A Collection of Interaction Patterns for Design the Visual Feedback of Interactive Applications

JAIME MUÑOZ ARTEAGA and GUSTAVO RODRÍGUEZ GOMEZ.
Coordinacion de Ciencias Computacionales (CCC)
Instituto Nacional de Astrofísica Optica y Electrónica (INAOE)
Calle Luis Enrique Erro No. 1, Sta. María Tonantzintla
72840 Puebla, México

Abstract: - The visual feedback is an important usability factor, it is the visual response given by an interactive application in order to assist on a graphical interface the user task. In general, the visual feedback is defined at implementation stage where the user’s requirements are difficult to take into account. Although the visual feedback is predominant in current interactive application, it lacks of design techniques to specify the visual responses based on the user’s requirements. This work proposes a categorization of interaction patterns aimed at supporting the design of visual feedback in function of usability factors. The goal is to offer a high level description of visual responses of an interactive application independent of any graphical environment.

Key-Words: - Design and interaction patterns, visual feedback, usability factors and interaction styles

1 Introduction
In an interactive application, the term visual feedback (or visual response) applies to any graphical form of communication directed from the application towards the user. The visual feedback helps to user to guide and to make easy his/her task. For example, this feedback could assists to user to answer the following questions: Where am I? Where can I go? What can I do? What is this object? [13]. In effect, the goal of visual feedback is twofold: the visualization of internal state of system, and the visualization of user actions [8]. For this, the visual feedback is presented in the user interface under different forms (e.g. windows, icons, maps and error messages) in order to establishing the most natural visual communication with the user. This kind of feedback must be informative, easy to understand and carry out in a properly period of time.

For design the visual feedback, it is necessary to take into account the utility and the usability factors of the underlying interactive application. The utility factor concerns to the right functionality of software application. The usability factor concerns to the facilities that offer an application to user to make easy his/her task. Although the visual feedback plays an important role in the human-computer dialog, the designer doesn’t have any mean to specify and to evaluate the visual information required for guide the user actions. A lot of work coming from the literature of software engineering and the Human Computer Interaction (HCI) has been devoted to study the different aspects of presentation of an interactive application, but the visual feedback rarely. For example in HCI, a great number of ergonomic rules have been proposed to use the visual feedback, but it lacks of context to apply them. In the other hand, the software engineering have proposed several visual programming techniques and CASE tools to get the software components of visual feedback, but this kind of tools software is strongly independently of programming language and the underlying graphical tool.

Indeed, the visual feedback is predominant in current interactive application and it lacks of design techniques to specify the visual responses based on the user’s requirements. In order to mitigate this problem, this works proposes to use the interactions patterns to specify in function of usability factors the successful experiences in the design of visual feedback. The goal is to offer a high level specification of visual responses of an interactive application independent of any graphical environment. Section 2 presents the problematic to design the visual feedback. Section 3 proposes as solution a collection of interaction patterns, most of them are described in detail. Finally, the Section 4 compares the presents work with other collections of interaction patterns.

2 Aspects to design the visual feedback
When the user uses an interactive application, the application could answer to user with multiple visual responses even if the user action is simple, this is described through the Seeheim architectural model.

The Seeheim model reveals the linguistic nature of the visual feedback identifying 3 main software modules:
1. The dialogue control module handles the syntactic aspects of the interaction and is responsible for the dynamic of the system.
2. The application interface module provides a semantic interpretation of the information received for the dialogue component.
3. The presentation module handles the lexical aspects of the interaction such in input as well in output.

The Seeheim model establishes that the multiple forms of visual feedback could come throughout all the software components of an interactive application (see fig.1). In effect, it is quite difficult to design the visual feedback without being immediately stuck into implementation details that heavily depend on the underlying graphic toolkit.

The software engineering has proposed design patterns to solve the abstraction of code for the visual feedback. By definition, a pattern is a solution to a recurrent problem within a specific context [1]. A pattern allows the communication of experiences and the knowledge in a specific area. A great number of patterns have been published since 1990. Nowadays there are a large diversity of new patterns for examples the scientific patterns [4], the architectural patterns and design patterns [7]. The design patterns are classified in a catalogue and they promote the reusability of software capturing the experience of successful designs of systems based on the object-oriented paradigm. In an interactive application is possible to identify a great number of design patterns corresponding to multiple visual feedback forms. Unfortunately, design patterns are not useful to design visual feedback, they do not take into account the specification of user interface components and the user requirements [5,8].

The works in the HCI literature have been proposed a great number of guidelines and ergonomic rules. Nevertheless, the large number, the simplicity and the generality of these rules ergonomics make difficult their use and their interpretation. Even worse, an ergonomic rule lacks of context that can determine what type of problem could resolve. For example the ergonomic rule: "All error message displayed to user must be coherent", it is not easy to identify what types of problem could solve this rules and event worst what it is the context where an error message is coherent?

An approached to design the visual feedback of interactive application that it takes into account the HCI and software engineering aspects is the interaction patterns. The interaction patterns are a recently specification techniques for design the external aspects of interactive applications. More specifically, the interaction patterns are successful solution given to recurrent problems about the design of user interfaces within a specific context [2,3]. The visual feedback is an important aspect of an interactive application; we propose the interaction patterns in order to describe the visual response of an interactive application based on the user requirements. Our proposal is close to Christopher Alexander’s idea about patterns concepts[1]: the visual feedback must be designed according to the needs, the satisfaction, and the acceptability of final users.

3 A Pattern Language for Design the Visual Feedback

The use of the interaction patterns can help to design the external aspect of an interactive application but it helps too to facilitate the communication with the people involved in the design and the development of the visual responses of an interactive system. We propose the next format for an interaction pattern.

| Name | The title of the pattern must be clear and concise about the concept to communicate |
| Problem | A description about the problem to solve in terms of user point of view |
| Usability Principle | The ergonomic criteria taking into account by the current interaction pattern. |
| Context | Description of situation to applied the patterns in functions of user needs. |
| Force | Specify the scope of solution |
| Solution | Clear description of the proposed solution. Other patterns could be part of the solution |
| Consequences | The effects if the current pattern is applied |
| Example | An illustrative sample of a successful solution.
In the previous table, the usability principle of an interaction pattern is based on the ergonomic criteria. The ergonomic criteria proposed by Dominique Scapin et al. [10] are used here because they fully conform the good properties for the visual feedback [13]. These ergonomic criteria are eight: the explicit control, the guidance, the compatibility, the work load, the adaptability, the consistency, the significance of the code and the error management.

According to Seeheim model (see fig.1) an interactive application could answer to user with multiple visual responses even if the user action is simple. In order to identify the different types of answer of interactive application, we propose three categories of interaction patterns for design the visual feedback based on the linguist nature of human-computer dialog:

1. **The semantic level** regroups the interactions patterns that give solution to the problems of visual feedback related to the visualization of system state.

2. **The syntactic level** regroups the interactions patterns that give solution to the problems related to visualization of system behaviour.

3. **The lexical level** is composed by the interactions patterns that display the information of user actions.

These three categories are put together to become a collection of interaction patterns. This collection is represented in the next figure.

In the figure 2 every node of the collection represents an interaction pattern. These nodes are ranged in a hierarchical order, the interaction patterns near of the root describe high level abstractions of visual feedback and the interaction patterns near of leafs describe detailed abstractions. This collection cover the large diversity of visual feedback than an interactive application could provide to support the user task. The collection of interaction pattern proposed here doesn’t try to be exhaustive; in fact the objective is to propose a base for the HCI designers who intend to develop new interaction patterns, particularly for the design of other types of feedback such sound, force and animation.

Due to the lack of space, this section presents only a part of the collection of interaction patterns showed in the fig. 2. The complete description of the collection is located in the web site http://cee.inaoep.mx/~jaime

---

### Table: Interaction Patterns for System State

<table>
<thead>
<tr>
<th>Name</th>
<th>System state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem</td>
<td>The user has difficulties to interpret and access the information of system state.</td>
</tr>
<tr>
<td>Usability principle</td>
<td>Information density and work load.</td>
</tr>
<tr>
<td>Context</td>
<td>When there is a large quantity of information of system state to be displayed in structured way. The information to be displayed should be selected in function of user task and also in function of the screen space.</td>
</tr>
</tbody>
</table>
| Force | • An interactive application must notify to user every changes of its state.  
• The visualization of system state can be personalized taking into account its structure, user task and user preferences. |
| Solution | Offer to user several options to access the information of the system state. |
| Consequences | In function of the structure of information, the system state could have multiple representations such as: **hierarchical, graphs, spatial and temporal**. |
| The user can evaluate the system state and afterward define his/her future action. |

### Table: Hierarchical Representation

<table>
<thead>
<tr>
<th>Name</th>
<th>Hierarchical representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem</td>
<td>How to show the information of system stated structured as a tree?</td>
</tr>
<tr>
<td>Usability principle</td>
<td>Work load and information density.</td>
</tr>
<tr>
<td>Context</td>
<td>The system state is composed of a large amount of information structured hierarchically.</td>
</tr>
<tr>
<td>Force</td>
<td>• The user can easily access the information moving through in a hierarchical structure.</td>
</tr>
</tbody>
</table>
A hierarchical representation motivates the user to discover new information.

**Solution**
Show the information in a hierarchical form where the entry point is the root node and the terminal node represents the elementary information.

**Consequences**
The hierarchical representation helps the user to have a right mental image of the system state.

**Example**
A graphical file system manager shows the distribution of documents in folders which can be either composed (those with the sign +) or simple (those with the sign -).

**Temporal feedback**

**Problem**
The user needs to monitor periodically the changes of system state.

**Usability principle**
Minimal actions and information density

**Context**
The user needs to measure in function of time the efficiency of his/her task.

**Force**
- Temporal feedback covers the visualization of information of system state which change in a period of time.

**Solution**
Display periodically the value of system in terms of units of time such hours, minutes and seconds.

**Consequences**
The user could estimate the next command offered by the system state.

**Example**
The visualization of system time of a computer is displayed in a textual and graphical form.

**Warning message**

**Problem**
The user executes irreversible or undesirable operations.

**Usability principle**
Protection to errors

**Context**
Alert the user when the user's information could be lost or when the user actions become ambiguous.

**Force**
- The application offers security to user task
- Alert the user about the irreversible operation that could be executed by the interactive application. Difficult

**Solution**
Alert the user, inform the nature of the problem, and offer to the user the opportunity to cancel the operation that cause the problem.
Consequences

Example

The warning message allow the system offer to user a secure environment

This warning messages alert the user that the required web site is not found.

Name | Notification
--- | ---
Problem | The user unknown the characteristics of services executed by the system.
Usability principle | User control
Context | In general, the notification answers the questions of what’s happening with the execution of system’s services, especially those are long and unfamiliar to user.
Force | • Give information about the accomplishment time of a service.
• The animation visual help to user’s comprehension.
• Give to user the control to go on or to leave the execution of a service at any state of execution.
Solution | Provide a notification giving the control to user over the execution of an operation. The notifications could be at the beginning, during and at the end of an operation.
A notification must be given within a time to support the current user task.
Consequences | Offer to user information about the delay time of a particularly long process to avoid that user becomes exasperated and leaves her/his task.

Name | Evolving notification
--- | ---
Problem | The user needs to have the control about the execution of the long operations.
Usability principle | User control
Context | This type feedback is applied when the time of this execution of a operation is more than 10 seconds.
Force | • Inform the user about the estimation and the accomplishment of a service
• Notify to user continuously about the status of execution for long operation.
Solution | Give the control to user to cancel long operations. In addition notify the user about the progression and the reaming time to achieve the operation.
Consequences

Example

In this example the user could stop at any moment the downloading operation.

Name | Activation
--- | ---
Problem | The user does not know why the services are active in a certain moment of the interaction
Usability principle | Explicit control
Context | The availability (or unavailability) of services must be predictable to user
Force | • Simple syntaxes are required for the set of services offered by the system.
Solution | In all moment of interaction the system must inform to user about the set of available and unavailable services. Two types of activation: proactive and reactive.
Consequences | The user can define his future actions based on current set of (and unavailable) availability services.

Name | Proactive activation
--- | ---
Problem | The user does not know what are the available (disable) services
Usability principle | Explicit control
Context | The interaction space needs to be continuously visualized by the final user.
Force | • This feedback displays the space of interaction at any moment of user interaction.
Solution | Graying out the unavailable widgets in order to avoid the user trigger an illegal action. When an action become legal the widget is put available to user.
Consequences | This technique is quite common, but presents the drawback of providing very little information on the reasons why a particular action is currently disabled, and gives even less clues on the possible ways to make it available.

Example

In the application of figure 5, the set of services available to the user is those that are associated to button “Close”, “Refresh” and “Help”

Name | Navigation
Problem: When the users access large quantities of information of the system, they could have frequently questions such as: where am I? and Where can I go?

Usability principle: Guidance and information density

Context: Display large quantities of information in a screen with a restricted space

Force: • The user needs to access in a structured way the large amount of information.
• The navigation inside the window’s component could consider as fine granularity dialog.

Solution: The information access could be distribute throughout logic units such as windows, dialog boxes, scrolling lists etc. The form to access the information could be direct or sequential navigation.

Consequences: In general, the navigation using windows become familiar to user in order to access large quantities of information.

Name: Direct navigation

Problem: The user has difficulties to access directly the current active window

Usability principle: Guidance and concision

Context: Given a large quantity of system’s information the user needs an access direct through the windows.

Force: • The navigation always displays the list of windows available to user

Solution: Offer to user a direct access to the current active window where is located system’s information.

Consequences: A direct access of windows also facilitates the access of the interactive objects (e.g. icons, items of menu, radio buttons) of the window selected by user.

Example: The focus is a feedback mechanism used frequently to access a direct access of current active window

Name: Sequential navigation

Problem: The user don’t have any means to access the next and previous windows

Usability principle: Guidance and concision

Context: This technique is very useful for the current browser of internet where the user access a large number of web sites.

Force: • This feedback takes into account the previous windows visited by user during a work section.

Solution: The sequential navigation of windows of an interactive application can be either toward back or forth from current window.

Consequences: Help to user to have a right mental image of the navigation space

Example: Current browser of internet have a button “Back” for to go to previous page and a button “Forward” for to go to next page.

Name: Dialogue

Problem: No there isn’t any facilities that make efficiently the development of user task

Usability principle: Guidance and coherence

Context: Web site, multimedia system, data warehouse, learning system that need to access a large quantities of information

Force: • Identifying how the system can help the user task requires a user centred design.
• Make easy the use of an interactive application

Solution: Offer a set of service that could facilitate the user task such as redo/undo and assistance services

Consequences: A better comfort in the information access

Name: Non-contextual feedback

Problem: User actions are not displayed immediately by the interactive application

Usability principle: Guidance and coherence

Context: In a system with direct manipulation the user actions need to be notified under 0. 2 seconds of time.

Force: This type of visual feedback requires a specification of a very fine dialog.

Solution: Show to user the low level interaction (e. g. a click, press a certain key, etc.) using the selecting, the pointing or the tracing mechanisms.

Consequences: The visual feedback of user actions assure user’s expectations of his/her task

Name: Pointing

Problem: It lacks the visualization of user actions for the cursor of physical devices such as the mouse, joystick and the keyboard.

Usability principle: Guidance and precision

Context: The manipulation of physical devices by
needs to be informed immediately in a response time average of 0.2 seconds.

**Solution**

Give a continuous display of user actions which manipulate physical devices of an interactive application.

**Consequences**

User actions could be represented by icons in order to click and move over a target objects instead of a complex syntax.

**Example**

The little arrow of the mouse and the little prompt the keyboard are visual forms of cursor used frequently for the pointing feedback.

<table>
<thead>
<tr>
<th>Name</th>
<th>Contextual feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem</td>
<td>It lacks a mechanism to get information about the object that the user can directly manipulate throughout the physical devices</td>
</tr>
<tr>
<td>Usability principle</td>
<td>Adaptability and precision</td>
</tr>
<tr>
<td>Context</td>
<td>When the response time to user should be under 0.2 seconds. This case is required frequently in the direct manipulation user interfaces</td>
</tr>
<tr>
<td>Force</td>
<td>This type of feedback is displayed when the cursor is over the graphical representation of an interactive object.</td>
</tr>
<tr>
<td>Solution</td>
<td>Get information from the current manipulated objects using the waiting, modifying and helping visual responses.</td>
</tr>
<tr>
<td>Consequences</td>
<td>A better precision for the user actions</td>
</tr>
</tbody>
</table>

**4 Related Work**

VisualFeedback is the name of the collections of interaction patterns proposed here, it is compared with other collections of patterns aimed to design graphical user interfaces.

<table>
<thead>
<tr>
<th>Interaction patterns</th>
<th>Semantic</th>
<th>Syntactic</th>
<th>Lexical</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brigton[9]</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>Ámsterdam[12]</td>
<td>5</td>
<td>15</td>
<td>4</td>
<td>24</td>
</tr>
<tr>
<td>Experiences[15]</td>
<td>4</td>
<td>18</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>Visualization[14]</td>
<td>19</td>
<td>18</td>
<td>4</td>
<td>31</td>
</tr>
<tr>
<td>PoInter[16]</td>
<td>3</td>
<td>7</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>VisualFeedback</td>
<td>11</td>
<td>10</td>
<td>6</td>
<td>27</td>
</tr>
<tr>
<td>CyberPatten[17]</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

According with the previous table, the VisualFeedback could be regrouped in the collections that have a large number of interaction patterns as the Common Ground and the Amsterdam collection. Most of the collections are devoted to specify general aspects of the user interfaces, but the VisualFeedback collection is the only collection of interaction patterns focused in the design of visual feedback of interactive application.

**5 Conclusion**

The purpose of this paper was to propose the interaction patterns as one specification technique that supports the design of visual feedback. The collection of interaction patterns proposed here identifies different abstractions level of the visual response given to user. In addition, most of interaction patterns have been described on detail with meaningful examples.

As a summary, the collection of interaction patterns proposed in this work helps to designer of an interactive application in the next aspects:

1. To describe the visual feedback in terms of usability factors.
2. To have a high level description of visual responses of an interactive application independent of any graphical operating system or language programming.
3. To classify the interaction patterns in function of the linguistic nature of the visual feedback

One expectation of the present work is to specify the feedback of multimedia application such as the sound and the animation. Other future work is the automatic generation of code of user interfaces for this kind of application using the patterns approach.

References: