Journal of the Korea Academia-Industrial Cooperation Society
Vol. 12, No. 12 pp. 5849-5855, 2011

http://dx.doi.org/10.5762/KAIS.2011.12.12.5849

User Interface Design Model for Improving Visual Cohesion

In-Cheol Park[1*] and Chang-Mog Lee[2]
[1]Division of Computer·Game, Howon University
[2]Division of Computer Engineering, Chonbuk National University

가시적 응집도 향상을 위한 사용자 인터페이스 설계 모델

박인철[1*], 이창목[2]
[1]호원대학교 컴퓨터게임학부, [2]전북대학교 컴퓨터공학부

Abstract As application development environment changes rapidly, importance of user interface design is increasing. Usually, most of designers are clustering by subjective method of individual to define objects that have relativity in design interface. But, interface which is designed without particular rules just adds inefficiency and complexity of business to user who use this system. Therefore, in this paper, we propose an object oriented design model that allows for flexible development by formalizing the user interface prototype in any GUI environment. The visual cohesion of the user interface is a new set of criteria which has been studied in relation to the user interface contents, and is founded on the basis of the cohesion of the interface as defined using basic software engineering concepts. The visual cohesion includes the issue of how each unit is arranged and grouped, as well as the cohesion of the business events which appear in the programming unit. The interface will become easier to understand and use if the business events are grouped by their inter-relevance within the user interface.

요 약 애플리케이션 소프트웨어 개발 환경이 빠르게 변함에 따라 사용자 인터페이스 설계의 중요성이 증가하고 있다. 일반적으로 대부분의 설계자들은 설계 인터페이스에서 상호 의존성 있는 객체들을 정의하기 위해 개인 각자의 주관적인 방법으로 그룹화한다. 그러나 특정한 규칙이 배제된 체 설계된 인터페이스는 이러한 시스템을 사용하는 사용자들에게 업무의 비효율성과 복잡성만 증가시킬 뿐이다. 그러므로, 본 논문에서는 인터페이스 프로토타입을 정형화 함으로써 어떠한 GUI 환경에서도 유연한 개발을 할 수 있도록 객체지향 설계 모델을 제안한다. 사용자 인터페이스의 가시적 응집도는 사용자 인터페이스 내용들과 연관된 연구를 해왔던 새로운 범주영역이며, 기본적 소프트웨어 공학 개념을 사용하는 것을 정의한 것으로서 인터페이스의 응집 원리에 기반한다. 가시적 응집도는 프로그래밍 단위로 나타나는 비즈니스 이벤트의 응집도 뿐 아니라 각 단위 객체가 정렬되고 그룹화되는 방법에 대한 결과를 내포한다. 따라서 인터페이스는 비즈니스 이벤트들이 상호 연관성으로 그룹화 된다면 이해하기 쉽고 사용하기가 더욱 용이해질 것이다.

Key Words : User interface, Business event, Interface prototype, Object grouping, Desgin model

1. Introduction

The design of a User Interface(UI), which is funda mental to the convergence of the different customers' requirements, and the communication required to support the complicated interaction between human beings and computers, requires very comprehensive and varied knowledge and experience[1]. The design of such a UI requires a graphics expert, requirement analyzer, system designer, programmer, technology (description) expert,

This paper was supported by Research Fund of Howon University, 2011.
[1*]Corresponding Author : Park, In-Cheol(icpark@howon.ac.kr)
Received October 12, 2011 Revised November 16, 2011 Accepted December 13, 2011

[Table 1] Comparison of studies into the automatic generation of a user interface

	TRIDENT	JANUS	GUIPS
Domain model	- data and task analytical model	- object model	- object analysis model founded on UML
Method of design	- extraction of user interface by the requirement analysis of task and function	- extraction of user interface from object model	- Extraction of user interface from scenario according to the requirement analysis
Method of specification	- data and function requirement specification	- data structured specification	- data specification of transition object
Characteristics	- application analysis of the interaction task and the decision on the task attributes by the user interface - activity chain graph(interaction of data and function) - deals with the static features of the user interface	- Non-abstraction class transmits to the user interface - Attributes and methods that have nothing to do with the user interface are distinguished in the process - Dynamic features of the user interface are not covered	- user-interaction interface modeling from scenario - object transition graph (interaction with interface) - prototype creation interface of user interface

social behavior expert and other experts, depending on the particular application[2, 3]. However, it is difficult to realistically engage experts in multiple fields in the design of a UI.

Therefore, it is necessary to research automatic designs for a UI which can meet the professional requirements of various fields. The use of Visual Cohesion (hereafter referred to as VC) in the design of a UI helps improve its quality by providing the designer or developer with a visual prototype prior to the embodiment of the system. Moreover, VC provides the standards needed to measure the appropriateness of the layout of the UI and its semantic contents. It is necessary to improve the comprehensibility of business tasks and the usability of the interface by clustering business events in such a way that they are semantically related one another[4]. This paper aims to look at the modeling techniques used to improve the VC. The purpose of the VC in this study is to improve the comprehensibility and usability of a business system by clustering business events in such a way that they are related to one another[5, 6, 13]. Therefore, this paper proposes 4 types of objects that can improve the VC of a UI prototype and discusses the techniques used to produce an object oriented design that performs the clustering of these objects, as well as discussing the method used to measure the VC.

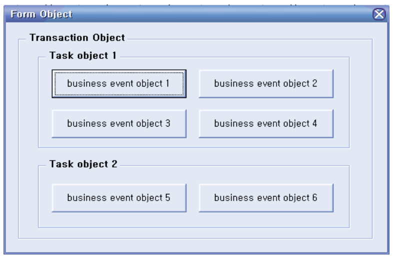
2. Related Works

GENIUS[11], JANUS[12], TRIDENT[13], GUIPS[11, 14] are some examples of studies related to the automatic creation of a user interface. Table 1 sums up the characteristics of recent studies into the automatic creation of a user interface.

3. Interface Design Model based on Classification

The object oriented model suggested in this paper is composed of 4 object models which can improve the VC of the UI, as shown in [Fig 1] : 1) business event object, 2) task object, 3) transaction object, 4) form object

model used to calculate the cohesion of the suggested object model and validate the improvement of cohesion compared to the existing design model. Therefore, the detailed objects of the UI are analyzed in terms of their similarity, relevance and transference of the business events in the UI in order to perform the clustering of the business events through the set of objects[12]. This is because the visualization of objects patterned by clustering can lead to an improvement in the VC of the business events in the UI.



[Fig. 1] Hierarchical structure of object oriented design model

3.1 Business Event Object

The design of the business event object is the stage in which the object which represents the User Interface Business Event Object (hereafter referred to as UIBEO) is designed. The control pattern of the business event which comprises the transference data of the business event is designed in the UI. In other words, the designing operation of the business event includes the design of the business event controls, such as the radio buttons, combo boxes, check buttons, etc. The rules used to design the business event objects of the UIBEO are as follows.

*<**Rule 1**> the business event which has the number of instant limited to one business event is UIBEO.*
*<**Rule 2**> if the number of instance that can be fed to one business event is not regular, it is not the UIBEO.*
*<**Rule 3**> the item that can have the instance less than 7 at the maximum is the UIBEO that can use the radio button.*
*<**Rule 4**> the business event that can have over 8 instances is the UIBEO that can use the combo box.*
*<**Rule 5**> the item that can feed the instance of choice is the UIBEO that can use the check button.*

This improves the cohesion of the business event by effectively modeling the function of the business event in the UI, and also enhances the reusability of instant data and functional cohesion within the UI.

3.2 Task Object

The clustering of task objects is the design stage in which the objects that represent the User Interface Task Objects (hereafter referred to as UITO) are created. It is a clustering stage that enables the user to distinguish the set of task objects by modeling the block label composed of task units if there are more than 2 input or output business events which are transferred when the event occurs. The following is the rule related to the clustering of the set of task objects of the business event that is transferred to a task.

*<**Rule 1**> the business event which has the number of instant limited to one business event is UIBEO.*
*<**Rule 2**> the output UITO which has more than one record is the StringGrid bloc.*
*<**Rule 3**> the node which has more than one continual input business event is UITO.*
*<**Rule 4**> the node which has more than one continuous output business event is UITO.*

The business events are clustered and labeled by set of task objects in order to improve the communication cohesion and VC of the business event in the UI.

3.3 Transaction Object

The design of the transaction object is the stage in which the object that represents the User Interface tRransaction Object (hereafter referred to as UIRO) is created. In other words, the UIRO is the clustering stage in which the group of business events composed of input-control-output events is grouped into the set of transaction objects. The transaction object is created by turning the request (input) and response (output) of the user into the block through one suite. The following is the rule used for clustering the set of transaction objects.

*<**Rule 1**> it is composed necessary of input task-button-output task, and the input task can be omitted if overlapping with the previous transaction.*
*<**Rule 2**> it can have more than one input task and output task.*
*<**Rule 3**> the input task is the beginning of UIRO, while the output task is the end of UIRO.*

The design of the transaction object, which is the stage in which the users are provided with the set of transaction objects, is the method of clustering the transaction objects that are grouped into 'task-control-output' tasks. This facilitates the understanding of the users by visualizing the transaction objects of the business events in the UI.

In other words, it makes it easier for the users to understand the set of transactions in the Interface, by clustering the 'input task-control-output' tasks into one object unit for the sake of visualization.

3.4 Form Object

The design stage of the form object serves to create the object that represents the User Interface Form Object (hereafter referred to as UIFO). This stage creates the form object by dividing the business events into the form in which they are presented in the UI. If the number of input/output business events exceeds 20 (criteria for human engineering) or the output form(or state) is selected in more than one input and it is necessary to make the user clearly understand as in the case of Interrupt, it divides the objects into multiple forms. The following is the rule used for clustering the set of form objects.

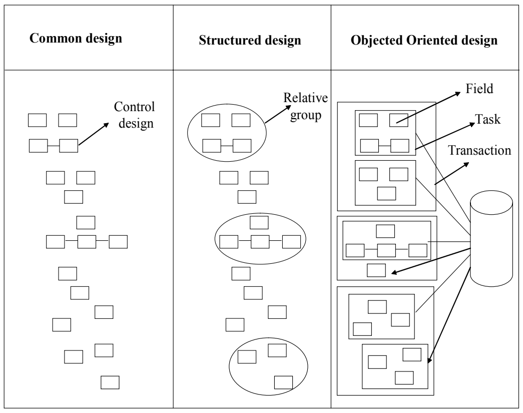
*<**Rule 1**> the input/output objects exceed 20, and in case of different task, are divided into other form object.*
*<**Rule 2**> if the response to the demand is alternative, it is divided into different form object.*
*<**Rule 3**> if the result of event is Interrupt, it is divided into new form object.*
*<**Rule 4**> if it is the abstract object with same task though it exceeds 20 items, it cannot be divided into other form object.*
*<**Rule 5**> one task object can be divided into form object, and the transaction object gathers to become form object.*

The efficient design of the form provides the support needed to facilitate the development of the program and its maintenance/repair, by making it easy to understand the business process and reducing the complexity of the software.

4. Evaluation of Proposed Model

4.1 Features of Proposed Model

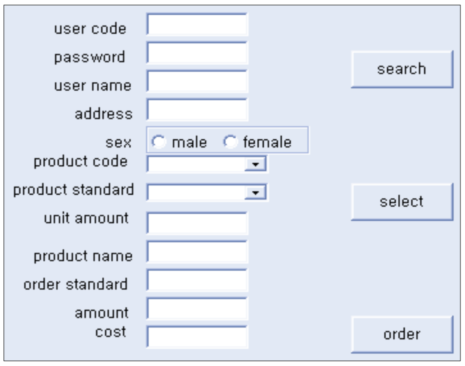
In this section, the existing common design, structured design and object oriented design proposed in this paper will be explained, in order to compare these different design models of the UI. Fig. 2 shows the structure of these evaluation models in order to facilitate the understanding of the design models.



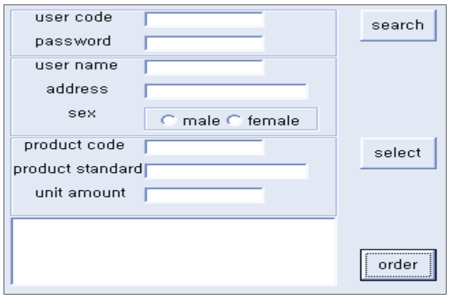
[Fig. 2] Comparison of the structure of the referenced design model

Business events in the general design model are not subject to clustering, and the designer arranges the business events (by him or herself). The designer determines only the control of the business events, depending on his or her skill. The structured design model of Constantine designs the control pattern of the business events like general designers, and the designer groups the business events according to the irrelevance, so that the user can understand the irrelevance to the work involved.

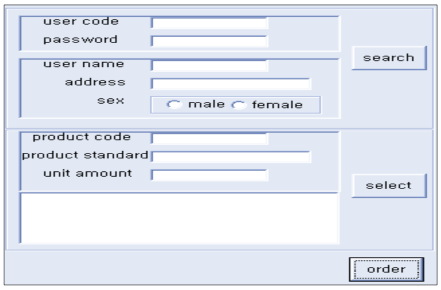
The object-oriented design model proposed in this paper applies the concept of the object-oriented design by classifying the business events into objects (business events, tasks, transactions, forms) in the UI.



(a) Generalized design model



(b) Structured design model



(c) Object-oriented design model

[Fig. 3] An example of design model for improving VC

Fig. 3(a) shows the general design model which designs only the control pattern of the business events. In other words, it designs only the control patterns, such as the properties, product codes, product standards, etc. The structured design model in Fig. 3(b) shows an example in which the control of the business events is designed and their grouping is performed based on the user information, product information and order information according to the relevance of the work involved. Fig. 3(c) is an example of the object-oriented design model visualized by the multi-dimensional grouping of 4 objects to which the object oriented design method is applied (business events, tasks, transactions, forms). The object-oriented design model is a method that enables the user to recognize the object group and sequence of tasks which are transferred to the database in the UI, as shown in Fig. 2.

4.2 Criteria for VC

The VC of the UI is a new set of criteria which has been studied in relation to the UI contents, and is founded on the basis of the cohesion of the interface as defined using basic software engineering concepts. These criteria represent the VC of the UI based on the extension of the already well-established software engineering concepts, used to assess the complexity of the UI, to its coherence. These criteria are based on the principle that semantically related elements in a large group of units can be combined, thereby promoting the understanding of each unit, reducing their inter-reliability and simplifying the overall structure[13]. The VC includes the issue of how each unit is arranged and grouped, as well as the cohesion of the business events which appear in the programming unit. The interface will become easier to understand and use if the business events are grouped by their inter-relevance within the UI.

Constantine proposed an equation to estimate the VC[13]. This VC is represented by the ratio of the number of pairs related to visual business events to the number of business events. The summation of the VC in the form and dialogue box is the summation of the VC in the group of all levels.

$$VC = 100 \times \left(\frac{\sum_{\forall l} G_l}{\sum_{\forall l} N_l (N_l - 1)/2} \right)$$

however,

$$G_l = \sum_{\forall i,j | i \neq j} R_{i,j}$$

[Formula 1] Equation of calculating VC

N_l represents the number of business events in group l, $R_{i,j}$, represents the semantic relevance(however, $0 \le R_{i,j} \le 1$) between the business events i and j in each group. If business events i and j are relevant, $R_{i,j}$. If no relevance exists, $R_{i,j}$=0. The number of VCs increases if the grouping among relevant business events is good. The equation used to calculate the VC is applied on the basis of the outcome of the design in section 4.1, and the VCs of the designed models are compared and evaluated. The referenced design model used for the evaluation is designed in the form of a visual prototype of a non-functional screen layout, and the basic visual properties and business events were designed with the same number (of visual properties and business events) in order to ensure the objective measurement of the result of this experiment. Table 2 shows the outcome of the calculation of the relevance by pattern of this referenced design.

[Table 2] Outcome of the calculation of relevance by referenced design model

Generalized design	$G_1=1,G_2=10,G_3=10,G_4=17$
	$N_1=2,N_2=5,N_3=5,N_4=16$
Structured design	$G_1=1,G_2=10,G_3=10,G_4=10,G_5=3,G_6=10,G_7=0$
	$N_1=2,N_2=5,N_3=5,N_4=5,N_5=3,N_6=5,N_7=6$
Object oriented design	$G_1=1,G_2=10,G_3=10,G_4=1,G_5=3,G_6=3,G_7=10,G_8=2,G_9=2,G_9=0$
	$N_1=2,N_2=5,N_3=5,N_4=2,N_5=3,N_6=3,N_7=5,N_8=3,N_9=3,N_{10}=3$

The VC of the general design model was $N=4$, $N_1=2$, $N_2=5$, $N_3=5$, $N_4=16$, VC=27 was the lowest VC. The VC of the structured design model was N=7, *and* $N_1=2$, $N_2=5$, $N_3=5$, $N_4=5$, $N_5=3$, $N_6=5$, $N_7=6$, VC=74 was the intermediate level of VC. The VC of the object oriented design model was N=10 *and* $N_1=2$, $N_2=5$, $N_3=5$, $N_4=2$, $N_5=3$, $N_6=3$, $N_7=5$, $N_8=3$, $N_9=3$, $N_{10}=3$, VC=89 was the highest VC. Therefore, the VC of the referenced design model can be compared as shown in Table 3.

[Table 3] Evaluation of VC of referenced design model

Name of referenced design model / VC	General design	Structured design	Object design
Calculated value	27	75	89

It was found that the cohesion of the object oriented design model proposed in this paper was much improved 14 points more than the Structured design(see the different value between Structured design and Object design of Table 3). The VC provides the criteria for reviewing the quality of the visual prototype and graphic design for the UI. Moreover, the VC provides the criteria for forecasting the user preference, the evaluation of the easiness and comprehensibility, the degree of response, and the quality of the graphic layout[13]. The automatic graphic layout of the objects ensures the most efficient grouping and the highest cohesion according to the modeling rule, regardless of the skill of the designer.

5. Conclusion

This paper studied the design rules and modeling technique of a UI that supports the user based on the improved VC. The findings of this study are as follows: First, the proposed method improves the VC by designing the objects of the UI on the basis of objects which are functional, consecutive and communicative. Second, it improves the user preference, easiness, comprehensibility, degree of response, and quality of the graphic layout on the basis of the improvement of the object based VC. Third, it improves the communicative, consecutive, and procedural cohesion of business events on the basis of the clustering of the UI objects. Fourth, it constitutes an object oriented designing method that can improve the comprehensibility of the business process and the usability of the UI on the basis of the visualization of the object pattern.

References

[1] Garcia, E., Sicilia, M.A., Gonzalez, L., Hilera, J.R., "Dialogue-Based Design of Web Usability Question naires Using Ontologies", Computer-Aided Design of User Interfaces, pp. 131-144, 2005.
[2] http://www.useit.com/alertbox/
[3] Dix A., "Design of User Interface for Web", Proceedings, User Interface to Data Intensive System, pp. 2-11, 1999.
[4] Constantine L.L., Biddle R., and Noble J., "Usage-centered Design Engineering: Models for Integration", IFIP international conference on software engineering, pp. 106-113, 2003.
[5] http://www.usernomics.com/user-interface-design.html
[6] http://www.bainbrdg.demon.co.uk/index.html
[7] Leszek A. Maciazek, "Requirements Analysis and System Design", Addison Wesley, pp. 244-270, 2001.
[8] H. Balzert, "From OOA to GUIs: The Janus System", IEEE Software, Vol. 8, No 9, pp. 43-47, February 1996.
[9] F. Bodart, A.-M. Hennebert, J.-M. Leheureux, I. Provot, and J. Vanderdonckt, "A Model-based Approach to Presentation: A Continuum from Task Analysis to Prototype", Proceedings of the Eurographics Workshop on Design, Specification, Verification of Interactive Systems, Carrara, Italy, Focus on Computer Graphics, Springer-Verlag, Berlin, pp.77-94, June 1994.

[10] M. Elkoutbi, I.Khriss, and R.K.Keller, "Generating User Interface Prototypes from Scenarios", Proc. of the 4th IEEE International Symposium on Requirements Engineering, pp. 150-158, 1999.
[11] Nerurkar U., "Web User Interface Design, Forgotten Lessons", IEEE Software, Vol. 18, No. 6, pp. 69-71, Nov.-Dec. 2002.
[12] Chidamber S. and Kemerer C., "A Metrics Suite for Objected-Oriented Design", IEEE Transaction on Software Engineering, Vol. 20, No. 6, pp. 476-493, 1994.
[13] Constantine, L. L. "Visual Coherence and Usability: A Cohesion Metric for Assessing the Quality of Dialogue and Screen Designs", Proceedings, Sixth Australian Conference on Computer-Human Interaction, IEEE Computer Society Press, 1996.
[14] http://www.useit.com/alertbox/
[15] W. K. Park, "Design and implementation of SOA based S/W services for dynamic behavior of embedded System", Journal of The Institute of Webcasting, Internet and Telecommunication, Vol 10, No 4, pp. 29~34, 2010.

In-Cheol Park [Regular Member]

- Feb. 1986 : Chonbuk National Univ., Computer and Statistic, MS
- Aug. 1998 : Chonbuk National Univ., Computer and Statistic, PhD
- Mar. 1990 ~ current : Howon Univ., Dept. of Computer·Game, Professor

<Research Interests>
Korean Information Processing, Information Retrival, Embedded System, Sematic Web

Chang-Mog Lee [Regular Member]

- Feb 2001 : Chonbuk National Univ., Computer and Statistic Information, MS
- Aug. 2005 : Chonbuk National Univ., Computer and Statistic Information, PhD
- Mar. 2009 ~ Feb 2010 : Chonbuk National Univ., Researcher(as a Post-Doctor) of BK21
- Mar. 2010 ~ current : Chonbuk National Univ., Dept. of Computer and Statistic Information, Instructor

<Research Interests>
CBSD, AOSD, Reverse Engineering Methodology, Information Security