User Interface Design Model for Improving Visual Cohesion

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

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.

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

1. Introduction

The design of a User Interface(UI), which is fundamental 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,
Table 1] Comparison of studies into the automatic generation of a user interface

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

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.
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 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](image-url) 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.
User Interface Design Model for Improving Visual Cohesion

(b) Structured design model

(c) Object-oriented design model

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

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.

\[
VC = 100 \times \frac{\sum G_i}{\sum N_l(N_l - 1)/2}
\]

\[G_i = \sum_{\forall j, j \neq i} R_{ij}\]

[Formula 1] Equation of calculating VC

\(N_l\) represents the number of business events in group \(l\), \(R_{ij}\), represents the semantic relevance (however, \(0 \leq R_{ij} \leq 1\)) between the business events \(i\) and \(j\) in each group. If business events \(i\) and \(j\) are relevant, \(R_{ij}\). If no relevance exists, \(R_{ij}=0\). 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.
**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**


---

**User Interface Design Model for Improving Visual Cohesion**

---

**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

---

**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, Semantic Web