	0.3	1
259C	198.8336	400	3400	0.82	2.58	0.27	1
259A	200	500	900	0.11	0.97	0.09	1
2598	200	500	3500	0.84	3.36	0.35	1
2595	200	500	2100	0.25	2.27	0.21	1
2593	200	500	1500	0.36	1.44	0.15	1
2591	200	400	2800	0.67	2.13	0.22	1
258F	200	400	2600	0.62	1.98	0.21	1
258C	200	500	5100	0.61	5.51	0.51	1



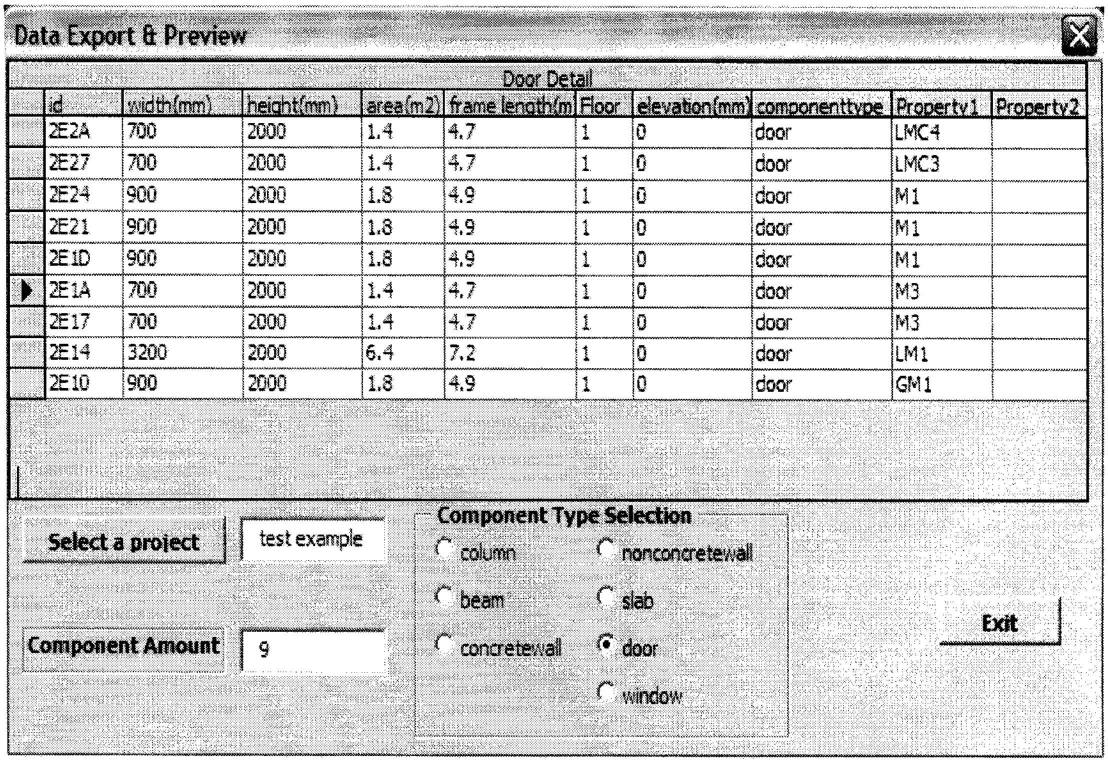
NonConcreteWall Detail

id	length(mm)	height(mm)	thickness(mm)	subtract door(m2	subtract window(m2	face area(m2)(si
2DF2	1550	2700	100	1.4		2.78
2DF1	2451.46	2600	100			6.37
2DEE	2400	2400	200	1.8		3.96
2DED	4000	2400	200			9.6
2DEC	5100	2500	200		4.5	8.25
2DEB	3700	2500	200			9.25
2DEA	1800	2500	200	1.4	0.75	2.35
2DE9	700	2500	200		0.9	0.85
2DE8	4600	2500	200			11.5
2DE7	4100	2500	200	6.4		3.85
2DE6	2950	2500	200		2.7	4.68

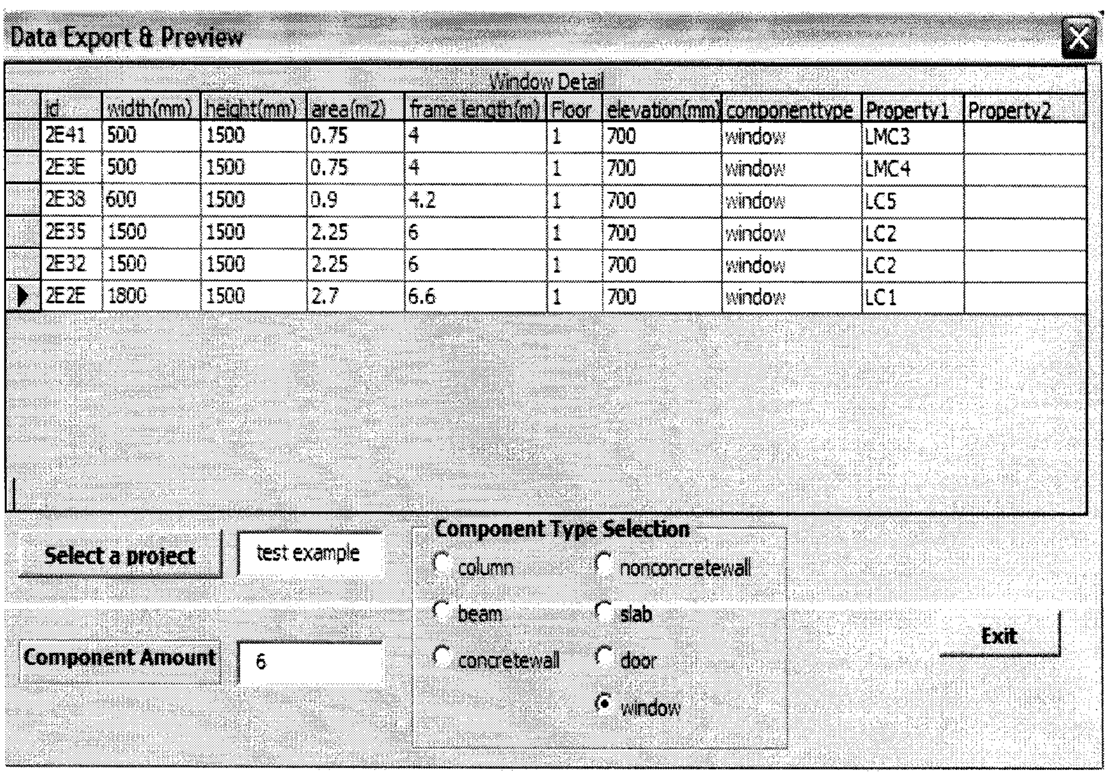


Slab Detail

id	perimeter(mm	thickness(mm)	area(m2)	formwork area(m2	volume(m3)	Floor	e
2690	5799.42	120	2.09	2.09	0.25	1	2
2689	7705.25	120	3.57	3.57	0.43	1	2
264F	8400	120	4.16	4.16	0.5	1	2
264E	5597.08	120	1.75	1.75	0.21	1	2
264D	6702.92	120	2.78	2.78	0.33	1	2
263A	7000	120	3.04	3.04	0.36	1	2
2633	11500	120	8.26	8.26	0.99	1	2
262C	9700	120	5.85	5.85	0.7	1	2
2625	13700	120	11.73	11.73	1.41	1	2
261A	11246.03	120	5.05	5.05	0.61	1	2
2610	25700	120	32.17	32.17	3.86	1	2



id	width(mm)	height(mm)	area(m2)	frame length(m	Floor	elevation(mm)	componenttype	Property1	Property2
2E2A	700	2000	1.4	4.7	1	0	door	LMC4	
2E27	700	2000	1.4	4.7	1	0	door	LMC3	
2E24	900	2000	1.8	4.9	1	0	door	M1	
2E21	900	2000	1.8	4.9	1	0	door	M1	
2E1D	900	2000	1.8	4.9	1	0	door	M1	
2E1A	700	2000	1.4	4.7	1	0	door	M3	
2E17	700	2000	1.4	4.7	1	0	door	M3	
2E14	3200	2000	6.4	7.2	1	0	door	LM1	
2E10	900	2000	1.8	4.9	1	0	door	GM1	



id	width(mm)	height(mm)	area(m2)	frame length(m)	Floor	elevation(mm)	componenttype	Property1	Property2
2E41	500	1500	0.75	4	1	700	window	LMC3	
2E3E	500	1500	0.75	4	1	700	window	LMC4	
2E38	600	1500	0.9	4.2	1	700	window	LC5	
2E35	1500	1500	2.25	6	1	700	window	LC2	
2E32	1500	1500	2.25	6	1	700	window	LC2	
2E2E	1800	1500	2.7	6.6	1	700	window	LC1	

Figure 5.24 Preview of the Populated Database

If users want to check the details of the populated project database, they can enter the directory in which the project file resides and open the database and its tables (see Figures 5.23 and 5.24). The Microsoft Access 2003 embedded tool "Analyze It with Microsoft Office Excel" (Figure 5.24) allows us to analyze each table in the database with the powerful features of Microsoft Office Excel. Figures 5.25 through 5.32 show some simple analyses such as summarization of columns in a table. This will let us know the quantity information of the total volume of columns, beams, slabs and form work, etc.

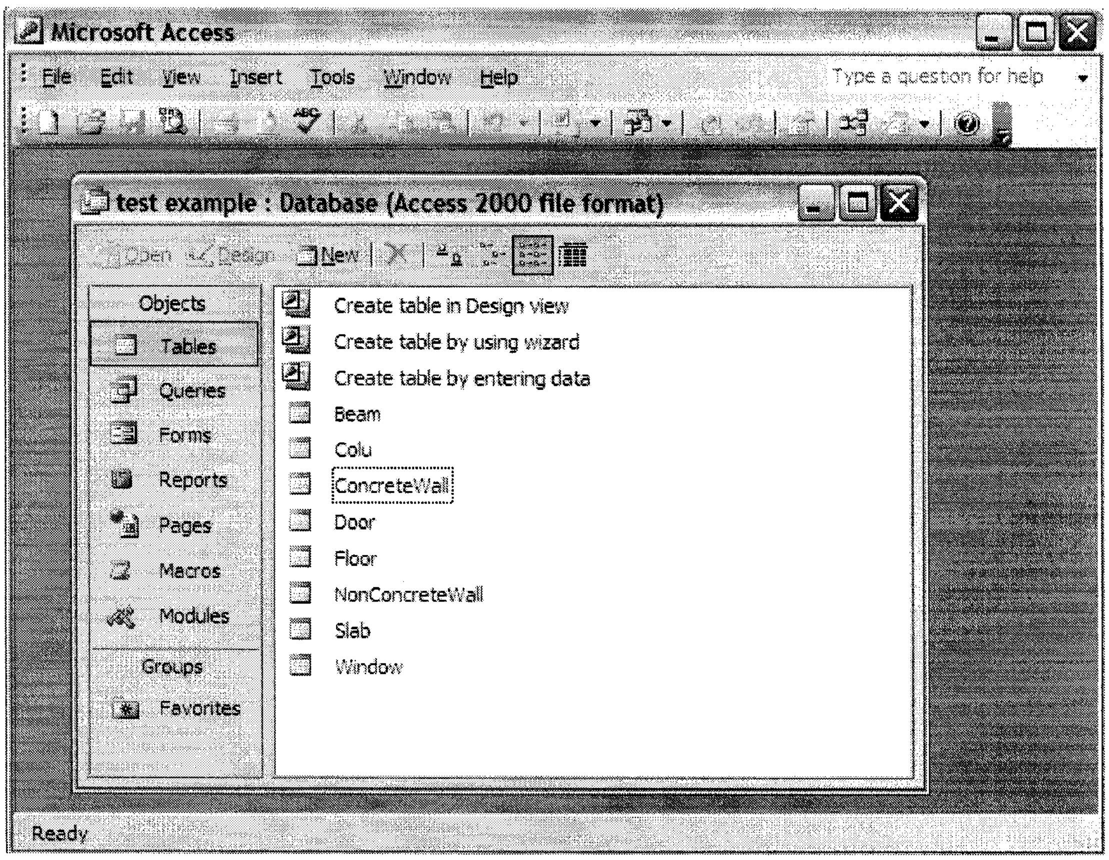
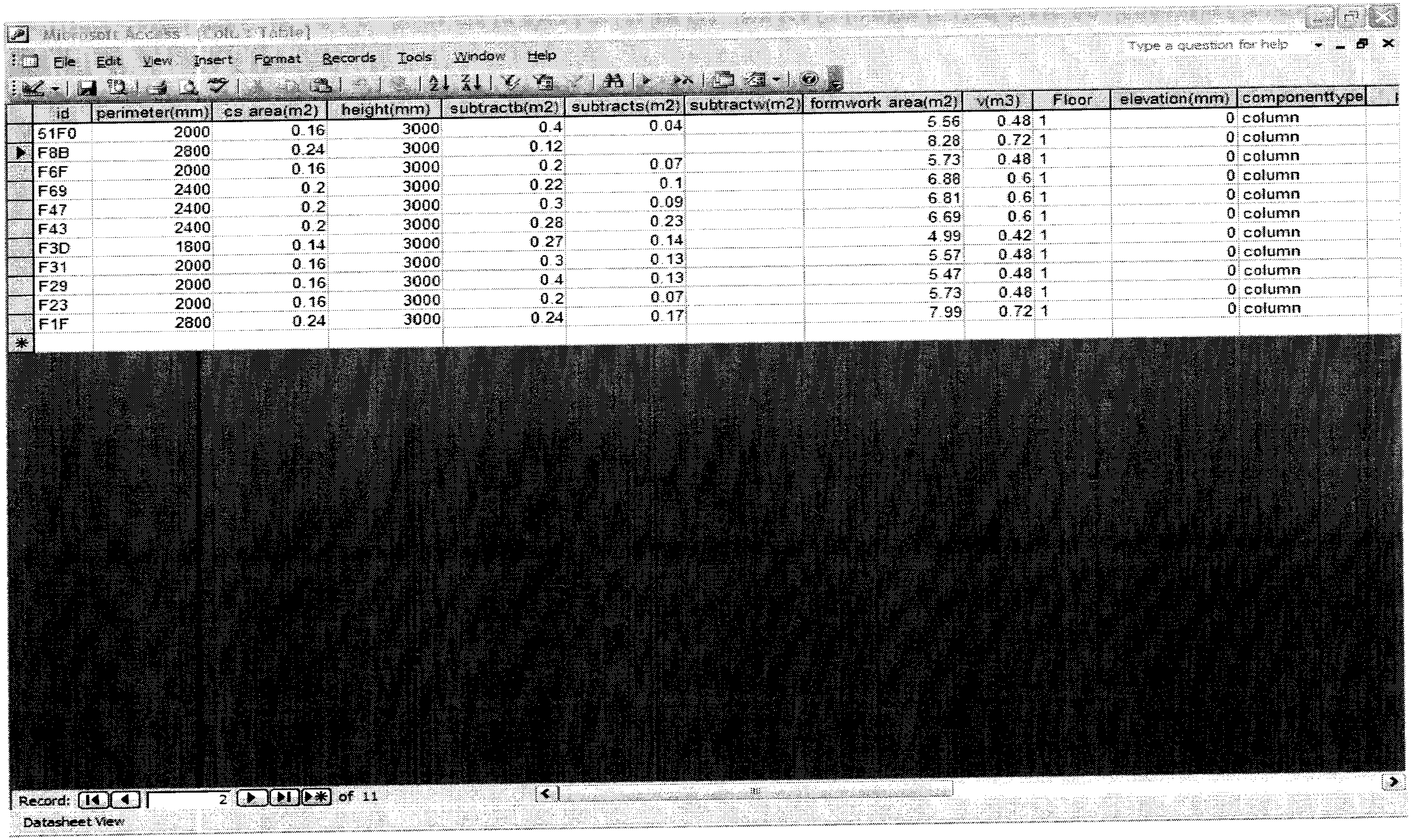


Figure 5.25 The Project Database



id	perimeter(mm)	cs area(m2)	height(mm)	subtractb(m2)	subtracts(m2)	subtractw(m2)	formwork area(m2)	v(m3)	Floor	elevation(mm)	componenttype
51F0	2000	0.16	3000	0.4	0.04		5.56	0.48	1	0	column
F8B	2800	0.24	3000	0.12			8.28	0.72	1	0	column
F6F	2000	0.16	3000	0.2	0.07		5.73	0.48	1	0	column
F69	2400	0.2	3000	0.22	0.1		6.88	0.6	1	0	column
F47	2400	0.2	3000	0.3	0.09		6.81	0.6	1	0	column
F43	2400	0.2	3000	0.28	0.23		6.69	0.6	1	0	column
F3D	1800	0.14	3000	0.27	0.14		4.99	0.42	1	0	column
F31	2000	0.16	3000	0.3	0.13		5.57	0.48	1	0	column
F29	2000	0.16	3000	0.4	0.13		5.47	0.48	1	0	column
F23	2000	0.16	3000	0.2	0.07		5.73	0.48	1	0	column
F1F	2800	0.24	3000	0.24	0.17		7.99	0.72	1	0	column

Figure 5.26 The Column Table

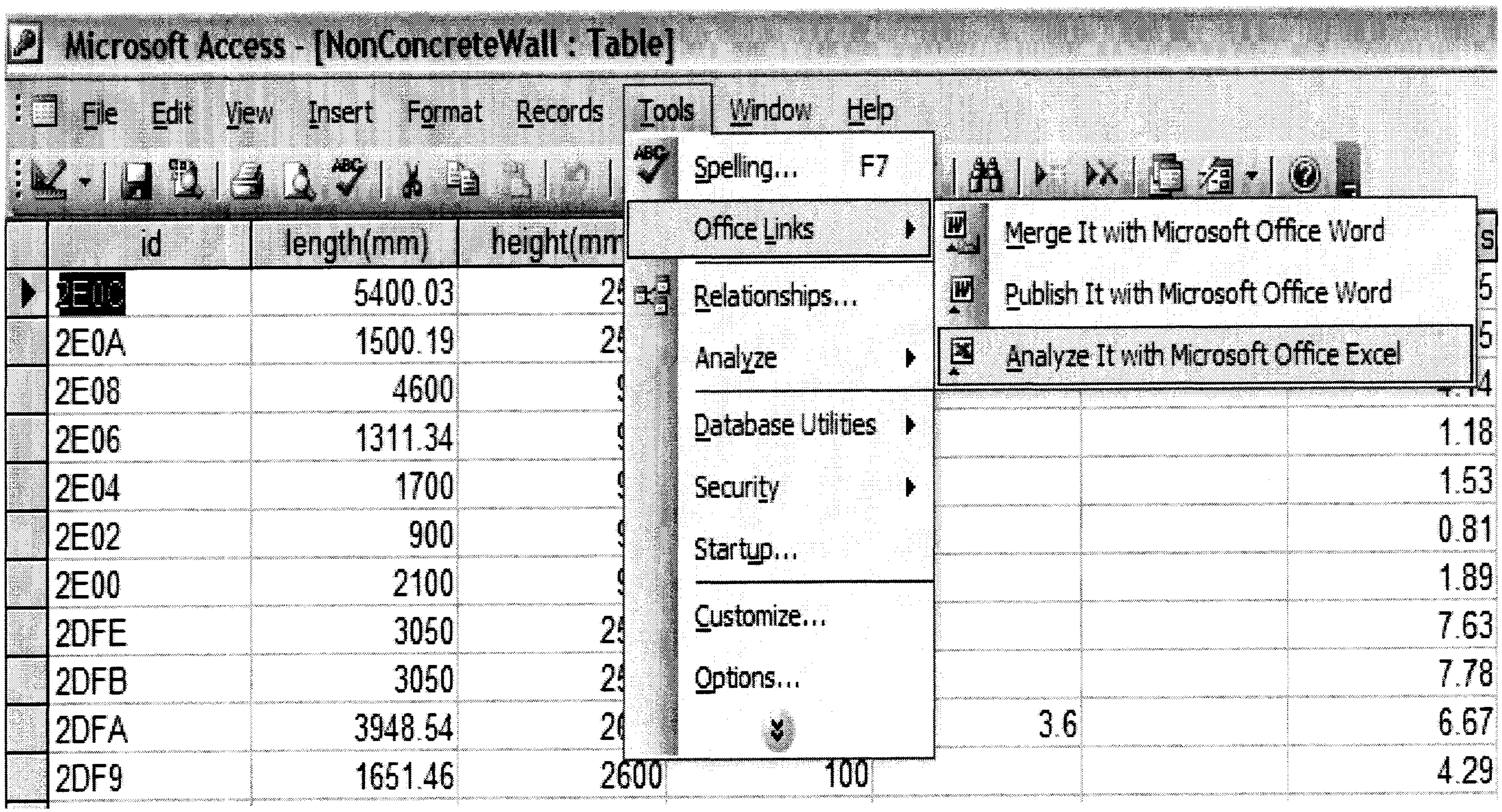


Figure 5.27 The Embedded Analyze Tool

Microsoft Excel - Colu

Next | Previous | Zoom | Print... | Setup... | Margins | Page Break Preview | Close | Help

id	perimeter (mm)	cs area (m2)	height(mm)	subtractb (m2)	subtracts (m2)	subtractw (m2)	formwork area(m2)	v(m3)	Floor	elevation (mm)	componenttype
51F0	2000.00	0.16	3000.00	0.40	0.11		5.49	0.48	1	0	column
F8B	2800.00	0.24	3000.00	0.12			8.28	0.72	1	0	column
F6F	2000.00	0.16	3000.00	0.20	0.07		5.73	0.48	1	0	column
F69	2400.00	0.20	3000.00	0.22	0.10		6.88	0.60	1	0	column
F47	2400.00	0.20	3000.00	0.30	0.09		6.81	0.60	1	0	column
F43	2400.00	0.20	3000.00	0.28	0.22		6.70	0.60	1	0	column
F3D	1800.00	0.14	3000.00	0.27	0.14		4.99	0.42	1	0	column
F31	2000.00	0.16	3000.00	0.30	0.13		5.57	0.48	1	0	column
F29	2000.00	0.16	3000.00	0.40	0.13		5.47	0.48	1	0	column
F23	2000.00	0.16	3000.00	0.20	0.07		5.73	0.48	1	0	column
F1F	2800.00	0.24	3000.00	0.24	0.17		7.99	0.72	1	0	column
Σ							**69.64**	**6.06**			

Figure 5.28 Analysis Result of Table Column

Microsoft Excel - Beam

Next | Previous | Zoom | Print... | Setup... | Margins | Page Break Preview | Close | Help

id	width(mm)	height(mm)	length(mm)	subtract(m2)	formwork area(m2)	v(m3)	Floor	elevation(mm)	componenttype	property1
25C7	150	300	1551.17	0.37	0.79	0.07	1	2700	beam	
25C5	150	300	1850	0.44	0.94	0.08	1	2700	beam	
25BF	200	600	4000	0.48	5.12	0.48	1	2400	beam	KL14
25BD	200	450	4300	1.03	3.7	0.39	1	2550	beam	KL13
25BB	200	450	3350	0.4	3.28	0.3	1	2550	beam	KL13
25B8	200	500	1400.09	0.17	1.51	0.14	1	2500	beam	KL15(1A)
25B6	200	500	5400	0.65	5.83	0.54	1	2500	beam	KL15(1A)
25B4	200	500	1323.02	0.16	1.43	0.13	1	2500	beam	KL12(1A)
25B2	200	500	3050	0.73	2.93	0.31	1	2500	beam	KL12(1A)
25B0	200	500	1600	0.38	1.54	0.16	1	2500	beam	KL11a
25AE	200	500	3700	0.44	4	0.37	1	2500	beam	KL11a
25AC	200	500	4600	0.55	4.97	0.46	1	2500	beam	KL11
25AA	200	500	1600	0.19	1.73	0.16	1	2500	beam	L6
25A8	200	500	4500	0.54	4.86	0.45	1	2500	beam	L3
25A6	200	500	4100	0.98	3.94	0.41	1	2500	beam	KL5
25A4	200	500	2950	0.35	3.19	0.3	1	2500	beam	KL5
259C	198.8336	400	3400	0.82	2.58	0.27	1	2600	beam	KL4
259A	200	500	900	0.11	0.97	0.09	1	2500	beam	KL3(1A)
2598	200	500	3500	0.84	3.36	0.35	1	2500	beam	KL3(1A)
2595	200	500	2100	0.25	2.27	0.21	1	2500	beam	KL2(1A)
2593	200	500	1500	0.36	1.44	0.15	1	2500	beam	KL2(1A)
2591	200	400	2800	0.67	2.13	0.22	1	2600	beam	L2
258F	200	400	2600	0.62	1.98	0.21	1	2600	beam	L1
258C	200	500	5100	0.61	5.51	0.51	1	2500	beam	kl1
Σ					70	6.76				

Figure 5.29 Analysis Result of Table Beam

Microsoft Excel - NonConcreteWall

Next | Previous | Zoom | Print... | Setup... | Margins | Page Break Preview | Close | Help

id	length (mm)	height (mm)	thickness (mm)	subtract door(m2)	subtract window(m2)	face area(m2) (single side)	formwork area(m2)	volume (m3)	Floor	elevation(mm)	componenttype	property1	property2
2DEE	2400	2400	200	1.8		3.96	8.9	0.79	1	0	nonconcretewall	exterior	
2DED	4000	2400	200			9.6	19.2	1.92	1	0	nonconcretewall	exterior	
2DEC	5100	2500	200		4.5	8.25	18.9	1.65	1	0	nonconcretewall	exterior	
2DEB	3700	2500	200			9.25	18.5	1.85	1	0	nonconcretewall	exterior	
2DEA	1800	2500	200	1.4	0.75	2.35	6.44	0.47	1	0	nonconcretewall	exterior	
2DE9	700	2500	200		0.9	0.85	2.54	0.17	1	0	nonconcretewall	exterior	
2DE8	4600	2500	200			11.5	23	2.3	1	0	nonconcretewall	exterior	
2DE7	4100	2500	200	6.4		3.85	9.14	0.77	1	0	nonconcretewall	exterior	
2DE6	2950	2500	200		2.7	4.68	10.68	0.94	1	0	nonconcretewall	exterior	
Σ						54.29		10.86					

Figure 5.30 Analysis Result of Table NonConcreteWall (exterior)

Microsoft Excel - NonConcreteWall

Next | Previous | Zoom | Print... | Setup... | Margins | Page Break Preview | Close | Help

id	length (mm)	height (mm)	thickness (mm)	subtract door(m2)	subtract window(m2)	face area(m2) (single side)	formwork area(m2)	volume (m3)	Floor	elevation(mm)	componenttype	property1	property2
2E0C	5400	2500	200.7291			13.5	27	2.71	1	0	nonconcretewall	interior	
2DFE	3050	2500	100			7.63	15.26	0.76	1	0	nonconcretewall	interior	
2DFB	3050	2550	100			7.78	15.56	0.78	1	0	nonconcretewall	interior	
2DFA	3949	2600	100	3.6		6.67	14.32	0.67	1	0	nonconcretewall	interior	
2DF9	1651	2600	100			4.29	8.58	0.43	1	0	nonconcretewall	interior	
2DF8	1950	2700	100	1.4		3.86	8.19	0.39	1	0	nonconcretewall	interior	
2DF7	1500	2500	100			3.75	7.5	0.38	1	0	nonconcretewall	interior	
2DF5	1800	2550	200	1.4	0.75	2.44	6.62	0.49	1	0	nonconcretewall	interior	
2DF4	1550	2550	100	1.8		2.15	4.79	0.22	1	0	nonconcretewall	interior	
2DF3	2600	2500	100			6.5	13	0.65	1	0	nonconcretewall	interior	
2DF2	1550	2700	100	1.4		2.78	6.03	0.28	1	0	nonconcretewall	interior	
2DF1	2451	2600	100			6.37	12.74	0.64	1	0	nonconcretewall	interior	
Σ						67.72		8.4					

Figure 5.31 Analysis Result of Table NonConcreteWall (interior)

Microsoft Excel - NonConcreteWall

Next Previous Zoom Print... Setup... Margins Page Break Preview Close Help

id	length (mm)	height (mm)	thickness (mm)	subtract door(m2)	subtract window(m2)	face area(m2) (single side)	formwork area(m2)	volume (m3)	Floor	elevation(mm)	componenttype	property1	property2
2E0A	1500	2500	200			3.75	7.5	0.75	1	0	nonconcretewall	Balcony	
2E08	4600	900	100			4.14	8.28	0.41	1	0	nonconcretewall	Balcony	
2E06	1311	900	100			1.18	2.36	0.12	1	0	nonconcretewall	Balcony	
2E04	1700	900	100			1.53	3.06	0.15	1	0	nonconcretewall	Balcony	
2E02	900	900	100			0.81	1.62	0.08	1	0	nonconcretewall	Balcony	
2E00	2100	900	100			1.89	3.78	0.19	1	0	nonconcretewall	Balcony	
Σ						13.3		1.7					

Figure 5.32 Analysis Result of Table NonConcreteWall (Balcony)

Microsoft Excel - Slab

Next | Previous | Zoom | Print... | Setup... | Margins | Page Break Preview | Close | Help

id	perimeter(mm)	thickness(mm)	area(m2)	formwork area(m2)	volume(m3)	Floor	elevation(mm)	componenttype
85FD	8400	120	4.16	4.16	0.5	1	2880	slab
2690	5799.42	120	2.09	2.09	0.25	1	2880	slab
2689	7705.25	120	3.57	3.57	0.43	1	2880	slab
264E	5597.08	120	1.75	1.75	0.21	1	2880	slab
264D	6702.92	120	2.78	2.78	0.33	1	2880	slab
263A	7000	120	3.04	3.04	0.36	1	2880	slab
2633	11500	120	8.26	8.26	0.99	1	2880	slab
262C	9700	120	5.85	5.85	0.7	1	2880	slab
2625	13700	120	11.73	11.73	1.41	1	2880	slab
261A	11246.03	120	5.05	5.05	0.61	1	2880	slab
2610	25700	120	32.17	32.17	3.86	1	2880	slab
Σ				**80.45**	**9.65**			

Figure 5.33 Analysis Result of the Table Slab

Microsoft Excel - Door

Next | Previous | Zoom | Print... | Setup... | Margins | Page Break Preview | Close | Help

id	width(mm)	height(mm)	area(m2)	frame length(m)	Floor	elev ation(mm)	componenttype
2E2A	700	2000	1.4	4.7	1	0	door
2E27	700	2000	1.4	4.7	1	0	door
2E24	900	2000	1.8	4.9	1	0	door
2E21	900	2000	1.8	4.9	1	0	door
2E1D	900	2000	1.8	4.9	1	0	door
2E1A	700	2000	1.4	4.7	1	0	door
2E17	700	2000	1.4	4.7	1	0	door
2E14	3200	2000	6.4	7.2	1	0	door
2E10	900	2000	1.8	4.9	1	0	door
Σ			**19.2**				

Figure 5.34 Analysis Result of the Table Door

Microsoft Excel - Window

Next | Previous | Zoom | Print... | Setup... | Margins | Page Break Preview | Close | Help

id	width(mm)	height(mm)	area(m2)	frame length(m)	Floor	elevation(mm)	componenttype	Property1	Property2
2E41	500	1500	0.75	4	1	700	window	LMC3	
2E3E	500	1500	0.75	4	1	700	window	LMC4	
2E38	600	1500	0.9	4.2	1	700	window	LC5	
2E35	1500	1500	2.25	6	1	700	window	LC2	
2E32	1500	1500	2.25	6	1	700	window	LC2	
2E2E	1800	1500	2.7	6.6	1	700	window	LC1	
Σ			9.6						

Figure 5.35 Analysis Result of the Table Window

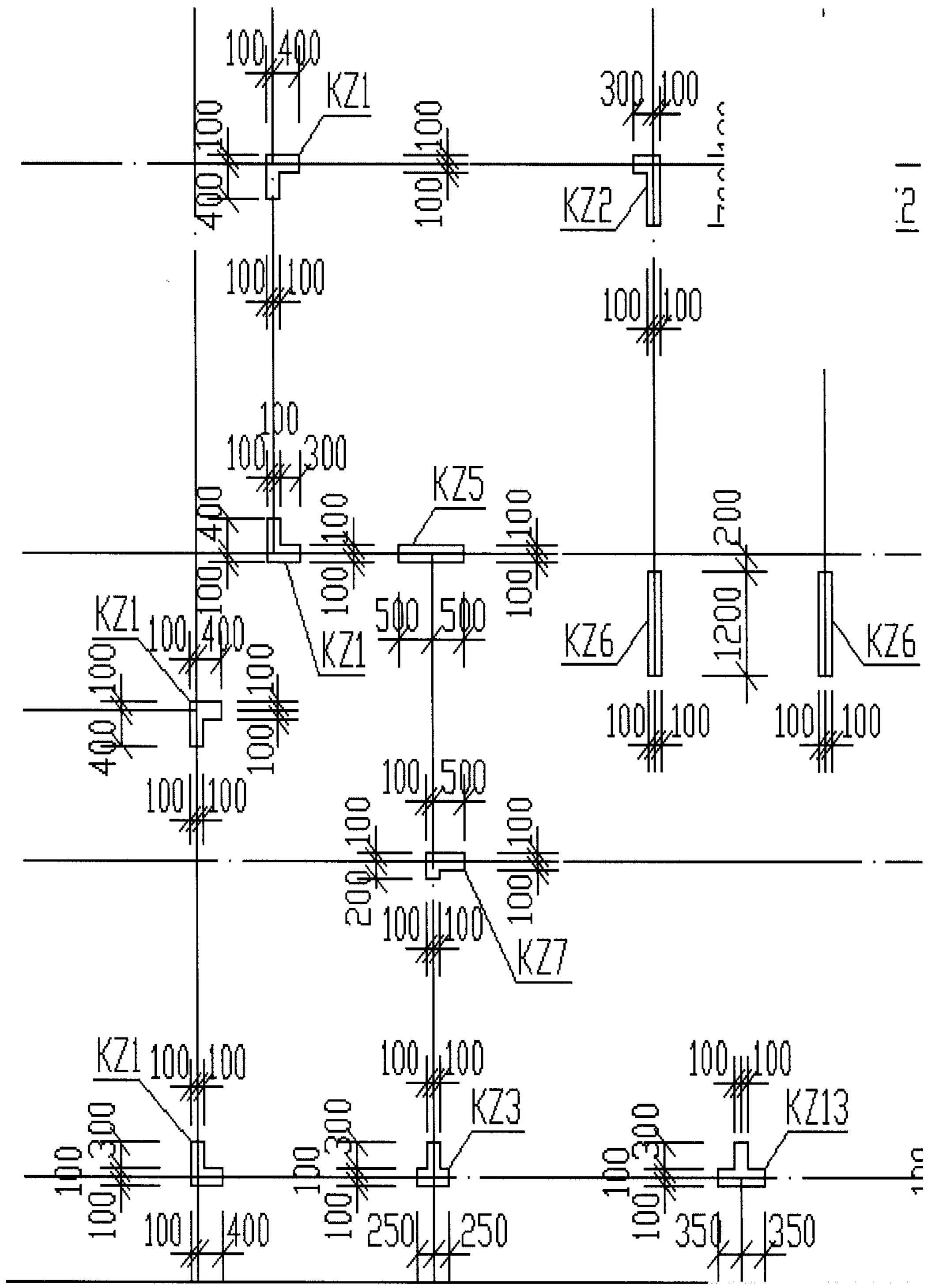


Figure 5.36 Column Plan

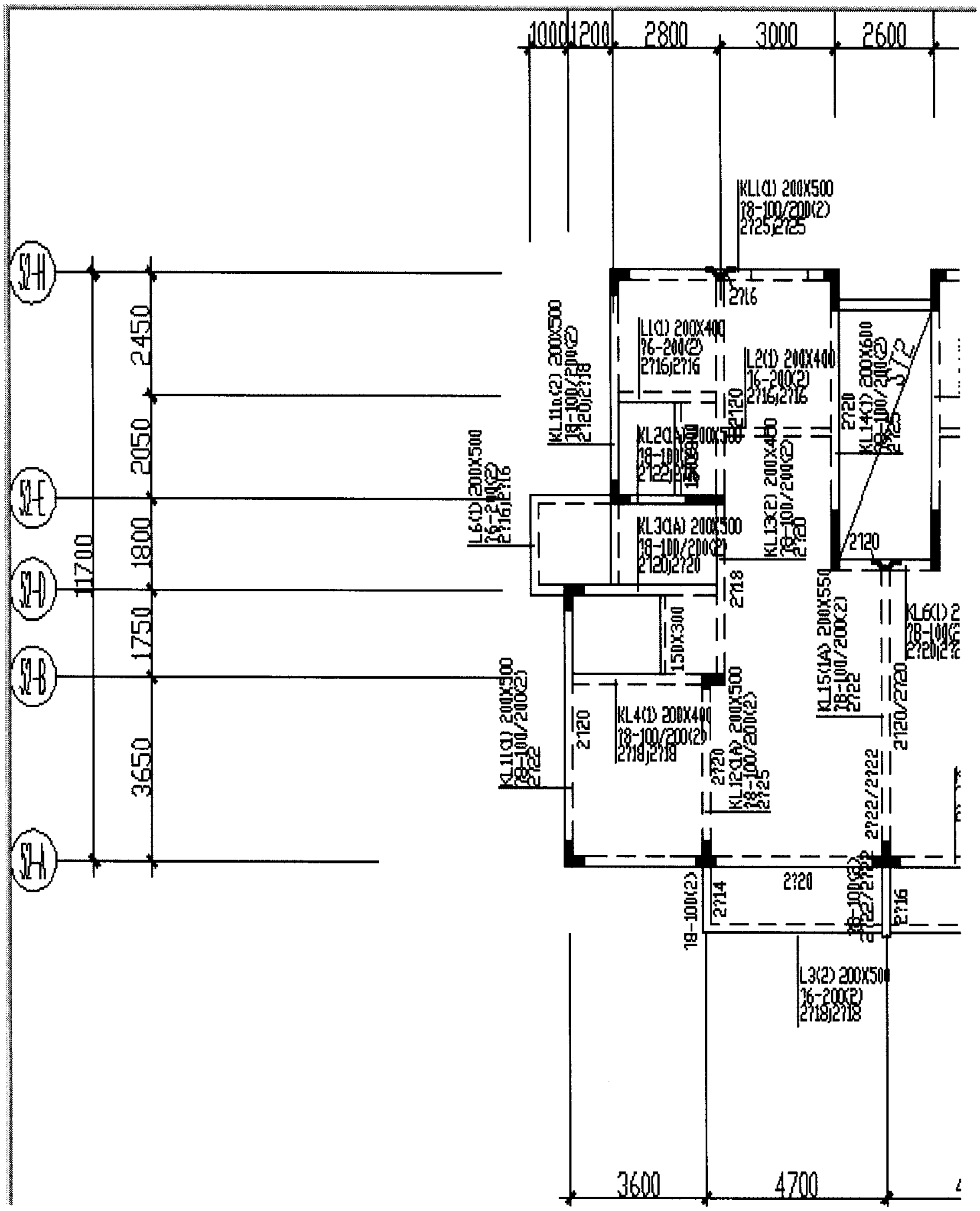


Figure 5.37 Structure Plan

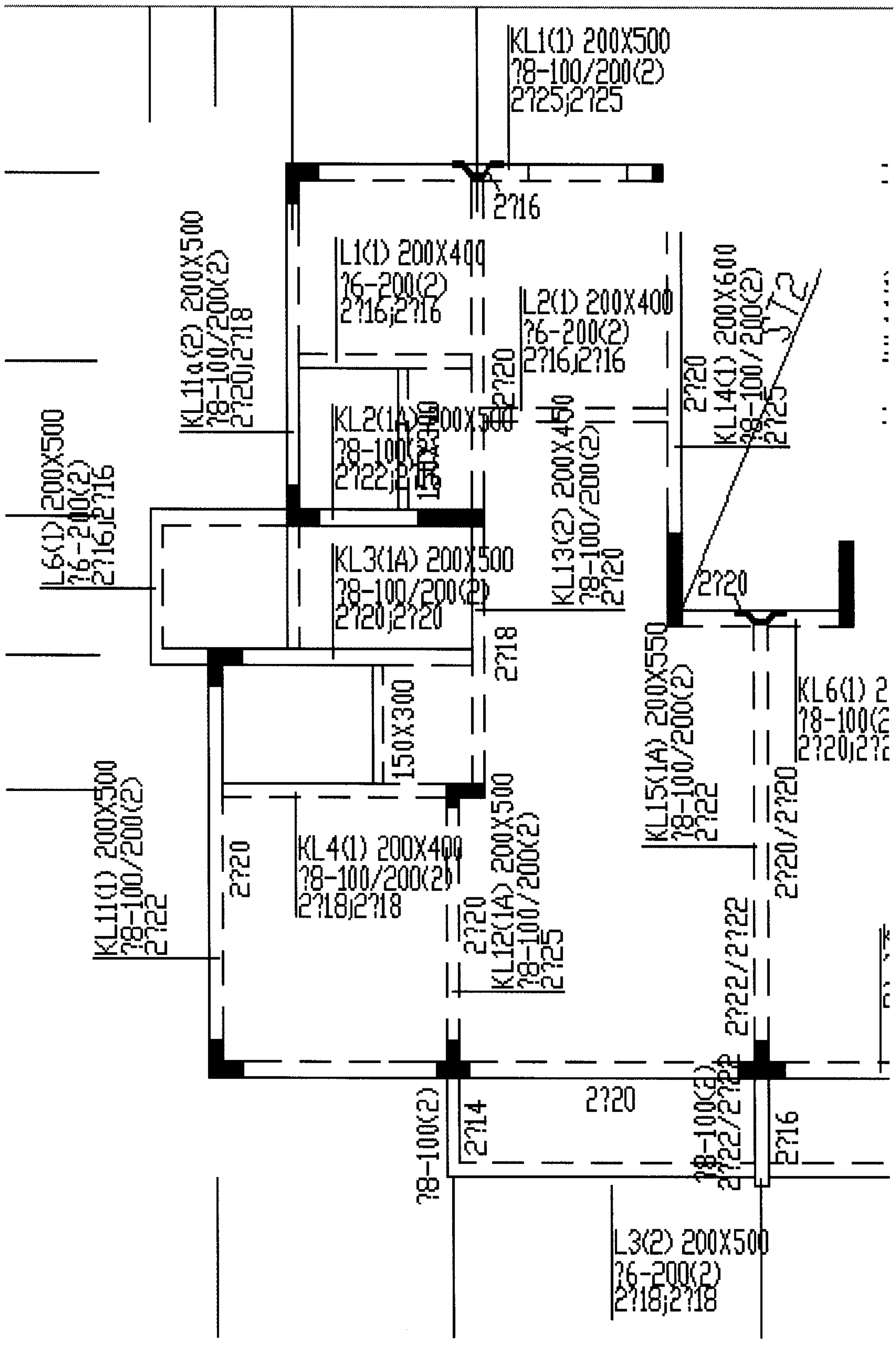


Figure 5.38 Structure Plan

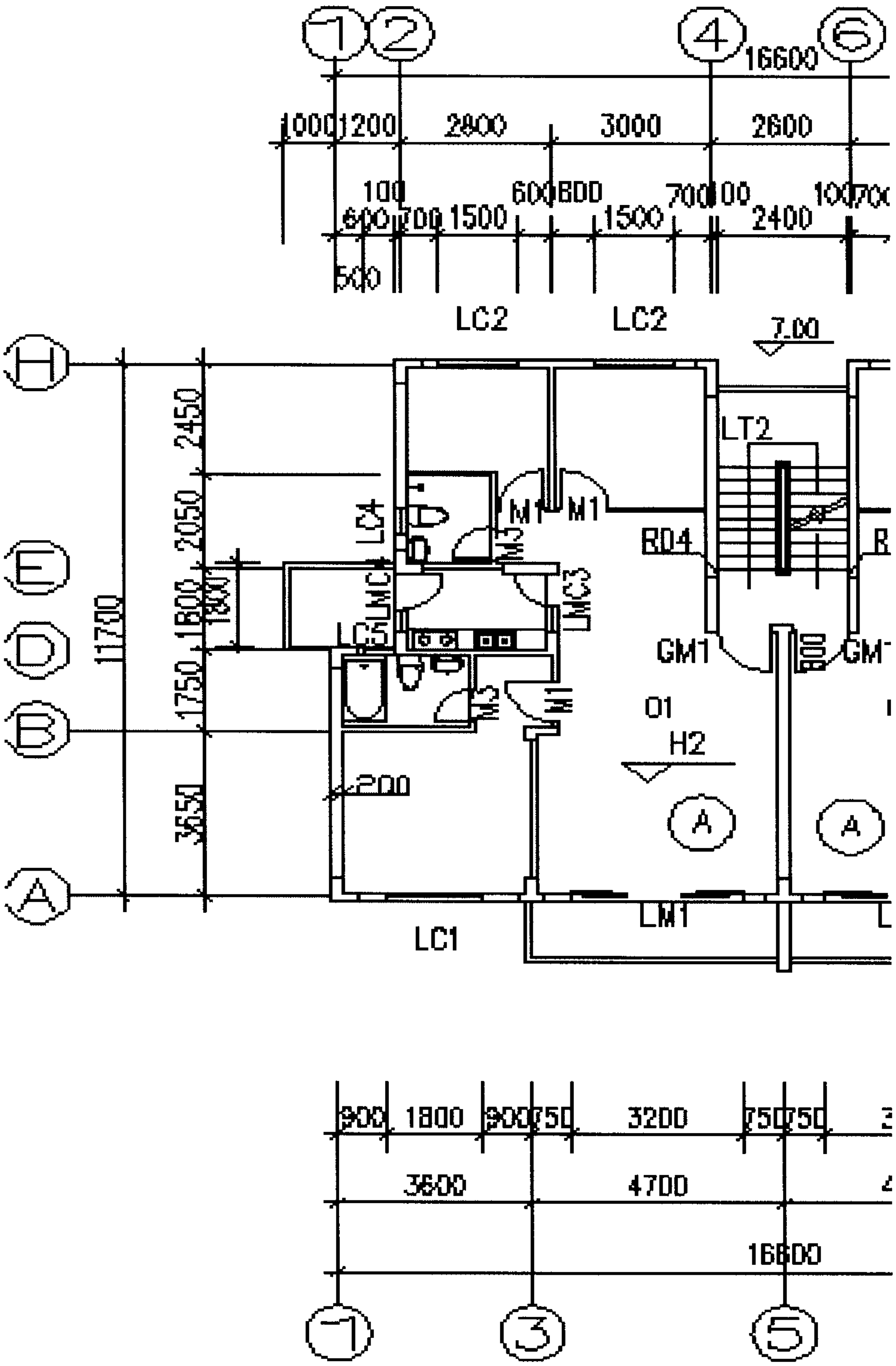


Figure 5.39 Architectural Floor Plan

Based on the drawings shown in Figures 5.36 through 5.39, the manual calculation is performed.

First, the volume of the columns is calculated thusly:

Table 5.1 Volume Calculation (Column)

Column	Area (m2)	Height (m)	Volume (m3)	Number	Sub Total (m3)
KZ1	0.5*0.2+0.3*0.2=0.16	3	0.48	4	1.92
KZ2	0.4*0.2+0.6*0.2=0.2	3	0.6	1	0.6
KZ3	0.5*0.2+0.3*0.2=0.16	3	0.48	1	0.48
KZ5	1*0.2=0.2	3	0.6	1	0.6
KZ6	1.2*0.2=0.24	3	0.72	2	1.44
KZ7	0.6*0.2+0.2*0.1=0.14	3	0.42	1	0.42
KZ13	0.7*0.2+0.3*0.2=0.2	3	0.6	1	0.6
Total					**6.06**

Comparing the result with the one in Figure 5.25, we see that the total volume of the columns is perfectly matched: **6.06** m^3.

Similarly, the comparison of the column formwork is also made. Here we pick one column as an example (Figure 5.40, Figure 5.41). This column intersects with 4 beams and has overlaps with two slabs. Thus, the intersecting and overlapping area should be deducted from the formwork area of the column. The calculation is illustrated in Table 5.2.

Table 5.2 Formwork Calculation (Column)

Column	Perimeter * Height(m2)	Deduction (beam)(m2)	Deduction (slab)(m2)	Deduction (concretewall)	Formwork (m2)
KZ1	2*3=6	4*0.2*0.5 =0.4	3*0.3*0.12 =0.108	0	5.492

Once again, when we compare the result with the one in Figure 5.25, find the record (id 51F0), and check the columns “subtractb”, “subtracts”, “subtractw” and “formwork area”, the numbers are **0.4, 0.11, 0** and **5.49,** respectively. The difference is merely because of the decimal places selected. So the results are also perfectly matched.

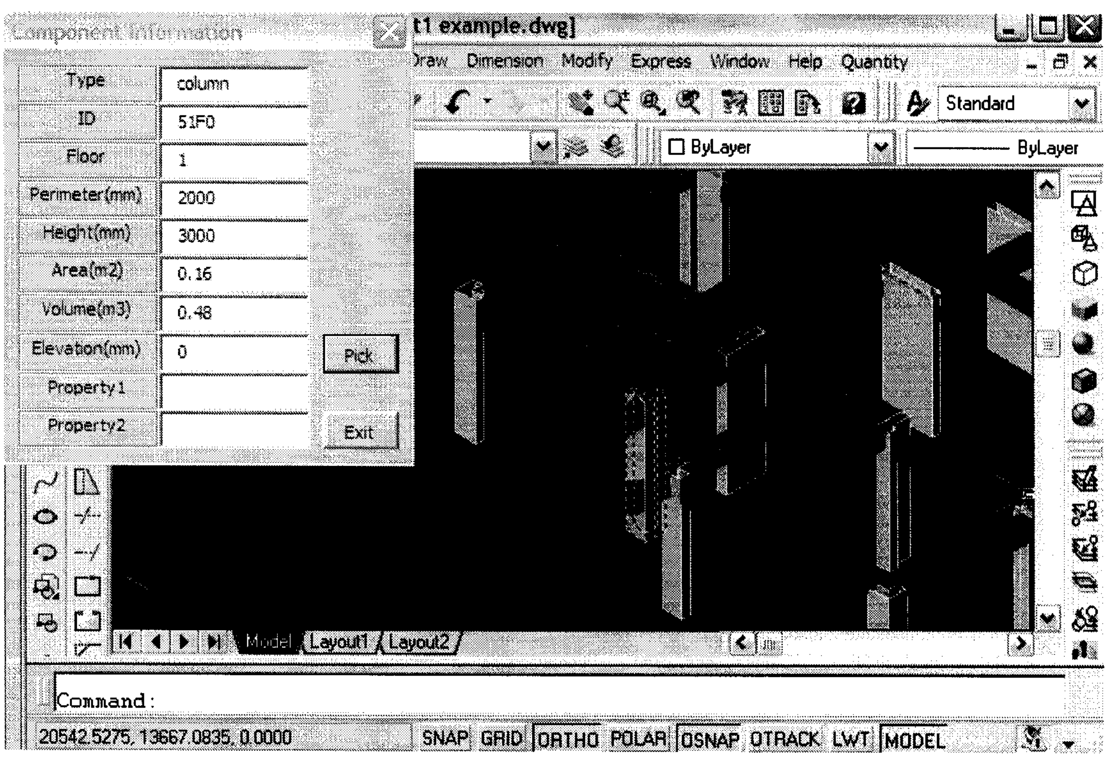


Figure 5.40 3D View of Selected Column

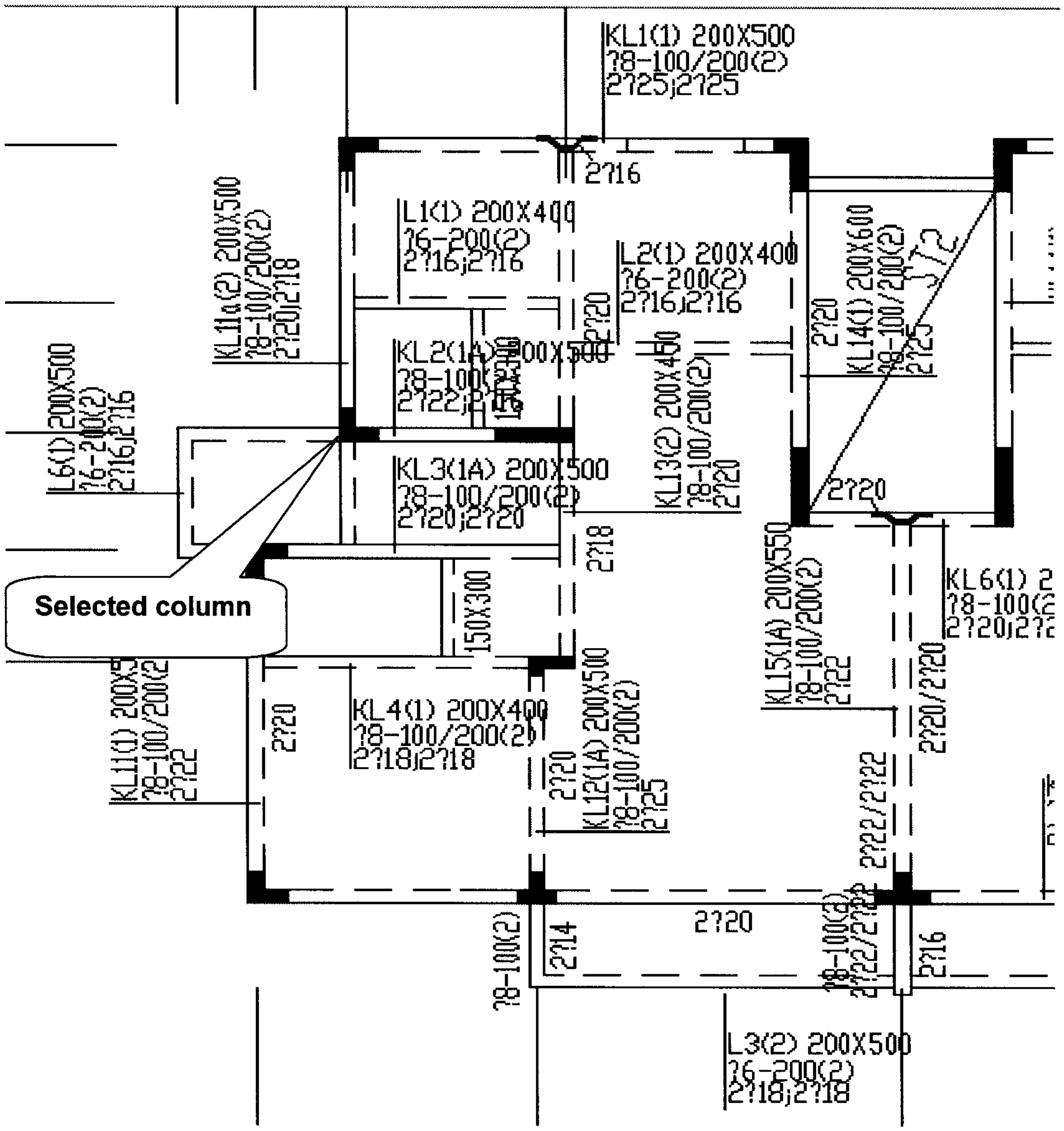


Figure 5.41 Structure Drawing

Following the same rules, comparisons between the results of computer model generated and those of manually calculated were made for the other components. The conclusion is that the results created by the developed model are accurate and the model can be applied in real world quantity take-off practice.

5.3 Summary

The practical and efficient features of the developed computer model have been presented by applying it to an example project. The results generated by the computer model were compared with the manually calculated results. The comparison proves the practicality and validity of the model and indicates that a job which may take weeks using traditional methods can be performed using the developed model in a few days.

CHAPTER 6

CONCLUSIONS

6.1 Summary

23 years ago, Autodesk first revolutionized the software industry with AutoCAD, which introduced drafting functions to the PC. Today, AutoCAD has become the world's most accessible and widely used design software. The designers who use AutoCAD are familiar with the benefits of producing graphic images with it. However, for estimators, AutoCAD is not being used effectively. The very specific and detailed information stored in CAD drawing files offers considerable potential to the construction estimator. If this information can be used to provide quantity take-offs, more accurate and detailed estimates may be attainable. Instead of manually measuring the drawings for quantities, the quantity take-off can be performed in a more automated way.

Above, a computer model has been developed to integrate AutoCAD with quantity take-off during the detailed cost estimation process. Directly based on the designer's AutoCAD 2D drawings, through interaction with a user, a customized AutoCAD module generates a 3D model of a project, and export the components' information into an external access database for storing detailed quantity take-off information. And in the mean time, the 3D model improves project visualization by allowing estimators to visualize the construction process as it would be actually built. This model makes quantity take-off more efficient and effective.

6.2 Research Contributions

This study has presented a new methodology in the domain of construction quantity take-off. Its major contributions are:

- The development of a unique computer model to integrate AutoCAD with detailed construction quantity take-off.
- Improvements to the current construction quantity take-off practice in terms of time, accuracy and cost. Time is always critical during quantity take-off; the developed model can greatly improve the productivity by easily extracting the quantity information from the 3D model based on 2D electronic drawings. In addition. the output will not have computation errors that can creep in during manual take-off calculationg. With the improved productivity and accuracy, the costs spent on quantity take-off will certainly decrease.
- Improvements to the communication in project management. Compared with the 2D paper drawing, an intelligent 3D AutoCAD model of a project improves visualization, hence making the communication easier and more effective.

6.3 Limitations

Since the proposed computer model is established based on a general building structure, there are some limitations with the current model. However, these limitations could be reduced gradually with improvements made in future work. The current limitations are:

- The proposed model is designed to help estimators perform quantity take-off.
- The proposed model is most suitable for concrete buildings.
- The types of components which can be modeled in the current system include: columns, walls, rectangular beams, slabs, windows and doors.
- The output of the model includes: the quantities of concrete, formwork (column, wall, beam and slab), windows, doors.

6.4 Future Work

The developments achieved in this thesis are based on a modularized, yet integrated, approach. Therefore, it has great potential for incorporating many improvements into the present prototype. The following is a list of recommended future endeavors:

- Link the quantity information to cost libraries such as R.S. Means or company-based cost libraries. By doing this, the model will allow the user to query not only the quantity information of a particular component, but also its cost information. Thus, the user can easily develop a complete and detailed cost estimation of the project.
- Expand the type of components to cover the whole process of a building. In this research, the foundations, staircases, finishing, insulation, etc. are not taken into account. But they can easily be incorporated into the present model using the conception and approach illustrated in this thesis.
- Expand the types of applicable structures to a wider range such as steel

structures, infrastructures.

- Integrate with other applications such as GIS, to improve the infrastructure management.
- Add the time factor to make the model equipped as a 4D Model based application which can help improve both cost estimation and project scheduling.

References

Adrian, J., **"Construction estimating: An accounting and productivity approach,"** 1982, Reston

Al-Hussein, M., **"An Integrated Information System for Crane Selection and Utilization,"** Ph.D. Thesis, Department of Building, Civil and Environmental Engineering, Concordia University, Montreal, Canada, 1999.

Autodesk website: [On-Line] **http://www.autodesk.com**

Clayton, M.J., Kunz, J.C., Fischer, M.A., and Teicholz, P., **"First drawings, then semantics,"** In Harfmann A.C. and Fraser M. (eds). Reconnecting: ACADIA'94. (The Association for Computer Aided Design in Architecture), 1994, pp. 13-26.

Duncan, W.R., "**A Guide to the Project Management Body of Knowledge,**" 1996, Library of Congress Cataloging-in-Publication Data.

Duvel, S. C., Schmidt, K., **"Integrating Relational Database Technology into the Construction Management Curriculum,"** Journal of Construction Education, Vol. 7, No. 2, summer 2002, pp.74-85.

Elmasri, R., Navathe, S. B., "**Fundamentals of Database Systems,**" 3rd Edition, 2001, Addison-Wesley Longman.

Fisher, M.A. and Froese, T., **"Examples and characteristics of shared project models,"** Journal of Computing in Civil Engineering, ASCE, Volume10. No.3, 1996, pp.174-182.

Halpin, D. W., **"Financial 8 Cost Concepts for Construction Management,"** 1985, John Wiley & Sons.

Hewitt, M., **"Representational forms and models of conception,"** Journal of Architectural Education, Volume 39, No.2, 1985, pp.2-9.

Jurkiewicz, W. J., "**Theory and Practice with Dual Entry in Project Cost Accounting and Control,**" Transaction of the AACE 43rd Annual Meeting, Denver, Colorado, U.S.A., June 1999, pp. CSC.05.1-CSC.05.8

Kibert, J. C. and Hollister, C. K., **"An Enhanced Construction Specific SQL,"** Automation in Construction, Vol. 2, No. 4, April 1994, pp. 303-312.

Kim, Y. S., OH, S. W., Kim, J. R., Sung, B. J., **"The Development of an Automated System Using 3D CAD and Relational Database,"** Proceedings of the 17th Symposium on Automation and Robotics in Construction XVII, September 2000, Taiwan, pp. 827-833.

Kitchens, M. "**Estimating and Project Management for Building Contractor**," 1996, ASCE Press.

Lewis, R. and Sequin, C., **"Generation of 3D building models from 2D architectural plans,"** Computer- Aided Design, Volume30, No.10, 1998, pp.765-779.

Lin, C.Y., "**A digital procedure of building construction"** In: Gero, J., Chase, S. and Rosenman, M. (eds.) CAADRIA2001, Key Centre of Design Computing and Cognition, University of Sydney, 2001, pp.459-468.

Mcfarlane, Scott, **"AutoCAD Database Connectivity,"** 2000, Thomson Learning.

Peurifoy, R. L. and Oberlender, G. D. **"Estimating Construction Cost,"** 5th edition, 2002, McGraw-Hill.

Roe, A. G., **"Using Visual Basic with AutoCAD,"** 2nd edition, 2001, Autodesk Press, Thomson Learning.

Sun, M. and Howard, R. "**Understanding IT in construction**" 2004, Spon Press, London.

Teicholz, P. and Fischer M.A. **"Strategy for computer integrated construction technology,"** Journal of Construction Engineering and Management, ASCE, Volume120, No.1, 1994, pp.117-131.

Teory, T. J., **"Database Modeling and Design,"** 2nd Edition, 1994, Morgan Kaufman Publishers, Inc.

Xu, J., **"CAD-Based Integrated Simulation Environment (CAD-ISE),"** PH.D. Thesis, Department of Civil and Environmental Engineering, University of Alberta, Edmonton, Canada, 2000.

Westney, R. E., **"The Engineer's Cost Handbook Tools for Managing Project Costs,"** 1997, Marcel Dekker, Inc.

Zahnan, L., **"CAD-Based Project Management Model,"** M.A.Sc. Thesis, Department of Building, Civil and Environmental Engineering, Concordia University, Montreal, Canada, 2001.