Journal of Modern Education Review, ISSN 2155-7993, USA
December 2016, Volume 6, No. 12, pp. 931–939
Doi: 10.15341/jmer(2155-7993)/12.06.2016/007
© Academic Star Publishing Company, 2016
http://www.academicstar.us

Study of a Pattern Language for App Design of User Awareness

Lee Cheng-Ta[1]*, Lee Lai-Chung*[2]

(*1. Department of Visual Communication Design, China University of Technology, Taipei, Taiwan*;

2. *Department of Interaction Design, National Taipei University of Technology, Taipei, Taiwan*)

Abstract: The objective of this study was to establish an app design method using pattern languages based on infographics and user awareness. In the use of HCI pattern languages, the intentions and motives of the users must be taken into consideration. In app design, various elements such as text, the configuration and design of colors, and even figure size and the most important infographics can all be expressed using pattern languages. When the user's consciousness transforms from data to visual images, today's infographic design can clearly express visual design characteristics. In other words, we hope that with the influence of the design pattern model of infographic figure and text messages on the subjective preferences of users, we can use pattern languages to establish design patterns that are not affected by subjectivity. The results will also provide future infographic design patterns.

Key words: pattern language, user awareness, infographics, design pattern

1. Introduction

Recent years have seen a significant increase in the popularization and utilization of smart mobile devices, and in human-computer interaction (HCI), patterns are being widely used to solve issues in HCI design and interfaces. Most HCI patterns adopt the end user perspective, which makes HCI pattern languages closer to the pattern languages proposed by Alexander (1977). However, interactive systems often lack a common language for mutual exchanges due to a lack of thinking, perspective, and values on the designer's part (Borchers, 2001). For the sake of cooperation, it is therefore crucial to find a common language, i.e., a pattern language.

In the use of HCI pattern languages, the intentions and motives of the users must be taken into consideration. In app design, various elements such as text size, graphic colors, and infographic design, which is the most important in message identification, can all be expressed using pattern languages. Pattern languages have been applied to software engineering, interactive design, game design, business management, and knowledge management and have served as a possible way to share tacit knowledge (Takashi, 2007). Thus, all patterns are still in ongoing development, which means that we must continue researching ways to improve pattern accuracy so as to enhance the designs of apps and mobile interfaces.

Lee Cheng-Ta, Lecturer, Department of Visual Communication Design, China University of Technology; Graduate Institute of Design, National Taipei University of Technology; research areas/interests: visual communication design, infographics, pattern language. E-mail: adahouse@gmail.com.

Lee Lai-Chung, Professor, Department of Interaction Design, National Taipei University of Technology. E-mail: f10666@ntut.edu.tw.

2. Literature Review

2.1 Origin and Meaning of Pattern Languages

The term pattern language was used by Christopher Alexander around the 1970s in theoretical research and practice regarding architecture. In his 1977 book *A Pattern Language*, Alexander stated that patterns can be used to solve recurring problems in certain areas of construction. Each pattern describes a problem that occurs repeatedly in our environment, followed by the core solution to the problem. It thus enables us to use the solution repeatedly, and each time is different. Alexander hoped that the existence of architectural pattern languages is meaningful not only to the architects but also to all users, which means that any individual can use pattern languages to create an energetic and unique home. Alexander once described his architectural pattern language as a set of various architectural patterns, each of which describes a recurring problem in the environment. Figure 1 shows the relationships among three pattern elements: (1) context, (2) problem, and (3) solution. Put simply, rethinking the problems in a certain context produces the solution to the spatial planning in the problem (Hughes, 2006).

Patterns and pattern languages are correlated but never equal to each other. Alexander (1979) explained the relationship between the two in detail. In short, a pattern is the solution to a recurring problem in an environment with certain conditions; in contrast, a pattern language is a set of patterns that are correlated with one another (Fig. 2).

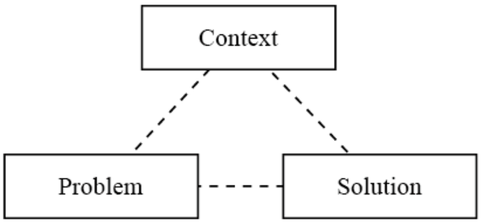


Figure 1 Basic Relationships in Pattern Languages

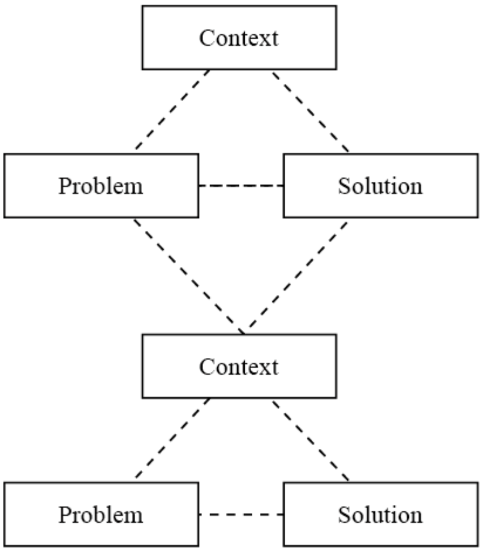


Figure 2 Repeated Occurrence and Mutual Strengthening

2.2 Mutual Relationship between Pattern Languages and User Awareness

In recent years, continuous progress has been made in HCI and context-aware technologies. With the latter, computers can provide users with automatic feedback on the environment and human behaviors (Lu, 2010). This is of particular significance because conservative statistics today show that nearly a thousand apps are added to Apple Store iTunes every day. If an app design focuses on integrating HCI and user context awareness and is people-centered, using pattern languages can establish a design model for the emotional awareness and intuition in everyday life.

Today, an increasing number of smartphone manufacturers are investing a considerable amount of funds to research user interfaces (UIs). However, whether UIs influence consumer attitudes is still not clear. Further research is required to know how smartphone users feel about interface design aspects including appearance, operational freedom, and whether it is easy to learn how to use it. Pattern language design models and app designs should also be adopted, and the three UI constructs analyzed to see whether they influence perceived ease of use, perception, and user attitude (Chi et al., 2011). It is hoped that the results of this study can provide reference to smartphone manufacturers on whether investments in UIs should continue.

Alexander provided a pattern language initiative for the future reference of smart mobile device developers and designs a set of design patterns for apps; it is also available for relevant personnel to solve similar problems (Ng & Wadhwa, 2014). One point that Alexander emphasizes with pattern languages is that it is not how one designs one’s own space but to provide some problems and solutions. The most suitable design method can be ultimately found by personally feeling one’s space and environment, presenting problems based on one's feelings, and then finding the solution among architectural pattern languages. Architecture varies with the location, so if an individual wants to fit in the environment, he or she cannot adopt only a single architectural pattern.

3. App Design Pattern Languages of User Awareness

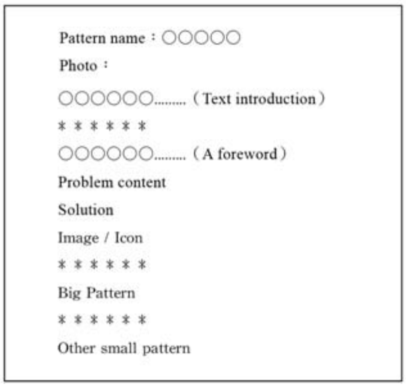
When progress was made from graphic design to interactive design and creative concepts were used, objects produced the principles of development; pattern modules and reused components influenced various ideas and developments, which naturally generated design patterns (Hoober & Berkman, 2011). In *A Pattern Language*, Alexander once referred to architectural pattern languages as “space seeds”, explaining that we cannot design flowers, only plant seeds, and that pattern languages are seeds of space. The book *A Pattern Language* itself is guide and operating manual. A total of 253 pattern languages, from regional planning to the designs of windowsills and entrances, provide questions and scientific answers regarding space. Each pattern is an assumption, and assuming that there are several solutions, people can re-grasp the languages of their living environments by understanding construction systems and design patterns and then conceive and develop their own languages. From the individual to the group, unique features will begin to develop once there is a common language. Patterns are the constituent elements of a language; they can be used to construct a dialogue or establish an organization to form a space. Although architectural construction and interactive designs have nothing in common, the same concepts can be followed. With regard to *A Pattern Language*, Hsieh (1995) stated that a pattern is a planning principle. One by one, it describes all of the commonly recurring problems, the circumstances before and after the occurrence of the problem, and the solutions. Patterns are not the only solutions. They are a type of assembly language. Such patterns can be developed into thousands of man-made environments with numerous variations in detail. Of course, this may not be the best form. So, one can only repeat the problem and revise the solution to find the best pattern language.

3.1 Influence on App User Awareness

The usability of a smart mobile device has the following characteristics: short residence time, quick response, and high portability. This means that apps must be designed to provide effective information quickly, which means the main purpose and features must be grasped quickly. For instance, a restaurant app should be designed to focus on reservations; a tool app should be convenient and useful, and the app of a logistics company should focus on shipping inquiries. User mentalities should be analyzed to identify the most important function. Examples include what design users prefer and what the factors of the influence of graphic patters on user awareness are (Chang, 2012). Research and development involving user awareness has shown that app design attributes, color configurations, interface designs, and app ion designs all influence user willingness to select, open, or download an app. App developers must clearly understand the individual differences among various types of users, including age, gender, learning habits, experience, educational background, culture, and the environment that they grew up in. Only then can they categorize users properly, understand their lives and work, try to reduce their cognitive differences, and produce an app design pattern language.

3.2 Generation of Pattern Languages

Whatever format is used to write a pattern language, the format should include the following four elements aside from its name (Vlissides, Coplien & Kerth, 1996): (1) the context, which describes the field, scope, or circumstances that the pattern is suitable for, (2) the problem, which explains the recurring problem in the environment in question, (3) the forces, which list the causes of the problem, and (4) the solution, which provides a concrete solution to the recurring problem and counters all of the forces. Each pattern needs at least one picture to represent it. When Alexander discussed the integrity of a pattern language, he mentioned two factors: morphology and function. A pattern language must have integral morphology and function in order to be complete. Thus, the forces and solution in a pattern represent its function (Chen & Cheng, 2003). However, patterns have a uniform format including a photo, a text introduction, a foreword, the problem content, and the solution, in said order. Patterns also have a sequential order starting from the primary and followed by the secondary pattern or detailed small patterns and their mutual associations. Figure 3 displays an example. In short, a pattern language is a set of solutions to these problems. Pattern languages must describe the scope and necessary details of each pattern and provide guidance enabling users to use the system by following the order of the patterns (Yuan, 2008).



Pattern name：○○○○○

Photo：

○○○○○○………（Text introduction）

＊＊＊＊＊＊

○○○○○○………（A foreword）

Problem content

Solution

Image / Icon

＊＊＊＊＊＊

Big Pattern

＊＊＊＊＊＊

Other small pattern

Figure 3 Alexander Architectural Pattern Language

Source: Alexander C., Ishikawa S. & Silverstein M. (1977), *A Pattern Language: Towns, Buildings, Construction*

In general, patterns should have their positive side, which means that the best pattern language should serve as the representative. In contrast, there is also the negative side, which comprises the so-called poorer patterns. Thus, the primary direction of research on design pattern languages for apps should be based on user awareness. Of course, when we study user awareness, we should also record and observe the variations produced by interactions involving individuals, the environment, everyday life, and culture. However, before this, we should clarify what patterns can be studied by users. Based on the method in which the several pattern languages below were written and generated, a pattern language can be created once the name, problem, and solution are confirmed (Figure 4). Nevertheless, this approach may be fruitless because a certain solution cannot be found Figure 5. At this point, the name pattern must be dismantled repeatedly. If a specific solution does not appear after problem generation, the pattern is split into two or more patterns rather than remain a pattern with variation. Other name patterns are then referred to, and the best pattern selected once a certain solution has been confirmed. Subsequently, all of the patterns that support this pattern are identified Figure 6.

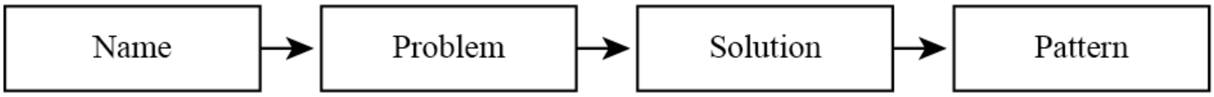


Figure 4   Generation of Pattern Languages

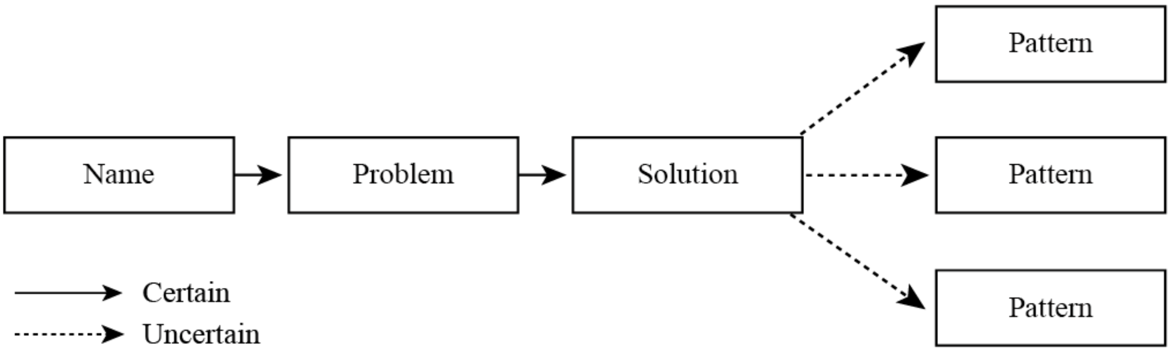


Figure 5   Uncertain Generation of Pattern Languages

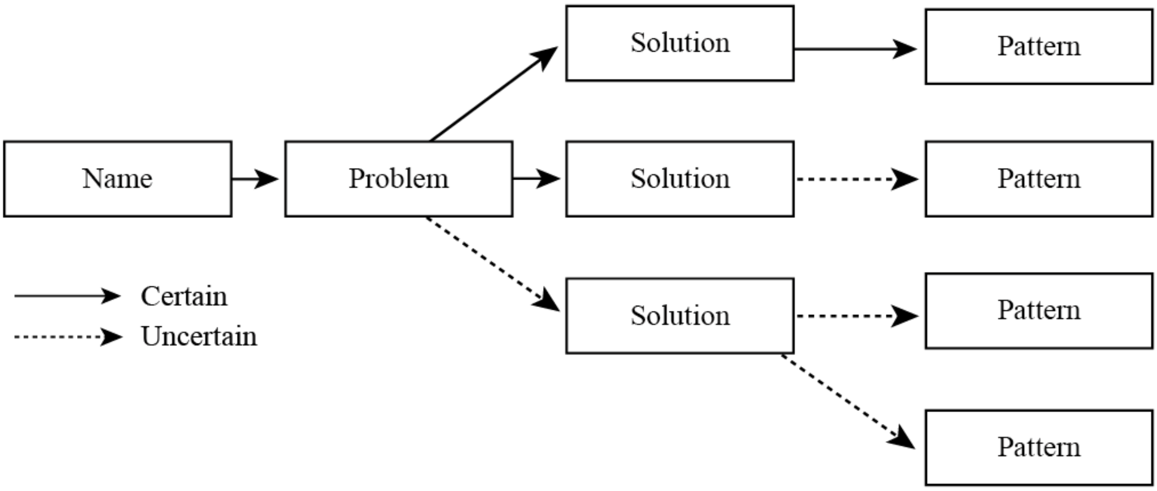


Figure 6   Generation of Best Pattern Language

App design problems may not be easy to understand. Basically, solutions still needed to be developed based on agile user experience (UX). However, agile principles were produced from the perspective of developers, which focuses on high efficiency and high quality. In developer circles, the most common misconception is that usability is a subjective feeling. Nevertheless, we often overlook user-centered design (UCD), which helps developers and designers create apps that fulfill user needs. Incorporating UCD and UX into the development process of pattern languages with the foundation of usability and HCI and putting the user in the development process will naturally eliminate a lot of the uncertainties in design and prevent perceptual errors that will create difficulties for app users in usage and understanding. UCDs must demonstrate that they are effective rather than a whim or personal preference, and this is completely the opposite of subjectivity. With good UCD, users will naturally generate positive feelings toward the app design (Lowdermilk, 2013). Finally, usability, UCD, and UX must be integrated to promote user awareness regarding the significance of app design patterns.

3.3 App Design Pattern Languages

The names of app design patterns roughly include composition, information presentation, forms and tables, text input, image display, color, and audio presentation. After obtaining the problem, solution, the aforementioned usability, UCD, and UX, as well as the context and other forces, the relationship graph of app design pattern languages can be produced Figure 7. This will also facilitate understanding on the importance of user awareness to app design patterns, from which information and corresponding patterns can be organized and categorized. Name patterns are selected from the categories to serve as the foundation of app design pattern language construction, and then the pattern language that we need can be established from the selected patterns. Below, we conduct a simple research analysis with a term relevant to app design as the name. With “name” as an example, we use the associations generated to obtain a basic pattern language Figure 7.

Name	Problem	Solution	User–centered design (UCD)	User experience (UX)	Context and other forces (Context/Forces)	Pattern
Text	Information is generally presented using text. Other presentation methods include figures and tables. The size of information images and the amount of information will be problems.	The concept of infographics can be incorporated to solve the problem of information and visual presentation. Interactive forms can also be used to solve the problem.	Quickly obtain the desired information and clearly see the needed information content in the image. Images with either vertical or horizontal spatial configurations can comprehensively present the information.	Easy to understand, the information updated swiftly and accurately. Of course, the content of text descriptions also has its influence.	The use of color text, the arrangement and application of figures and tables. Emotions and user perceptions will also create varying results.	The basic pattern of the language in a design pattern.

Figure 7 Association Graph and Walkthrough of App Design Pattern Language Generation

The app design patterns in the example above need to be categorized, which means starting from the name and then establishing the information language patterns in the design patterns as the categories (Figure 8). For instance, patterns must be produced one by one based on the text, proportions, order, spacing, theme, user groups, and color of the app contents, each of which is then categorized. Of course, the pattern details derived from the categorization above must still be revised, which is followed by repeated questioning and then finding the solution. After multiple revisions, it will be suitable for any app design pattern.

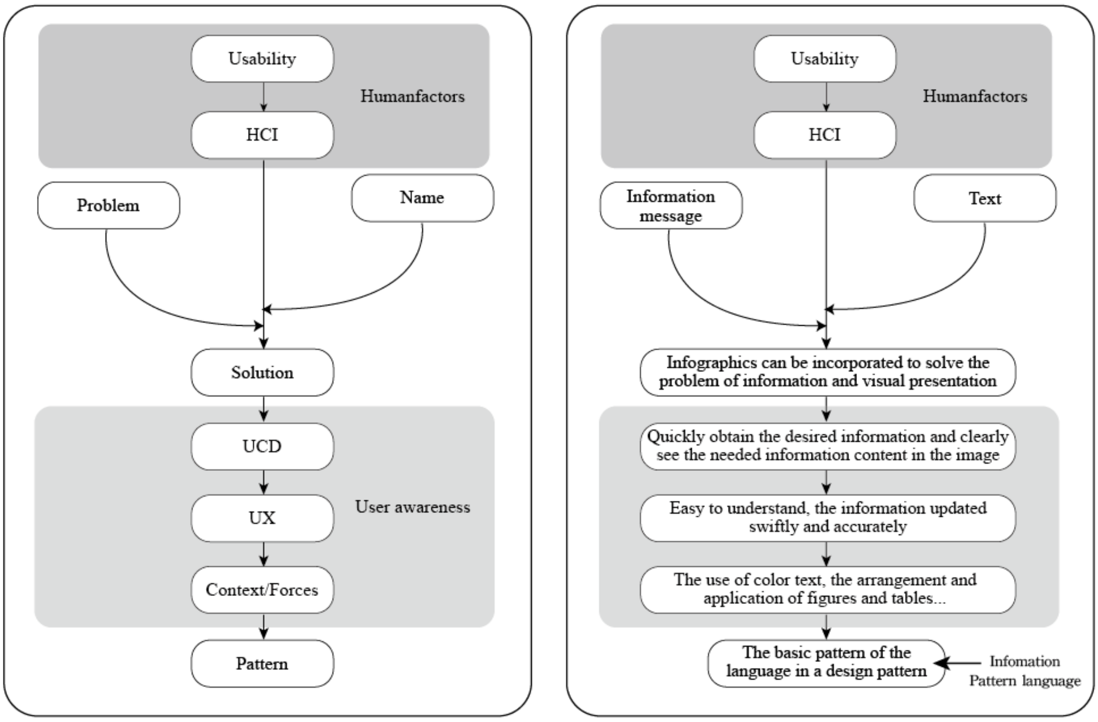

Usability
Humanfactors
HCI
Problem
Name
Solution
UCD
UX
User awareness
Context/Forces
Pattern
Usability
Humanfactors
HCI
Information message
Text
Infographics can be incorporated to solve the problem of information and visual presentation
Quickly obtain the desired information and clearly see the needed information content in the image
Easy to understand, the information updated swiftly and accurately
The use of color text, the arrangement and application of figures and tables...
The basic pattern of the language in a design pattern
Infomation Pattern language

Figure 8 Generation Relationship Graph of Pattern Languages

3.4 Experiment Design for App Design Patterns

3.4.1 Color Setting

When designing app UIs, a simple description must be given with regard to the names of included patterns, the background of the problem, the solution, the forces, and the context Figures 9 and 10. With color setting as an example:

Pattern Name	Solution
Color Setting	The three basic attributes of color are brightness, saturation, and hue. These determine the feelings that are presented by interface design. Referring to color matching reference books and color samples are also design solutions. Naturally, the effects will be more significant if UX is included.

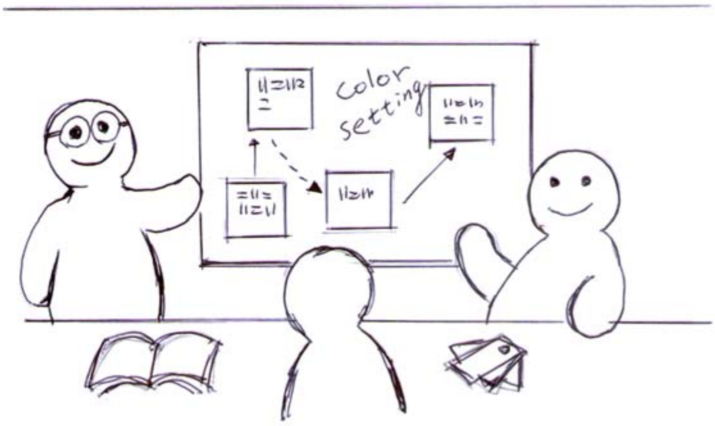


Figure 9 Information Engineers, Visual Designers and Other Relevant Personnel Discussing Color Setting

3.4.2 Context

When information engineers and visual designers discuss color setting, they must include their own UX. If they do not understand UX or UI, it will create difficulties in UI designing in the future. For color setting, they will have issues with one another regarding communication and configuration. However, in the contextual environment, the problem must be greater than the context. Thus, the occurrence of context after problem generation should not be limited to the perceptions of one or two individuals.

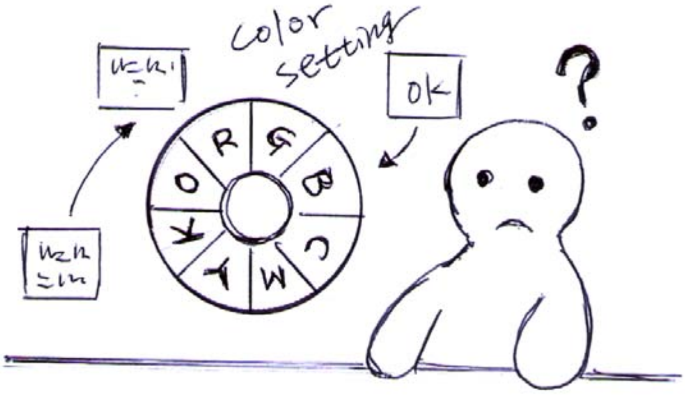


Figure 10 UI Designer Proposes Primary Solutions Regarding the Background of Color Setting Problems

3.4.3 Solution

During discussions, designers propose color applications to solve the color setting problems. Initially, they may use color charts, Pantone, or other color–related books to work out a solution for color setting. However, this approach cannot solve every color setting problem. For instance, if the user is color blind to some extent, overlooking this in app UI designs will lead to the loss of a large user group. Thus, including UX in continual design discussions is necessary to avoid falling in the so-called expertise trap.

3.4.4 Results

When the same color combination is used, it can still vary depending on the brightness, saturation, and hue of the colors. The area of the colors used can also lead to different perceptions. The guidance presented by IU design feels that flow also changes what matters must be taken into consideration during UI designing. Subjective judgment, objective solutions, age, emotions, and experience are all factors of preferences that may lead to difficulties during design.

4. Conclusion and Future Development

The objective of this study was to establish some basic methods with regard to app development involving UCD, usability, and UX using the influence of user awareness on app design pattern languages. With regard to app development and design, speed and function are not enough to satisfy users. For developers and users, interactive forms with new patterns, visual design, pattern languages that provide complete information, and cognitive perceptions are more worthy of in-depth investigation than the design of the product itself. From another perspective, mobile devices have seen significant changes in the last several years; three of the six major mobile devices in 2011, namely WebOS, Symbian, and BlackBerry, have already disappeared. Another change is that developers used to be able to create a single interface that could work on multiple systems. Today, the circumstances are the opposite: various design specifications have formed for iOS, Android, and Windows Phone (Neil, 2014), particularly in terms of design pattern languages. Pattern designs cannot solve major design problems of the products themselves. However, what is certain that following common design patterns can at least ensure that users will be able to find and use app functions. Designing a practical and useful app can only be done by repeated confirmation, experimentation, and name pattern deconstruction. After problem generation with no specific solution, patterns can be decomposed into multiple patterns to find the final solution and the best pattern. This is the final result of this study to find the best app design pattern language.

References

Borchers J. O. (2001). "A pattern approach to interaction design", *Ai & Society*, Vol. 15, No. 4, pp. 359–376.

Alexander C. (1970). *Notes on the Synthesis of Form*, Oxford University Press.

Alexander C., Ishikawa S., Silverstein M., Jacobson M., Fiksdahl-King I. and Angel S. (1977). *A Pattern Language*, Oxford University Press.

Alexander C. (1979). *The Timeless Way of Building*, Oxford University Press.

Alexander C., Siverstein M., Angel S., Ishikawa S. and Abrams D. (1988). *The Oregon Experiment*, Oxford University Press.

Chang P. C. (2012). "The study on user cognition of stylization of icon design on the smartphone", unpublished master's thesis, National Taipei University of Technology, Taipei County.

Chen C. T. and Cheng Y. C. (2003). "A pattern language for web-based learning authoring environment", *TANET–2003*, Taipei, pp. 365–371.

Chi et al. (2011). "Principle of mobile platform message design — A conception and development of inductive analysis framework", in: *The 19th International Conference of Advertising and Public Relation*, Taipei.

Hughes M. (2006). "A pattern language approach to usability knowledge management", *Journal of Usability Studies*, Vol. 2, No. 1, pp. 76–90.

Hoober S. and Berkman E. (2011). *Designing Mobile Interfaces*, O'Reilly Media, Inc..

Hsieh P. C. (1995). "A Study on pattern of urban openness space", master's thesis, National Cheng Kung University, Tainan County.

Lowdermilk T. (2013). *User-Centered Design: A Developer's Guide to Building User-Friendly Applications*, O'Reilly Media, Inc..

Lu J. S. (2010). "A study of combine interactive sensors and context-aware for human computer interaction design", unpublished master's thesis, National Taichung University of Science and Technology, Taichung County, pp. 1–118.

Neil T. (2014). *Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps*, O'Reilly Media, Inc..

Ng S. and Wadhwa B. (2014). "A pattern language for mobile application development", in: *Asian PLoP 2014: 3rd Asian Conference on Pattern Languages of Programs*.

Takashi Iba (2007). "Creation toward quality without a name — Sociological analysis of pattern language", in: *1st International Workshop on Software Patterns and Quality (SPAQu'07)*, p. 37.

Vlissides J. M., Coplien J. O. and Kerth N. L. (1996). *Pattern Languages of Program Design 2*, Addison–Wesley Longman Publishing Co., Inc..

Yuan M. S. (2008). "Externalizing tacit knowledge in independent documentary film production via pattern language", *Journal of Library and Information Studies*, Vol. 6, No. 1/2, pp. 57–82, available online at: http://squall.cs.ntou.edu.tw/UserInterface/DesignOfHCI.html.