4-21-1993

The Investigation of Object-Oriented Methodology as an Enabling Technique for End-User Application Development

Eng Chong Lim
Western Michigan University, eng.chong.lim@limnetwork.net

THE CARL AND WINIFRED LEE HONORS COLLEGE

CERTIFICATE OF ORAL EXAMINATION

Eng Chong Lim, having been admitted to the Carl and Winifred Lee Honors College in 1991, has satisfactorily completed the senior oral examination for the Lee Honors College on April 21, 1993.

The title of the paper is:

"The Investigation of Object-Oriented Methodology as an Enabling Technique for End-User Application Development"

Dr. Judy Yaeger
Business Information Systems

Dr. Arlen Gullickson
Evaluation Center

Mr. Ralph Yingling
Haworth College of Business
Computer Services

THE INVESTIGATION OF OBJECT-ORIENTED METHODOLOGY AS AN ENABLING TECHNIQUE FOR END-USER APPLICATION DEVELOPMENT

April 21, 1993

Presented by:
Eng Chong Lim

Presented to:
Dr. Judy Yaeger, Chairperson
Dr. Arlen Gullickson, Advisor
Mr. Ralph Yingling, Advisor

THE CARL AND WINIFRED LEE HONORS COLLEGE

CERTIFICATE OF ORAL EXAMINATION

Eng Chong Lim, having been admitted to the Carl and Winifred Lee Honors College in 1991, has satisfactorily completed the senior oral examination for the Lee Honors College on April 21, 1993.

The title of the paper is:

"The Investigation of Object-Oriented Methodology as an Enabling Technique for End-User Application Development"

Dr. Judy Yaeger
Business Information Systems

Dr. Arlen Gullickson
Evaluation Center

Mr. Ralph Yingling
Haworth College of Business
Computer Services

Table of Contents

Acknowledgments

I would like to thank Dr. Judy Yaeger for her guidance, support, and patience in assisting me in this study of object-oriented design, Dr.Arlen Gullickson for allowing me the chance to design the Evaluation Center's project and staff accounting information system, and Mr. Ralph Yingling for his confidence in my ability as a future Computer Information Systems professional.

Gratitude also goes out to Jean Wyer of the Lee Honors College for her counsel and encouragement to finish my senior honors thesis.

Special thanks to the staff of the Evaluation Center, namely, Mary Ramlow and Robin Jung for patiently going through with me the procedures of the Center. Most importantly, I wish to express exceptional gratitude to Elissa Stringer of the Evaluation Center for her valuable advice which I shall always remember and use in my career as a systems professional.

Above all, I would like to dedicate this work to my family (father Dr. Lim Hong Kuan, mother Mak Geok Tin, sisters Lim Sok Ngoh, Lim Siok Kim, brother Lim Eng Teik) in Malaysia for their emotional support, advice, and belief in my judgment in my undertakings.

Overview

Purpose
This research studies the value of object-oriented methods to design information systems that can then be developed and implemented by end-users.

Background
Traditional techniques of systems design focus on the processes of the system. This approach leads to technical design specifications that confound end-users, requiring the Information Systems (IS) Professional to wholly develop the system. In contrast, the newer object-oriented methodology approaches systems design in a product-oriented fashion which relates more closely to the way end-users commonly think. This feature of object-oriented methodology raises the possibility for end-user development and implementation of a system from a design developed with those techniques. Therefore, this study investigates whether it is possible to use object-oriented methods for the initial design of a system by an IS Professional, with the resulting design so clear and intuitive to the end-user that he/she can develop the system easily from design, using modern application development tools.

Rationale
This study has a general and a specific value. Its general value is that it will make a contribution to research on object-oriented methodology by focusing on an aspect not considered by past researchers: the extent to which the methodology enables end-user system development. The specific value of this research is the development of an information system design for WMU Evaluation Center; hypothetically that design will be one which the Center's end-users will be able to develop and implement.

Source of Methodology
Object-oriented design methods were used to design this software. These principles are similar to those used by programmers, with the exception that the end-users will apply them outside-in, in contrast to inside-out by programmers.

The methodology comes from articles and books pertaining to object-oriented technology, methods, and systems design. For example, the framework of this research came from Elizabeth Gibson's Object Behavior Analysis, which she asserts results in requirements specifications that "can help to produce a clear, understandable object-oriented application structure" [Gibson, 1990].

The basis for this research was the application of object-oriented methods and tools, including development software currently available that facilitates applications system development. This study used one such development tool, a software package called ObjectVision® Pro, to investigate the ease with which an end-user can progress from object-oriented design specifications to a fully developed and implemented information system.

Object-Oriented Design Research

Object-Oriented Technology came into being with the advent of object-oriented programming languages and has since stimulated wide interest for its capabilities to solve present inadequacies in developing information systems. Development of object-oriented design methods soon ensued to deliver the promise of the technology to the programmer-developer. Research of this technology is currently focusing on four areas: languages, databases, operating systems, and environments [Blackford, 1990].

Object-oriented design methods research primarily has emphasized programmer benefits. For example, *Communications of the ACM* devoted its entire September issue to the discussion of object-oriented design as a systems development plan for programmer-developers. The viability of using such an approach for end-user application development and implementation has been overlooked, a possibility I discovered while browsing the literature on object-oriented systems development.

Methodology

The WMU Evaluation Center has recently upgraded its computer facility with a newly installed computer network; the center requires new project and staff accounting software to replace its existing system, which lacks integration and functionality. This research project provides a design of the new system as a whole; end-user development will initially focus on those modules which will provide functionality that the present system lacks. The center will phase in new portions of the system as the end-user completes their development.

Design proceeded according to the first three stages of Object-Oriented Systems Life Cycle model [Henderson-Sellers et al., 1990] consisting of five stages. First, the Requirements stage, using Object Behavior Analysis [Gibson, 1990], determined the requirements specifications of the software and the end-user. Second, the Design stage produced the design with the aid of schematic charts and/or diagrams. Third, the Development stage let me work with the end-user in co-developing the software, employing the design and an object-oriented tool called ObjectVision® Pro.

This development cycle follows the fountain model where the stages overlap and may be iterated a number of times. However, this research focused on the first four stages; these phases will provide information on the extent a design based on object-oriented methods enables the end-user to develop and implement application software.

Timeline

Work began in early November. The determination of the requirements specifications of the information system and the end-user took approximately two months. In the next month, I designed the information system based on the requirement specifications. However, due to time and other constraints I was not able to work with the end-user to codevelop the information system.

Evaluation

Since this is a field study, involving a single case, formal, statistical evaluation was impossible. Thus, this research should be viewed as a pilot study, investigating the value of this approach.

At appropriate points in the development process, the end-users was to be interviewed about their comfortableness with the development process. The interviews were to assess the users' sense of success in moving from design to complete application development. In addition, user-developed modules were to be evaluated based on the design specifications for correctness and accuracy.

Closure

This research culminated in an object-oriented design of WMU Evaluation Center's project and staff accounting information system, consisting of schematic diagrams/charts, and other documentation. The results of this research are indicated in the following pages.

Object-Oriented Theory

About Object-oriented Technology

Object-oriented technology embodies three basic mechanisms: objects, messages, and classes [Taylor, 1990, 127].

Objects

The one characteristic of object-oriented technology that strongly supports this investigation is the concept of object, which the lay person can easily understand.

An object contains properties and behaviors. In computer terms, properties consist of data and behaviors consist of methods.

An object is more than a container of data and methods; it has attributes such as encapsulation and data abstraction. The former allows an object to protect its data from corruption; only its methods can access its data. The latter allows an object to create new data types; thus, raising the developer's thinking from the level of data storage to the level of the problems he is trying to solve [Taylor, 1990, 34]. However, encapsulation and data abstraction were not incorporated in the information system design in this paper, because of the limitations of the tool (ObjectVision) used.

Messages

An object communicates with another object through a common message interface that allows polymorphism, the ability to hide implementation of one object from another object.

Classes

A class is a general prototype defining the methods and variables for a specific type of object. Objects in a given class, called instances of that class, are identical in form and behavior but hold different data in their variables [Taylor, 1990, 132].

By defining subclasses of one another, inheritance enables the subclasses for a given class to make use of the methods and variables of that class, resulting in code reusability.

The design in this paper did not include classes because ObjectVision does not support them.

Object-oriented Analysis and Design

The power of object-oriented technology is the use of the same conceptual model in analysis, design, and development; they are thought in terms of object types, objects, and their required behaviors. This characteristic of object-orientation offers the possibility for the end-user to develop an information system from the design created by the IS professional because there is a "common thinking interface" between the end-user and the IS professional.

On the contrary, in traditional methodologies, conceptual models used for analysis, design and development differ from one another [Martin, 1992, 70]. Consequently, the end-user would not be able to use the IS professional's design without having had systems development training.

Another benefit of object-orientation is the result of a design that is in a natural language that the end-user can easily understand. Moreover, the design in terms of objects and their behaviors has in common in the way people commonly think.

Conversely, traditional design produces diagrams such as data-flow diagrams that use symbols that only the trained systems developer could comprehend. Also, this design reduces people to think at the level of the computer.

Object-oriented design is not concerned with the technical implementation of the object's characteristics. When used with a CASE tool, an object-oriented design lets the end-user to specify the desired result and leave the technical details to be generated by the CASE tool.

The Evaluation Center

The Evaluation Center is a research and development unit reporting to the Dean of Western Michigan University's College of Education. The Center engages in research, development, dissemination, service, instruction, and leadership in the advancement of the theory and practice of evaluation, as applied to education and human services.

The Center manages the accounting of the eighteen projects it is currently involved in using a semi-automated accounting system, a combination of DOS-based applications consisting of Quattro Pro, WordPerfect, and Budget Master. However, there are still many functionalities its present system does not provide. For example, the current system is unable to automatically show the status of each project. Additionally, duplicate data are being entered into the system, promoting redundancy and the likelihood of error in the data entry process.

The Center presently relies on the reports generated by its accounting system and needs an on-line system where the users of the accounting information can be apprised of the immediate status of the projects.

To manage the increasing information flow, the Center installed a new network, which is loaded with MS DOS, Microsoft Windows 3.1, and Quarterdeck's Desqview. With this regard for future information increase, the Center requires that the new system be extensible to accommodate changes in information needs. Since the Center has a small staff, it is essential for the staff to be able to modify the system easily, while maintaining system integrity and security.

Because of the above factors, Dr. Arlen Gullickson assigned Elissa Stringer to find an intern to develop this information system. Through Dr. Judy Yaeger, I volunteered, with the stipulation that I use object-oriented methodology to support the investigation of my thesis, to develop a design of the accounting information system.

Application of Object-Oriented Methodology

Object-Orientation and The Evaluation Center

Considering the requirements of the EC project and staff accounting information system, object-orientation would be the ideal methodology to be used in its design and development.

Object-oriented methodology provides the Evaluation Center with a systems design that gives a clear understanding of the requirement specifications of the information system. The design is also intuitive to the end-user that will enable him/her to develop the information system.

To complement object-oriented design, an object-oriented development tool would have to be selected. Since object-oriented programming languages require systems development training, an object-oriented CASE tool would be the choice. However, such a tool is too costly and requires too much computer resources which the Evaluation Center does not have. Therefore, a cheaper alternative, ObjectVision Pro 2.1 for Windows, without true object-oriented characteristics was chosen.

The selection of ObjectVision as the development tool represents a change in the Evaluation Center's information system from the DOS platform to that of Windows. This does not pose a problem because the users of the accounting information system are high-ranked staff, who have Windows capable personal computers.

About ObjectVision Pro 2.1

ObjectVision Pro 2.1 for Microsoft Windows is a visual form-based CASE tool that was designed for non-programmers. Each form can be linked to multiple database files and is created with objects such as fields, texts, graphics, buttons, and dialog boxes. In a way the form itself is an object.

Since there is no programming involved, a call to a "subroutine" is actually a call to a form. The forms can be layered on top of one another; thus, forming a hierarchy of forms.

Through the event tree, each object responds to certain events such as a click of the mouse button. Through decision trees, business rules could be applied to the application.

ObjectVision offers extensibility by the creation of dynamic link libraries (DLLs). This feature serves to support the creation of functions that later may be found lacking in ObjectVision.

ObjectVision was chosen to use in this research because of its graphical user interface which allows a person (end-user) to develop an application without knowing programming and to do

rapid prototyping. Also, ObjectVision offers the potential to integrate all the functionalities of an accounting information system. In addition, the completed application offers a user-friendly interface; commands are issued by pointing the arrow cursor at the desired button and clicking the mouse.

Due to the nature of ObjectVision being not a true object-oriented development tool, we designed the EC Project and Accounting Software by outlining primary objects and specifying their required behaviors and properties. This method will produce a clearer understanding of the application, concerned with solving problems the application is set to solve. In contrast, traditional methods would dwell on the low-level concerns such as solving problems of data types.

Designing an Object-Oriented Information System

According to David Taylor, an object-oriented information system is built in three distinct layers:

- **Developing a model** of the business identifying the required objects and their relationships to one another.
- **Designing the classes** supporting these objects and accomplishing their responsibilities within the model.
- **Building applications** on top of the model using rapid prototyping.

Responsibility-Driven Design

Traditionally, information systems were designed using either procedure driven or data driven approaches. However, Taylor suggests using the responsibility driven approach because internal structure of the objects would be ignored. This approach would especially be important in designing the EC IS based on the capabilities of ObjectVision as a CASE tool. Since developing an ObjectVision application is based on forms, details (internal structure) of the objects are hidden.

The responsibility-driven approach is achieved through object behavior analysis, OBA. The product of OBA is the formulation of a requirements specifications, in terms of required behavior, of the information system to be developed.

According to Elizabeth Gibson, there are five steps to OBA [Gibson, 1990, 246]:

- Identify the behaviors
- Define the objects
- Classify the objects
- Identify relationships
- Model the processes.

Identify the Behaviors
To determine system behaviors of the EC information system, the following script was produced.

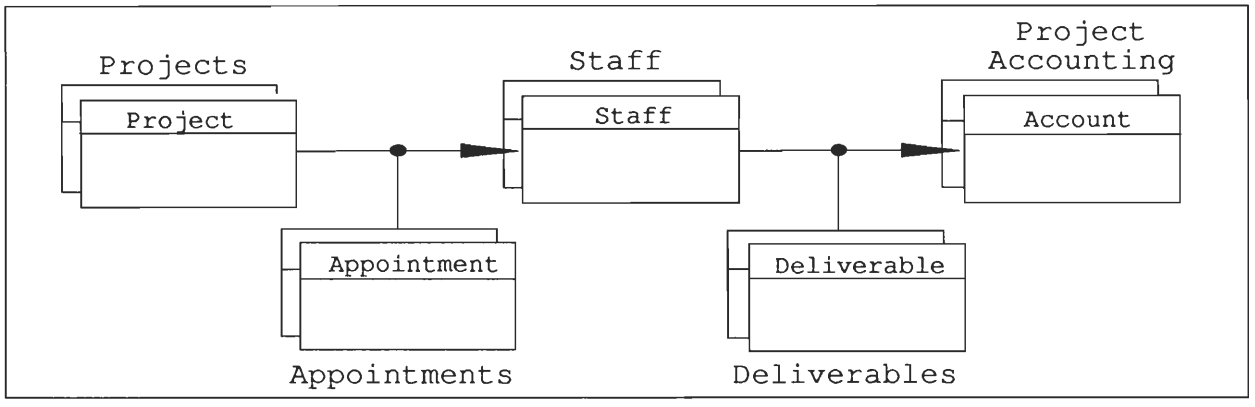
Agent	Action	Recipient	Result
Project	Assigns	Appointment	Appointment determines who is available to work on project
Appointment	Selects	Staff	Staff responds that he/she is available to work on project
Staff	Produces	Deliverable	Deliverable results in expenditures in salary, telephone charges and computer charges
Deliverable	Updates	Project Account	Project account changes to reflect expenditures on deliverable
Project Account	Reports	Status	Status of project changes

Define, Classify the Objects and Identify Relationships
From the above script, the following objects and their behaviors/ responsibilities were identified:

Object	Behavior/Responsibilities
Project	Hold, report, revise: • Name • Begin date • End date • Proposed budget • Staff assigned
Appointment	Find staff available for project (Resource planning) Select staff Determine staff is available for project Assign appointment level to staff Report personnel cost for projected project
Staff	Record hours worked on project Record phone usage Hold, report, revise: • Staff name • Appointment level Hold, report, revise account(s): • Account number and name • Begin and termination dates • Date of last renewal • FTE assigned to account • Salary
Deliverable	Hold, report, revise: • Name • Designated delivery date • Actual delivery date • Staff responsible Indicate if staff responsible is not assigned to project
Project Accounting	Hold, report, revise (Gold sheet information): • Staff name • Project name • Actual time (FTE) allocation by date of effort Indicate if staff has proper allocation/ approval for project expenditure of funds Report project proposed budget and actual expenditures

Model the Processes

From the results of OBA, the following model of the Evaluation Center was derived.



Model of the Evaluation Center

Referring to the above model, each of the objects depicted represents an individual form in ObjectVision. The behaviors/responsibilities of the objects will be performed by the next layer of forms/objects.

Results of Investigation

Evaluation of Design as Applied to ObjectVision

From the stand point of the end-user, the above design of the EC information system is in a natural language which is easily understood. The end-user would just have to transform the simple language of the design into an ObjectVision information system. However, despite ObjectVision's intuitive graphical development environment, end-user who is not familiar with the Microsoft Windows "object-oriented" operating environment would find difficulty implementing the design.

Due to Windows' simplicity, object attributes are hidden from the user. For example, when ObjectVision is launched, a simple blank work area with limited menu choices is brought up to the end-user; thus, little guidance is presented to the end-user to proceed. In other words, ObjectVision was designed for the IS professional and does not provide the end-user with a "workshop" feature which could steer the end-user through the process of development.

Other problems were encountered in using ObjectVision. It was unexpected that ObjectVision would run slowly in a network environment. In addition, its graphical property adds to slowing down the performance of an ObjectVision application. As a result, data entry would have to be done using a regular DOS relational database such as Paradox.

There is also a problem with the methodology used in the above design: the object-oriented methodology currently available was designed for programmers who mostly program with an object-oriented programming language (OOPL). Unlike OOPL, ObjectVision is a visual development CASE tool and there is no code to show the flow of the application. As a result, the systems analyst would have to conceptualize information systems development in a different way: design of the information system in ObjectVision would have to be modified not to include features of OOPL such as classes, but to include ObjectVision's hierarchy of forms.

Next Steps

Since the end-user would have difficulty developing the information system with ObjectVision, there are three alternative ways to implement the above design:

- Train the end-user to use ObjectVision; currently ObjectVision does not have a feature to guide the end-user with the development.
- Replace ObjectVision with a true object-oriented development tool, preferably one that has a development "workshop". However, presently there is no development tool designed for end-users.
- Recruit an IS professional to develop the information system using ObjectVision.

Conclusion

Object-oriented design offers many benefits to enable the end-user to use the IS professional's design to develop an information system. To enable the end-user to use the design effectively, he/she would need to use an object-oriented development tool that guides him/her with the development. In addition, the development tool would have to have the capability to utilize object-oriented properties such as encapsulation, data abstraction, and classes. However, the development tool, ObjectVision, used in this investigation does not possess such properties. Moreover, an object-oriented developmental tool for end-user is unavailable. As a result, to implement the design the end-user has, to some extent, to be trained in systems development.

In spite of the absence of an object-oriented development tool for end-users, object-oriented design bridges the gap between the thinking level of both the developer and the end-user. In the future, when an object-oriented tool for end-user is available, the end-user could develop the information system while leaving the analysis and design to the IS professional. Consequently, IS professionals could focus on solving other problems.

Bibliography and References

Blackford, John. "The Story of "O"",*Personal Computing*, vol. 14 no. 6 (June 19, 1990), pp. 83-86.

Booch, Grady. *Object-Oriented Design with Applications*, Reading, MA: Addison-Wesley Publishing Company, 1991.

Cattell, R.G.G. *Object Data Management*, Reading, MA: Addison-Wesley Publishing Company, 1991.

Cox, Brad J. "There Is a Silver Bullet: A software industrial revolution based on reusable and interchangeable parts will alter the software universe",*Byte*, vol. 15 no. 10 (October 1990), pp. 209-218.

Gibson, Elizabeth. "Objects--Born and Bred: Object Behavior Analysis is a step-by-step, object-oriented approach to analysis", *Byte*, vol. 15 no. 10 (October 1990), pp. 245-254.

Graham, Ian. *Object-Oriented Methods*, Wokingham, England: Addison-Wesley Publishing Company, 1991.

Harmon, Paul. "Object-Oriented Fourth Generation Language and CASE Products: Part I", *Object-Oriented Strategies*", Vol. II, No. 12 (December 1992), pp. 1-11..

Henderson-Sellers, Brian and Edwards, Julian M. "The Object-Oriented Systems Life Cycle", *Communications of the ACM*, vol. 33 no. 9 (September 1990), pp. 142-159.

Korson, T. and McGregor, J.D. "Understanding Object-Oriented: A Unifying Paradigm", *Communications of the ACM*, vol. 33 no. 9 (September 1990), pp. 40-60.

Martin, James and Odell, James. *Object-Oriented Analysis and Design*, Englewood Cliffs, NJ: Prentice-Hall, 1992.

McClure, Steve. *Object Technology: A Key Software Technology For The '90s*, International Data Corporation White Paper.

Taylor, David. *Object-Oriented Technology: A Managers Guide*, Reading, MA: Addison-Wesley Publishing Company, 1990.

Taylor, David. *Object-Oriented Information Systems: Planning and Implementation*, New York, NY: John Wiley & Sons, Inc., 1992.

Wirfs-Brock, Rebecca J. and Johnson, Ralph E. "Surveying Current Research in Object-Oriented Design", *Communications of the ACM*, vol. 33 no. 9 (September 1990), pp. 104-124.