

Learning and Studying Interaction Design through Design Patterns

Miguel Carvalhais
University of Porto, Department of Design
Portugal
mcarvalhais@fba.up.pt

ABSTRACT

The creation of a course on Interaction Design at the School of Fine-Arts of the University of Porto presented a good context for the deployment of a pedagogical program built around the study and the writing of design patterns for Interaction Design. This paper presents a report of this experience, of why design patterns were used and how they were studied and created by students.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces – *Graphical user interfaces (GUI), Screen design, Theory and methods*

K.3.2 [Computers and Education]: Computer and Information Science Education – *Computer science education, Curriculum*

General Terms

Design, Human Factors, Theory.

Keywords

Interaction Design, GUI, HCI, Education, Communication Design, Pattern Languages, Design Patterns.

1. INTRODUCTION AND CONTEXT

At the School of Fine Arts of the University of Porto (FBAUP) we have been faced with a growing demand for studies in Interaction Design and in the design of digital systems, both in undergraduate and post-graduate levels. At the forefront of the market demands for design graduates, we currently find web design, software design, hybrid and digital media skills and, even in the traditional areas of communication design such as corporate identity or editorial design, we nowadays feel the need for not only basic skills but very often for a high expertise in digital systems, be it in the design of corporate identities that need to be deployed in digital contexts or in the design of such media as eBooks, DVDs, etc.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission.

Preliminary versions of these papers were presented in a writers' workshop at the 15th Conference on Pattern Languages of Programs (PLoP).

PLoP'08, October 18-20, Nashville, TN, USA.

Copyright 2008 is held by the author(s). ACM 978-1-60558-151-4

In spite of this, most of the skills that communication design students develop during their undergraduate training at FBAUP are hardly oriented towards digital technologies. Digital tools are of course learned and used, but besides some very specific courses dedicated to e.g. Multimedia Design (4th year, undergraduate), Web Design (2nd, 3rd or 4th years, undergraduate) and Introduction to Programming (1st year, undergraduate), there is hardly the time to closely study digital media, much less their technical aspects. Digital systems are studied as tools, and most often they are not specifically studied as a field of work *per se*, but more as resources to be used in developing projects in other fields, such as print, audiovisuals, etc.

Interaction Design skills need to be built upon a broader procedural literacy that has long been defended as indispensable for any practitioner of new media. As early as in the 1970s, Ted Nelson defended that “the more you know about computers, the better your imagination can flow between the technicalities, can slide the parts together, can discern the shapes of what you would have these things do.” [quoted in 14] Authors as Stephen Wilson [24], Moira Cullen [7], Gregory Ulmer [20], Brenda Laurel [10], Joseph Weizenbaum [23] and the aforementioned Ted Nelson [13], to name just a few, made strong cases for the necessity of studying programming, not a single specific language, but rather, as Michael Mateas puts it, studying “the more general tropes and structures that cut across all languages”, garnering a “general competence in computation as the medium for representing structure and process.” [11]

Structure and process are, therefore, the matter with which *form* is created in digital systems. If the “ultimate object of design is form” [1], it is not enough to be able to shape the matter with which it is created, to use the tools that create it, or to read and write the language with which it is summoned, but it is also essential that there is a “programmatically clarity in the designer’s mind and actions”, so that she can “trace [the] design problem to its earliest functional origins and be able to find some sort of pattern in them.” [1]

It is then necessary to be literate in procedural phenomena, and also in the culture of the design project for interactive systems, to be literate in the means through which cross-disciplinary undertakings can be developed and accomplished.

In the school year of 2006/2007 we were asked by the Design Department at FBAUP to organize a course on Interaction Design for the second semester of the first year of the master courses. In this context, two immediate problems were identified: (1) How to study Interaction Design with students that, in spite of having an extensive knowledge of communication design, have little or no skills in programming and have mostly never developed complex projects for digital media; and (2) how to study Interaction Design

by placing the emphasis on digital systems as a broad field of work and not on particular systems — personal computers, handheld devices, web sites, games, installations, kiosks, etc — so that the class could be as useful as possible to a group of students coming from different backgrounds, with varying interests, perspectives and sub-fields of specialization.

Following examples found in the literature that we compiled for this course and also the experience gathered from consulting colleagues that teach engineering and computer programming, we found a possible answer in the study and development of design patterns for Interaction Design.

2. WHAT ARE PATTERNS

Christopher Alexander's *Notes on the Synthesis of Form* [1], is an indispensable treatise on the project of design. In this book Alexander presented a very solid, methodical and systematic approach to the problems of design and, although he wasn't focusing on the issues of designing for digital systems, he nevertheless considered the difficulties posed by very complex contexts and problems. When discussing methods to explore the conceptual order required by complex design problems, he used strategies that can also be found in the fields of computer sciences and artificial intelligence, most notably, the subdivision of (complex) problems in several simpler ones, and the further subdivision of these into even simpler problems, defining "goals" for the project that add up to the higher-level goal that answers the problem, that defines the form that *fits the context*.

If the object of design is form, Alexander warns us that this form could not be the result of a process that uses any arbitrarily chosen formal order, because forms that are created this way, although in some sense originated by the problem, do not manage to fully answer it. The core of the design problem is to achieve a good, if not perfect, fitness between the form being designed and the context that called for it, [1] so that the form can be the actual solution to the problem that is defined by the context.

When dealing with complex contexts, the subdivision of the higher-level problem gives room to a multitude of lower-level problems that call for various strategies of design. Whenever one of the lower-level problems is adequately solved, the designer garners knowledge and thus, if she wishes to make any future use of this knowledge, she is faced with the question of how to represent it and how to effectively explore it in behalf of other contexts — how to represent the connections that helped her reach the solution of the problem. The designer needs to develop what Marvin Minsky called a "knowledge-based problem-solving system" [12], if she ever wants to be able to return to the proven solutions of identifiable and recurring problems in any of the various levels of the program or of its realization, as defined by Alexander in *Notes on the Synthesis of Form* [1].

In *The Timeless Way of Building* [2] and *A Pattern Language. Towns, Buildings, Construction* [3], Alexander proposes a system to do this systematic gathering of knowledge about the multiple bits and pieces that are brought together in the resolution of the design project. He develops the basic concepts of the *pattern* and of the *pattern language*, and not only defines the pattern format — the common structure for the development of individual patterns — but also proposes the idea of a larger framework of patterns that are articulated in a language. This language is defined by causal connections in a hierarchy developed from

patterns with a high-level and broader scope, towards patterns with a lower-level and narrower focus.

Alexander presents patterns as definers of a common vocabulary for all the agents of design and as time-savers and record-keepers of good and tested solutions for recurring problems. According to him, a language of patterns should (1) list and name the most common problems in a given field of design, (2) describe the main characteristics and the most effective solutions to a given problem, (3) help the designer to move along the problems in a logical way and, finally, (4) allow many alternative routes through the process of design.

Each of the design patterns in the language should be the "abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts" [15], or, as Brad Appleton suggests, "a named nugget of insight that conveys the essence of a proven solution to a recurring problem within a certain context amidst competing concerns" [4]. In *The Timeless Way of Building* [2], Christopher Alexander summarizes that, (1) a pattern expresses the relation between a certain context, a problem, and a solution; (2) as an element in the world, each pattern is a relationship between a certain context, a certain system of forces which occurs repeatedly in that context, and a certain spatial configuration which allows these forces to resolve themselves; (3) as an element of language, a pattern is an instruction, which shows how a configuration can be used, over and over again, to resolve the given system of forces, wherever the context makes it relevant; and finally that a pattern is (4) at the same time a thing and the rule which tells us how and when to create that thing — it is both a process and a thing; both a description of a thing and a description of the process which will generate it.

Each of the patterns in a language should share a common format that includes at the very least, (1) a name, which is itself a short description of the pattern; (2) a problem description that outlines the conflict of forces in a given context; and (3) a discussion that explains the problem in more detail that describes evidences for the existence of the problem and finally points to the methods to solve the problem.

Design patterns are then a way to systematize good practices and proven solutions to several (many, or all) of the problems that designers face in any given field of activity. Design patterns document the problem that demands a solution, following it with the discussion and explanation of the solution that is being presented. Design patterns try to understand the forces in conflict at the core of the problem and to propose a set of values to steer the designer towards a decision that can be the best for the particular context that is being worked.

Design patterns should be sufficiently abstract to be more or less universal in their usefulness, but simultaneously specific enough to be of practical use, therefore, design patterns should present and explain their contexts of applicability, referring to other patterns whose solution they can help to develop and likewise reference further patterns that can be used in its solution.

All of this should be done in simple and synthetic documents, written carefully but economically while keeping in mind that each of the patterns is just a piece in the language, that its value is not independent from the remaining patterns and from the way it can be articulated with them.

This paper aims to affirm the usefulness of the system of patterns in the development of a methodology for the study of design, particularly, for the field of Interaction Design.

3. WHY PATTERNS IN DESIGN EDUCATION?

A system of design patterns allows the simultaneous study of both the general problems of, and the specific solutions for, the digital systems that both interest students and present most of the professional demands they will face. Design patterns can prospectively fit systems that are yet to be designed, or that cannot be experienced by the students or in the class, due to technical, temporal, or economical limitations.

Design patterns can be developed with varying scopes: from very broad-range and high-level patterns dedicated to the cognitive forces of the users or to information architecture, all the way down to very narrow-ranged and focused patterns, dealing with more specific issues, such as portable devices, web sites, interactive installations, games, or sub-classes of problems of these issues.

Patterns can also be used as (1) tools for the systematic gathering of knowledge about Interaction Design at multiple and contrasting levels of details, that is, as tools for the theoretical study of Interaction Design; (2) tools for the design of digital systems of variable complexity; (3) tools to guide the implementation stage of these projects; (4) tools for the analysis and the critical review of design projects; (5) reference tools for the usage and commissioning of design for digital systems; and, finally, as (6) a system for the creation of a vocabulary for communication design and parallel disciplines. Therefore, a language of patterns can be a valuable resource whatever the future needs of the students may be, both in academic or practical design work, as well as a good resource for those professionals from multiple disciplines with whom students will need to work in the collaborative development of digital systems.

Christopher Alexander envisioned most of these usages when he first studied the design project and suggested the language of patterns for architecture and urbanism in the 1970s. Three of his works from this period have been very important foundations for the development of this course, the first of them, *Notes on the Synthesis of Form* [1], due to the insights on the design project, the other two due to the system of patterns they presented. These patterns, theoretically introduced in *The Timeless Way of Building* [2] and published in *A Pattern Language. Towns, Buildings, Construction* [3] have, amongst other things, provided a comprehensive and very well articulated example of a complete language of patterns for reference. The fact that they focus on another field of design and not on Interaction Design was actually beneficial, as it allowed students to understand the concept of a pattern language and its relevance in a sufficiently abstract level, focusing on the instrument and not on its contents, thus leading the way to the development of the class work.

4. PATTERNS FOR INTERACTION DESIGN

Having established this ground to develop the course, we searched experiences and references for patterns for Interaction Design. At the time of the initial preparation of the course, in 2005/2006, we used two main books. A third reference was released in 2007 and was then included in the bibliography and lectures. These three

books were not by any means the only references in the bibliography (available at <http://tinyurl.com/5rykut>), which also compiled online references and several other resources, with themes that ranged from design (communication, visual, sound and interactive) to media theory and computer sciences, etc.

A Pattern Approach to Interaction Design [6] was, to our knowledge, the first book that used patterns to systematically present knowledge gathered in the field of Interaction Design. It is an interesting work both for the preliminary study of patterns and pattern languages but also because of the parallel development of more than one language of patterns. Borchers presents a language for interactive musical exhibitions and extensively debates the usage of patterns in other disciplines. He then subdivides the Language into three clusters of patterns that can be seen as three parallel Languages: (1) a Musical Pattern Language, (2) a HCI Pattern Language, and, (3) a Software Pattern Language. This multiple approach is very interesting as it shows the resourcefulness and malleability of a system of patterns, although on the other hand it may limit the scope of the patterns to a somewhat too specific usage. Maybe as a further consequence of this “dispersion” of the author, we have found that the number of thirty-two patterns in the book somewhat limits its practical usefulness.

Borchers’ approach is nevertheless appealing due to the macrostructure that he proposes for the organization of the patterns, something that was perhaps the most significant contribution of his book to our ongoing work. Borchers initially uses a structure that classifies patterns in nine levels according to *scale*, much in the same way that Alexander did: (1) Society, (2) Multiple users, (3) Social position, (4) System, (5) Application, (6) UI Structure, (7) Components, (8) Primitives, and, (9) Physical properties [6]. Alternatively he later proposes a simpler structure, based on *process*, split in only six levels: (1) Culture and society, (2) Environment, (3) Role of the user, (4) Use, (5) Navigation, and, (6) Structural levels, further subdividing tasks into (a) retrieval, (b) monitoring, (c) proactive and reactive controlling, (d) construction, (e) transaction, (f) modifications, (g) calculation, (h) workflow, and (i) communication [6]. We will later see how we have found that a simpler macrostructure was seemingly more adequate to the development of a pattern language for Interaction Design, and how fewer levels have allowed us to relate patterns more easily in the resolution of practical problems.

The second book that we initially studied was *Designing Interfaces, Patterns for Effective Interaction Design*, by Jenifer Tidwell [19], a revised and expanded version of her 1998 collection. This is a far more comprehensive collection of patterns than Borchers’, and it is also more universal in its intent, not focusing in a somewhat specific field of work. The larger part of the ninety-three patterns that Tidwell presents became the core material for the organization of the lectures in the first implementation of the course, and the structure of her language defined the macrostructure of the classes, organized in seven main topics covering: (1) Users, (2) Information Architecture, (3) Navigation, (4) Visual Layout, (5) Actions and Commands, (6) Information Graphics, and, (7) Forms and Controls. Builders and Editors and Visual Style and Aesthetics, the remaining two sections of Tidwell’s book were not turned in discrete sections in the classes, the first due to its high specificity and the later due to the overlap with the students’ background training and the subsequent redundancy and repetition of content it could bring to

the lectures. Visual style and aesthetics were almost permanently discussed during the classes, usually relating them to the patterns that were presented, but most of the time not through the development of specific patterns for this.

Tidwell's patterns are in general very complete, although they sometimes lack visual references or examples and extensive cross-referencing of other patterns. The macrostructure of the language is clear, but the definition of some of the sections leads to some redundancy or to difficulties in the categorizing of the individual patterns in the structure. The microstructure of the individual patterns is very loose, something that in general simplifies them, but that can also be a peril for the internal consistency of the language. In general, being more broad in scope than Borchers', Tidwell's work proved to be more valuable as a tool for instruction, precisely because it touches multiple aspects of the field of Interaction Design, and it mixes platforms, tools and systems in the problems and examples discussed in the patterns.

Besides these two books, we referenced the work on design patterns for object-oriented programming, *Design Patterns: Elements of Reusable Object-Oriented Software* by Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, [9] not so much as a work-book but as another good example of how patterns can be used in a field that is so close to ours as (information and communication) designers but that is simultaneously so alien to most of the students in the class.

We used some online references and collections of patterns, mainly Martijn van Welie's collection [22], Tom Erickson's collection of references in *The Interaction Design Patterns Page* [8] and the Yahoo! Developer Network's *Design Pattern Library* [25], all of which were included in the references given to the students as part of the course's syllabus. These three collections, although of variable overall utility in the context of the classes became important inspirations to the platform we adopted for the development of the work in the course and to the system used to compile examples for each of the patterns developed. As an example, the possibility of commenting on the patterns that is offered by *A Pattern Library for Interaction Design* alerted us to the potential benefits of collaborative work and public discussion that later led us to choose a wiki platform. A more recent platform, *Pattern Tap* [17], has been referenced, but, by having its main focus on the visual cataloguing of design solutions and not really in the development of patterns, it didn't prove to be useful in this context. Other collections, e.g. *Stimmt AG - Uni Basel Pattern Catalogue* [18] or Christian Behrens' *Info Design Patterns* [5] were also not included, mostly due to time constraints in the course.

The third book to be added to the main references of patterns was *The Design of Sites: Patterns for Creating Winning Web Sites*, by Douglas K. van Duyne, James A. Landay, and Jason I. Hong [21]. This book is obviously oriented towards website design and it structures each of the patterns in a somewhat simpler (and eventually more versatile) way, reducing the fundamental structural sections of the patterns to just (1) Name, (2) Background, (3) Problem, (4) Forces, (5) Solution, and, (6) reference of other patterns. The macrostructure of the language is divided into a larger and more complex number of major sections: (1) Genre, (2) Navigation framework, (3) Homepage, (4) Writing and content management, (5) Trust and credibility, (6) Basic e-commerce, (7) Advanced e-commerce, (8) Help to fulfill tasks, (9)

Page layout, (10) Search, (11) Facilitation of navigation, (12) Speeding up the site, (13) Mobile web.

This complex structure can be justified by the extreme specialization of this language, that unlike Tidwell's, doesn't propose to be more or less universally applied to any (or most) of the digital systems, but just to one particular sub-class of systems.

While both Borchers and Tidwell dedicate initial sections or chapters of their books to the roles or behaviors of users, van Duyne et al. do not. However they elevate to a major section of their language something that Tidwell just briefly mentions in the introduction to her book: the idea of genre, or what Tidwell identifies as "idioms" of the interfaces.

This third book also raises issues regarding the way in which we can devise patterns that are very specific and that address questions pertaining to single-purpose problems of sub-fields of digital systems. We have been discussing this topic in the classes since the work started and have been experimenting with possible solutions that may allow us to formulate as patterns some specific interface solutions for applications or web sites — something very often found in the Yahoo! Design Network's *Design Pattern Library*, for example.

This last book's contributions to the course and to the language have been limited so far. This was mostly due to time constraints. However, parts of it and of its patterns have been integrated into the class, have inspired ongoing patterns, and are crucial references for the current work in progress.

5. USAGE IN THE CLASS

While many of the patterns from the references have been used in the lectures, a more ambitious proposal was presented to the students in the two semesters during which the Interaction Design course has been offered.

Based on the idea of a pattern language and on those patterns discussed and presented in the classes, the students were challenged to collaboratively develop a pattern language for Interaction Design. This assignment was presented as the main evaluation project in the course, replacing the more usual final written test. The proposal was to develop a broad language for the study and development of interaction design projects, for their analysis and critical review, a language that would not lead students to a forced specialization in a particular sub-field of Interaction Design but that could benefit from the existing specializations.

Asking students to be involved in the creation or development of the patterns turned out to be advantageous in several ways:

(1) Patterns are seen as living entities, not as static recipes for producing instant results. They are seen as a collection of phenomena in the world of design, as models of good practices that can be collected and studied beyond the more direct or simple visual layer and that can, and should be, continuously revised and expanded;

(2) As students are asked to write patterns instead of just reading them, they are encouraged to see patterns as both a theoretical resource and as a framework for creation;

(3) The investment in the work of writing the patterns leads the students to study the language much more closely than they otherwise would;

(4) Students create the theoretical tool, not only study or use it, thus coming closer to the matters being studied, actively researching the bibliography that is provided and other references that they may need to track down to document specific patterns;

(5) Students are encouraged to bring their individual backgrounds to the class and learn through an active, collective, and social process. This has proven results in the motivation of the students, a good example is the way that some students from the art courses that enrolled in the course as an elective have actively participated and developed very interesting work in this context.

Inspired by the previously mentioned online platforms, a wiki was setup to organize the patterns and allow tracking of the works in development. Each student was requested to either (1) start a new pattern, or, to (2) develop one or more of the existing patterns by expanding or reviewing its contents, images, references, notes or examples. Furthermore, students were encouraged to use the "discussion" pages to comment on the work in progress by their colleagues, thus generating an open discussion that contributed to the improvement of the patterns and an overall qualitative increase of the collection.

At the start of the first offering of the course (the second semester of the 2006/2007 year), the wiki contained no patterns, and all of those patterns that were referenced from Borchers, Tidwell and the other authors were only discussed and presented orally in the classes. Short summaries of these were posted in the online notes from each of the lectures.

Students were encouraged to develop their patterns directly in the wiki, starting with the information they had (most usually a name, the description of a problem, maybe references) and to work on the pattern or patterns during the semester, keeping a visible work in progress that could benefit the development of all the other patterns. This would ideally allow all the students to cross-reference other's works, to check data, and continuously develop a work-in-progress format for the patterns. The idea of keeping this public work-in-progress also tended to lead the students to develop the patterns more thoroughly, checking citations and keeping records of all the information while, ideally, converging towards a somewhat unified "style" of writing, something that would probably be very difficult if all the work was developed individually and compiled only at the end of the semester. Furthermore, all students were encouraged to edit and amend patterns of their colleagues, adding useful information whenever they could do so sensibly. Their work was not evaluated quantitatively but rather qualitatively and the "history" of the wiki allowed checking the context and relevance of all the contributions. Registration in the wiki is limited to the students and teacher of the course, read-only access is unrestricted.

The global coherency of the style of writing of the patterns was more visible in the second edition of the course, as students didn't have to start from scratch but rather from the work developed during the first offering. Some interventions in those first patterns brought them closer to the patterns of the second class, but a few remained in a style that is still not totally consistent with the rest of the collection.

Work on the patterns was developed over the course of ten weeks and during this period the patterns in progress were regularly discussed in the class, in various round tables during which the work in development was presented and debated.

Throughout the second course, both because the number of students in the class had more than doubled since the first semester and because the sixty-two patterns on the wiki were already enough to allow it, we experimented with a second exercise. The class was asked to use the patterns in the wiki in the development of a work of analysis and criticism of interactive systems and tools. Each group of two or three students was requested to select a whole or partial system or tool, and to develop its critical analysis, starting by listing those patterns that were observable in the object of the analysis, and proceeding to elaborate how these patterns provided a solution for the context for which the system had been designed, how they were articulated in a pattern language specific of the object.

This second exercise was not only intended as a test for the students and the collection of patterns, but also as a further contribution to the work of developing the collection. By using the patterns in the extensive analysis of real-world examples, some of their shortcomings would perhaps be made more apparent to both students and teacher, and a revision would be possible. Problems were indeed identified, in references, lack of examples, and many other small details that were at the time revised, not only in the patterns that the class was working by then, but also in patterns developed during the previous class, which were edited if necessary.

6. STRUCTURE OF THE LANGUAGE AND THE PATTERNS

After several discussions in class, the macro hierarchy of the language was defined in what was perhaps one of the simplest possible way, dividing the patterns into as few as 6 groups: (1) User, (2) System, (3) Application, (4) Output (human perception), (5) Input (system perception), and (6) Primitives. Adopting this simplified structure aimed at creating a hierarchy that was not so much based on scale or process but rather, to be able to organize the patterns as modules that could be freely connected in relations that were not imposed by the hierarchy. While studying the language we have been finding that sometimes the same pattern can be used at multiple levels in the specific language of patterns built for a project, therefore, we concluded that it wouldn't be really adequate to divide them so clearly in the index of patterns. The discussion regarding this index is still ongoing and thus it is subject to change as more patterns are developed and languages for specific projects are built.

The goal was not to create one or many specialized languages with independent patterns (one language for the web, one for mobile devices, one for operating systems, etc.) but rather to create a collection of patterns from which multiple languages could be created whenever necessary, thus sharing knowledge and data between specific fields that otherwise eventually would not share information.

The microstructure was also discussed and tuned over time. At the moment of writing, each pattern is structured around (1) Name, (2) Validity, (3) Context with references, (4) Headline of the problem, (5) Description of the problem, including notes and examples when necessary, (6) Solution, including notes and examples when necessary, (7) Diagram, and, (8) References.

Great care was put in the observation of a solid and regular structure for the patterns, trying to always develop all the sections of their structure (even the diagram, that being a seemingly smaller section of the pattern is extremely important due to how

much and how well it is able to synthesize the core of the pattern). This wasn't achieved in all patterns, due to limitations of time and other constraints, or to the fact that students were unable or unwilling to finish some of the patterns. The nature of this collective work allows for this openness, as incomplete patterns are always seeds for the future work of students.

We insisted on the importance of limiting the extension of the patterns, appealing to economy, trying to make the patterns as easy and fast to read as possible. Whenever necessary, notes about specific issues or multiple examples can be added, but preferably outside the main document of the pattern, always as individual, hyper-linked, pages.

7. EXAMPLE PATTERN

The current collection includes sixty-two patterns in various levels of completion. The following pattern, included in the "Application" group, is presented as an example of the work developed by the students:

7.1 Name

Right and Left Alignment

7.2 Validity

**

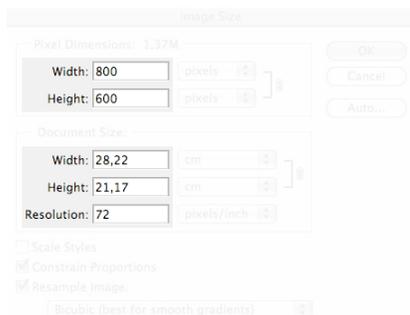


Figure 1. Image size dialog from Photoshop CS3.

7.3 Context with References

...defines an axis for separation and alignment in two column tables or forms. The items in the left column (usually legends) align to the right; the items in the right column (usually contents) align to the left...

7.4 Headline of the Problem

How to compose pairs of items in a list?

7.5 Description of the Problem

When we need to compose text input fields, selectable choices or other input controls, there is a big diversity of possible elements to use and complex layout decisions necessary to fit together both the controllers and the legends that identify them and that therefore need to be composed near to the controllers. This problem is also felt when composing non-editable contents where several pairs of content and legend can be composed in a two-column structure.

When reading left-to-right, the visual flow would naturally suggest to align both content and legends to the left, however, when aligning all the elements on the left column to the leftmost

among them, these will tend to be composed further from their counterparts in the right column, depending on their length and whether wrapping of the text is or isn't possible.

As stated by the Proximity Principle of Gestalt's theory, the pairs would maybe not be perceivable, once the unevenness of the white space between columns would compromise the correct interpretation of the pairings between contents and legends. Formally, both columns would be understood as consistent groups of information but the two alignments to the left would reinforce the vertical relations between different elements and not those horizontal relations one would need to emphasize.

Composing the controllers immediately after the end of each of the legends would result in a very asymmetric composition and the unforeseeable horizontal placement of each controller would have a negative effect in the usability of the system.

7.6 Solution

Visually group all the elements in a block that is split by a clear vertical axis. Align the contents of the left column to their right and the contents of the right column to their left, therefore creating clearly identifiable pairs of contents.

RIGHT AND LEFT ALIGNMENT should be used when it is necessary to apply a two-column structure to a list of contents and their legends. By using this pattern, we will align the legends to their rights and the contents to their lefts (or vice versa), thus minimizing the white space that separates them. This spine structure will lead the reader to more immediately perceive visual pairs, due to the principle of proximity. The spacing between legends and contents should be uniform, reinforcing the central axis and making the better use of the continuity principle of Gestalt. The visual flow is therefore driven from the top to the bottom of the list.

When composing editable contents, the legends should be composed on the left column so that the user can read them just before reaching the controllers for editing, knowing beforehand what these can be used for. In the case of non-editable contents this arrangement is not so important.

In specific cases, when the legends are long, it can be justifiable to limit the width of the column where they are composed, wrapping the text in two or more lines, if the context allows it.

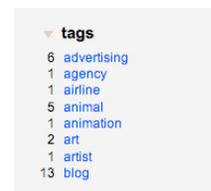


Figure 2. List of tags in del.icio.us

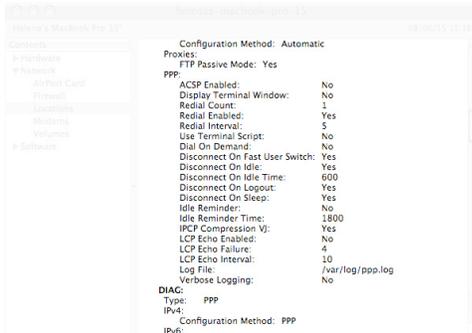


Figure 3. Example of not using the pattern: Mac OS X System Profiler.

7.7 Diagram

No diagram was produced for this pattern.

7.8 References

(a) STRIPED LINES is a possible solution to analogous problems, especially when more than two columns are necessary; (b) This pattern can be articulated with TITLED SECTIONS in the composition of complex contexts; (c) When dealing with editable elements, KEYBOARD ONLY, KEYBOARD SHORTCUTS and JUMP TO ITEM should be considered in order to facilitate the navigation and editing of the controls.

7.9 Authors

Cristina Braga, Helena Borges, Jonas de Andrade, 2008

8. INTERDISCIPLINARY BENEFITS

This pattern language for Interaction Design is still in an early stage of development and is still a work in progress that because of its nature will remain a work in progress. Its main usage so far has been in the Interaction Design course, but it has already been used in other contexts, both in practical project scenarios and in research projects. A masters student of FBAUP developed her thesis on Usability and Accessibility in Web Design, using these patterns as a framework, and steering the development of the project to expand the language with the addition of new patterns and the revision of patterns that are specific to this field, broadening their references, citations, and examples. This work should help to add a substantial number of patterns to the language, bringing local focus and enriching the language.

On a less theoretical and more practical front, the language has been used by the Laboratory of the Design Department of FBAUP during the development of several digital projects, in the course of which the language has been proving its usefulness both as a support tool for the development of the projects and as a general reference in the implementation and maintenance stages. Using the patterns in real-life scenarios has allowed us to test them and to find limitations and areas of possible improvement that otherwise would maybe have passed unnoticed.

The patterns have also been used as references in the Web Design I and II courses at the same school, and have started to be used in both theoretical and practical works developed by the students in those courses. Some of the students are enrolled in both Web Design and Interaction Design courses and this lead them to share the system with their colleagues. The adoption of the patterns by these classes also proved its usefulness in practical design contexts.

9. CONCLUSIONS AND FUTURE WORK

We hope this work in progress can continue to prove its usefulness as a tool for learning, studying and practicing Interaction Design.

As previously noted, this collection of patterns is intended to be a work in progress, not only because of the way its development is tied to an ongoing course but also due to the constant changes in the field it studies — new devices, systems and interfaces need new patterns or force revisions in the existing patterns. The completeness of the collection is one of the main objectives, it is necessary to develop more patterns, especially in the lower-levels, but the completeness of each of the patterns is also an important focus: more examples are necessary, and more specialized and organized examples, maybe related to specific platforms or operating systems but also historic examples.

Another issue to consider is translation. The official language of the courses in FBAUP is Portuguese and the majority of the post-graduate students are either Portuguese or native speakers of Portuguese. The average proficiency in English of the students varies a lot, and there is a sizable percentage of the students to whom the use of English as a working language can be a big problem. We chose to develop this language in Portuguese, knowing in advance that this option would cause two main problems:

(1) The translation of names, keywords and technical terms generates frequent discussion. In most cases, it was decided to keep English names and expressions whenever there wasn't any viable and commonly accepted Portuguese translation. This was the case with such names as “breadcrumb”, “framework”, “wizard”, or “tooltip”. Pattern names have been created in Portuguese except in cases when the pattern name was or included some of these English terms. As Caroline Schubiger points, the discipline of Interaction Design is still “in the process of defining itself and formulating a specific vocabulary” [16]. This is particularly true for us, using English and Portuguese words in a confusing balance, most of the times being uncertain of the reasons why either language is chosen in naming conventions. Through the names of the patterns and the normalization of terms over the collection, this vocabulary can be discussed and we can hopefully contribute to its normalization.

(2) The limitation of the usefulness of the language to the minority of non Portuguese-speaking students in FBAUP or outside the sphere of Portuguese-speaking countries, as some exchange students at both undergraduate and post-graduate levels.

We believe that in order to make this collection as useful as possible, we will need to devise a strategy for its translation and for the maintenance of parallel Portuguese and English versions of all the patterns. This work is perhaps not adequate to the students of the Interaction Design class, but may be interesting to develop in collaboration with exchange students, students from foreign-languages courses — especially if they are studying technical translation. It also remains to test whether it will then be possible to keep ongoing synchronized revisions of both language versions.

Once it is open to unrestricted editing, the language has the potential to involve agents from outside FBAUP, both in other schools of the University of Porto and other universities or from outside the academic circles. This may of course demand a bigger

and more regular user base that can assure the ongoing editorial control of the language.

We think it is still too early to open the registration in the wiki to a broader number of users. A first step may be to try to collaborate with another school, sharing access and editing with its students, which could allow us to test the development of the language with a broader base of users. After this is tested we hope to eventually make editing access unrestricted.

It could be argued that a unified pattern language for all fields of communication design could be possible, should a concerned effort be developed by designers and design researchers. As this pattern language for Interaction Design is being developed, we have often been faced with questions that are relevant and that address problems shared by other fields of design. These have been so far addressed in the context of Interaction Design, but could eventually be formulated in ways that would make them useful in other contexts. The development of both more specific sections of the language (dedicated to Accessibility or Usability, or dedicated to specific media or platforms) or of patterns that can address other fields of communication design, can lead to the necessity of revising not only the macrostructure of the language but also to the development of new or alternative systems for indexing the patterns, bringing the benefits of this approach to other fields of design that have, in some cases, lacked such a methodical and extensive systematization of knowledge.

“Some of the most crucial steps in mental growth are based not simply on acquiring new skills, but on acquiring new administrative ways to use what one already knows.” [Papert’s Principle, quoted in 12]

10. ACKNOWLEDGMENTS

We would like to acknowledge the wonderful inputs, the dedication and careful reviews of our shepherd Christian Kohls, whose contribution to this paper was essential. The discussion and notes of all the participants of the “Software and People” writers’ workshop at PLoP’08 significantly improved the final version of this work. University of Porto’s Professor Ademar Aguiar’s enthusiasm and motivation was essential at all times. Without Joana Cunha’s efforts at FBAUP, we wouldn’t have been able to attend PLoP’08. Our participation in PLoP’08 was supported by FCT – Fundação para a Ciência e Tecnologia.

11. REFERENCES

- [1] Alexander, C. *Notes on The Synthesis of Form*. Harvard University Press, Cambridge, Massachusetts, 1964.
- [2] Alexander, C. *The Timeless Way of Building*. Oxford University Press, New York, 1979.
- [3] Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. and Angel, S. *A Pattern Language. Towns, Buildings, Construction*. Oxford University Press, New York, 1977.
- [4] Appleton, B. *Patterns and Software: Essential Concepts and Terminology*, 2000.
- [5] Behrens, C. *Info Design Patterns*, 2008.
- [6] Borchers, J. *A Pattern Approach to Interaction Design*. Wiley, Chichester, 2001.
- [7] Cullen, M. Future Teaching. in Heller, S. ed. *The Education of a Graphic Designer*, Allworth Press, New York, 1998, 31-37.
- [8] Erickson, T. *The Interaction Design Patterns Page*, 2004.
- [9] Gamma, E., Helm, R., Johnson, R. and Vlissides, J.M. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1995.
- [10] Laurel, B. *Computers as Theatre*. Addison-Wesley, Reading, Massachusetts, 1993.
- [11] Mateas, M. Procedural Literacy. Educating the New Media Practitioner. *On The Horizon. Special Issue. Future of Games, Simulations and Interactive Media in Learning Contexts*, 13 (1).
- [12] Minsky, M. *The Society of Mind*. Picador, London, 1988.
- [13] Nelson, T.H. Computer Lib / Dream Machines. in Wardrip-Fruin, N. and Montfort, N. eds. *The New Media Reader*, The MIT Press, Cambridge, Massachusetts, 2003, 301-338.
- [14] Reas, C. and Fry, B. *Processing: a programming handbook for visuals designers and artists*. The MIT Press, Cambridge, Massachusetts, 2007.
- [15] Riehle, D. and Züllighoven, H. Understanding and Using Patterns in Software Development. *Theory and Practice of Object Systems*, 2 (1). 3-13.
- [16] Schubiger, C. Interaction Design: Definition and Tasks. in Buurman, G.M. ed. *Total Interaction: Theory and Practice of a New Paradigm for the Design Disciplines*, Birkhäuser, Basel, 2005, 341-351.
- [17] Smith, M. and Pollock, C. *Pattern Tap*, 2008.
- [18] Stimmt AG - Uni Basel. *Stimmt AG - Uni Basel Pattern Catalogue*, 2008.
- [19] Tidwell, J. *Designing Interfaces. Patterns for Effective Interaction Design*. O’Reilly, Sebastopol, California, 2005.
- [20] Ulmer, G.L. *Heuristics: The Logic of Invention*. The Johns Hopkins University Press, Baltimore, Maryland, 1994.
- [21] van Duyne, D.K., Landay, J.A. and Hong, J.I. *The Design of Sites: Patterns for Creating Winning Web Sites*. Prentice Hall, Upper Saddle River, New Jersey, 2007.
- [22] van Welie, M. *A Pattern Library for Interaction Design*, 2007.
- [23] Weizenbaum, J. *Computer Power and Human Reason: From Judgment to Calculation*. W. H. Freeman and Company, San Francisco, 1976.
- [24] Wilson, S. *Information Arts*. MIT Press, Cambridge, Massachusetts, 2002.
- [25] Yahoo! Developer Network. *Design Pattern Library*, 2005-2008.