

Patterns in interaction

A review of current works

By
J Rambrijesh (02625010)

Guided by
Prof Anirudha Joshi

June 2004

IDC IITBombay

The special project titled
“Patterns in interaction design: Review of current works” by Rambrijesh is
approved as partial fulfillment of the requirements for the post graduate
degree in Visual communication

Project Guide

Chairperson

Internal Examiner

Industrial Design Centre
Indian Institute of Technology
Bombay

Acknowledgments

I like to thank Prof. Anirudha Joshi for his views and guidance.

Special thanks to Kiran, shalaka, Anu, Anand, Rashmin for their viewpoints and timely support, for his valuable input to the project.

I like to thank my dad, mom and my brother for their constant support.

Thanks IIT for giving access unlimited internet and access to online digital libraries like ACM which gave update information on the subject.

Contents

Abstract.....	01
Interface design projects at IDC.....	02
Patterns Introduction.....	05
Patterns review	
-Patterns in architecture.....	08
-HCI patterns.....	11
-Web design patterns.....	18
Pattern supported approach.....	20
Patterns to pattern language.....	21
Application of patterns in design.....	23
Tail piece.....	28
Bibliography.....	29

Abstract

01

This study attempts to understand the patterns approach for interaction design, tracing its roots from architecture, and looks at how it has been adopted by computer programmers while highlighting the recent interests show by people from the HCI domain.

The paper also tries to establish the merits and limitations of patterns approach and speculates on the use of patterns to aid the 'expert evaluation' in usability testing. It suggests the use of patterns as a means of delivery mechanism in place of specifications and the use of interaction design patterns to enrich the project life cycle.

With study of examples from IDC projects and the web, an attempt has been made to suggest how 'Patterns approach' can be used to understand the fields of user interface design and IxD and its design implications.

Key words

Patterns, Pattern language, User interface design, Expert evaluation, Specifications, HCI.

Interface design projects

At IDC most interface design projects (web design, CBT etc) begin with a broad scope and are only able to accomplish the first iteration.

For many students, it's the first attempt with an interface project, most time is spent up front, understanding markets, positioning the product, making initial concepts, trying out different concepts, learning to make interfaces, finalize one concept and detailing it out. It rarely gets to the stage where it can be fully prototyped and tested for effectiveness in the assumed market. Most projects stay as an academic exercise only.

In the past 5 years there have been about 25 web design projects, almost 15 multimedia/ CBT, 5 mobile phone redesign projects etc. In most of the projects, a few common mistakes are observed and similar navigation is attempted.

The project outputs suggest negligible learning from the previous projects of the same genre. Each time, things are reinvented as if they were done for the first time.

Can there be better learning, by avoiding these common mistakes?
Can the design cycle be accelerated by learning from previous attempts?



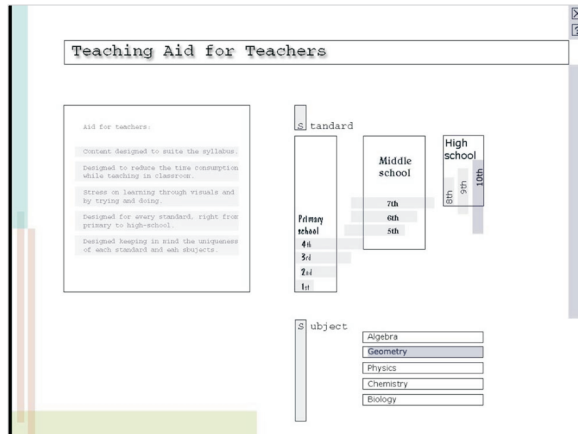
Operating system interface for illiterates and semi-literates



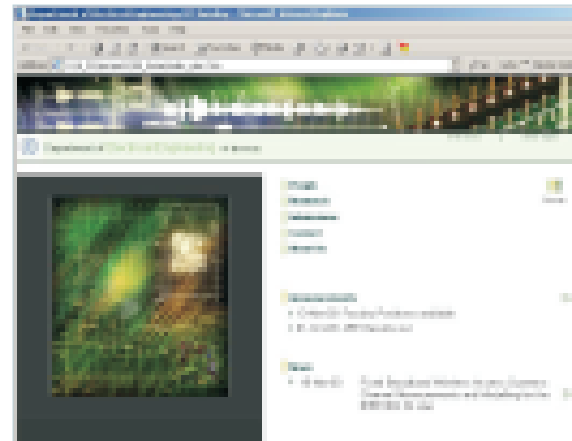
Sharing personal media for children.- an electronic interface.

Interface design projects

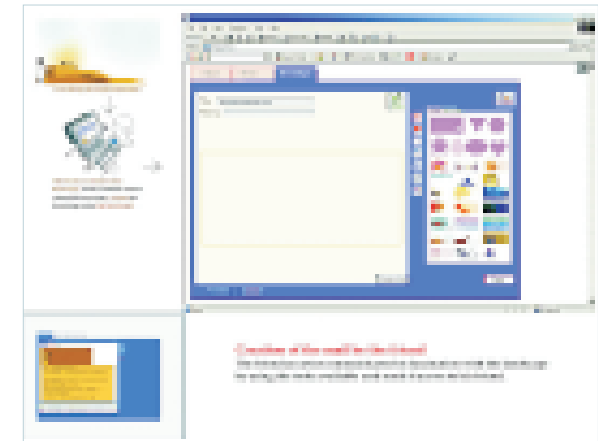
Capturing the user interface design lessons learned from a completed project is crucial to reach a level of finish and avoid repeating mistakes committed earlier.



Teaching Geometry in class - CBT



Web template for engineering colleges.



Personalized emails- conceptual interface.



Redesign of Department of tourism website

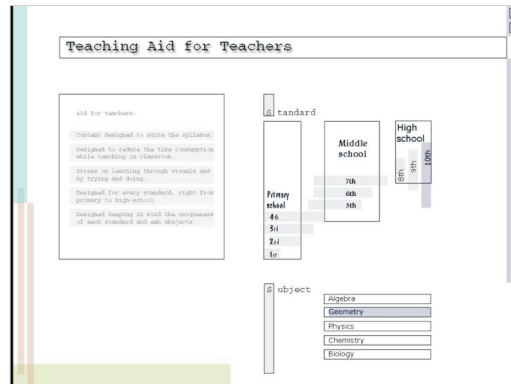


Redesign of Indian railways website.



Information design for eye care.

Common mistakes



There is no doubt that some mistakes proves very valuable for learning... Lets look at some common mistakes done by students in their interface design projects at IDC.

Repeated mistakes

A great deal of time, cost and effort is wasted in taking decisions which have previously proven fruitless.

Reinvent the wheel.

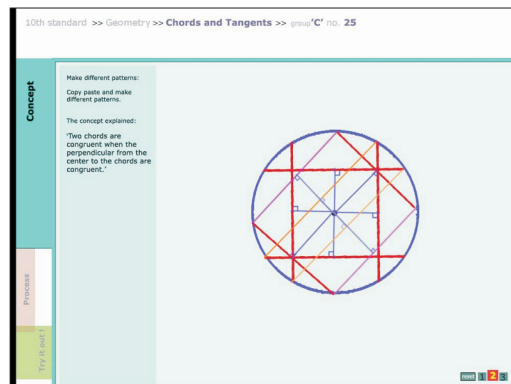
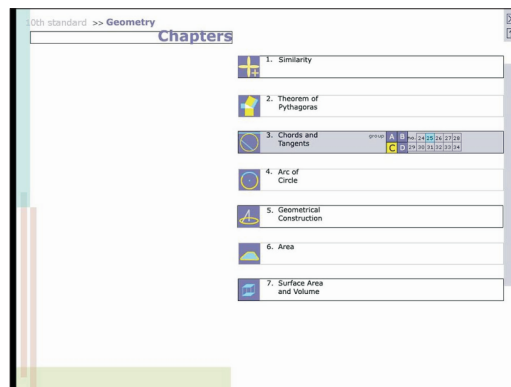
The common forces acting on the given situation are not understood up-front, so we tend to work out solutions similar to previous projects and the wheel is reinvented every time.

The awareness disparity

There are a lot of very effective case examples, but are not widely known, interfaces being a relatively new field. The awareness about successful examples affects the design and thought process of students and the unacquainted students remain deprived. This disparity is significantly visible in the projects

Common knowledge base?

There have been so many mobile phones, CBTs and website projects done, but there is no common knowledge base to refer to. Each time it is 'start from the scratch'!



Is there any ways in which these repeated mistakes can be avoided?

Teaching Geometry in class - CBT

Introduction

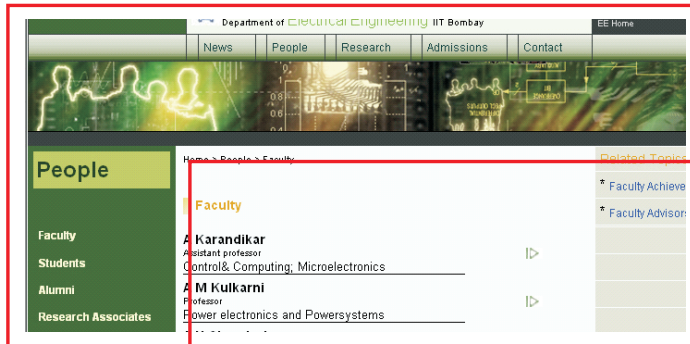
Today's Context:

Design is all about finding solutions. Unfortunately, designers often reinvent things which have already been done in the past. Even if a good example is found, it is hard to know why things were done a certain way.

On the other hand, engineering disciplines have handbooks that describe successful solutions to known problems. For instance, automobile designers don't design cars using the laws of physics every time. Instead, they reuse standard designs with successful track records.

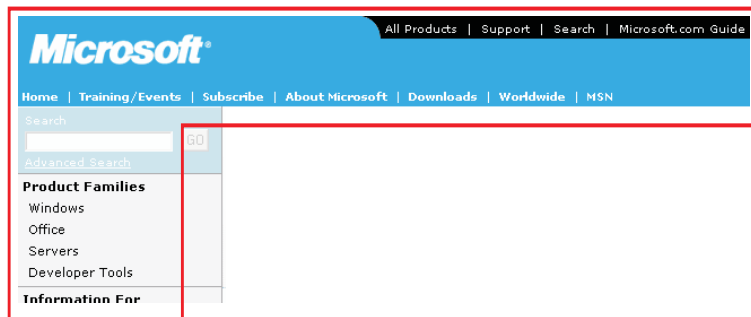
There are numerous discussions worldwide at conferences, in magazine articles, about design methods, processes and effective ways of knowledge transfer. How can the HCI community help inexperienced designers move away from clumsy designs and labor-intensive processes towards the state of confidence, without spending years learning it the hard way?

When we are talking about efficiency, consistency and institutionalizing design, 'patterns approach' seems to offer some potential in these areas. In this paper, we are looking at a 'Patterns approach', tracing its origin from architecture, review the current works and looking at various possibilities offered by this method.

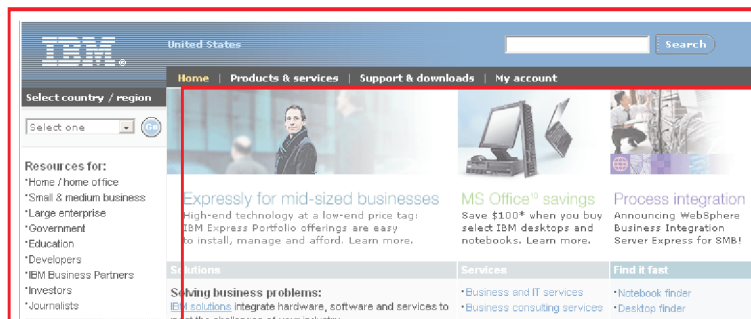


Web template for engineering colleges. (Done at IDC 2003)
A output created after 5 months of exploration.

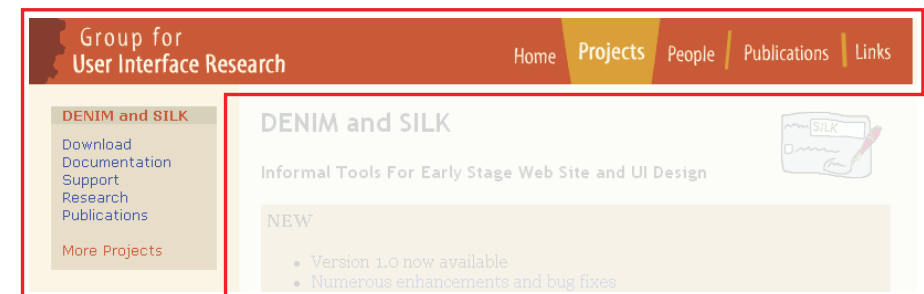
Standard navigation bar



<http://www.microsoft.com>



<http://www.ibm.com>



<http://guir.berkeley.edu/>

Introduction

What are design patterns?

Patterns are pockets of knowledge! ' *Pattern* ' is a format for describing a solution to a design problem- the problems that have occurred time and time again, and have being solved time and again by designers. Patterns describe practical solutions to these problems and a clear picture of how it can be applied in different situations.

Dictionary meaning.

- A model or original used as an archetype.
- A plan, diagram, or model to be followed in making things
- To serve as an example for; also, to parallel.

An interaction design pattern is a design pattern in the discipline of HCI. They document proven solutions for recurring design problems in a systematic and understandable way.

HCI Design Patterns have many advantages:

- 1) Patterns can be used to effectively teach design fundamentals to newcomers into the HCI field (ex. For UI design)
- 2) Patterns can reduce complexity and enable the easier understanding of a given problem.
- 3) Capture the best practices for follow-up projects.
- 4) Inform and establish a common vocabulary within the current user interface design team for a project
- 5) Provide a valuable source of information, supporting both the analysis of the current situation and the design of the new system.

They are presented consistently, are easy to read, and provide background reasoning. The format provides information about the problem at hand, the context, a solution and also the **Rationale** behind the solution.

Patterns effectively share information, highlight design rationale, capture sound practice, and provide a standardized design vocabulary.

Application areas:

The application of this pattern is broad: wherever there is an interaction between a person and a software artefact, patterns can be used. Patterns support a broad spectrum of activities, ranging from the most absorbing interactive products to products with no interactivity.

Websites, game development, usability reviews/ testing, user interface design (software interfaces, speech interfaces, multimodal interfaces), computer based training (CBT), online manuals, control panel designs, 3d environments, collaborative spaces, kiosks designs, e-learning initiatives etc....its application areas are really endless.

To relate to patterns, let's trace back to its origin in architecture and see some popular patterns works, Jennifer Tidwell's HCI patterns collection-common ground, and wilie's Amsterdam web design patterns.

Patterns in architecture

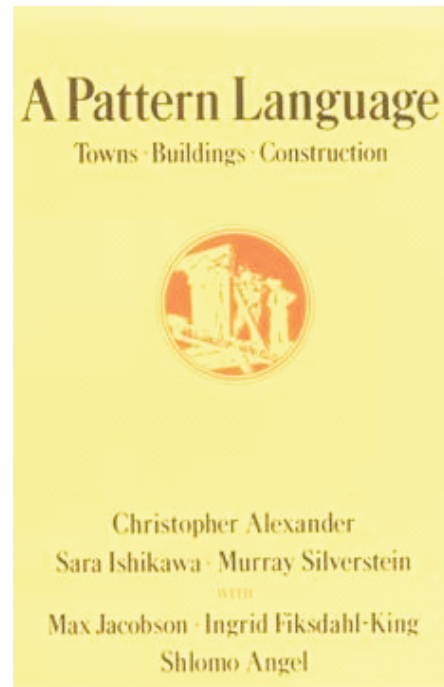
08

Early attempts in patterns

In 1970's, Christopher Alexander compiled his landmark book '*The Timeless Way of Building*'. His thesis was that one could achieve excellence in architecture by learning and using a carefully-defined set of design rules, or patterns. He also states that thought, a quality of a well-designed building is hard to put into words, the patterns which make up that building are remarkably simple and easy to understand by laymen.

The patterns that he and his colleagues defined (published in 'A Pattern Language') are an attempt to codify generations of architectural wisdom. They are not abstract principles that require you to rediscover how to apply them successfully, nor are they overly specific to one particular situation or culture. Instead, they are somewhere in between. A pattern describes possible good solutions to a common design problem within a certain context, by describing the invariant qualities of all those solutions.

"Each pattern expresses a relation between a certain context, a problem, and a solution."



Pattern language Christopher Alexander

09

Alexander's intention was to capture the essence of successful solutions to recurring design problems in architecture and to create a language. This language gives the person who uses it, the power to create an infinite variety of new and unique buildings, just as his ordinary language gives him the power to create an infinite variety of sentences.

The speciality of these individual pattern is that it talks about a problem which occurs over and over again in our environment, then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same ways twice.

His patterns gives us a impression that they are prescriptions for making buildings. But his patterns describe a situation in a broad manner, he talks about various forces acting on the building. Yet he retains the essence of the specific example, give an holistic idea of handling similar situation.

To quote him *"Patterns provide design solutions that are concrete enough to immediately put into practice, with good results, and yet are sufficiently abstract to apply to countless situations, limited only by the imagination and skill of the pattern user."*

He wrote 253 interconnected patterns Ranging from Region, town -> neighborhoods -> cluster of buildings -> buildings -> rooms -> alcoves -> details of construction.

Patterns in architecture

All patterns were described in this fashion:

Picture- shows an archetypal example.

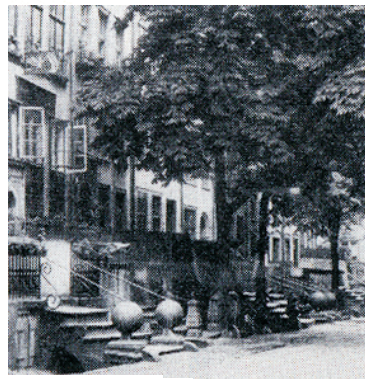
Introductory paragraph- set the context for the pattern/ explains how it can help complete a larger patterns

Essence of the problem 2-3 lines

Body of the problem- empirical background, evidence of its validity, different ways of manifestation so on.

The solutions- the heart of pattern, explains the physical & social relationships that needs to be built to solve the problem in the context.

Solution diagram- shows solution with main components labeled.



Example: "Entrance Transition"

"thrives on a sense of exclusion from the world."

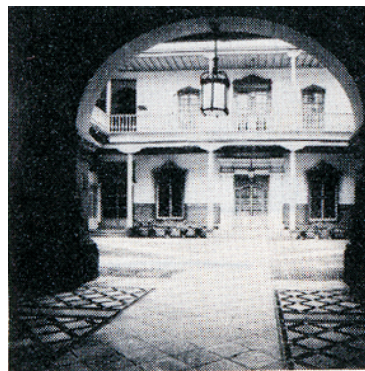
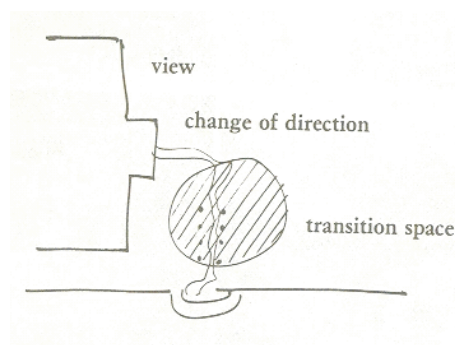
"Make a transition space between the street and the front door. Bring the path which connects street and entrance through this transition space, and mark it with a change of light, a change of sound, a change of direction, a change of surface, a change of level, perhaps by gateways which make a change of enclosure, and above all with a change of view."

Some more examples:

Child caves : children love to be in tiny, cave-like places **Connected play:** If children don't play enough with others children during the first five years of the life, there is a large chance that they will have some kind of mental illness.

Intimacy gradient : Unless the spaces in a building are arranged in a sequence which corresponds to their degree of privateness, the visits made by strangers, friends, guests, clients, family will always be a little awkward.

Waist high shelf, six- feet balcony, Flexible office space



From Architecture to HCI

Software engineering picked up the patterns concept from architecture. Their community identified a large number of software design patterns, collated them, discussed them in conferences and published.

Their main interests in patterns were their reusable nature. It also offered them platform for documenting and knowledge transfer. This helped them to create 'usable' software consistently and faster.



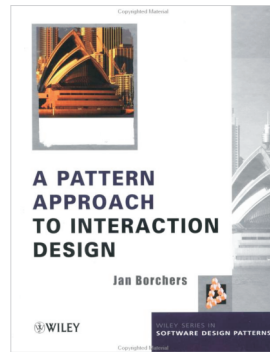
"The pattern discipline has become one of the most widely applied and important ideas of the past decade in software architecture and design."

-James Coplien

Timeline

Taking off from the Alexanderian patterns, there were attempts made by other fields as well to document their knowledge base

Interaction design patterns differ from software design patterns. Interaction design patterns are describing problems end-users have, as opposed to problems encountered by system architects and coders. Interaction design patterns are centered around users. There are different formats of interaction design patterns have been proposed, lets look at a few to get a historic perspective, in the timeline chart.



Todd Coram and Jim Lee 1996

One of the earliest work, follows the same structure and sections as in Alexander. They presented papers about patterns in software Programming conference, 1996.

Jan Borchers 1999

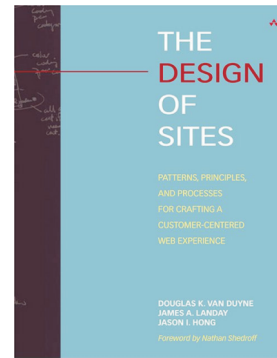
An HCI pattern presented in pure Alexandrian form. He went about making a best seller Book with these patters called "Design patterns"

Jenifer Tidwell May 1999

Jenifer has developed one of the earliest, largest-scale and most influential HCI pattern collections, Common Ground.

September 1999

The Usability Group started making usability patters



Jason Hong& Landay April 2000

A purely narrative form. (This is an early version which has subsequently been developed and extended, and compiled in a book: "The Design of Sites": Patterns, principles and processes for crafting a customer-centered web experience

Welie April 2000

Simple descriptions, with examples- his Design Patterns page "Amsterdam Patterns Collection" is now expanding, covers web, GUI & mobile.

Jenifer Tidwell, May 2002

UI Patterns and Techniques.

Sari Laasko, Sep 2003

User Interface Design Patterns

Please refer to last section- References for the weblinks

Tidwell's HCI patterns

Tidwell (in 1999) was the first to make an elaborate human-computer interface pattern language titled "Common grounds". Common Grounds is a set of interrelated patterns, provides powerful and generic design guidance, which convey knowledge about good design. . It is written in a format that is consistent and easy to read and understand,

These patterns were intended as a language that would aid individual interface designers in their day-to-day work.

They were described in the following format: Example, context, forces, solutions, relationships.

She identified some 80 patterns and she organized its collection around the following categorical questions:

- What is the basic shape of the content?
- What is the basic shape of the actions taken with the artifact?
- How do the content or available actions unfold before the user?
- How does the artifact generally use space and the user's attention?
- How is the content or action organized into working surfaces?
- How can the user navigate through the artifact?
- What specific actions should the user take?
- How can the user modify the artifact?
- How can the artifact be made visually clear and attractive?
- How else can the artifact actively support the user?

Tidwell's HCI patterns

Lets look at an example of a pattern set, which contains patterns which belongs to same context.

How do the content or available actions unfold before the user?

Navigable Spaces is highly interactive, letting users move through the artifact at their own pace, and in their own directions.

Overview Beside Detail also lets users work at their own pace, but presents a more structured, two-level view of information or actions.

Step-by-Step Instructions prescribes a sharply limited set of actions to the users, usually in a linear fashion, moving progressively through stages.

Small Groups of Related Things loosely organizes visual content into distinct groups, usually done hierarchically, so that a user can see both the "big picture" and fine detail.

Series of Small Multiples organizes visual content into a single (possibly long) series of discrete images, bringing out differences between them by emphasizing their basic continuity; it can be used to implement a High-Density Information Display.

Hierarchical Set is a way of implementing a High-Density Information Display with a strict tree-like organization.

Tabular Set also implements High-Density Information Display, but as a table; again, users can see both the big picture and fine detail.

Chart or Graph is yet a third implementation of High-Density Information Display, but a more graphic one than the other two.

Optional Detail On Demand lets users reveal hidden actions or content at their own discretion.

Disabled Irrelevant Things blocks certain actions or content according to the current state of the artifact, usually driven by user interaction.

Pointer Shows Affordance temporarily reveals possible actions according to the user's explicit focus of attention.

Short Description temporarily reveals content according to the user's explicit focus of attention.

To understand each of the patterns, lets look at two examples of step by step instructions and Map of Navigable Spaces

Tidwell's HCI patterns

The patterns contained in Tidwell's work address the general problem of how to design a complex interactive software artifact. She has grouped them based on the actions which can be performed on or with the artifacts. Looking at all the interrelated patterns often works well to support the individual pattern.

What is the basic shape of the content?

- Narrative
- High-density Information Display
- Status Display

What is the basic shape of the actions taken with the artifact?

- Form
- Control Panel
- WYSIWYG Editor
- Composed Command
- Social Space

How does the content or available actions unfold before the user?

- Navigable Spaces
- Overview Beside Detail
- Step-by-Step Instructions
- Small Groups of Related Things
- Series of Small Multiples
- Hierarchical Set

- Tabular Set
- Chart or Graph
- Optional Detail On Demand
- Disabled Irrelevant Things
- Pointer Shows Affordance
- Short Description

How does the artifact generally use space and the user's attention?

- Sovereign Posture
- Helper Posture
- Background Posture

How is the content or action organized into working surfaces?

- Central Working Surface
- Tiled Working Surfaces
- Stack of Working Surfaces
- Pile of Working Surfaces

How can the user navigate through the artifact?

- Map of Navigable Spaces
- Clear Entry Points
- Color-Coded Sections
- Go Back One Step
- Go Back to a Safe Place

What specific actions should the user take?

- Convenient Environment Actions
- Localized Object Actions
- Actions for Multiple Objects
- Choice from a Small Set
- Choice from a Large Set
- Sliding Scale
- Editable Collection
- Forgiving Text Entry
- Structured Text Entry
- Toolbox

How can user modify the artifact?

- User Preferences
- Personal Object Space
- Scripted Action Sequence
- User's Annotations
- Bookmarks

How can the artifact be made visually clear and attractive?

- Iconic Reference
- Calm Grid
- Repeated Framework
- Few Hues, Many Values

How else can the artifact actively support the user?

- Good Defaults
- Remembered State
- Interaction History
- Progress Indicator
- Important Message
- Reality Check
- Demonstration
- Quick Access
- Familiar Quantity

Tidwell's HCI patterns

Example: Step by step instructions



Possible usage scenarios:

Wizards, Installation instructions, Recipes, Repair manuals , Getting cash from an ATM

Context:

A complex task,
Limited time/ knowledge/ attention or space.
it's meaningless to show all the action possibilities at once.

Problem:

Unfold in a way that does not overwhelm or confuse them,
- instead guides them to a successful task completion?

Forces:

- Requires choices/information from the user,
- task can't be performed automatically,
- user want/need to understand what he is doing.
- afraid of doing something wrong
- prefer that the actions to be explicitly spelled out.

Solution:

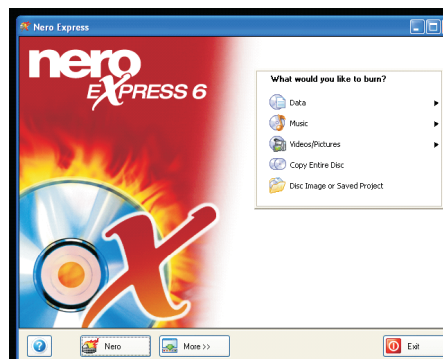
- Walk the user through the task one step at a time,
- giving very clear instructions at each step.
- Use visual similarities in all the steps
- maintain the user's sense of flow
- keeping it short



Geocities "web page wizard" online application



IDBI ATM interface



Nero Express CD Writing application

Tidwell's HCI patterns

Example: Map of Navigable Spaces

Possible usage scenarios:

Book's table of contents, Web site map, Geographic map, Selection area for a suite of applications, as with the PalmPilot or a network computer, Windows Explorer

Context:

The artifact (or its content) can be organized into distinct spaces which are semantically linked to each other

Problem:

How can the artifact help a user navigate effectively and remain oriented?

Forces:

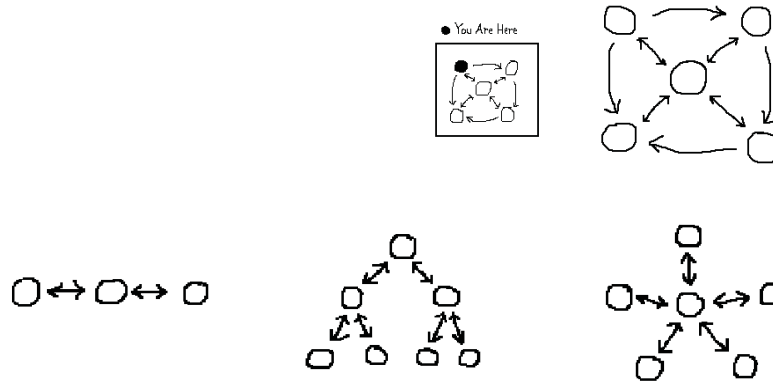
- user should know where they are at any time
- user wants to know where they can go next, & how to get there.
- user should be able to see the interrelationships between the Navigable Spaces & also the overall structure.
- user may need a way to jump from place A to B, but A may not have a direct link to B

Solution:

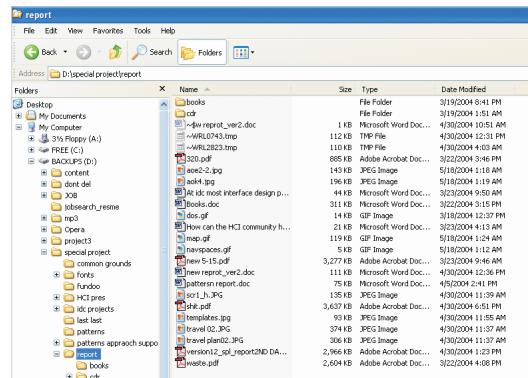
- Provide a map or diagram of the *Navigable Spaces* relevant to the artifact.
- If the spaces are related via a hierarchy, use a *Hierarchical Set*
- if they form a flat set, show them in a linear fashion
- Some sets of spaces there are so large that you don't want to show them all at once. Optional Detail On Demand may also help.

Notes:

- For Web sites and computer interfaces, if you use the common idiom of putting the map on the left side of the screen and a view onto the chosen "space" on the right! It works well, and many people are familiar with it by now.



Screenshot from 'Age of empire'



Screenshot from 'Window Explorer'

Wilie compiled these web site patterns as a part of this Phd and published it in his website. Its popularly know as the Amsterdam design patterns. Given below are the lists of patterns he has identified, the list is appended as and when new patterns are identified.

Web Design patterns

Site Types

- [My Site](#)
- [Portal](#)
- [Commerce Site](#)
- [Community Site](#)
- [Branded Promo Site](#)
- [Corporate Site](#)
- [News Site](#)
- [Museum Site](#)
- [Automotive Site](#)
- [Web-based Application](#)
- [Travel/booking Site](#)
- [Multinational Site](#)
- [Artist Site](#)

Navigation

- [Main Navigation](#)
- [Bread crumbs](#)
- [Double tab](#)
- [Meta Navigation](#)
- [Split Navigation](#)
- [Repeated Menu](#)
- [Faceted Navigation](#)
- [Teaser Menu](#)
- [Header-less Menu](#)
- [Fly-out Menu](#)
- [Directory](#)
- [Trail Menu](#)
- [Scrolling Menu](#)
- [Shortcut Box](#)
- [Image Menu](#)
- [Guided Tour](#)
- [Image Browser](#)
- [Minesweeping](#)
- [Overlays](#)
- [Doormat Navigation](#)
- [Outgoing Links](#)
- [Panning Navigator](#)
- [Map Navigator](#)

User Experiences

- [Shopping](#)
- [Community](#)
- [Learning](#)

Searching

- [Simple Search](#)
- [Advanced Search](#)
- [Search Results](#)
- [Search Area](#)
- [Sitemap](#)
- [Topic Pages](#)
- [Search Tips](#)
- [Site Index](#)
- [Help Wizard](#)
- [FAQ](#)

Ecommerce

- [Shopping cart](#)
- [Login](#)
- [Registering](#)
- [Product Comparison](#)
- [Product Configurator](#)
- [Product Advisor](#)
- [Premium Content Lock](#)
- [Newsletter](#)
- [Case study](#)
- [Booking process](#)
- [Store Locator](#)
- [Virtual Product Display](#)

Basic Page Types

- [Homepage](#)
- [Contact Page](#)
- [Processing Page](#)
- [Printer-friendly Page](#)
- [Article Page](#)
- [Blog Page](#)
- [Product Page](#)
- [Forms](#)
- [Input Error Message](#)

Managing Collections

- [Column Filter](#)
- [Sortable Table](#)
- [List builder](#)
- [Favourites \(Collector\)](#)
- [Parts Selector](#)
- [Sorting](#)

Visual Design

- [Alternating Row Colours](#)
- [Grid-based Layout](#)
- [Colour-coded Areas](#)
- [Liquid Layout](#)
- [Font Enlarger](#)
- [Center stage](#)

Page Elements

- [News box](#)
- [Home](#)
- [Language Selector](#)
- [Country Selector](#)
- [Hotlist](#)
- [Customization Window](#)
- [Poll](#)
- [Footer Bar](#)
- [Forum](#)
- [News ticker](#)
- [Constrained Format](#)

Basic Interactions

- [Tabbing](#)
- [Paging](#)
- [Wizard](#)
- [Enlarged Clickarea](#)
- [Action Button](#)
- [Stepping](#)

A pattern written about SHOPPING experience.

Shopping Experience



[Add to Wishlist](#)

From www.bn.com

- Problem** Users want to look for products of interest and potentially purchase them
- Use when** You are building a web site where you sell products, typically an [E-commerce Site](#) site but it can also be a site with paid content. The sort of products that you are trying to sell may vary a lot, ranging from books, electronics, to holiday and clothes. Some products can be delivered directly by downloading it and others will have to be delivered 'later' by some logistical process. No matter what product you are trying to sell, there are well known aspects to shopping that apply to all products and to all ways of shopping.
- Solution** **Create an online shopping experience that matches off-line shopping experiences**

Shopping involves several fundamental activities that apply to both online and offline shopping activities. These activities needs to be supported for each type of product and domain. How to do that best is largely domain dependent, but some basic ideas can be defined:

 - **Discovering.** People need to know what they can buy in the store, as far as they don't already know it. Even if they have been in the store before they need to be informed of new products that are for sale. Even if there are no new products to sell, there may be products that should be brought under the users attention because of other reasons e.g. because they are discounted, very popular etc. Use [Hotlist](#)
 - **Browsing.** Most people like to browse through the store for seeing what they have and whether something attracts their attention. Browsing is made easier when products are categorized in ways that customers expect them to be. The categories allow them to browse in a specific manner that is a bit more directed than no structure at all. Use structured navigation such as a [Double Tab Navigation](#) with [Breadcrumbs](#) so that people are fully aware of where they are and where they can go to.

<http://www.welie.com/patterns/showpattern.php?patternid=shopping>

- **Comparing.** Often people do not know exactly which product they want. They may have several options that they want to compare using a [Product Comparison](#) or [Product Configurator](#).

- **Trying.** When people try a product they want to make sure it is the right product for them. Trying is all about 'seeing' certain aspects of the product. In many cases it is even possible to 'interact' with the product by 'virtually touching it', seeing close-ups, table of contents or a preview of a part of the object. Sometimes it may also be possible to try the real thing with some limitations on the use of it. In other words, create a [Virtual Product Display](#)

- **Asking Opinions.** Many shops have shop assistants that help customers to find the right product for them. Online this is difficult to achieve but one could create [Product Advisor](#) or collect recommendations/ratings/comments of other people that bought the product.

- **Choosing.** Choosing is not the same as buying. Customers may choose several products and before they actually start buying, discard several of them at the last minute. Give them a place to keep products they may want to buy such as a [Shopping Cart](#) or wish list

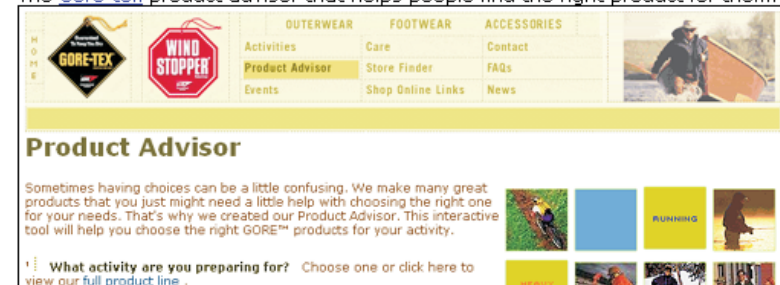
- **Recommending.** Many times when people shop they do not find anything for themselves but they find something that might interest a friend or relative. In that case they want to recommend it to others or letting others know about the existence of the product. Use [Send-a-Friend Link](#) or list of recommended products.

Besides finding a product people like, there are other factors that give people greater satisfaction with their purchase:

- Having struck a bargain
- The cheapest price
- A unique product
- The product comes with exceptional service
- The product is more durable
- The product is quickly delivered

Why People know the off-line shopping experience very well. The essentials of shopping should be taken into account for online shopping as well since they have little to do with the medium itself. The goal is to find the appropriate way to sell particular products in the web while paying attention to all aspects of the shopping experience.

More Examples The [Gore-tex](#) product advisor that helps people find the right product for them:



Pattern supported approach

Patterns not only focus on screen design issues but can look at the overall user-oriented interface design process. The PSA (Pattern Supported Approach) addresses patterns not only at the design phase, but also 'before' design phase. Most UI design patterns tend to be general, and larger in number but domain specific patterns are very few specific in nature, not lot of work is done in this area.

Here is a brief overview of the PSA patterns:

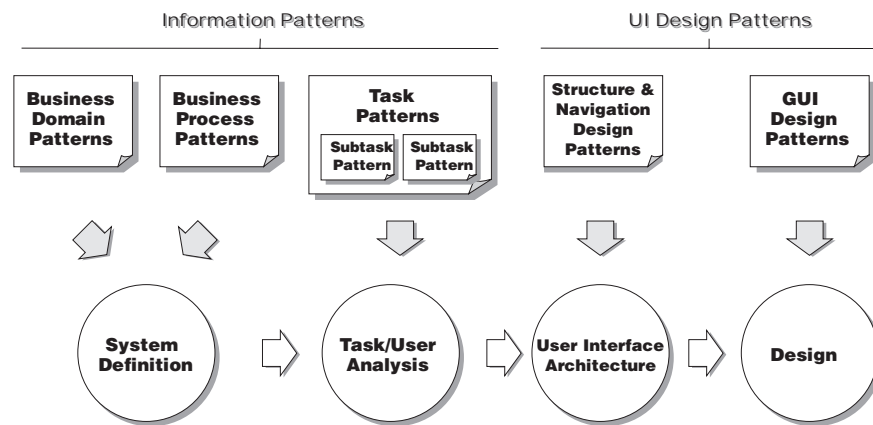
-Business Domain Patterns describe the type of business, its goals, plus the typical actors and business processes involved. They provide a starting point for initially defining the system design by pointing to relevant Business Process patterns and thereby to Task patterns. They help communicate the System Vision.

-Business Process Patterns describe typical processes and actors involved in the delivery of services/goods in compliance with the business goals. They narrow down the system definition and point to specific Task patterns to be considered.

-Task Patterns are used for capturing and passing on knowledge about the task, typical users, and their work context from previous similar projects, and for suggesting an appropriate interaction design solution. They point to Structure & Navigation Design patterns that describe solutions that have proven suitable for the task type in previous designs.

-Structure & Navigation Design Patterns describe ways to structure information and implement navigation in order to support the user's task. This design is based on the information described in the Task patterns.

-GUI Design Patterns document GUI design issues based upon information described in the Task Patterns and Structure & Navigation Design Patterns. They are based on the work of Tidwell (1999).



The PSA Framework

Patterns to pattern language

Patterns can exist in all scales, patterns can be placed in a clear hierarchical order.

The effectiveness of a set of networked patterns is much higher than its constituent individual patterns. A quick overview of the specific domain is obtained by glancing through the pattern network.

A large, well-organized collection of design patterns can work together as a kind of language with a rich set of characteristics and its own grammar.

The figure below only shows a graph centered around shopping experience. A complete graph could have more than 200 connected patterns. The different levels and associated patterns can be shown in a graph of connected patterns.

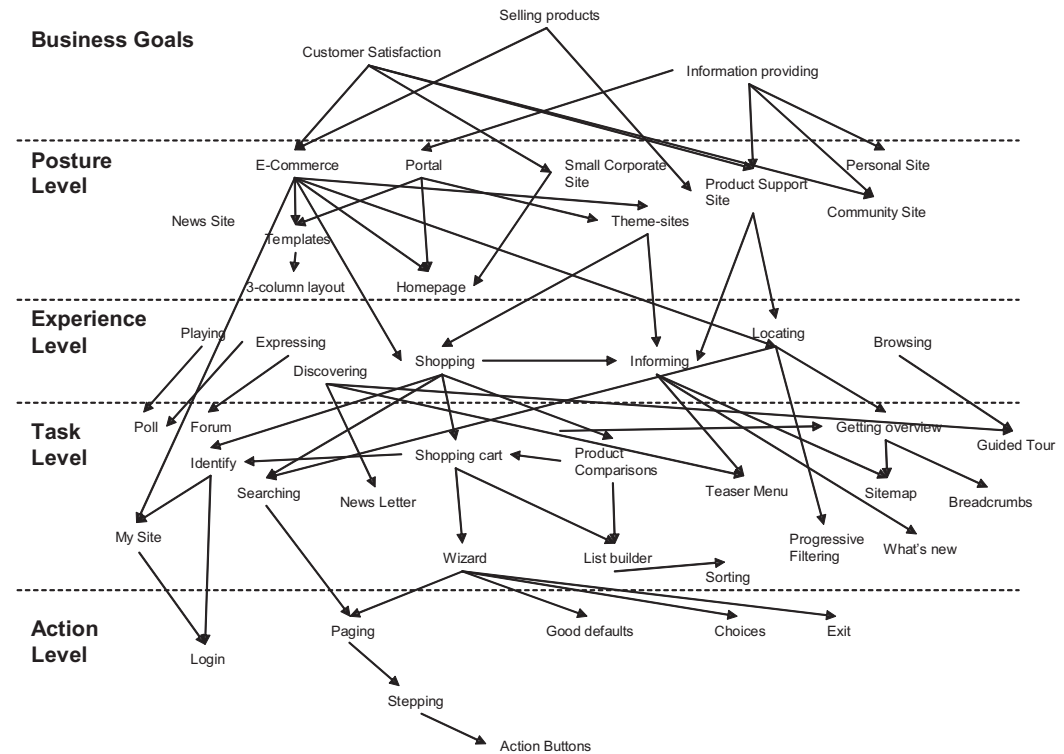
Network of patterns, give a quick overview

the links between the patterns are almost as much a part of the language as the patterns themselves.

-Christopher Alexander

“Relationships are at the heart of the pattern language because they create actual additional value over single patterns.”

-Christopher Alexander



A partial pattern language for web design (centred around "shopping")

(Source: wilie's patters)

Patterns, Guide lines, Style guide 22

The most common misconception about patterns is that, it is very similar to guidelines and style guide. Patterns are not as specific as style guides or guidelines, nor are they as general as a design principle; they can be placed somewhere in-between. To demystify this, let us look at what exactly are guidelines and style guides.

Guidelines:

Ex: Macintosh human interface guidelines or Microsoft XP visual guidelines

These are too much tailored to the use of a certain set or toolkit of use interface objects.

Guidelines forms an instant opinion about look and feel, character, values advocated, some standards, and corresponding intentions too. More than just colors and logos, they tell about identity and overall character of the product

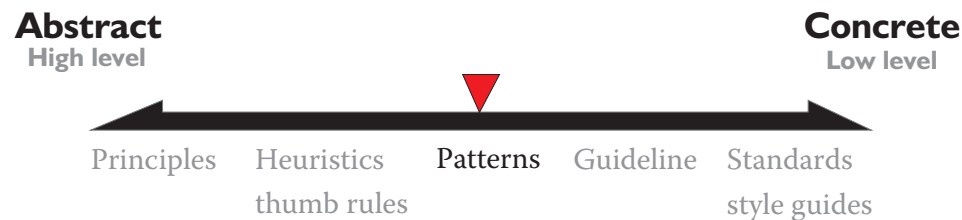
this guide contains the latest information examples of best practice. These examples are intended to offer new perspectives on the way our visual identity can be applied.

Style Guide

A publication which specifies details of writing style, visual treatment required by a particular company or professional organization, including such matters as punctuation, capitalization, and rules for citing references, tone, colour pallets etc

Style guides, are a very efficient medium to ensure consistency within a single user interface environment. They are concrete, and constructive, but do not carry over well to a different environments.

HCI pattern languages are not Style Guides. HCI pattern languages are not Golden Rules not guidelines. HCI pattern languages are not standards. But HCI pattern languages are about Human Computer Interaction, its all about defining a context very clearly.



Application of patterns in design

As discussed earlier, patterns as a medium to capture ideas and solutions within the domain is essential to make better systems & interfaces. Having understood the various kinds of patterns, let us now see how it can be applied to design.

It can help in familiarizing with a few successful methods; UI patterns cuts down the learning time up-front and lets them concentrate on the essentials (design).

Its more like a ready made case study sheet with logics explained. It gives students clear understanding of the contexts, problems and solutions, with a series of successful examples to relate to.

The following pages discuss the following issues.

- Can patterns support creative thinking?
- Can complement expert reviews?
- Can be used as a specifications?
- Role of patterns in a typical design cycle.

Patterns Support Creativity

Most of us think that patterns approach kills creativity. Is building on the experiences of others and reusing existing design knowledge is a ``disregard for originality''?

These are some points for us to consider:

Using a set of pattern collections to learn about user interface design could help designers quickly relate to many of those patterns which are relevant to their design projects. This helps a thorough understanding of concrete problems that have been tackled earlier, with successful examples.

Patterns approach breaks down a complex problem (abstract) into its various components (concrete), on which the designers can work on the crucial and unique problems.

Patterns are a valuable source of information, supporting both the analysis of the current situation and the design for new avenues.

Patterns give a quick platform on which one can start growing, providing mature creative solutions.

Patterns have a unique characteristic of not being specific, limiting or restricting and yet provide a space for creative application.

“Patterns provide design solutions that are concrete enough to immediately put into practice, with good results, and yet are sufficiently abstract to apply to countless situations, limited only by the imagination and skill of the pattern user.”

Finally, I like to mention that our minds is a medium which gives creative sparks that jumps between the patterns and the given situation in hand, to arrive at the optimal solution for given situation.

Heuristic evaluation

Heuristic evaluation is a form of usability inspection where usability specialists judge whether each element of a user interface follows a list of established usability heuristics.

A heuristic is a guideline or general principle or rule of thumb that can guide a design decision or be used to critique a decision that has already been made. To aid the evaluators in discovering usability problems, there is a **list of heuristics** which can be used to generate ideas while critiquing the system.

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

(from Jakob Nielsen and Robert L. Mack, editors, Usability Inspection Methods.)

Expert evaluation

Expert evaluation is similar to heuristics evaluation, but does not use specific heuristics.

Patterns to compliment Expert evaluation

The purpose of any evaluation is to uncover usability problems. Instead of 9-10 very basic predefined heuristics, if the criteria were reviewed from time to time based on patterns relevant to the interface in consideration, it would result in a better evaluation of the system.

Heuristics are meant to guide one in finding the problems, and patterns can support much more robust, relevant and accurate evaluation. Once all the expert comments are collected, the design team can determine solutions based on comments and referred patterns.

Design process

Overview of typical project cycle.

A project cycle can be divided into a few key phases- initial data collection phase, analysis phase, iterative design phase and delivery phase.

The analysis phase consists of user study, through user interviews, surveys and comparative market research etc. There is a range of informal and formal methods that can be used to get a clear understanding of the context.

Once the analysis phase is done, and the goals & objectives are clearer, the iterative design phase starts. It consists of brainstorming, ideation, making rough prototypes, user testing and feedback. At the end of the iteration design phase, a finalized product is arrived at.

In the delivery phase along with the final design solution, interface design specifications are produced. This document is written so that the team members have a working document to incorporate changes and to guide the development on the product on a later date if required.

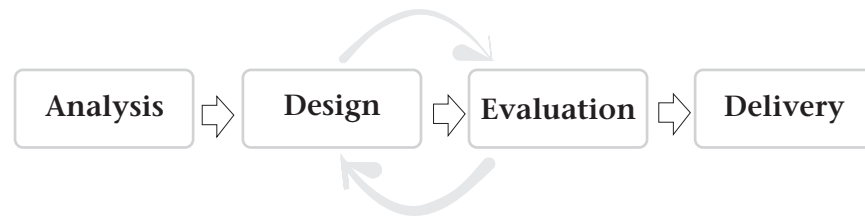
Use of patterns can enhance project cycle:

In most real life projects, the scope changes in a matter of a few years. The product may have altered requirements such as additions, modifications or redundancy. In any case a redesign is called for.

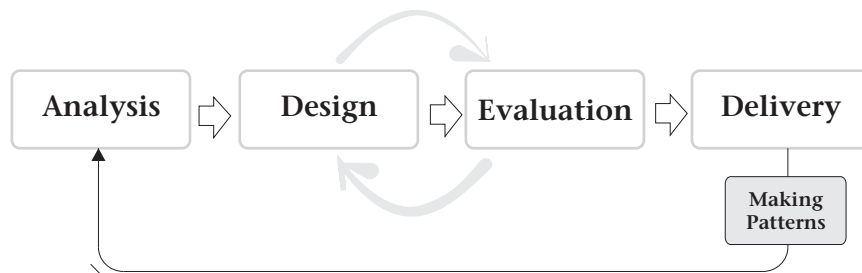
The specifications document as part of the final design solution is inadequate to help the redesign attempt as they are mere description of how to make changes to the product. It does not cite any of the design rationale or the tradeoffs considered.

Patterns approach can complement design specifications:

Patterns, on the other hand, can give a design lead since it can highlight the rationale and tradeoffs made. Hence can loop back into the analysis phase directly, thereby reducing the time taken for redesign while carrying along the merits of the previous iterations.



Typical project cycle



Pattern supported project cycle

Some hurdles.

Application of patterns is a skill by itself. It is not a ready made tool to create wonders. Even though it is evident patterns are useful, there is still no precise way of their application.

Patterns identification

“Architecture connects people to their surroundings in an infinite number of ways, most of which are subconscious.”

Process of pattern discovery is very tough, be it architecture or IxD. Most patterns are based on successful working examples and systems. It's very easy to point-out mistakes in a system, but is very tough to clearly establish something that is working fine. In Alexander's own words, “identifying good patterns is as hard as doing nuclear physics”.

Pattern writing

Once patterns are identified, **pattern writing** is another challenging activity. It needs to have the right kind of abstraction and yet be specific enough to associate with the project.

It will take a while before mature collections will be available. Mature patterns come only by iteration and refining; patterns needs to be refined as per changing times. Apart from patterns which are based on core principles of interactions, patterns based on specific contexts need to be redefined as per changing contexts. The patterns framework should be validated from time to time, additions made and revised.

Patterns presentation

Once we have a considerable amount of patterns in a particular domain of work, we should try cross referencing and grouping them to form a language. Pattern network would be better way of storing patterns rather than isolated patterns.

The pattern network thus created would offer a scope for having **multiple organization and grouping** of individual pattern collections.

Patterns by Alexander and Tidwell are lengthy and elaborate; if they have to take strides into the design community, they should be **precise**, and understandable by professionals and non-professionals alike, **enriched with examples**.

"A pattern language is nothing more than a precise way of describing someone's experience..."

-Christopher Alexander

Further:

Apart from project and research done on various subjects at IDC, there is a need to develop a system of storing design rationale and logics as patterns in specific domains and our cultural context for further reference and use.

We should also support global efforts towards codifying and refining our knowledge in the field of patterns, since HCI design patterns are put in the public domain. They are iterated and refined by public forums and practitioners.

Imagine a situation where all design knowledge is codified or captured as patterns. Can patterns then be used as a specification tool? It could also help in evaluations based on relevant patterns.

Tail piece:

Pattern-based design is a relatively recent approach that needs further research.

However, the thoughts presented in this paper are preliminary. This study does not recommend a shift in the way we design, but to point out the advantages, and versatility of the tool. The few speculations discussed are not tested.

There were some unanswered questions at various levels:

What specific kind of project would benefit most from the usage of patterns? Exactly how and when to use patterns?

How to use combinations of patterns?

How to validate the patterns used?

Is it feasible to create an exhaustive universal pattern catalogue which can be applied to users of different background pursuing different situations and goals?

Bibliography

Books:

Alexander, C. *The Timeless Way of Building*. New York: Oxford University Press, 1979.

Alexander, Christopher, Sara Ishikawa, Murray Silverstein, Max Jacobsen, Ingrid Fiksdahl-King, and Shlomo Angel. *A Pattern Language*. New York: Oxford University Press, 1977.

Cooper, Alan. *About Face 2.0: The Essentials of User Interface Design*. Foster City, CA: IDG Books Worldwide, 2003.

E-Books

Åsa Granlund., Daniel Lafrenière., and David A. Carr., *A Pattern-Supported Approach to the User Interface Design Process*, appeared in Proceedings of HCI International 2001 9th International Conference on Human-Computer Interaction, New Orleans, 2001.

Web links:

Common Ground:
A Pattern Language for Human-Computer Interaction.
http://www.mit.edu/~jtidwell/common_ground.html

Jan Borchert's HCI patterns page
www.hcipatterns.org

Amsterdam Patterns Collection. Martijn Van Welie
<http://www.welie.com/patterns>

the Diemen pattern repository- open source.
[Http://www.visiblearea.com/cgi-bin/twiki/view/Patterns/About_design_patterns](http://www.visiblearea.com/cgi-bin/twiki/view/Patterns/About_design_patterns)

Resources related to pattern languages for interaction design
http://www.pliant.org/personal/Tom_Erickson/InteractionPatterns.html