Architectures On-Demand for Any Domain Using Stable Software Patterns

FIVE DAYS SEMINAR PROPOSAL

DR. MOHAMED E. FAYAD
Professor of Computer Engineering
Computer Engineering Dept., College of Engineering
San José State University
One Washington Square, San José, CA 95192-0180
Ph: (408) 924-7364, Fax: (408) 924-4153
E-mail: m.fayad@sjsu.edu, mfayad@vrlsoft.com
http://www.engr.sjsu.edu/fayad

ABSTRACT

The lightening growth of technology coupled with the need for a quick turnaround time and demand for low production cost, are exerting a tremendous pressure and an immense desire for software developers, to create new and improved designs to respond to a rapidly changing business environment. Enterprises and developers must try to invest in building stable architectures that can be immediately adapted in many different ways to meet the new and emerging challenges. These kinds of architecture types are called architectures on demand, as they can be adapted depending on the situation to meet the future requirements and changes in creating the system. The primary focus of this seminar is to show how software stability concepts are used to develop on-demand architectures.

The seminar attempts to focus on three key aspects: (1) Enduring Business Themes (EBTs), or business goals and transformations, which we call stable analysis patterns; (2) Business Objects (BOs), or business process designs, which we call stable design patterns; and (3) Industrial Objects (IOs), or application objects. Both EBTs and BOs form a stable core, and, therefore provide architectures on demand for any domain. We shall call these architectures as “stable architectural patterns.”

Our approach is based on an iteratively refined and practically defined framework arrangement that we call the “software stability model,” which consist of three major phases:

1. **Enduring Business Themes (EBTs)** represent the goals of the business and govern business transformation. We shall also call them “**Stable Analysis Patterns**”. Their main impact on business design is many and includes a technique for dividing an enterprise into self-contained business components, in order to identify new opportunities for innovation and improvement, and prioritize business transformation activities. These critical components may span a single department of an enterprise or multiple departments inside an enterprise or an ecosystem of a hub of partner enterprises. These goals will be ultimately achieved through the external adaptation of stable analysis patterns.
2. **Business Objects (BOs)** are the capabilities, or the “workhorses,” of the enterprise. BOs are called “**Stable Design Patterns.**” These usually contain the workflow of the business, business process designs and infrastructure, like hooks, a gang of four pattern, and other similar concepts. BOs mainly focus on the operational aspects of business transformation. They always seek to design and implement flexible processes that integrate cross-divisional and cross-enterprise processes, as well as core process activities. BOs are the keys to business and support processes, such as finance and human resources. Flexibility is realized by separating business logic, application logic and data management in order to reduce and improve the time taken to evaluate new or modified processes. This phase focuses only on business logic and has many extension points called “hooks.”

3. **Industrial Objects (IOs)** represent application logic. Application logic can benefit from standards-based virtualization, automation and integration technologies through hooks that are provided by an on-demand operating environment. Such technologies enable the alignment of IT enabled processes with business priorities, improve asset utilization and reduce the total cost of ownership of the infrastructure, through the instantiation of many applications using the architecture on demand concept.

EBTs and BOs are **stable software patterns** and a combination of EBTs and BOs are known to form the **core knowledge** for a given domain. The core knowledge for any domain is called a **Stable Architectural Pattern** and this can be extended and adapted through hooks. The quality and merit of stable architectural patterns creates competitive advantages through differentiation and productivity. It also integrates partners in order to increase adaptive capabilities.

The seminar answers the following questions:

1. How can we achieve software stability over time and later extend the lifespan of software products in any given domain?
2. How can we build a stable architecture that can be adapted on demand to meet future changes and evolving technologies?
3. What are the relationships between software architecture and software that has been stable over a period of time?
4. What are the relationships between software that has been stable over time and business objects (BOs)?
5. How does one build timeless architectures for any field of knowledge?

The seminar is also designed to discuss several issues that are related to building systems from patterns. This seminar will also answer the following questions:

1. Can the various claims related to building any system from patterns stand?
2. What do we exactly mean when we say “systems of patterns”?
3. What are the different claims that are related to patterns composition, and are they really true?
4. If you would like to build a stable system from patterns, how do you select patterns?
5. What kind of patterns do you choose for building your system from patterns?
6. How do you compose and design patterns?
7. Are there any guidelines and procedures for patterns composition?
8. What other ways will help you build any system from patterns?
TABLE OF CONTENTS

1. Software Stability Model
   1.1 Why Software Stability Modeling?
   1.2 The Importance of Software Stability
   1.3 Problems Facing Today’s Software Development
   1.4 What is Software Stability Modeling?
      1.4.1 Enduring Business Themes
      1.4.2 Business Objects
      1.4.3 Industrial Objects
   1.5 Software Stability Advantages
      1.5.1 Contributing to Stable Architectures
      1.5.2 Aiding in Understanding the Problem?
      1.5.3 Providing Accurate Problem Modeling
      1.5.4 Aiding Engineering of a System of Patterns
   1.6 Software Stability Challenges
   1.7 Documentation Template
   1.8 Pattern Specifications
   1.9 Case Studies

2. Design Session-I

3. Stable Analysis Patterns
   3.1 Patterns: Background and Definition
   3.2 Analysis Patterns: Overview and History
   3.3 General Classifications of Analysis Patterns
   3.4 Our Classification – 3 Groups
   3.5 Major Challenges Facing Analysis Patterns
   3.6 Stable Analysis Patterns
   3.7 Stable Analysis Patterns vs. MetaModels & Data Models
   3.8 Samples of Analysis Patterns: Negotiation, Automation, Searching, etc.

4. Design Session-II

5. Stable Design & Architectural Patterns
   5.1 Problem Space vs. Solution Space
   5.2 Why Design & Architectural Patterns?
   5.3 The Importance of Design and Architectural Patterns
   5.4 Design Patterns Classifications
   5.5 Problems Facing Today’s Design Patterns.
   5.6 What are Stable Design and Architectural Patterns?
   5.7 Samples of Design Patterns: AnyAccount, AnyProject, AnyParty, etc.

Design Session-III

© M.E. Fayad, PhD, 2000-2006
A DESCRIPTION OF TEACHING METHODS: WILL THE COURSE CONSIST OF PRESENTATIONS, DEMONSTRATIONS, GROUP WORK, DISCUSSION, ETC., OR A COMBINATION?

A combination! In addition, course participants will also form teams and get involved in participating in hands-on design sessions that last from 30 minutes to one hour each for every session. A half-day tutorial will consist of one design session and a full-day tutorial will have two design sessions, and a three-day seminar will have 6 design sessions.

WHO SHOULD ATTEND

This seminar is targeted specifically at both computer and software researchers, pattern researchers and users, framework developers, and computer professionals. A short overview on current trends on software stability and creation of architecture on demand for any domain will be provided as an introduction to the seminar. The presentation will then shift on defining the various constituents of these two amazing subjects. These definitions will first take the form of enduring concepts and their properties in a targeted domain and then be transformed as Stable Software Patterns. Although the ability, skill and knowledge to understand basic object-oriented concepts, software engineering principles, software modeling techniques, (OMT, UML, or any object-oriented method) and software architecture, would be of great help, when observing the realization of Stable Software Patterns, no particular or specific knowledge will be needed to follow the conceptual and/or theoretical developments.

PRESENTERS' BIOGRAPHY

MOHAMED FAYAD is a Full Professor of Computer Engineering at San Jose State University from 2002 to till date. He was a J.D. Edwards Professor, Computer Science & Engineering, at the University of Nebraska, Lincoln, from 1999 to 2002, and an associate professor at the computer science and computer engineering faculty at the University of Nevada, from 1995 - 1999. He has 15+ years of industrial experience.

Dr. Fayad is a Senior Member of the IEEE, a Senior Member of the IEEE Computer Society, a Member of the ACM, an IEEE Distinguished Speaker, an Associate Editor, Editorial Advisor, a columnist for The Communications of the ACM and his column is Thinking Objectively, a columnist for Al-Ahram Egyptians Newspaper (2 million subscribers), an Editor-In-Chief for IEEE Computer Society Press - Computer Science and Engineering Practice Press (1995-1997), IASTED Technical Committee member on Software Engineering (2001-2004), a general chair of IEEE/Arab Computer Society International Conference on Computer Systems and Applications (AICCSA 2001), Beirut, Lebanon, June 26-29, 2001, and the president of Arab Computer Society (ACS) from April 04 to present.
Dr. Fayad was a guest editor on nine theme issues:

- CACM's OO Experiences, Oct. 1995,
- CACM's Software Patterns, Oct. 1996,
- CACM's OO Application Frameworks, Oct. 1997,
- ACM Computing Surveys – OO Application Frameworks, March 2000,
- International Journal on Software Practice and Experiences, July 2001,
- IEEE Transaction on Robotics and Automation -- Object-Oriented Methods for Distributed Control Architecture, October 2002, and

He has written and published a number of articles in many journals and magazines, such as IEEE Software, IEEE Computer, JOOP, ACM Computing Surveys and CACM on OO software engineering methods, experiences, aspect-oriented programming, internet & web applications, enterprise and application frameworks, design patterns, and management. He has given tutorials and seminars on OO Technologies and Experiences at many conferences and presented various seminars in the USA and several countries, such as Hong Kong (April 96), Canada (10 times), Bahrain, Saudi Arabia, Egypt (12 times), Portugal (Oct. 96, July 99), Finland (July 99), Mexico (Oct. 98), Argentina (3 times), Chile (00), Peru (02), and Spain (02), Brazil (04).

Dr. Fayad has received an MS and a Ph.D. in computer science from the University of Minnesota at Minneapolis. His research topic was OO Software Engineering: Problems & Perspectives. He is the lead author of several Wiley books like:

- Transition to OO Software Development, August 1998,
- Building Application Frameworks, Sept., 1999,
- and 3 new books in Progress: Stable Analysis Patterns, Stable Design Patterns, Stable Patterns in Action.

YOU WILL GET THE FOLLOWING ITEMS:

1. 1000+ Pages Notes that cover the entire seminar coursework.
2. One Book ($75.00):
   Please check: www.wiley.com/compbooks/frameworks
3. CACM's Theme Issue - Software Patterns, Oct. 96
5. 45 Pages White paper