WEBIST 2007 Conference Presentation

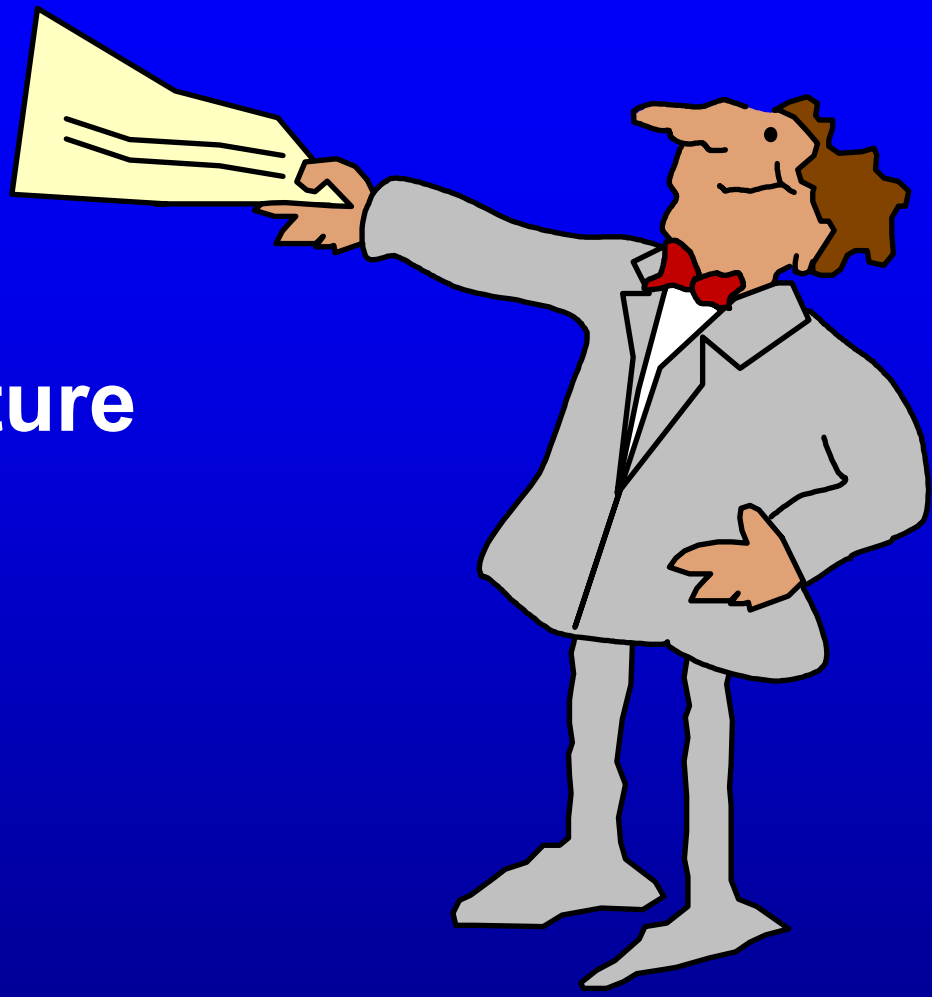
PATTERN-ORIENTED ARCHITECTURE FOR WEB APPLICATIONS

The 3rd International Conference
03-06 March 2007, Barcelona, Spain

Mohamed Taleb &
Ahmed Seffah &
Alain Abran

Agenda

- Introduction
- Related Work
- The proposed Architecture
- Conclusion



Introduction

- The Internet and its languages offer major opportunities to develop a new generation of Web software systems architecture .
- These new Web software systems are highly interactive, platform-independent and run on the client Web browser across a network .
- Provide a pool of proven solutions to many recurring Web design problems like example :
  - decoupling the different aspects of Web applications such as the business logic, the user interface, the navigation and information architecture
  - isolating platform specifics from the common concerns to all Web applications.

Introduction (Cont.)

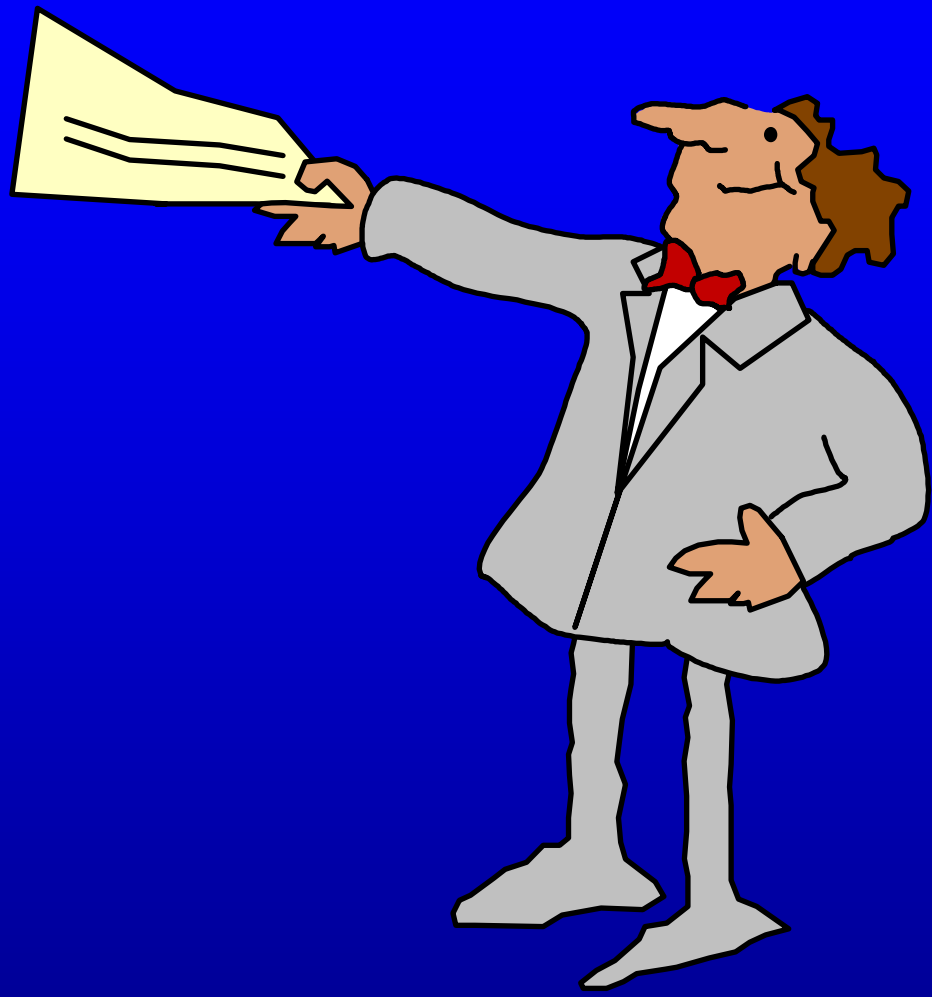
- To address these issues, a proposed architecture and the related list of patterns aim to provide a pool of proven solutions :
  - A list of patterns for a pattern-oriented architecture for Web application

- These individual patterns could then be combined at different levels of abstraction into heterogeneous structures that can be used as building blocks in the development of Web applications.

- 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
- Related Work
- The proposed Architecture
- Conclusion



Related Work

- MVC model (3-tier architecture)
- Model Core J2EE patterns (5-tier architecture)
- Zachman model (multi-tier architecture)

Related Work (Cont.)

- MVC Pattern (3-tier architecture)
  - Is commonly used to structure web applications that have significant processing requirements
  - This makes them easier to code and maintain
  - Is used here to describe the core components of Web applications architecture whereas MVC is seen as 3-tier architecture that often is used by Web applications designers to maintain multiple views of the same data.
  - At the design level, the MVC pattern hinges on a clean separation of three types of objects:
    - Model: for maintaining data;
    - View: for displaying all or a portion of the data;
    - Controller: for handling events that affect the model or view(s).

Related Work (Cont.)

- Model Core J2EE patterns (5-tier architecture)
  - Building on the MVC pattern, the Java Sun team proposed a 5-tier architecture (www.developpez.com) to model Core J2EE Patterns Architecture (www.sun.com)
  - Java also provides support for the implementation of the Model-View-Controller architecture using the Observer Interface and the Observable classes that together implement the observer pattern
  - The Observable class represents an observable object, or "data" in the model-view paradigm. It can be sub-classed to represent an object that the application wants to have observed.
  - An observable object can have one or more observers. An observer may be any object that implements the Observer interface.

Related Work (Cont.)

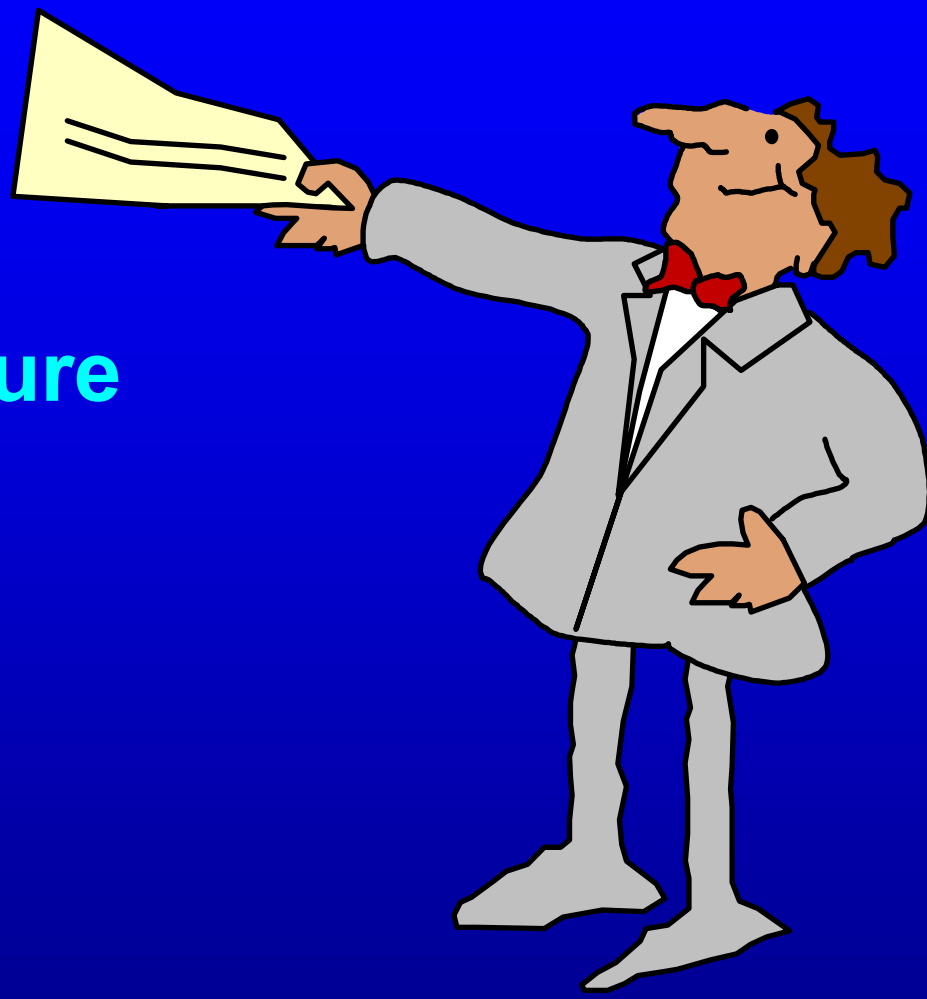
- **Zachman model (multi-tier architecture)** (Zachman, 1987) **and** (Sowa and Zachman, 1992)
  - Zachman proposed an Enterprise Architecture schema in which he depicted two distinct dimensions in a matrix:
    - The columns classify answers to questions such as: What (Data), How (Function), Where (Network), Who (People), When (Time) and Why (Motivation).
    - The rows classify the audience's perspectives: scope, owner, designers, builder, trades and functioning organization.
  - This gives 36 cells that uniquely classify portions of the organization.
  - The columns in the Zachman framework represent different areas of interest for each perspective and describe the dimensions of the systems development effort

Related Work (Cont.)

- **Summary of these architectures classification**
- The MVC architectural pattern is not a sufficient solution that fully addresses these issues. Other patterns are required to:
  - **Encourage the designer to consider other aspects of the dialogue which are very important for the user, such as assistance or error management;**
  - **Facilitate the use for the interface descriptions whereas they are of great importance to the designer (Booch, Rumbaugh and Jacobson, 1999), (Myers, 1986), (Myers and Buxton, 1986) and (Meyer, 1990).**
- It can be observed that Web architecture need to represent at six different levels (Navigation, Interaction, Presentation, Visualization, Interoperability and Information Patterns).

Agenda

- Introduction
- Related Work
- The proposed Architecture
- Conclusion



The Proposed Architecture (Cont.)

Zachman theory or set of concepts is used to propose a 6-tier architecture of a Patterns-Oriented generic classification schema for a Web Software Architecture.

	WHAT (Data)	HOW (Function)	WHERE (Network)	WHO (People)	WHEN (Time)	WHY (Motivation)
Navigation	✓	✓		✓		✓
Interaction		✓	✓	✓	✓	✓
Presentation	✓	✓				✓
Visualization	✓	✓		✓		✓
Interoperability	✓	✓	✓	✓	✓	✓
Information	✓	✓	✓	✓	✓	✓

Table 2: Patterns-Oriented generic classification schema for Web Software Architecture

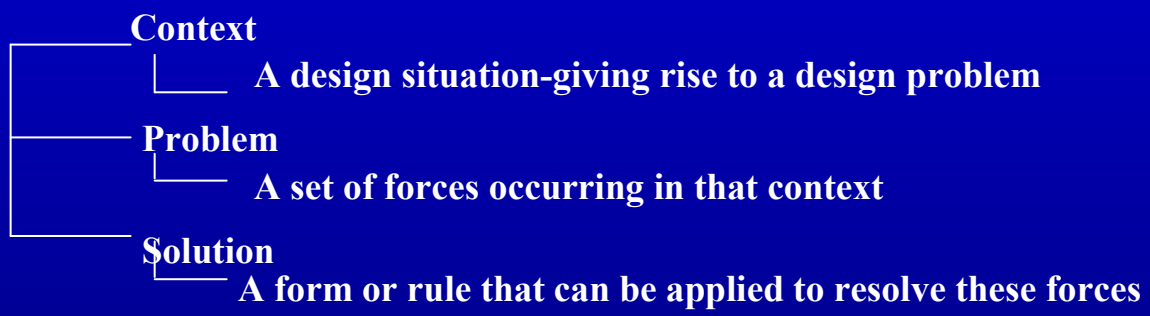
The Proposed Architecture (Cont.)

Patterns taxonomy

- A number of Web pattern languages have been suggested:
  - Van Duyne's "The Design of Sites" (Duyne, Landay, and Hong, 2003),
  - Welie's Interaction Design Patterns (Welie, 1999),
  - Tidwell's UI Patterns and Techniques (Tidwell, 1997) play an important role
- In addition, specific languages such as :
  - Laakso's User Interface Design Patterns (Laakso, 2003)
  - the UPADE Web Language (Engelberg and Seffah, 2002)

Related Work

- **Why a Pattern? Because of some limitations of IUs development**
  - Complexity of the models and their notations
  - Lack of tool support
  - Lack of reuse
- **Solution is the pattern : 3 main elements**



The Proposed Architecture (Cont.)

Architectural Level	Category of Patterns	Examples of Patterns
Navigation	**Navigation Patterns** This category of patterns implements proven techniques for navigating within and/or between a set of pages and chunks of information.	- Shortcut pattern - Bread Crumb pattern - Index Browsing pattern
Interaction	**Interaction Patterns** This category of patterns focuses 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.	- Search pattern - Executive Summary pattern
Presentation	**Presentation Patterns** This category of patterns provides 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, and tables.	- Home Page pattern - List pattern - Table pattern

Table 3: Patterns-Oriented taxonomy schema for Web Software Architecture

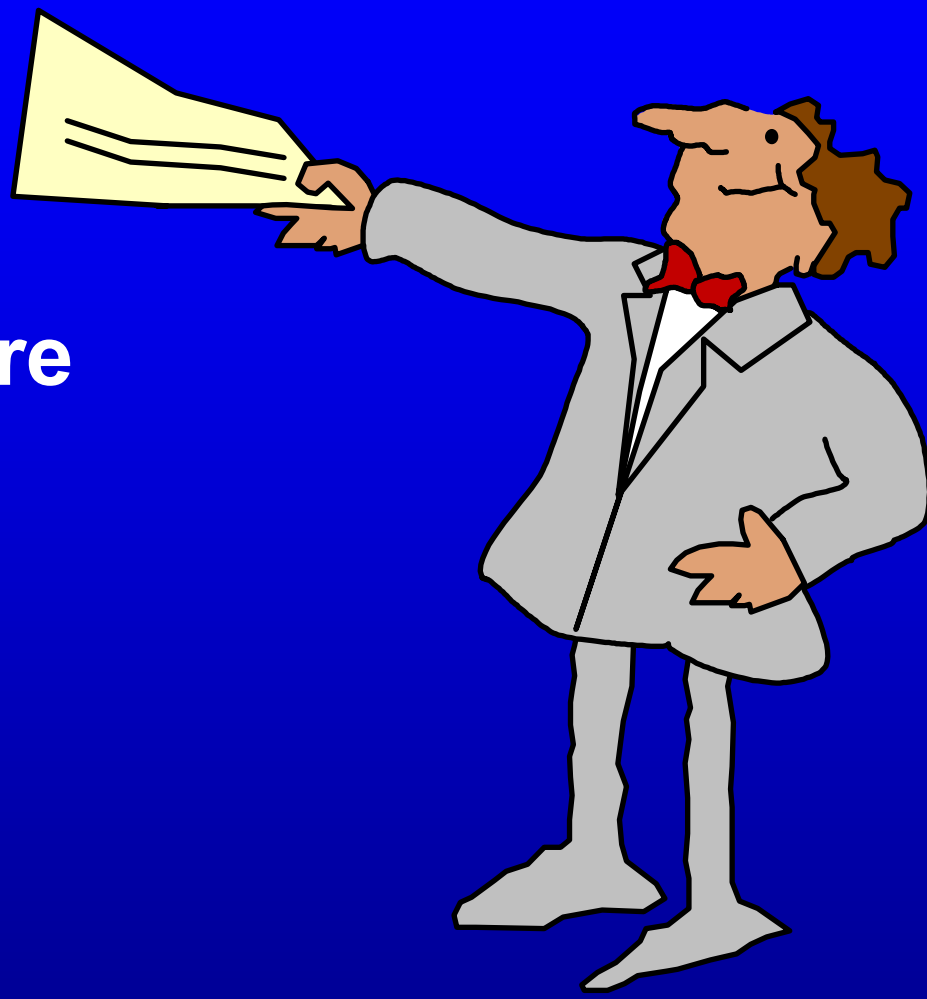
The Proposed Architecture (Cont.)

Visualization	**Visualization Patterns** **This category of patterns suggests different visual representations and metaphors for grouping and displaying information in cognitively accessible chunks. They mainly define the format and content of the visualization, i.e., the graphical scene and, as such, relate primarily to data and mapping transforms.**	**- Favourite Collection pattern** **- Bookmark pattern** **- Frequently Visited Page pattern** **- Navigation Space Map pattern**
Interoperability	**Interoperability Pattern** **This category of patterns aims to decouple the different layers of a Web application. In particular, between the content, the dialog and the views or presentation layers. These patterns are generally extensions of the Gamma design patterns such as MVC (Model, View and Controller) observer, command actions patterns. Communication and interoperability patterns are useful patterns to facilitate the mapping of design between platforms.**	- Adapter pattern - Bridge pattern - Builder pattern - Decorator pattern - Façade pattern - Factory pattern - Method pattern - Mediator pattern - Memento pattern - Prototype pattern - Proxy pattern - Singleton pattern - State pattern - Strategy pattern - Visitor pattern
Information	**Information Patterns** **This category of patterns describes 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**	**- Sequence pattern** **- Hierarchy pattern** **- Grid pattern**

Table 6: Patterns-Oriented taxonomy schema for Web Software Architecture

Agenda

- Introduction
- Related Work
- The proposed Architecture
- Conclusion



Conclusion

- **Summary**
  - we have identified and proposed six categories of patterns, together with examples, for a pattern-oriented architecture for Web applications to resolve many recurring Web design problems.
  - Examples of such problems include:
    - Decoupling the different aspects of Web applications such business logic, the user interface, the navigation, and information architecture
    - Isolating platform specifics from the common concerns to all Web applications.
  - Our discussion focused on the way to specify a Pattern-Oriented Architecture using particularly patterns.
- **Future Works**
  - Requires the classification of each pattern and the illustration of each one of them in UML class and sequence diagrams.
  - Some relationships must be defined between patterns in order to compose them together to create some models based on composed patterns.

References

- (Buschmann, Meunier, Rohnert, Sommerlad and Stal, 1999)
  Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. & Stal, M., 1996. A System of Patterns: Pattern-Oriented Software Architecture. West Sussex, England, John Wiley & Sons.
- (Zachman, 1987)
  Zachman John A.. A Framework for Information Systems Architecture. IBM Systems Journal, vol. 26, no. 3. IBM Publication G321-5298.
- (Sowa and Zachman, 1992)
  Sowa J.F. and Zachman John A, 1992. Extending and Formalizing the Framework for Information Systems Architecture. IBM Systems Journal, vol. 31, no. 3. IBM Publication G321-5488.
- (www.developpez.com)
  Architecture multi-tiers, Retrieved 2006, [Online] available at: http://java.developpez.com/archi_multi-tiers.pdf
- (Duyne, Landay and Hong, 2003)
  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.

References (Cont.)

- (Welie, 1999)
  Welie, M.V., 1999. The Amsterdam Collection of Patterns in User Interface Design - http://www.cs.vu.nl/~martijn/patterns/index.html .
- (Tidwell, 1997)
  Tidwell, J. Common Ground, 1997. A Pattern Language for Human-Computer Interface Design,. http://www.mit.edu/~jtidwell/common_ground.html
- (Engelberg and Seffah, 2002)
  Engelberg D., and Seffah A., 2002. Design Patterns for the Navigation of Large Information Architectures. 11th Annual Usability Professional Association Conference Orlando, Florida, July 8-12, 2002.
- (Laakso, 2003)
  Sari A. Laakso, 2003. Collection of User Interface Design Patterns University of Helsinki, Dept. of Computer Science, September 16, 2003. http://www.cs.helsinki.fi/u/salaakso/patterns/ .
- (Coram and Lee, 1998)
  Coram T., and Lee J., 1998. Experiences – A Pattern Language for User Interface Design,, at http://www.maplefish.com/todd/papers/experiences .

References (Cont.)

- (Lynch and Horton, 1999)
  Lynch P.J, and Horton S., 1999. Web Style Guide: Basic Design Principles for Creating Web Sites. New Haven and London: Yale University Press.
- (Nielsen, 1999)
  Nielsen J., 1999. Designing Web Usability: The Practice of Simplicity. New Riders
- (Garrido, Rossi and Schwabe, 1997)
  Garrido A., Rossi G. and Schwabe D., 1997. 'Pattern Systems for Hypermedia', Pattern Language of Programming Conference
- (Booch, Rumbaugh and Jacobson, 1999)
  Booch G., Rumbaugh J. and Jacobson I., 1999. *The Unified Modeling Language User Guide*, Addison-Wesley
- (Myers, 1986)
  Myers B. A., 1986. Visual programming, programming by example, and program visualization: A taxonomy. In Proceedings of the ACM CHI'86 Conference on Human Factors in Computing Systems; ACM New York, pp. 271-278; April 1986

References (Cont.)

- (Myers and Buxton , 1986)

  Myers B. A. & Buxton W., 1986, Creating highly-interactive and graphical user interfaces by demonstration, International Conference on Computer Graphics and Interactive Techniques, Proceedings of the 13th annual conference on Computer graphics and interactive techniques, Pages: 249 – 258 .

- (Meyer , 1990)

  Meyer B., 1990. Conception et programmation par objets pour du logiciel de qualité, Inter-Éditions, Paris.

- (www.sun.com)

  Core J2EE Patterns, Retrieved 2006, [Online] available at: http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html

Questions Period

Gracias / Thank you / Merci