Generative Pattern Languages for Interaction Design

Martijn van Welie[1,2]
martijn@welie.com

[1], Vrije Universiteit, Faculty of Sciences
Department of Computer Science
Section "Information Management & Software Engineering"
Sub section "Human Computer Interaction, Multimedia & Culture"

[2], Satama Interactive, Amsterdam

Introduction

A pattern by itself is just a small piece of the entire design knowledge "puzzle". Each pattern describes a proven solution to a problem in a certain design context. When all the pieces of the puzzle are "put together", we can see the how an entire body of design knowledge is unfolded. The understanding of this puzzle is *the* long-term goal in pattern-research. For it will show the paved roads of design, but it will also say when the road should be abandoned in search of new and innovative solutions. It is therefore important to create a meaningful and practical way to structure pattern languages in our field. In my vision, only qualitive pattern languages in our field will be able to demonstrate the usefulness of patterns in respect to other design aids.

Alexander's original idea

When Alexander wrote his book on architecture design patterns, it did not just contain patterns; the patterns formed a language. His language was hierarchical and started out on the level of cities, then neighborhoods, houses until the level of windows or seats was reached. In Alexander's idea, the language actually "generated" the design by traversing from high level patterns to the lowest level of patterns. From the design of cities down to the design of window seats, a hierarchy of *scale*.

The question is now whether we can create a similar sort of pattern language for Interaction Design. One big difference with architecture is that user interfaces are not strictly hierarchical in a geometrical sense. There is certainly a 2D display involved but what is shown on it varies over *time*. Therefore, a strict hierarchy based on the usage of screen estate is not possible for interaction design. However, the hierarchical nature of architectural patterns can also be interpreted as a hierarchy of *problems*. The highest level of problems are broken up in smaller problems for which solutions appear to exist. They just happen to map directly to a geometrical metaphor in architecture, working from large areas to small areas. The important thing to understand is that such a problem-hierarchy approach can be applied to other domains as well.

A pattern language for Interaction Design

If we try to apply Alexander's idea for a pattern language of scale, we must adopt the interpretation that 'scale' means scale of problems rather than geometry. In Interaction Design there is certainly a 'scale hierarchy' of problems. We may not always be explicitly aware of it but it is the hierarchy we mean when talking of *top-down design*. Usually design is a top-down activity where we start with gaining understanding of the users and their tasks, the client's wishes, technical environment, business context etc. Taking the example of web design, design continues by laying out the foundations of the application in terms of the site concept, information architecture, and basic functionality. The concept outlines the basic characteristics of the site that will be filled in later on up to the point were individual screens and widgets are laid out. Such a top-down approach will 'generate' a design when patterns are available at all levels.

When looking at such a networked set of patterns, we can also see layers of patterns emerging, when going from high level patterns to lower level patterns. These layers are rough delineations of the typical levels that are encountered in design. The levels I have identified so far are *posture*, *experience*, *task* and *action*.

Posture Type patterns

Every site or application is there for a *purpose* or has a reason for existence, for commercial sites there are usually business goals to be achieved while other sites have more *personal* or *social* goals. Proper design has its foundations in understanding why the design project is started in the first place. These stated business goal feeds into the choice for a 'kind of site' that is adequate..

From experience we know that many sites are actually quite similar in the sense that they serve the same goals and have a structurally similar site concept. We call this a site '*posture*' or 'type'. For example, small corporate sites are often so similar that we can write patterns describing them, the same goes for news sites, community sites and so on. We can define several of such site postures that can be taken as a basis for new site design projects. Patterns that describe such typical site postures are therefore called *posture patterns*. Van Dyne et al (2002)[1] also describe several of these site postures.

A posture patterns describes what the essentials of that posture are: what kind of site structure is usually used, which elements typically make up the homepage but also the main *experiences* that such a site is supposed to offer. It is like deciding whether you are going to design a 'sports car', a 'family saloon car', a '4x4' or a 'city car'. Each of these have specific characteristics and experiences that together form a type of car.

Many sites can be directly derived from the known postures but it is also common to design a site as a mix of postures. When a posture has been selected, several lower-level posture patterns will help to define concept level issues such as homepage design, promotion areas, navigation, templates etc. User research or contextual inquiry will help designers to decide which postures are most relevant.

Experience patterns

From the basic posture and from your user research, designers will have to determine what the main *user goals and tasks* are that need to be supported and to what extent. We will call this the '*experience*'. The user experience is not just about tasks and goals but also about how the users reach their goals using a site concept, how they perceive the site and whether it gives them the appropriate satisfaction. Experiences should therefore be understood as a broader design for which we are designing. The experience-level patterns describe common experiences and which lower level patterns can be used to create that experience. Typical experiences are activities such as "shopping", "playing", "browsing", "information gathering", "problem solving" or "sharing thoughts". When describing for example 'shopping' it is necessary to describe what it is even without taking into account the technology we are using. This includes browsing products, comparing product, asking advice etc. When we understand how shopping works we can then add references to lower level patterns that can be used to create the experience. See http://www.welie.com/patterns/shopping.html for an example of an experience pattern. Experiences are the high level goals for which the users come to a site. When comparing it to car design again, experiences can include 'sporty driving behavior' or 'luxury feeling'.

Every site type has a primary experience that it wants to offer. For example, an e-commerce site is primarily for a *shopping* experience. However, secondary experiences may include *community-building* or *information gathering*. The interaction designer needs to balance these experiences and create a consistent user experiences for the entire site. In practice, this will mean that an e-commerce site will use some elements from secondary experiences. In a similar way, a news site may use elements from a shopping experience for dealing with premium paid for content.

Task patterns

The task level is the level where we start to see most concrete and well-known patterns such as SHOPPING CART or PRODUCT COMPARISON. These will point to lower-level task patterns such as WIZARD or SET BUILDER that are needed in high level task patterns. Task patterns are describing solutions to small user problems that are part of a higher level "experience". Typically a task pattern describes a series of interactions on one or more objects for solving a problem. Such a series corresponds to a task sequence needed to achieve a task goal. Task patterns are relatively domain independent. The posture and experience patterns set the context specifics and the task patterns are used to fill in the blanks. Task patterns can often be 'drawn' using flow diagrams and sketches.

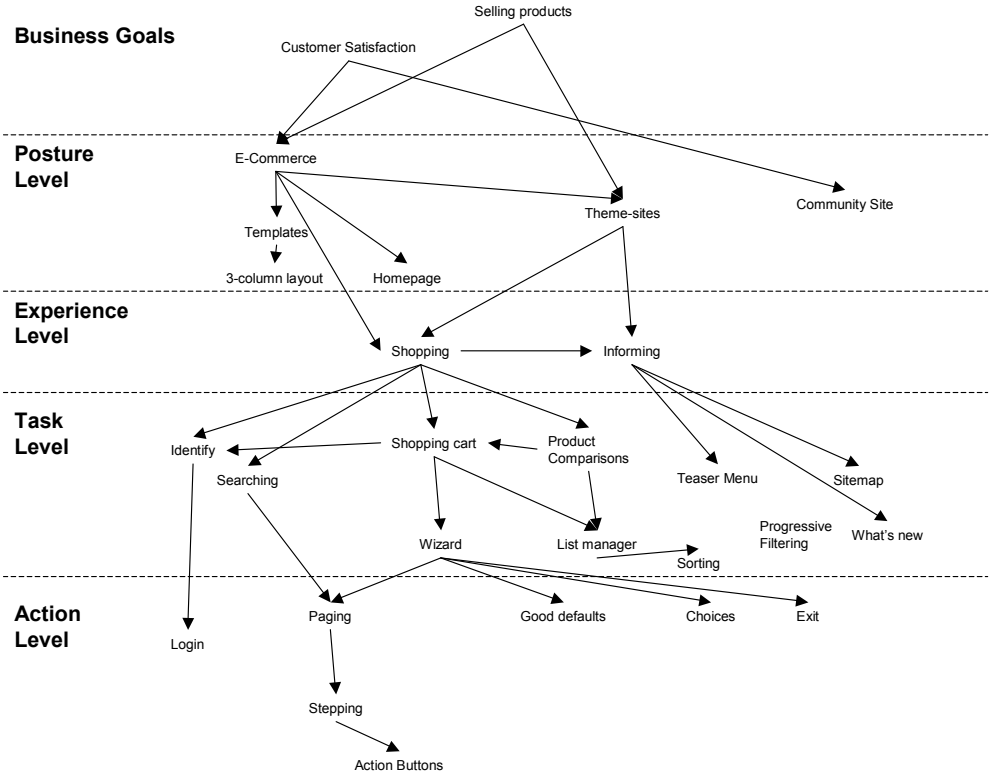
---

[1] The Design of Sites, van Duyne, Landay, Hong, 2002

Action patterns

The action level patterns that are patterns that not really related to a clearly defined task. A PUSHBUTTON or CLEAR EXITS are not really related to tasks, they are actions that are meaningful in real tasks such as "order", "go the next step" etc. We call these "action patterns" and they are often similar to widgets. They occur is almost all task patterns and are the lowest level of building blocks we still want to call a pattern. The solutions described in them are usually specific uses of well known widgets or describe custom-made widgets.

The different levels and associated patterns can be shown in a graph of connected patterns, see figure below. In the graph all types of inter-pattern relationships are use shown using a directed arc. Actually, the relationships are contained in the patterns themselves, every time a reference is made to an other patterns as part of a context or solution statement. The figure below only shows a partial graph centered around the SHOPPING experience. Because of the complexity of the domain, a complete graph could have more than 200 patterns that are all connected leading to a, perhaps, not so comprehensible diagram.



Conclusions

I have outlined a possible structure for a generative pattern language in Interaction Design. The next step would be to actually write an entire pattern language that follows this structure. Only then can we see if this really works and if it leads to a more effective pattern language.