Electronic Commerce Studies
Vol. 2, No.1, Spring 2004
Page 75-94

An Object-Oriented Analysis Method for Customer Relationship Management Information Systems

Jyh-Jong Lin
Chaoyang University of Technology
Ming-hang Lee
Chaoyang University of Technology

Abstract

For the advances of Internet technologies in recent years, Electronic Commerce (EC) has gained many attentions as a major theme for enterprises to keep their competitiveness. Amongst all possibly desired endeavors for the EC, research has shown that effective management of customer relationships is a major source for keeping competitive differentiation. Therefore, it is commonly recognized as an important goal for an enterprise to promote its management of customer relationships through a prospect information system on the Internet to achieve the so-called Business-to-Customer EC. In this paper, we propose an object-oriented analysis method for the development of such a Customer Relationship Management Information System (CRMIS). The approach starts from the identification of prospect customers and their desired behaviors under preferable execution environments, and ends with the specification of system- internal objects/entities that collaborate to satisfy these behaviors and environments. The method is a use case driven approach with UML utilized and extended as its tool. To illustrate, the method is applied to an exemplified CRMIS for house agency.

Keywords: electronic commerce, customer relationship management, information system, object-oriented analysis, method

1.Introduction

For the advances of Internet technologies in recent years, Electronic Commerce (EC) has gained many attentions as a major theme for enterprises to keep their competitiveness. Amongst all possibly desired endeavors for the EC, research has shown that effective management of customer relationships is a major source for keeping competitive differentiation. Therefore, it is commonly recognized as an important goal for an enterprise to promote its management of customer relationships through a prospect information system on the Internet to achieve the so-called Business- to-Customer EC. Also, as a common recognition, such a Customer Relationship Management Information System (CRMIS) that realizes the B- to-C EC application needs to explicitly capture and manage for prospect customers their desired behaviors under preferable execution environments.

In the literature, many discussions related to CRMIS have been presented such as personalization methods and customer decision support systems; it is therefore no lack of technical solutions about CRMIS. Nonetheless, any thorough analysis and design methods for CRMIS, which may contributively result in the system effectively satisfying the requirements of prospect customers under their preferable execution environments, are still few nowadays; such methods are explicitly needed in that it has well been recognized in the literature that analysis and design are important in developing a computer-based application where analysis plays a more significant role for collecting user requirements about the application domain (e.g., desired behaviors and execution environments of the application) -- failure to identify appropriate requirements usually results in late delivery, poor quality, and high maintenance costs. In general, system analysis can be accomplished by using function- or data- or object-oriented methods where the development of object-oriented ones is specifically motivated by the drawbacks and problems in the other two kinds: the significant features and benefits of object-oriented techniques such as inheritance of object specifics and information abstraction/hiding in an object would make the system constructed easy to understand, maintain, and reuse.

As CRMIS concerns especially its effectiveness on comprehensibility and maintainability for satisfying customers’ (often complex but changeable) requirements, it is therefore not uncommon in our knowledge to take advantage of object-oriented techniques for enhanced analysis and design of a CRMIS. Amongst those existing object-oriented methods, the well-known use case driven one in has already been ascertained by many researches and implementations for its robust process and resultant sound UML artifacts. Therefore, in this paper, we propose such a use case driven method that extends UML notations for specifically supporting an enhanced analysis for CRMIS. The approach starts from the identification of prospect customers and their desired behaviors under preferable execution environments; those artifacts identified will be explicitly specified in the use case and activity diagrams adapted from UML. With desired behaviors, the method ends with the specification of system-internal objects/entities that collaborate to satisfy these behaviors. For illustration, the method is applied to an exemplified CRMIS for house agency.

This paper is organized as follows. Section 2 presents our method that results in the creation of four diagrams, including the use case, activity, class, and sequence diagrams. The method is then illustrated in Section 3 by applying it to the analysis and specification of a CRMIS for house agency. Finally, Section 4 has the conclusions and our future work.

2.The analysis method

Our method is use case driven with the following steps: (1) creating a use case diagram that specifies the prospect customers of a CRMIS and their desired behaviors of using the system under preferable execution environments; (2) creating an activity diagram that presents how such behaviors proceed interactively to satisfy the expectations of these customers; (3) modeling a class diagram that describes any system-internal objects/entities that collaborate together to support these behaviors; and (4) modeling a sequence diagram that specifies how such objects/entities collaborate to support these behaviors. As mentioned above, these diagrams are UML based with desired extensions for the analysis of a CRMIS.

2.1 creating use case diagram

The first step in our method is to specify with a use case diagram what users desire such that different users can use various functions under different execution environments that best fit what they really need. Six concepts that result in the extensions on UML are employed in creating a use case diagram suitable for a CRMIS. As illustrated in Figure 1, these concepts include classifying users, categorizing users execution environment, building client unit, and specifying trigger event, customization, and personalization.

2.1.1 classifying users

Classifying users is a generic concept in customer relationship management (CRM). When designing a CRMIS, we should comprehend first who gets benefits from the functions and features of the system. Therefore, it is better to classify prospect users of the system into different kinds in advance before the design work gets started. Therefore, as shown in Figure 1, we may classify general users into various special kinds where each one may be further classified into some lower-level special kinds. As a common knowledge for such a resultant generalization-specialization structure, a lower-level kind of users could possess (i.e., inherit) the characteristics of its parent kind(s) of users.

2.1.2 categorizing users execution environment

On the Internet, many media can be used for providing users with effective interfaces such as cell phone, PDA, and game box. Since the hardware and operating systems (i.e., execution environments, or say, platforms) that support these media are different, it is needed when designing a CRMIS to consider all possibly preferable execution environments that prospect users of the system may use, and then categorize them into some special kinds as if these users may be

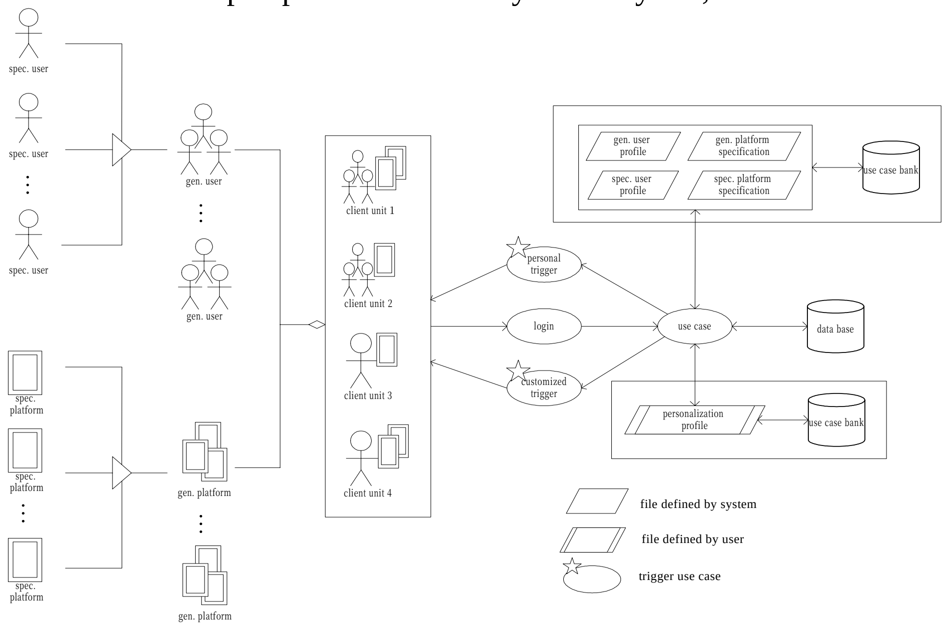

Figure 1: use case diagram

classified into corresponding special kinds. As above, this also results in a generalization-specialization structure: general execution environments are categorized into some special kinds where each one is further categorized into some lower-level special kinds.

2.1.3 building client unit

Client unit is a mixture of classified users and categorized execution environments. For instance, in Figure 1, four kinds of client units can be built up by the mixture of two kinds of users and two kinds of platforms. In client unit 1, in particular, it specifies the situation that general users will use general (all available) platforms. For this most general situation, of course, we need to make the system support well all available platforms possibly used by each prospect user. In contrast, client unit 3 specifies that a special kind of users will use a particular kind of platforms. For this most special situation, the system should be designed more focusing on how to provide these special users with desired and convenient functions under such a special kind of execution environments.

2.1.4 specifying trigger event

Trigger event is a special use case that enacts users to accomplish some specific operations that may result further in other use cases to be performed. It can be used in an active or passive way to stimulate users: activeness means that it may send messages to these users without any external forces, while passiveness represents that other use cases or events are needed to make it trigger these users. In CRMIS, trigger event is an important service that delivers some useful messages and information to users without unnecessary delays. As shown in Figure 1, a customized trigger event is specified to model such a desired service for a kind of users, and a personal trigger event is specified for a specific user of this kind.

2.1.5 specifying customization

With various kinds of client units through the categorization of users and platforms, the system is then able to offer customized functions under preferable platforms for different kinds of users. The objective of customization is to focus on special kinds of users and the users of these kinds will be offered with these customized functions for their use under their preferable platforms. Therefore, the system can purposefully offer different functions to its users based on their characteristics and expected platforms. As shown in Figure 1 that four kinds of client units are specified in our example, the system is therefore customized to have four kinds of profiles or specifications that provide guidance for determining suitable use cases for these four kinds of client units respectively.

2.1.6 specifying personalization

After achieving customization, the system may be further refined as a personalized system if it allows users to designate their preferable functions by themselves. For this purpose, the system, as shown in Figure 1, is specified with a personal profile for each specific user that records the functions (use cases) designated by this user.

2.2 creating activity diagram

As defined in UML, an activity diagram can be used to specify the possible workflow of system functions. Therefore, for our needs, we can utilize it in this step to specify the workflow of the critical customization and personalization processes presented in the previous step. However, the mechanisms of the activity diagrams in UML are not sufficient for this purpose. As shown in Figure 2, we therefore follow as well the six concepts used above in creating a use case diagram to create our UML based activity diagram with the adapted mechanisms described below.

In Figure 2, a rectangle with solid lines is used to hold use cases to be selected by system users. For the rectangles with dotted lines, they specify a range that holds use cases obtained from such resources as use case bank and customization/personalization profiles. For instance, use cases in the top rectangle are obtained from the use case bank; the two profiles that fill in the middle and bottom rectangles will control these use cases to determine if they are retained during the customization and personalization processes.

-- specifying customization and personalization workflow

Here in our activity diagram, customization and personalization processes are modeled by first retrieving use cases from the use case bank. These use cases are then determined, based on the customization and

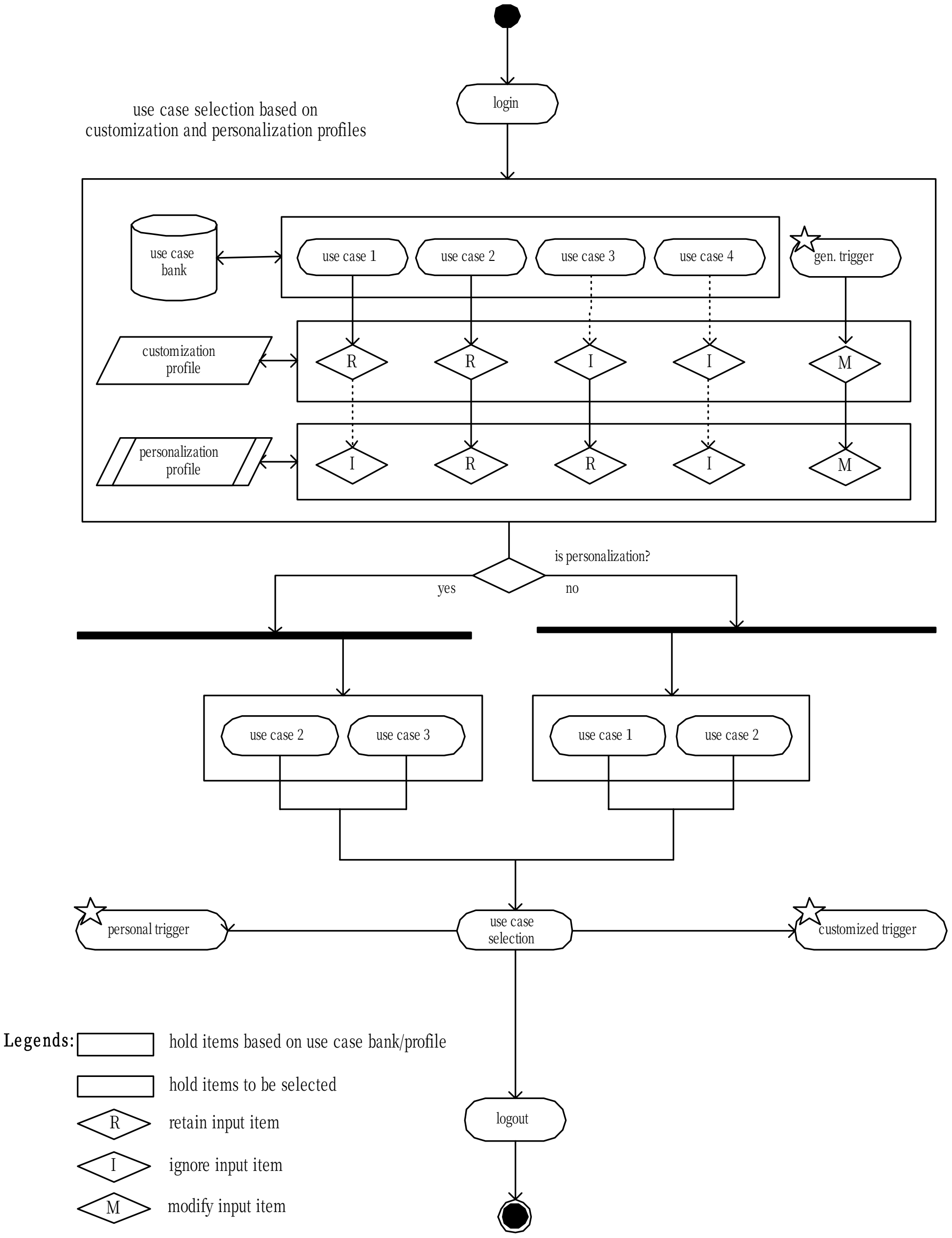

Figure 2: activity diagram

personalization profiles, if they should be retained as desired for a prospect kind of users or a specific user of this kind. Two decisions will be made consecutively based on the customization and personalization profiles respectively, and there are three possible choices to select at each decision: (1) retain the use case, (2) ignore the use case, and (3) retain but modify (part of) the use case. The arrows entering decisions represent selections on relevant use cases: a solid arrow means that the relevant use case is retained (with or without modification) at the customization level for a prospect kind of users or at the personalization level for a specific user of this kind; a dotted arrow means that the relevant use case is ignored at the customization level since this kind of users do not need it, or at the personalization level the specific user does not need it. For illustration in Figure 2, use cases 1 and 2 are retained, together with the general trigger event that is modified as a customized one, at the customization level for a kind of users; then, at the personalization level, use cases 2 and 3 are retained, together with the customized trigger event that is modified as a personal one, for a specific user of this kind; these results are specifically shown at the lower part of Figure 2.

It should be noticed that in decisions for personalization, users might have not defined their personal profiles although the system needs these files to decide what use cases should be retained for them. Therefore, in such a situation that users have not defined their personalization profiles, the system will support only use cases retained at the customization level for each of these users. For those users with their personal profiles, personalization decisions will be offered specifically to each of them.

2.3 modeling class diagram

A class diagram is used to describe any system-internal objects/entities that collaborate together to support desired system behaviors (here for a CRMIS, to support the workflow of the customization and personalization processes). In UML, it can be derived from the use case diagram with three stereotypes: boundary, entity, and control classes - a boundary class represents an interface used to interact the system with an actor as a bridge; an entity class models the information and associated behaviors in the real world; and a control class provides the desired behaviors for accomplishing one or several use cases.

In addition to these three kinds of class, however, a new kind class of ‘trigger control’ is introduced in our class diagram. This is because in our use case diagram, trigger event is a special use case that can be used in an active or passive way to make other use cases to be performed. That is, in addition to the three UML stereotypes, we employ a new one to represent a particular control class that is responsible for the execution of a trigger event. Figure 3 shows the class diagram derived from the use case diagram in Figure 1. It is noticed that as shown in Figure 3, various relationships may occur between classes such as association and inheritance relationships. As a common recognition for object-oriented paradigm, these relationships (together with other features such as information hiding in individual classes) are particularly useful for making the system constructed much easier to understand, maintain, and reuse.

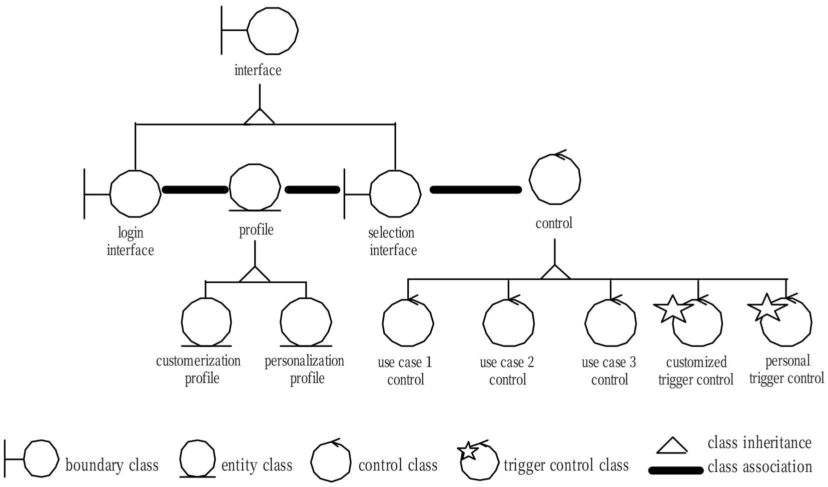

Figure 3: class diagram

2.4 modeling sequence diagram

With classes defined and specified for creating objects to support desired system behaviors, it is now possible to depict a sequence diagram that specifies how such objects collaborate to support these behaviors. Figure 4 is our sequence diagram where there are three differences from the traditional UML sequence diagram: (1) client unit is used that replaces actor; (2) trigger control object is specifically used for accomplishing the execution of a trigger event in an active or passive way; and (3) dotted arrows pointed to a rectangle are used to specify which objects are required for customization and personalization -- those (boundary and control) objects in the rectangle are needed to support customized behaviors (customized use cases) for a prospect kind of users or personalized behaviors (personalized use cases) for a specific user of this kind. As illustrated in Figure 4, after a client unit logs in the system, the login interface object will verify the customization and personalization profiles to determine which objects are required for desired customized and personalized behaviors: control objects for use cases 1 and 2 and the customized trigger event are required to support the customized behaviors, and control objects for use cases 2 and 3 and the personal trigger event are required to support the personalized behaviors. With required control objects, use cases 1 and 2 and the customized trigger can then be performed to support the customized behaviors for a prospect kind of users, and use cases 2 and 3 and the personal trigger can be performed to support the personalized behaviors for a specific user of this kind.

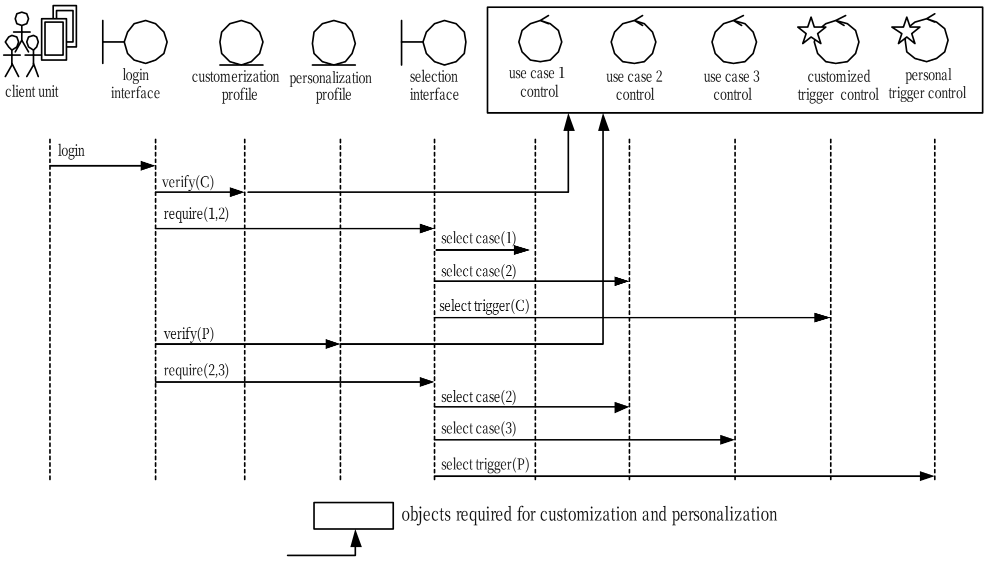

Figure 4: sequence diagram

3. an example－analysis of house agency information system

Providing customers with desired information is very important in selling houses. Many house agency companies have already shown houses information on their web sites with convenient searches for their customers. Although it is obviously desirable for different kinds of customers to retrieve different styles of information by various search ways, almost all of existing web sites provide only useless or disordered houses information and hence no means to their customers. For these shortcomings, we present in this section how search functions can be constructed much useful and meaningful for different kinds of users by applying our analysis method.

3.1 modeling search functions with use case diagram

3.1.1 classifying users

Classifying users may be achieved by various indices. For instance, before designing a house agency information system, we may need to comprehend first how those users, who will use and hence benefit from this system, operate well computers. Therefore, it is possible to classify these users into various kinds based on the degree of their familiarity with operating computers; for each kind, appropriate system functions may then be offered. For example, for the users that are unfamiliar with operating computers, we may make the system to provide a helper or a wizard to help their use of the system. In contrast, many of these helps are unneeded for those who are already familiar with operating computers.

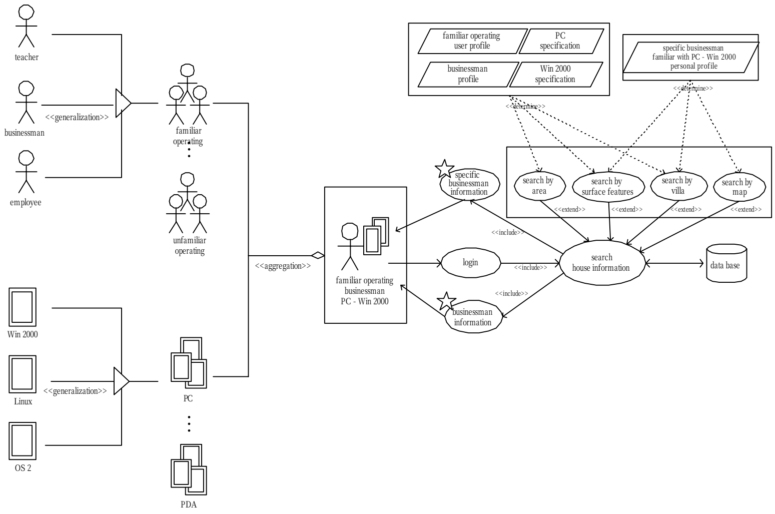

Figure 5: use case diagram for house agency information system

In addition to the above index, these kinds of users may be further classified by their occupation: different occupations may affect their buying ability, styles of information desired, and search ways preferred. Therefore, with such two indices considered, users can be classified twice for the system to provide each kind with appropriate functions. In our example as shown in Figure 5, all of the users who are already familiar with operating computers are further classified by their occupation into three lower-level kinds, including teacher, businessman, and employee.

3.1.2 categorizing users execution environment

Many techniques for connecting to the Internet are already available nowadays. However, these techniques impose different media for providing users with effective interfaces such as cell phone, PDA, and game box. Since the hardware and operating systems that support these media are different, it is needed when designing a CRMIS to consider all possibly preferable execution environments that prospect users may use, and then categorize them into some special kinds as if these users may be classified into corresponding special kinds. This results in a generalization-specialization structure as shown in Figure 5 for our example: the possible platforms for a house agency system includes PC and PDA where PC can be further categorized into Win 2000, Linux, and OS 2.

3.1.3 building client unit

Based on the results of classifying users and categorizing execution environments, we can then derive various client units each of which addresses a particular kind of user and a possible execution platform. At each client unit, corresponding profile(s) and platform specification(s) are explicitly featured that contain information about desired functions (use cases) for those users that this client unit addresses. For our example in Figure 5, the client unit illustrated addresses the kind of businessman who is familiar with operating computers under the platform PC – Win 2000. Therefore, at this client unit, desired functions for the businessman familiar with operating computers will be offered that are suitable for execution under PC – Win 2000.

3.1.4 specifying trigger event

With client units identified, trigger events are then specified to model possible desired services that deliver useful messages and information to the businessman without unnecessary delays. As shown in Figure 5, a customized trigger event is specified to address such a desired service for the kind of businessman stated above, and a personal trigger event is specified for a specific businessman.

3.1.5 specifying customization

With trigger events identified above, desired functions for each client unit then are identified that can be offered by many ways. One better offering is to use/adapt existing functions that had been constructed when developing similar systems in the past. In our knowledge, this is feasible by using a use case bank that saved all successful use cases of similar systems constructed in the past. When this time developing a new system, use cases in the use case bank that are possibly desirable in the new system can be retrieved first from the use case bank.

Then, with all possibly desirable use cases identified, we consider which of them are actually needed for each client unit and hence retained for those users that this client unit addresses. If it is needed, some modifications to (part of) these use cases may be performed in order to provide most suitable functions for these users. In our method, this is achieved by screening the customization (i.e., businessman familiar with PC – Win 2000) profile that contains features of these users to help determine what use cases will be actually desirable. In our example, three search functions: ‘search by area’, ‘search by surface feature’, and ‘search by villa’ are offered to the kind of businessman for their needs of searching desired houses. Note that the ‘search by area’ function will be replaced by the ‘search by map’ function during the personalization process that we will discuss below.

3.1.6 specifying personalization

Based on the customization for a kind of user, the system may provide further a personalized service for each specific user where he/she can designate his/her preferred functions by him/herself. For this purpose, the system utilizes a personal profile for each user that stores the functions defined by the user him/herself. That is, after customized use cases have been determined, the personal profile for a specific user is examined to identify which of them are actually preferable for this user (new functions may also be added if designated by the user). For our example in Figure 5, this means that based on the personalized profile for a specific businessman familiar with PC – Win 2000, one (‘search by area’) of the three customized search functions is replaced with the ‘search by map’ function for satisfying the personal needs of the specific businessman.

3.2 modeling search functions with activity diagram

With use cases identified above, an activity diagram is then specified to describe the workflow of the customization and personalization processes in the system. Figure 6 shows the resultant activity diagram derived from the use case diagram in Figure 5. In particular, discussions about the customization and personalization features in the activity diagram are presented as follows.

-- specifying customization and personalization workflow

In this process, all use cases related to searching houses are first retrieved from the use case bank. Through the customization profile for the businessman, some of these use cases may be retained or modified for these users. As Figure 6 shows, once these users login the system, use cases for them are determined by the customization profile, which include the three ‘search by area’, ‘search by surface feature’, and ‘search by villa’ functions. After the customization is achieved, these use cases are considered then by the personalization profile for determining which of them are preferable for a specific businessman. As shown in Figure 6, based on the personalization profile, the ‘search by area’ function is replaced with the ‘search by map’ function for satisfying the personal needs of the specific businessman.

3.3 modeling search functions with class diagram

At this step, a class diagram is derived from the use case diagram to describe the system-internal objects/entities that collaborate together to support desired system behaviors (here in our example, to support the workflow of the customization and personalization processes for the businessman). Figure 7 shows the class diagram derived from the use case diagram in Figure 5.

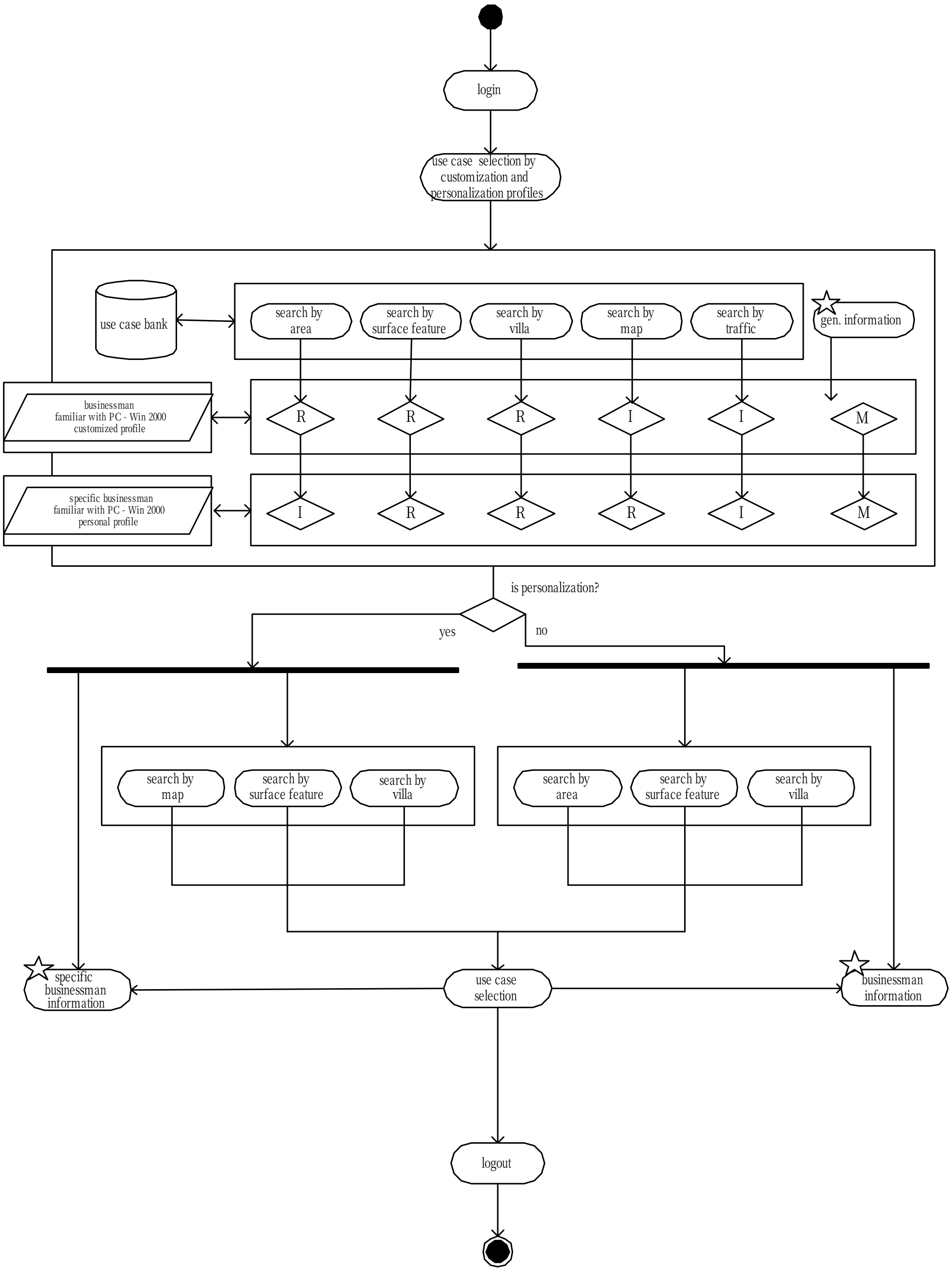

Figure 6: activity diagram for house agency information system

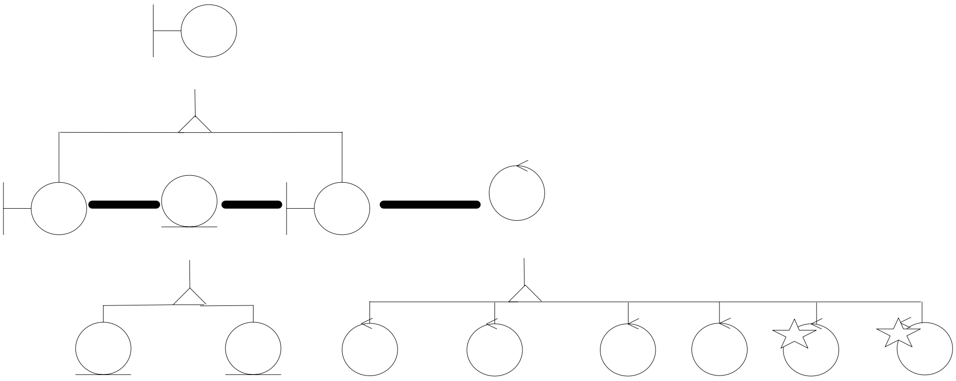

Figure 7: class diagram for house agency information system

3.4 modeling search functions by sequence diagram

Figure 8 is a sequence diagram for our example search functions for the businessman. Like the activity diagram, the sequence diagram can express the control flow of the customization and personalization processes. The dotted arrow and the rectangle frame display together what class objects are needed for the desired customized and personalized behaviors. When a businessman enters the system, a verification activity is performed to check if there exists a customization profile for the kind to which he/she belongs as well as a personalization profile for him/her individually. If there exists a customization profile, the customized trigger control object would provide him/her actively with ‘customized information’ that is more general and suitable for any businessman. Then, if there exists a personalization profile, the personal trigger control object would provide him/her further with ‘personal information’ that is more specific and suitable only for him/her individually.

4.conclusions and future work

In this paper, we extend UML notations with customization and personalization features for constructing a CRMIS. This is useful for alleviating the lack of a comprehensive modeling method for the analysis of a CRMIS that would contributively result in the system effectively satisfying the requirements of prospect customers under their preferable execution environments. Based on CRM concepts, our method focuses on the desired behaviors of customers to emphasize the influence of various client units on the analysis of the system. As a result, the system serves not only in a more general manner kinds of users, but also in a more specific manner specific users of these kinds and their preferable execution environments. In addition, we employ six enhanced concepts to achieve the customization and personalization purposes. In our knowledge, all of these features are critical for an analysis method to make the resultant CRMIS truly satisfy customers' requirements.

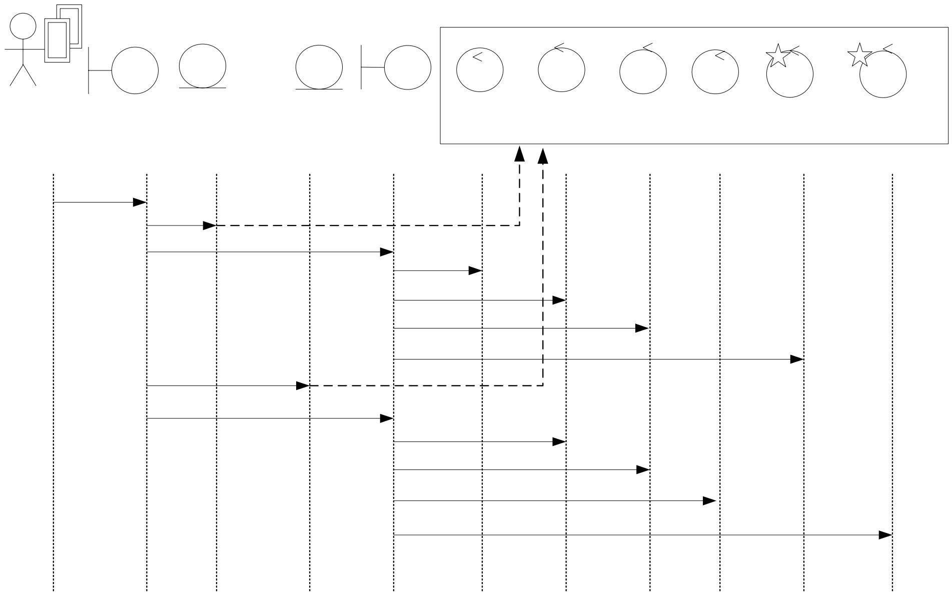

Figure 8: sequence diagram for house agency information system

For illustration, we apply our method to the analysis of search functions for a house agency information system. In fact, not only for this system that is a typical kind of CRMIS, is our method also useful for other kinds of CRMIS. Actually, when analyzing any kinds of CRMIS, our concepts of client unit, customization, and personalization are much helpful for making these systems truly satisfy customers' requirements. Furthermore, although our method is purposed for the analysis of CRMIS, our six concepts result in some extensions on UML diagrams that can actually be used in other application domains like organizational Intranet systems. We will explore next the usability of these concepts and UML-extensions in analyzing such Intranet systems. Finally, based on these concepts, work for analysis is much more robust than any other ways doing before. However, for surviving in the competitive Internet/Intranet environments, promoting the quality of customer services becomes an

businessman
familiar with
PC - Win 2000

login

famili
c

login

essential issue. Therefore, it is demandable for analysts to use a more robust analysis way to develop systems that truly satisfy customers' requirements. Our work presents a possible discussion on this need.

As our future work, we will continually explore the design work for CRMIS based on the artifacts of our current analysis method. It is common to recognize that in the design phase, system architecture plays a major role where possible communication standards on the Internet must be considerably embedded. These standards may include for example XML, SOAP, UDDI, and WSDL that are used together to provide customers with the so-called Web services from multiple service providers across the Internet. Therefore, in our design work, we will particularly focus on how those (boundary, entity, and control) objects identified in the analysis phase can provide desired behaviors under the Internet environment with these standards embedded.

Acknowledgments

The material presented in this paper is based on work supported by Chaoyang University of Technology under Grant CYUT 91-M-004. The authors are grateful to K.Y. Chang and C.Y. Chen for their helps in preparing the artwork of this paper.

References

Adomavicius and A. Tuzbilin, “Using Data Mining Methods to build Customer Profiles,” *IEEE Computer*, Feb. 2001, pp. 74-82.

Cingil, A. Dogac, and A. Azgin, “A Broader Approach to Personalization,” *CACM*, vol. 43, no. 8, Aug. 2000, pp. 136-141.

Kramer, S. Noronha, and J. Vergo, “A User-Centered Design Approach to Personalization,” *CACM*, vol. 43, no. 8, Aug. 2000, pp. 45-48.

Mobasher, R. Cooley, and J. Srivastava, “Automatic Personalization Based on Web Usage Mining,” *CACM*, vol. 43, no. 8, Aug. 2000, pp. 142-151.

Mulvenna, S. Anand, and A. Buchner, “Personalization on the Net using Web Mining,” *CACM*, vol. 43, no. 8, Aug. 2000, pp. 123-125.

Ba, S. et al., “Using client-broker-server architecture for Intranet decision support,” *Decision Support Systems*, vol. 19, 1997, pp. 171-192.

O'Keefe and T. Mceachern, “Web-based Customer Decision Support Systems,” *CACM*, vol. 41, no. 3, March 1998, pp. 71-78.

Cameron, “An Overview of JSD,” *IEEE Trans. on Software Engineering*, vol. 12, no. 2, 1986, pp. 222-240.

Jackson, *System Development*, Prentice-Hall, 1983.

Yourdon, *Modern Structured Analysis*, Prentice-Hall, 1989.

Hull, R. and King, R.“Semantic Data Modeling: Survey, Applications, and Research Issues,” *ACM Computing Surveys*, vol. 19, no. 3, 1987, pp. 201-260.

Peckham J. and Maryanski, F. “Semantic Data Models,” *ACM Computing Surveys*, vol. 20, no. 3, 1988, pp. 153-190.

Hayes, F. and Coleman, D. “Coherent Models for Object-Oriented Analysis,” *ACM OOPSLA Conference*, 1991, pp. 171-183.

Rumbaugh, J. et al., *Object-Oriented Modeling and Design*, Prentice-Hall, 1991.

Shlaer S. and Mellor, S. *Object-Oriented Systems Analysis*, Yourdon Press, 1988.

Jacobson, I. Booch, G. and Rumbaugh, J. *The Unified Software Development Process*, Addison Wesley, 1999.

Jacobson, I. Booch, and G. and Rumbaugh, J. *The Unified Modeling Language Reference Manual*, Addison Wesley, 1999.

Booch, G. Jacobson, I. Rumbaugh, J.*The Unified Modeling Language User Guide*, Addison Wesley, 2002.

Quatrani, T. *Visual Modeling with Rational Rose 2000 and UML*, Addison Wesley, 2001.

Extensible Markup Language (XML)， http://www.w3.org/TR/xml11.

C. Goldfarb and P. Prescod, The XML Handbook, Prentice-Hall, 1998.

Simple Object Access Protocol (SOAP)， http://www.w3.org/2002/ws.

Universal Discovery, Description, and Integration (UDDI) , http://www.ibm.com/services/uddi/ standard.html.

Web Services Description Language (WSDL)， http://www.w3.org/TR/wsdl.