DVS-IS 2006 Conference Presentation

PATTERN-ORIENTED DESIGN COMPOSITION AND MAPPING FOR CROSS-PLATFORM WEB APPLICATIONS
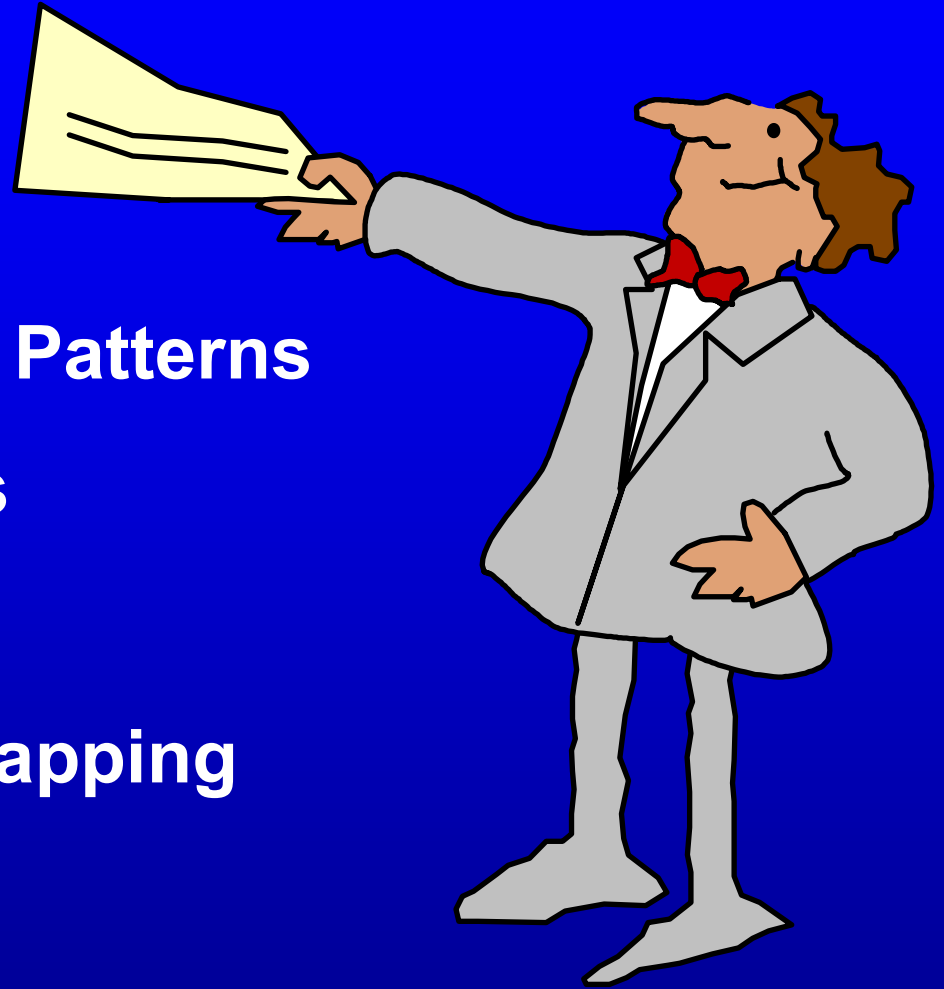
The XIII International Workshop
26-28 July 2006, Trinity College Dublin, Ireland

Mohamed Taleb &
Ahmed Seffah &
Homa Javehery

Agenda

- **Introduction**

- **Background Work**

- **The proposed Web Design Patterns**

- **Pattern Composition Rules**

- **An illustrative Example**

- **Pattern-Oriented Design Mapping**

- **Mapping rules**

- **Conclusion**

Introduction

- **Different kinds of computers and devices are used for interacting with such applications.**

- **One of the major characteristics of such cross-platform Web applications is that they allow a user to interact with the server-side services and contents in various ways.**

- **Web applications for small and mobile devices are resource constrained and cannot support the full range of Web application features and interactivity because of the lack of screen space or low bandwidth.**

- **One important question is how to develop and deploy the same application for different platforms – without "architecturing" and specifically writing code for each platform, or learning different languages and the many Web design guidelines that are available for each platform?**

Agenda

- **Introduction**

- **Background Work**

- **The proposed Web Design Patterns**

- **Pattern Composition Rules**

- **An illustrative Example**

- **Pattern-Oriented Design Mapping**

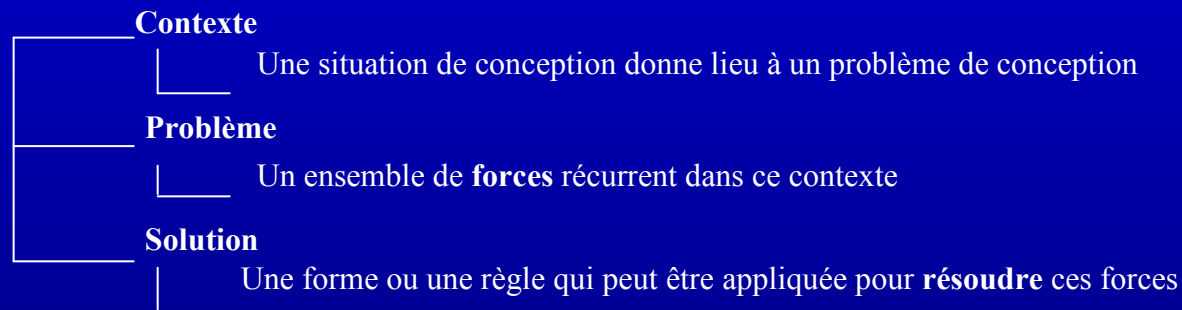- **Mapping rules**

- **Conclusion**
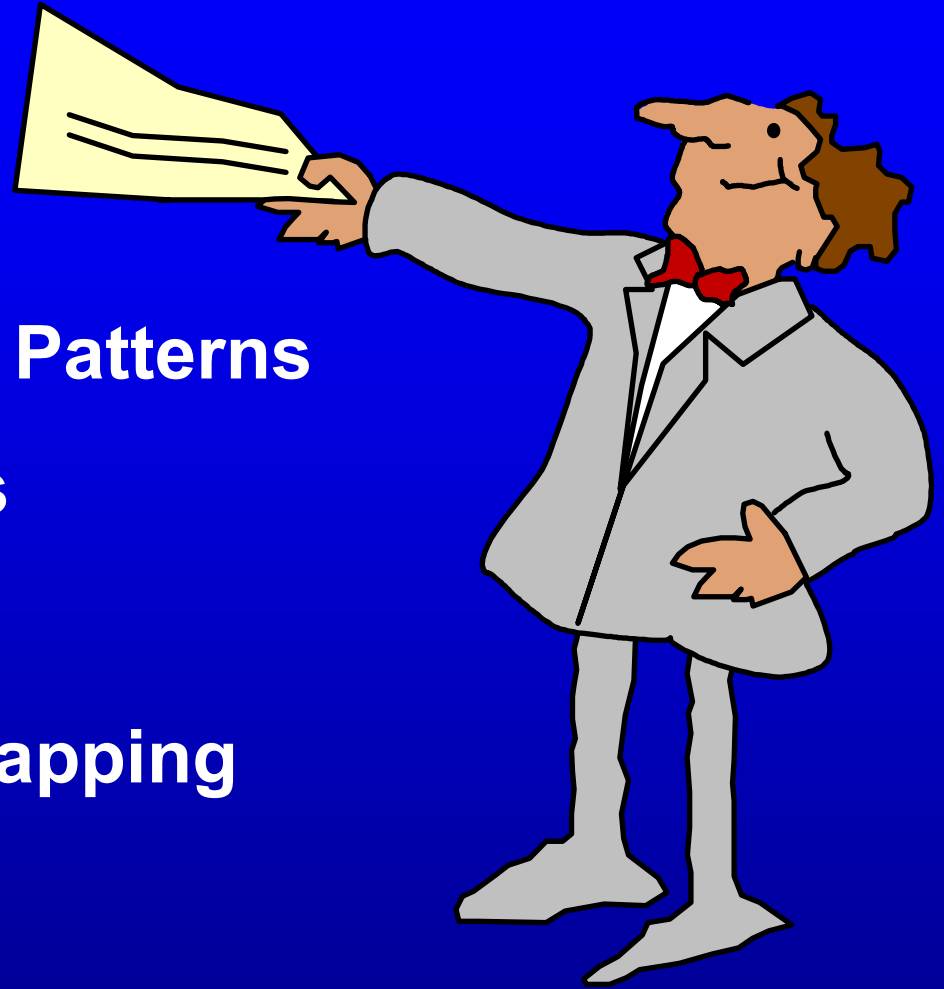
Background Work

- **Patterns**

- **Pattern-Oriented Design (POD)**

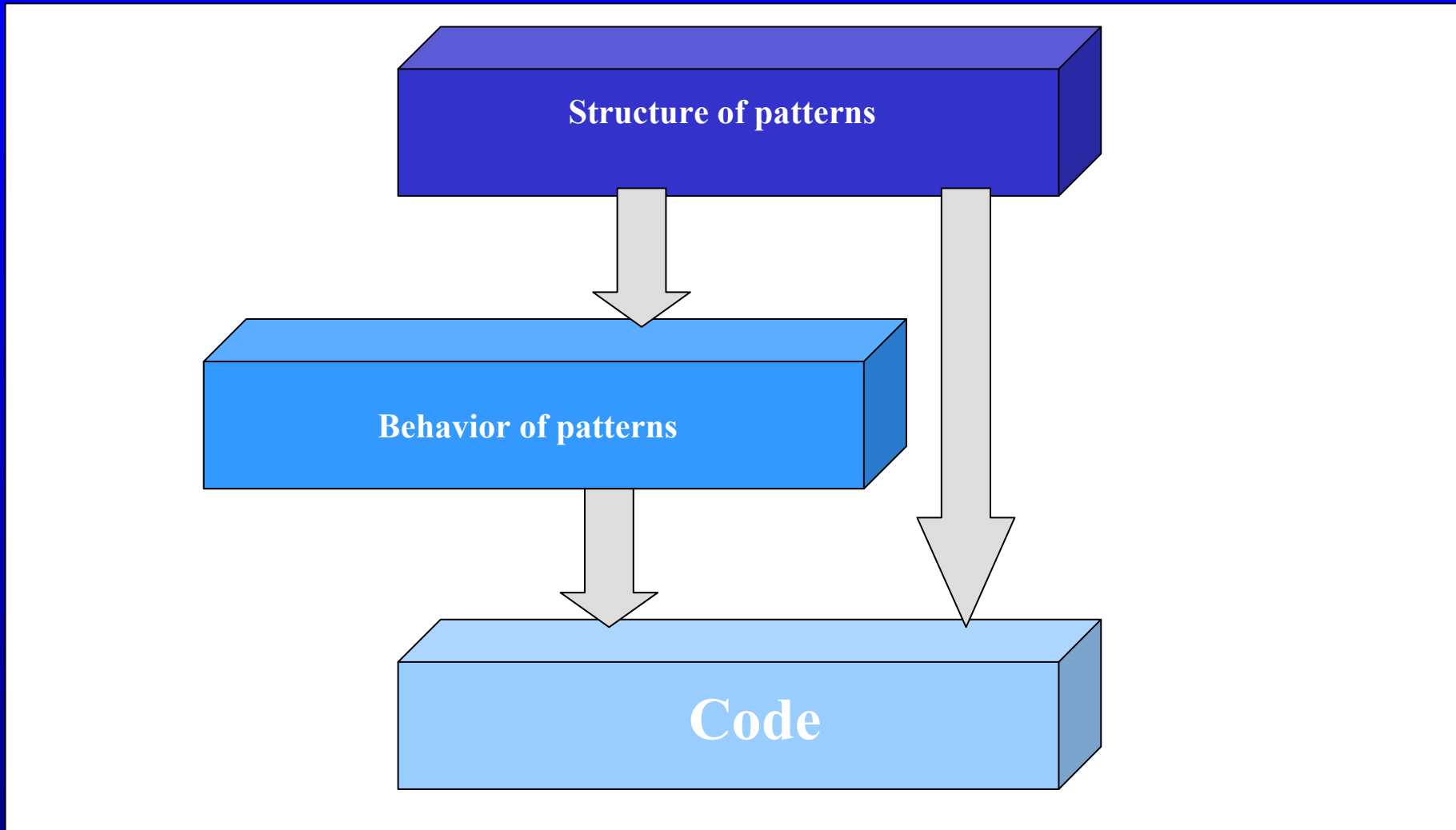- **Pattern-Supported Approach (PSA)**

Background Work

- **Why Pattern? Because of some limits UIs development**

  - **Complexity of models and their notations**

  - **Lack of the support tools**

  - **Lack of the reuse**

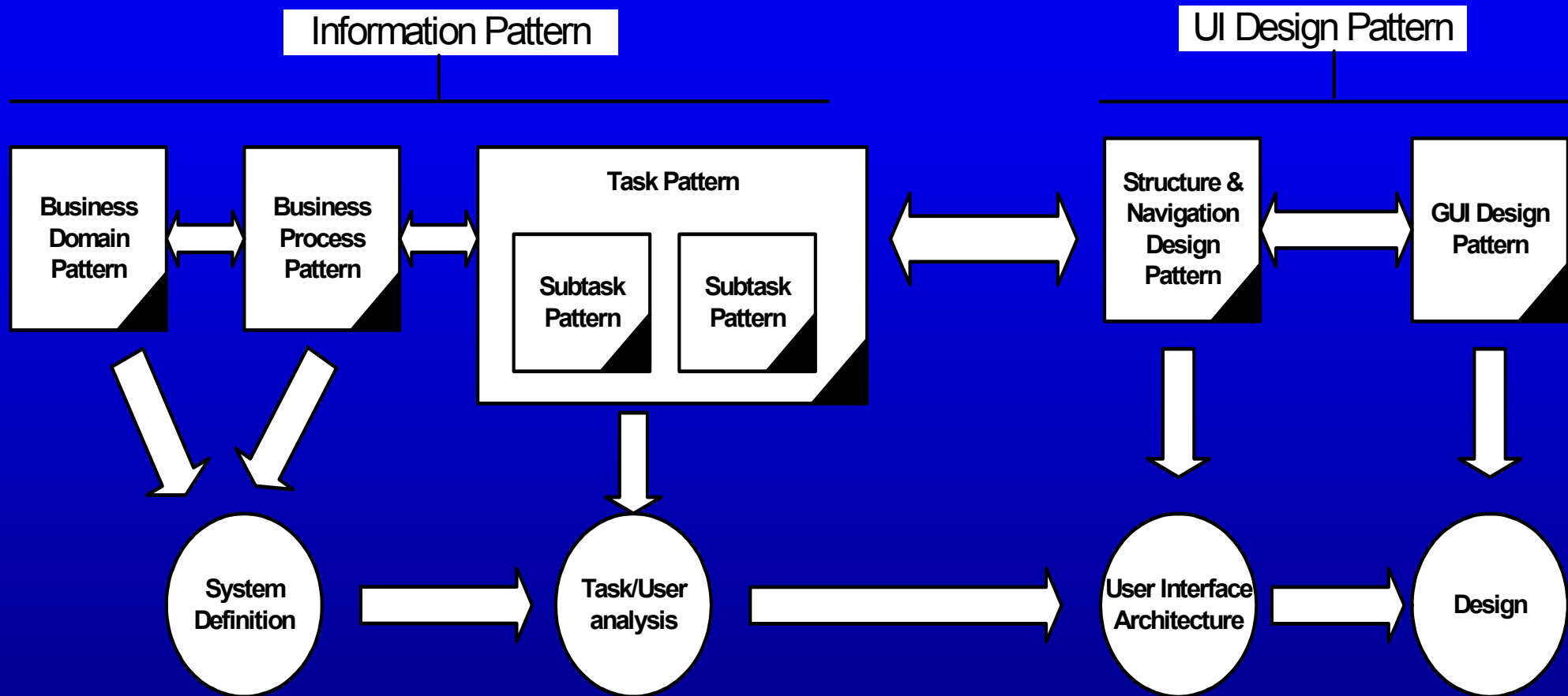- **Solution is the pattern : 3 principals elements**

Contexte

Une situation de conception donne lieu à un problème de conception

Problème

Un ensemble de **forces** récurrent dans ce contexte

Solution

Une forme ou une règle qui peut être appliquée pour **résoudre** ces forces

Background Work (Cont.)

- **Patterns-Oriented Design (POD)**

Structure of patterns

Behavior of patterns

Code

Composition techniques of Patterns-Oriented Design [7]

Background Work (Cont.)

- ## Pattern Supported Approach (PSA)

Information Pattern

UI Design Pattern

		Task Pattern		Structure & Navigation Design Pattern	GUI Design Pattern
Business Domain Pattern	Business Process Pattern	Subtask Pattern	Subtask Pattern		

System Definition → Task/User analysis → User Interface Architecture → Design

The PSA Framework with relationship between PSA patterns [10]

Background Work (Cont.)

- **Summary of these methodologies classification**

Methodology	Characteristics and criteria						
	Solution	Proven concept	Relationship	Human component	Structural technique	Behavioral technique	Tasks and Subtasks
Patterns (is not a methodology)	Yes	Yes	Yes	Yes	No	No	No
POD	Partially	Yes	No	No	Fully	Fully	No
PSA	Partially	Yes	No	No	No	No	Fully

Agenda

- **Introduction**

- **Background Work**

- **The proposed Web Design Patterns**

- **Pattern Composition Rules**

- **An illustrative Example**

- **Pattern-Oriented Design Mapping**

- **Mapping rules**

- **Conclusion**
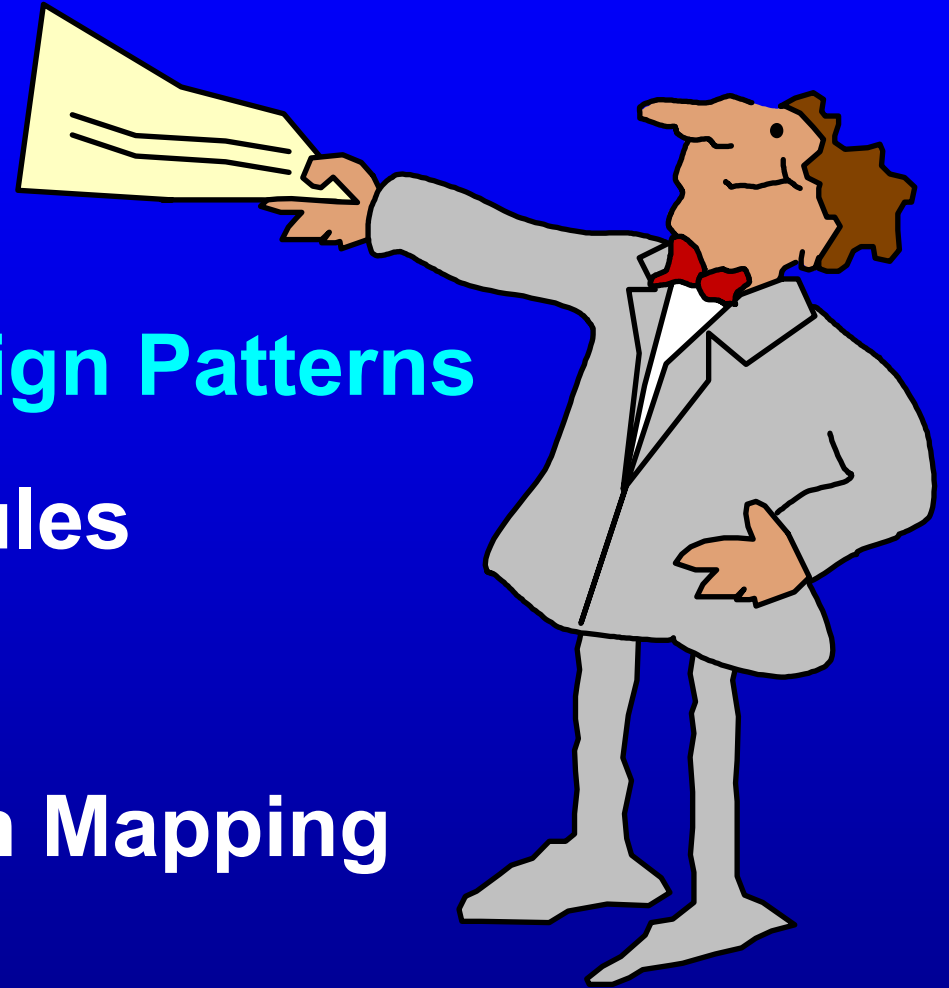
The Proposed Web Design Patterns

There are at least six types of design patterns that can be used in Web applications engineering :

Tiers for Separation of Concerns *Logical*	Applicable Patterns
Client User Interaction, UI **devices** presentation, **Tier**	Navigation and Interaction
Presentation Single sign on, session management, Content creation, Format and delivery	Presentation and Visualization
Business logic, services transactions, data,	Interoperability
Integration Resource adapters, legacy, external **Tier** systems, rules, engines workflow	Interoperability
Resource Sources, data and external services **Tier**	Information

A Pattern-Oriented Architecture of a Web Application

Navigation patterns implement proven techniques for navigating within and/or between a set of pages and chunks of information.

Interaction Patterns focus on the interaction mechanisms that can be used to achieve tasks and the visual effects they have on the scene, as such they relate primarily to graphical and rendering transforms

Presentation patterns provide solutions for how the contents or the related services are visually organized into working surfaces, the effective layout of multiple information spaces and the relationship between them. These patterns define the physical and logical layout suitable for specific Web pages such as home page, lists, tables, and forms.

Visualization Patterns suggest different visual representations/metaphors for grouping and displaying information in cognitively accessible chunks. They define mainly the form and content of the visualization i.e. the graphical scene, and as such relate primarily to data and mapping transforms.

Interoperability patterns aim to decouple the different layers of a Web application. In particular, between the content, the dialog and the views or presentation layers as outlined in Figure 1. These patterns are generally extensions of the Gamma design patterns such as MVC (Model, View and Controller) observer, and command action patterns.

Information patterns describe different conceptual models and architectures for organizing the underlying content across multiple pages, servers and computers. Such patterns provide solutions to questions such as which information can be or should be presented on which device.
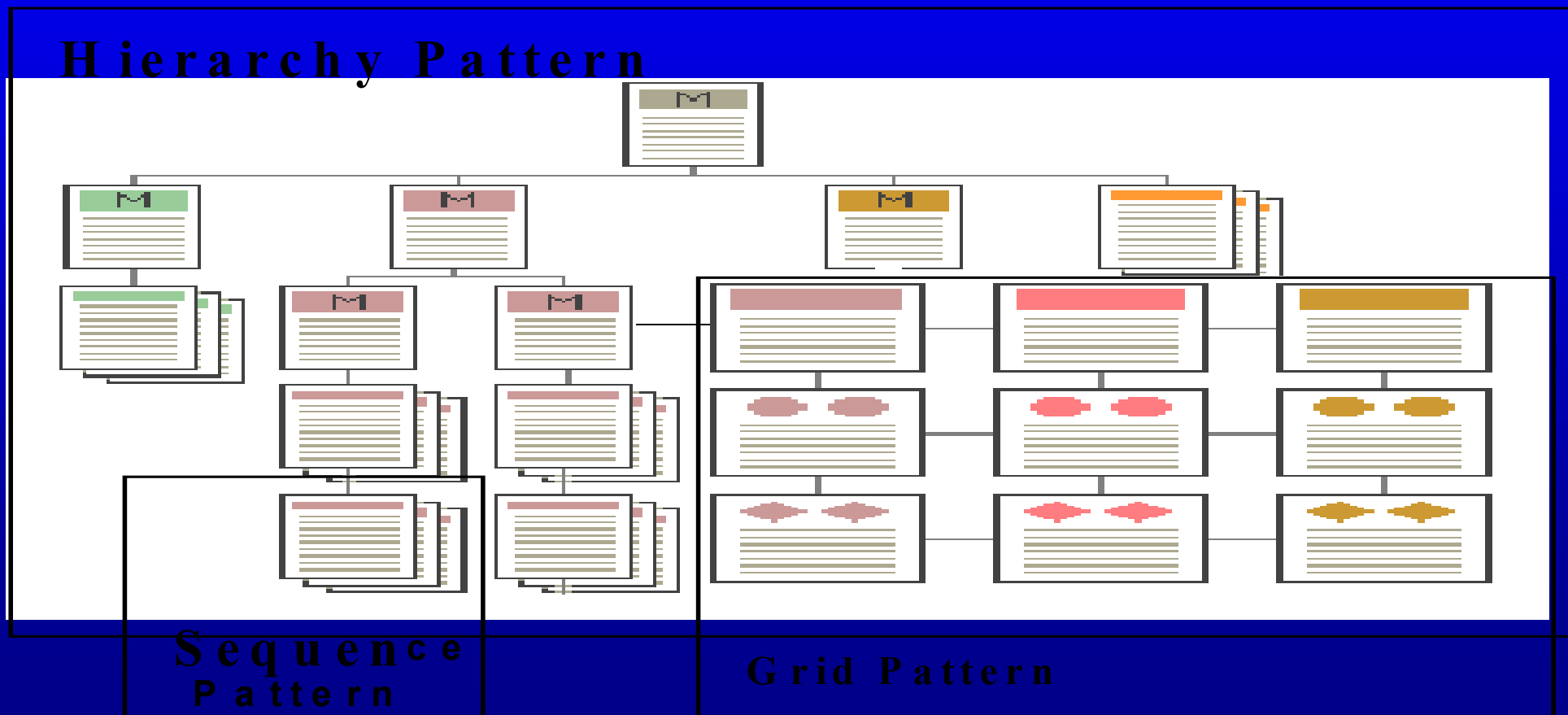
The Proposed Web Design Patterns(Cont.)

As illustrated in this Figure, several patterns need to be combined to organize the whole content in terms of information of a complex Web application.

The *sequence pattern* organizes a set of interrelated pages in a linear narrative.

The *hierarchy pattern* is particularly well suited to Web application content, because Web sites should always be organized as offshoots of a single Home Page [1].

The *grid pattern* organizes many procedural manuals, lists of university courses, or medical case descriptions.

A Combination of Information Architecture Patterns

The Proposed Web Design Patterns(Cont.)

Navigation patterns are fundamental in Web design since they help the user navigate easily and clearly between information chunks and pages.

Shortcut Pattern. Lists the frequently visited pages or used services. They are generally embedded in the home page and help experienced users find their favorite information and services with one mouse click.

Dynamic Path Pattern (or Bread Crumb) is a very useful pattern that indicates the entire path since the user accessed the Web application.

Index Browsing Pattern allows a user to navigate directly from one item to the next and back. The ordering can be based on a ranking

Example of structural patterns

The Proposed Web Design Patterns(Cont.)

Favorites Collection, *Bookmark*, *Frequently Visited Pages*, *Preferences* and *Navigable Spaces Map* patterns are some of the information visualization patterns for solving another complex design problem.
As presented in this Figure, these patterns are generally composed to provide a comprehensive map to a large amount of content that cannot be reasonably presented in a single view
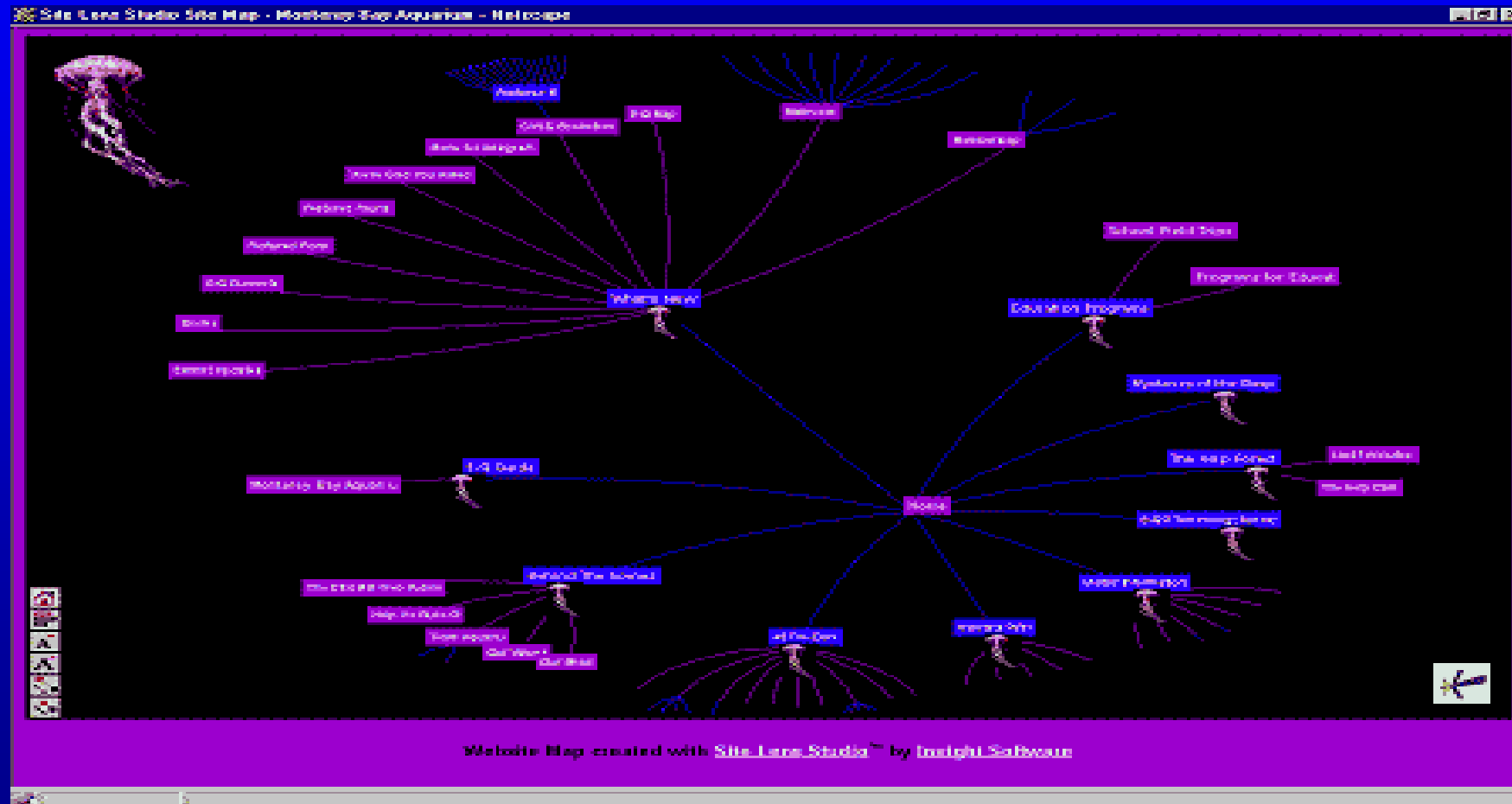
The Navigation Spaces Map Patterns

Agenda

- **Introduction**

- **Background Work**

- **The proposed Web Design Patterns**

- **Pattern Composition Rules**

- **An illustrative Example**

- **Pattern-Oriented Design Mapping**

- **Mapping rules**

- **Conclusion**
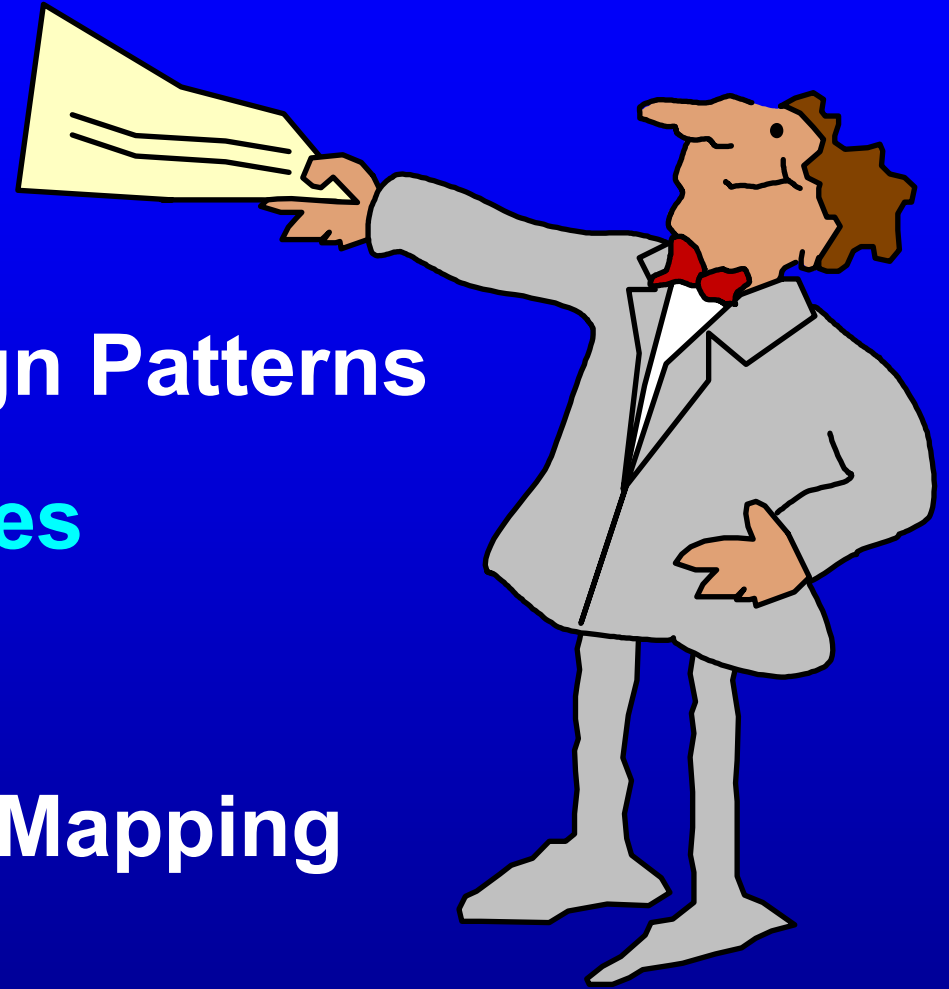
Patterns Composition Rules

- **Based on Zimmer's work, we define five types of relationships between patterns.**

  - **Similar (X, Y)** if and only if X and Y can be replaced by each other in a certain composition. This means that X and Y are patterns of the same category and they provide different solutions to the same problem in the same context. As illustrated in Figure 5, *Index Browsing* and *Menu Bar* patterns are similar. They both provide navigational support in the context of a medium size Web site.
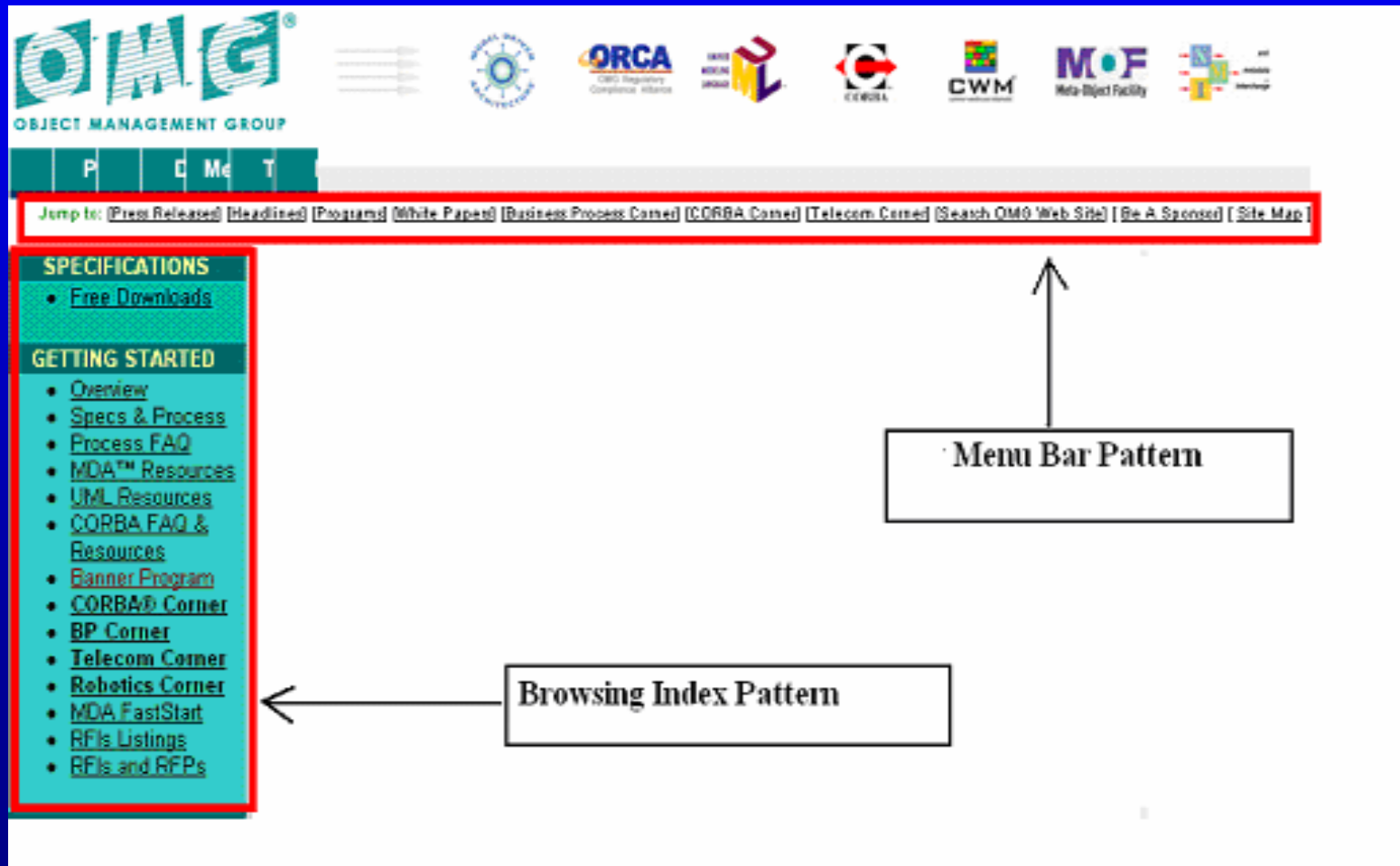
Patterns Composition Rules (Cont.)

- **Competitor (X, Y)** if X and Y cannot be used at the same time for designing the same artifact relationship that applies to two patterns of the same pattern category. Two patterns are competitors if and only if they are similar and interchangeable. For example, the *Web convenient toolbar* and *Index Browsing* patterns are competitors. The *Index Browsing* pattern can be used as a shortcut toolbar that allows a user to directly access a set of common services from any Web page. The *Convenient Toolbar* that provides the same solution is generally considered more appropriate

- **Super-ordinate** (X, Y) is the basic relationship to compose several patterns of different categories. A pattern X is a super-ordinate of pattern Y means that pattern Y is used as a building block to create pattern X. An example is the *Home Page* pattern which is generally composed of several other patterns

- **Sub-ordinate** (X, Y) if and only if X is embeddable in Y. Y is also called super-ordinate of X. This relationship is important in the mapping process of pattern-oriented design from one platform to another. For example the *Convenient Toolbar* pattern is a sub-ordinate of the *Home Page* pattern for either a PDA or Desktop Web application. Implementations of this pattern are different for different devices, as will be discussed in the next section.

- **Neighboring** (X, Y) if X and Y belong to the same pattern category (family). For example, neighboring patterns may include the set of patterns to design a specific page such as a home page.

Agenda

- **Introduction**

- **Background Work**

- **The proposed Web Design Patterns**

- **Pattern Composition Rules**

- **An illustrative Example**

- **Pattern-Oriented Design Mapping**

- **Mapping rules**

- **Conclusion**
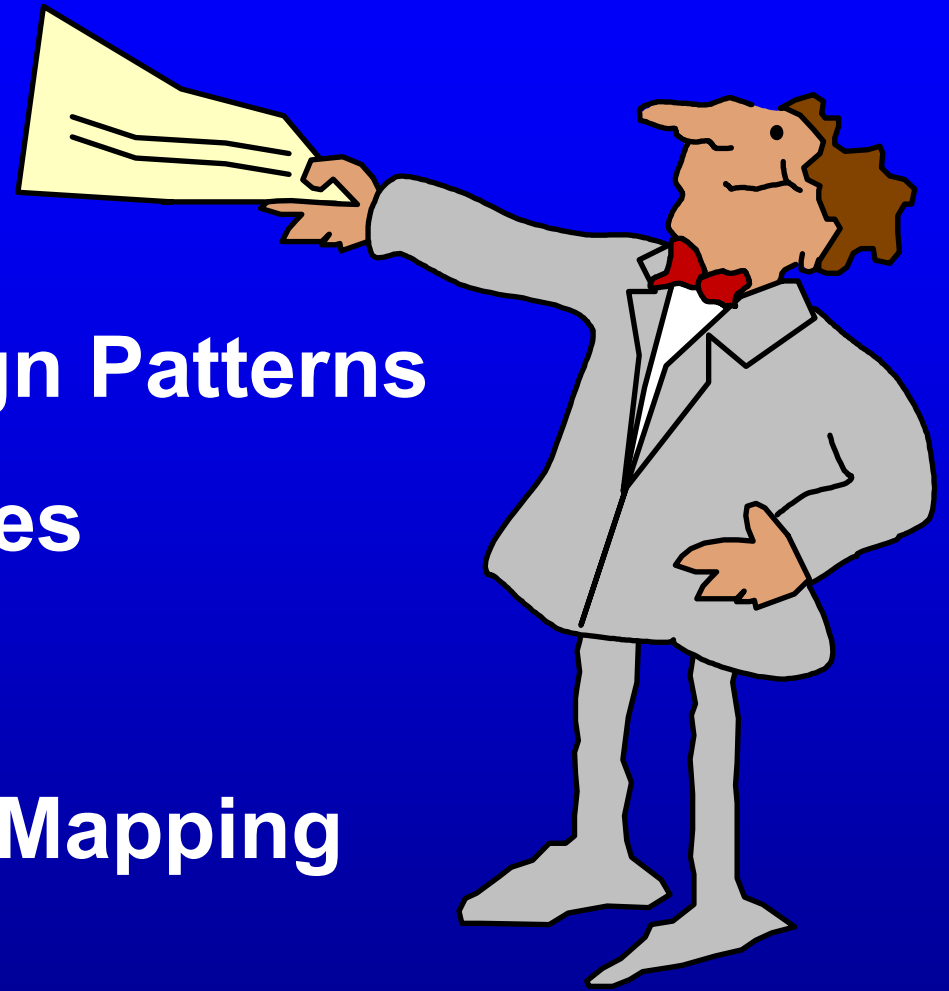
An illustrated Example
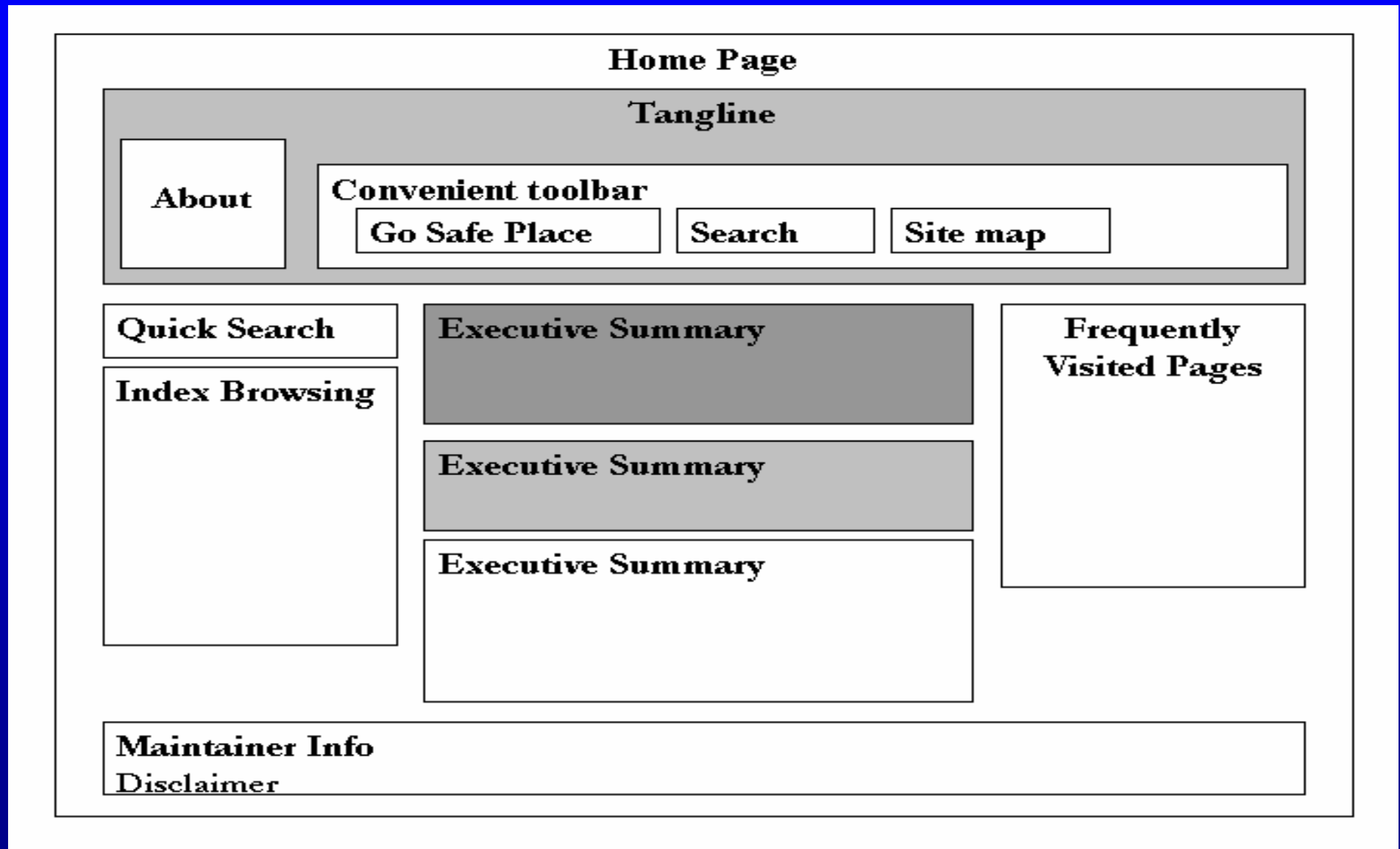
An illustrated Example

1. *Tangline Pattern*. The home page must explain what the site does and what makes it unique among competitors.

2. *Web Convenient Toolbar Pattern*. This navigation design pattern provides access to the most common services such as help, feedback, etc.

3. *Search Pattern* is important for any big Web site. When users want to search, they typically scan the home page looking for "the little box where I can type," so your search should be a box.

4. *Frequently visited pages Pattern*. Users will often remember good articles, products, or promotions.

5. *Site Map Pattern*. This information visualization design pattern summarizes the structure of the underlying content architecture.

6. *About Pattern*. Looking for specific information is rarely a user's first task, but sometimes people do need details about who you are. An "About <company-name>" pattern is the best way to link users to more in-depth information than can be presented on the home page.

7. *Executive Summary Pattern*. Don't just provide a link to some pages behind the home page. Show some of your best or most recent content. This can be achieved via the use of different instances of the executive summary pattern.

8. *Index Browsing Pattern*. Your home page should offer users a clear starting point for the main pages that users can undertake when visiting your site.

9. *Disclaimer Pattern*.

10. *Maintainer Info Pattern*. Sometimes pages are wrong; even the best proofreaders make mistakes. Therefore, a home page provides a link to the page's maintainer..

11. Go *Safe Place Pattern*. This pattern allows the user to go back to the home page from any page. To familiarize the user with it, this pattern should be visible on the home page.
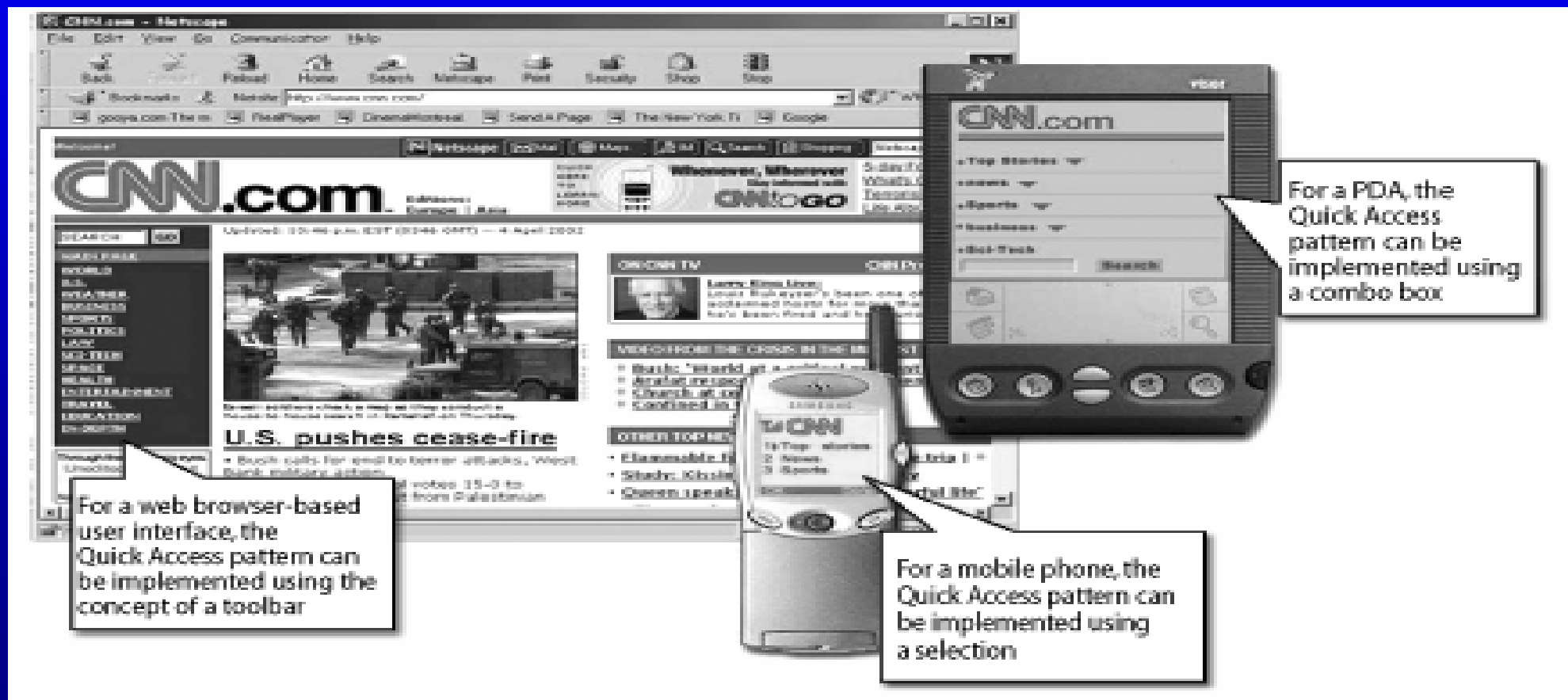
Agenda

- **Introduction**

- **Background Work**

- **The proposed Web Design Patterns**

- **Pattern Composition Rules**

- **An illustrative Example**

- **Pattern-Oriented Design Mapping**

- **Mapping rules**

- **Conclusion**
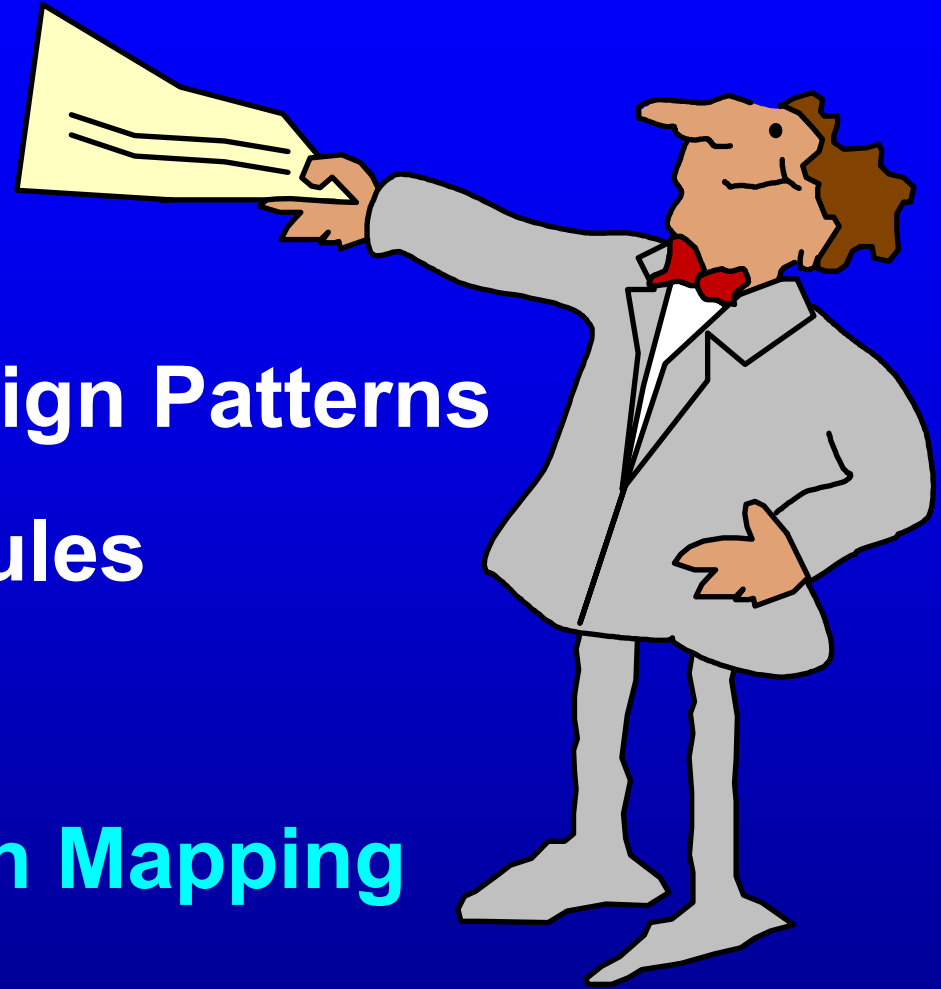
Pattern-Oriented Design Mapping

This Figure illustrates different mappings of the *Quick Access pattern* for three different platforms:

1. For a web browser on a desktop, it is implemented as an *index browsing toolbar*.

2. For a PDA, the Quick Access pattern can be implemented as a *combo box*.

3. For a mobile phone, the Quick Access pattern is implemented as a *selection* [4]. Patterns descriptions should provide advice to pattern users for selecting the most suitable implementation for a given platform

For a PDA, the Quick Access pattern can be implemented using a combo box

For a web browser-based user interface, the Quick Access pattern can be implemented using the concept of a toolbar

For a mobile phone, the Quick Access pattern can be implemented using a selection

The Web Convenient Toolbar pattern Implementations and Look & Feels for Different Platform 23

Pattern-Oriented Design Mapping (Cont.)

This Figure illustrates some of the navigation design patterns as used in the home page of a desktop-based Web application.

Once these patterns are identified in the desktop-based Web application, they can be transformed or replaced by others in a PDA version.
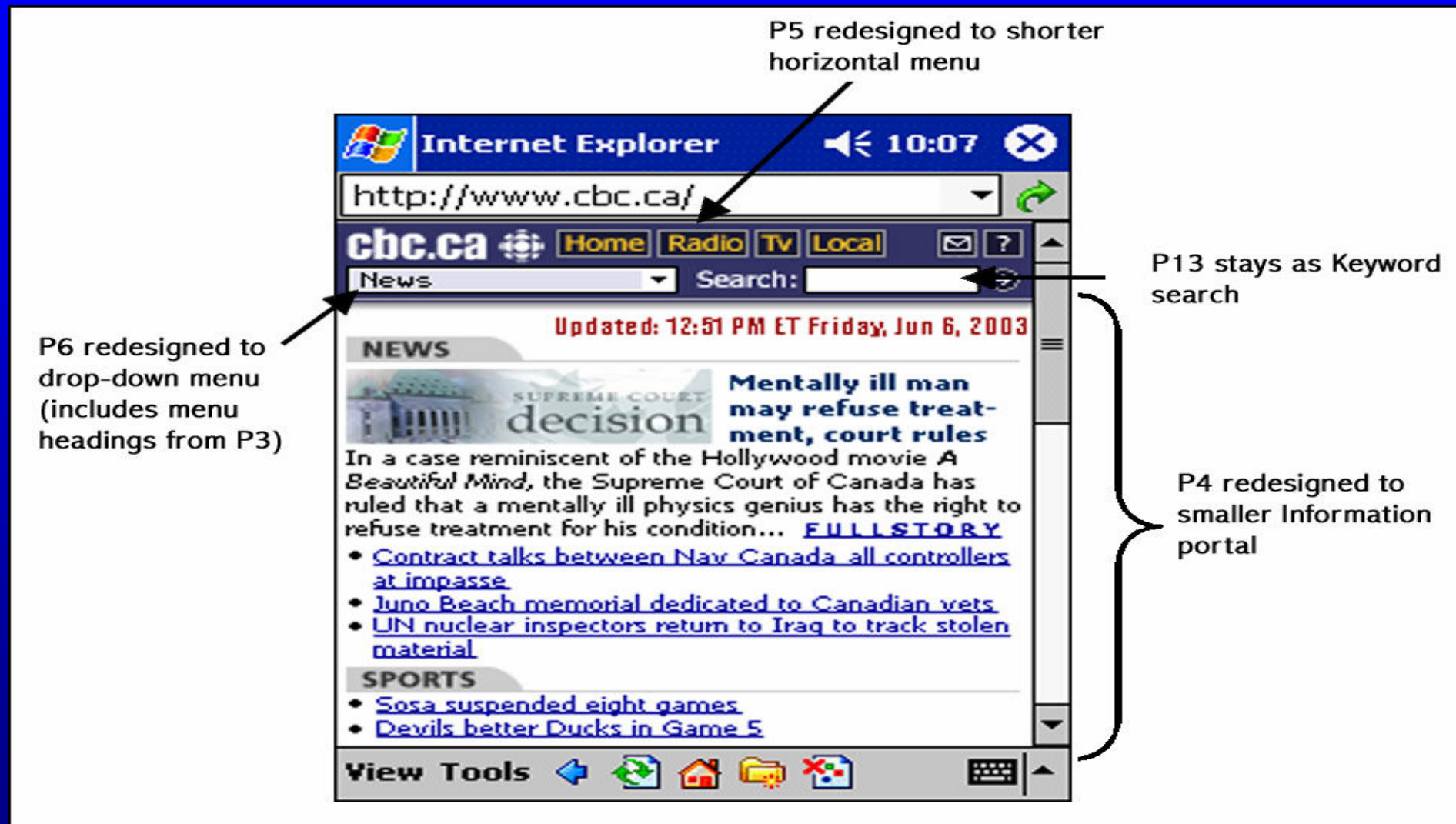

Patterns Extracted from the CBC News

Pattern-Oriented Design Mapping (Cont.)

Figure 9 demonstrates the redesigned interface of the CBC (Canadian Brodcasting Corporation, www.cbc.com) site for migrating to a PDA platform.

Migration of the CBC site to a PDA Platform using Pattern Mapping

Pattern-Oriented Design Mapping (Cont.)

In short, what we have just illustrated in this section and the examples of figures can be characterized in the form of architecture of composed pattern-oriented design mappings
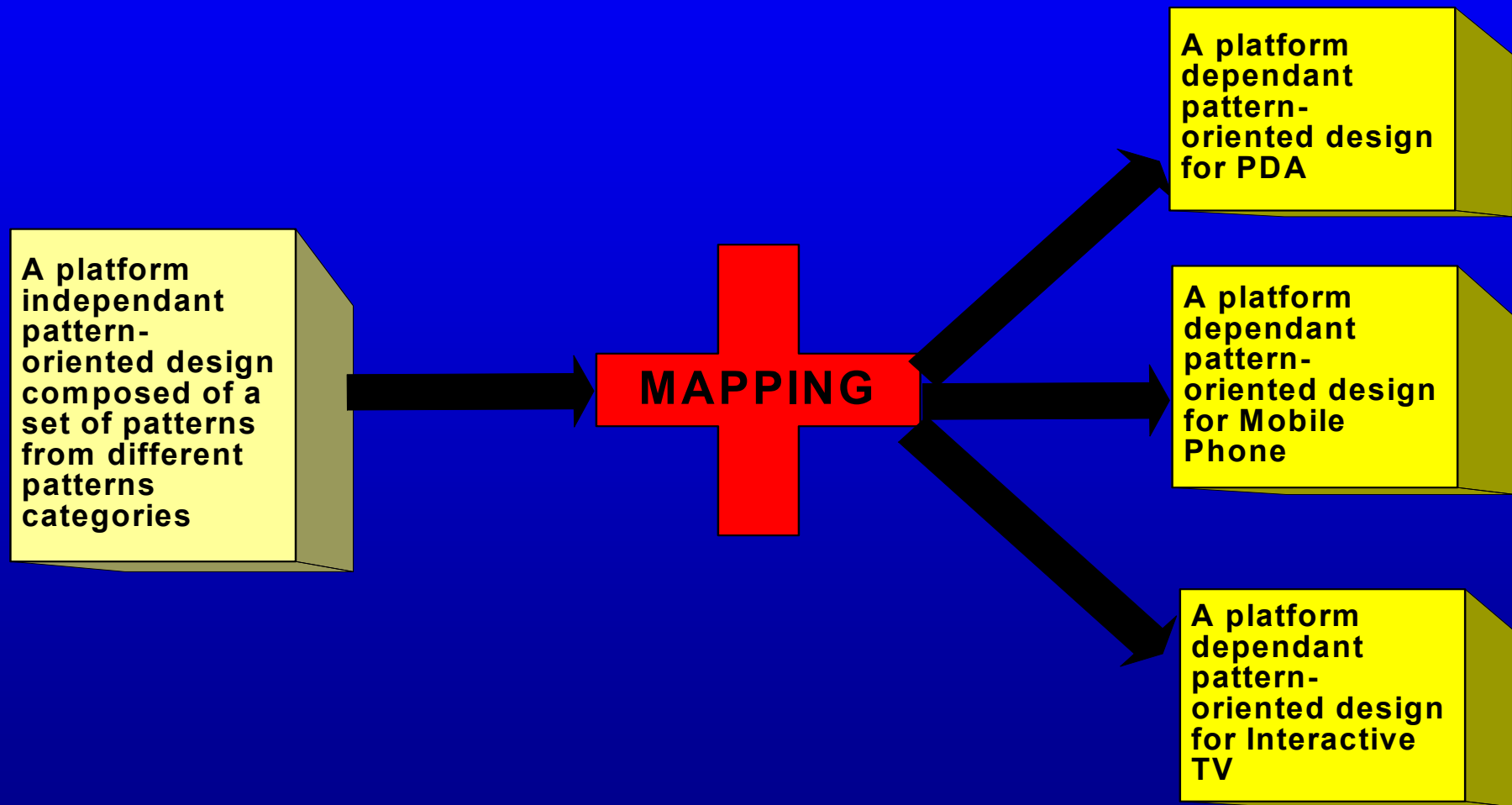
A platform independant pattern-oriented design composed of a set of patterns from different patterns categories

MAPPING

A platform dependant pattern-oriented design for PDA

A platform dependant pattern-oriented design for Mobile Phone

A platform dependant pattern-oriented design for Interactive TV

Pattern-Oriented Design Mapping Architecture

Agenda

- **Introduction**

- **Background Work**

- **The proposed Web Design Patterns**

- **Pattern Composition Rules**

- **An illustrative Example**

- **Pattern-Oriented Design Mapping**

- **Mapping rules**

- **Conclusion**
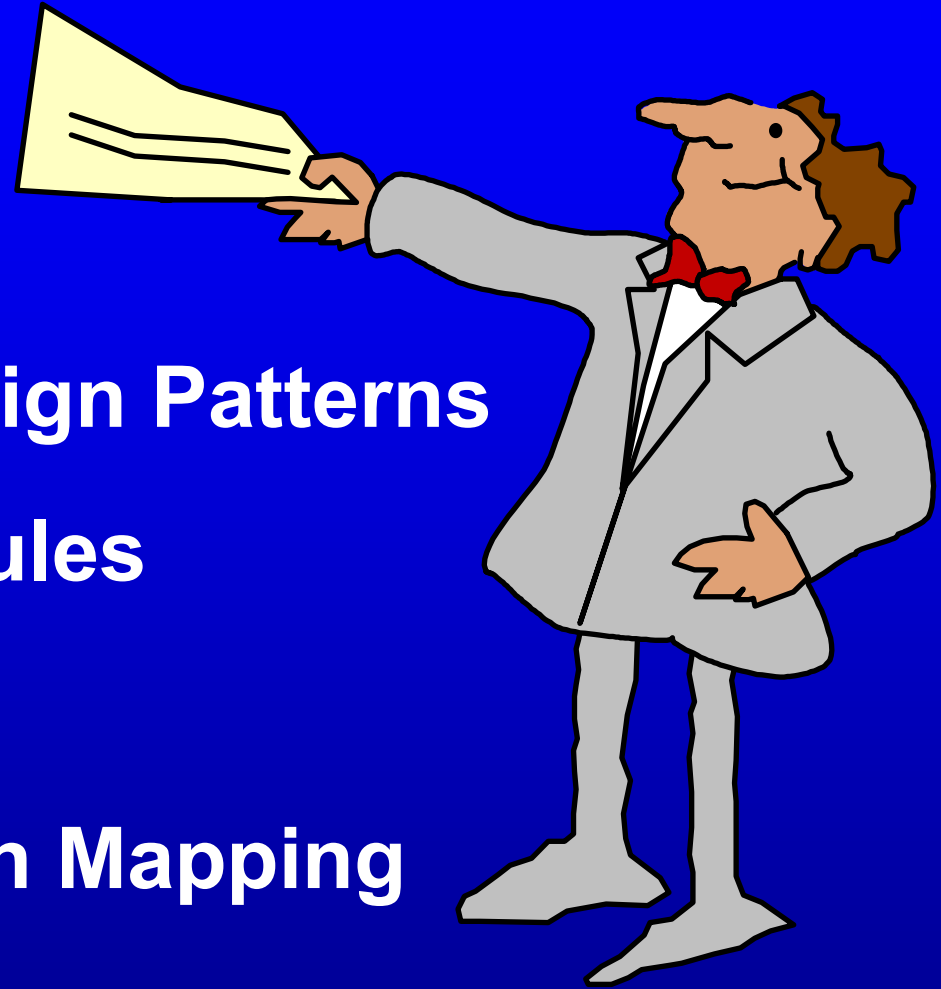
Mapping Rules

The following is the list of mapping rules we suggest:

1. Identical. For example, drop-down menus pattern can usually be copied without transformation from a desktop to a PDA.

2. Scalable changes to the size of the original design or to the number of items in the original design. For example, a long horizontal menu pattern can be adapted to a PDA by reducing the number of menu elements.

3. Multiple of the original design, either simultaneously or sequentially in time. For example, a single long menu can be transformed into a series of shorter menus.

4. Fundamental change to the nature of the original design pattern while replacing it generally by another one. For example, permanent left-hand vertical menus pattern are useful on desktop displays but are not practical on most PDAs. In transformation to a PDA, left-hand menus normally need to be replaced with an alternative such as a drop-down menu pattern.

Examples of Patterns Mapping

HCI pattern in desktop display	Type of transformation	Replacement pattern in small PDA display
P.1 Bread crumbs	Scalable or fundamental	P.1s - Shorter bread crumb trail; P.15 - Drop-down "History" menu.
P.2 Temporary horizontal menu	Scalable or fundamental	P.2s - Shorter menu; P.6 - Link to full-page display of menu options ordered vertically
P.3 Temporary vertical menu in content zone	Identical, scalable or fundamental	P.6 - Temporary vertical menu in content zone; P.6s — Shorter temporary vertical menu; or P.15 - Drop-down menu
P.4 Information portal	Scalable	P.4s - Smaller information portal
P.5 Permanent horizontal menu at top	Scalable or fundamental	P.5s — Shorter horizontal menu at top; P.6 - Link to full-page display of menu options ordered vertically
P.6 Permanent vertical menu at left	Fundamental	P.15 - Drop-down menu
P.7 Progressive filtering	Identical	P.7 - Progressive filtering
P.8 Shallow embedded vertical menus	Identical or fundamental	P.6m — Sequence of temporary vertical menus in content zone; P.8 - Shallow embedded vertical menus
P.9 Sub-site	Scalable or fundamental	P.6 — Temporary vertical menu in content zone; P.9s — Smaller sub-site;
P.10 Container navigation (3 containers)	Scalable or fundamental	P.10s - Container navigation (2 containers); P.7 - Progressive filtering
P.11 Deeply embedded vertical menus	Multiple or fundamental	P.6m — Sequence of single-level menus; P.8m - Sequence of shallow embedded menus
P.12 Alphabetical index	Scalable	P.12s - Alphabetical index (less items per page, or smaller index)
P.13 Key-word search	Identical	P.13 - Key-word search
P.14 Intelligent agents	Identical	P.14 - Intelligent agents
P.15 Drop-down menu	Identical, scalable or fundamental	P.15 - Drop-down menu; P.15s — Shorter drop-down menu; Hyperlink to P.6 — Temporary vertical menu in content zone
P.16 Hybrid navigation: Key-word search	Identical or scalable	P.16s - Hybrid approach with smaller or less deeply embedded menus

Agenda

- **Introduction**

- **Background Work**

- **The proposed Web Design Patterns**

- **Pattern Composition Rules**

- **An illustrative Example**

- **Pattern-Oriented Design Mapping**

- **Mapping rules**
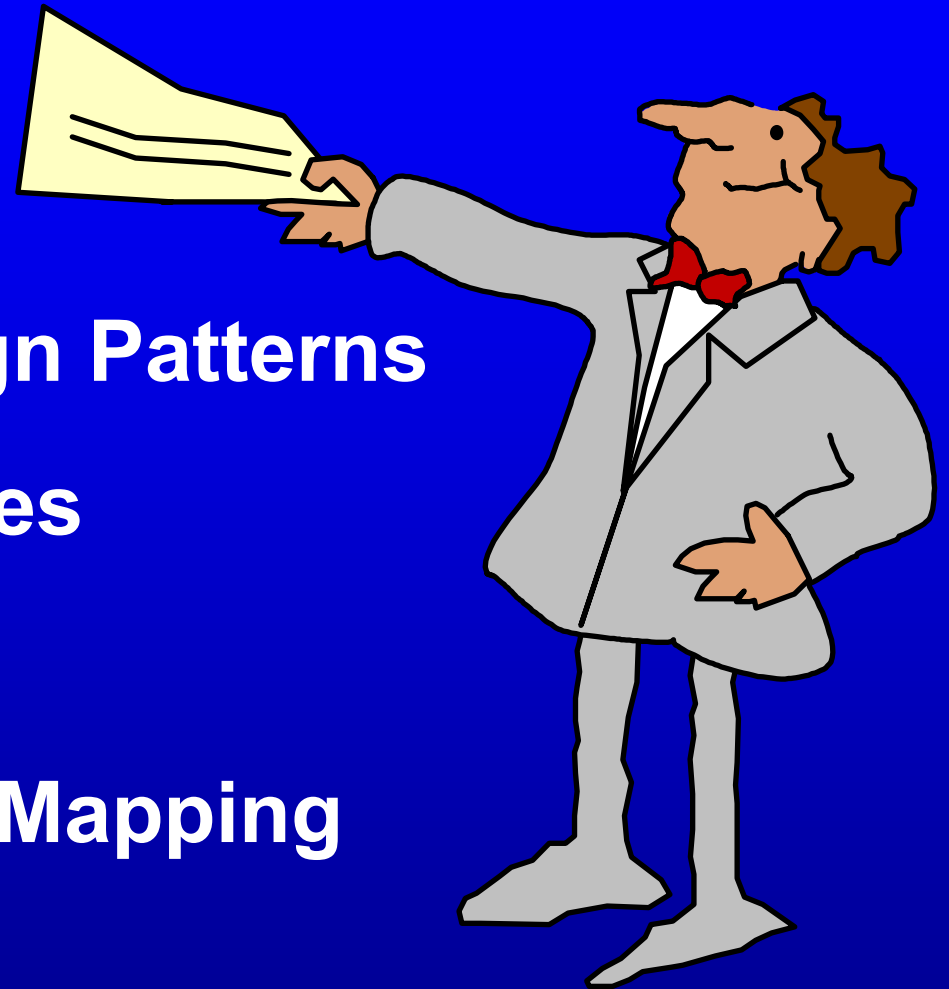
- **Conclusion**

Conclusion

- ## Summary
  - our discussion focused on the way to combine and map several types of patterns to create a pattern-oriented design for a cross-platform web application.

  - In POD, cross-platform Web application developers can exploit the composition and mapping relationships and the underlying best practices to derive a concrete design from a pattern-oriented design.

  - We can explore others forms of Web applications such as transactional applications, e-commerce, etc.

- ## Future Work
  - needs to be done for exploring:

    - (1) the scalability of the approach to given multiple patterns, platforms and platform design guidelines,

    - (2) the strategies for automating the POD approach as well as the reuse of patterns.

References

- [1] Lynch, P.J, and Horton, S. Web Style Guide: Basic Design Principles for Creating Web Sites. New Haven and London: Yale University Press. 1999.

- [2] Gamma, E., Helm R., Johnson R., and Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.

- [3] Tidwell, J. Common Ground: A Pattern Language for Human-Computer Interface Design, 1997. http://www.mit.edu/~jtidwell/common_ground.html

- [4] Welie, M.V. The Amsterdam Collection of Patterns in User Interface Design 1999- http://www.cs.vu.nl/~martijn/patterns/index.html

- [5] Engelberg, D., and Seffah, A. Design Patterns for the Navigation of Large Information Architectures. 11th Annual Usability Professional Association Conference Orlando, Florida, July 8-12, 2002.

- [6] Duyne D. K. van, Landay, J. A, and Hong J. I. The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience. Addison-Wesley, 2003.

- [7] Sherif Yacoub, Hany Ammar, 2003, *Composition of Design Patterns*, Addison Wesley Professional, ISBN 0-201-77640-5, 416 pages, Hardcover, Germany

References (Cont.)

- [8]   Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiskdahl-King, S. Angel. A Pattern Language, Oxford University Press, New York, 1977

- [9]   F. Buschmann, What is a pattern? Object Expert vol. 1(3), pp17-18, Mars/April 1996.

- [10]   Åsa Granlund, Daniel Lafrenière, David A. Carr, 2001, 'A Pattern-Supported Approach to the User Interface Design Process', Proceedings of HCI International 2001 9th International Conference on Human-Computer Interaction, August 5-10, 2001, retrieved on 2005, New Orleans, USA.

- [11]   Walter Zimmer. Relationships between design patterns. New York, NY, USA, 1995. ACM Press / Addison-Wesley Publishing, pp 345–364.

References (Cont.)

- [17]   **Walter Zimmer. Relationships between design patterns. New York, NY, USA, 1995. ACM Press / Addison-Wesley Publishing, pp 345–364.**

- [18]   **Sari A. Laakso. Collection of User Interface Design Patterns University of Helsinki, Dept. of Computer Science, September 16, 2003.**
  http://www.cs.helsinki.fi/u/salaakso/patterns /

Questions Period

Thank you