Towards Stable and Adaptable Software Architectures

One-Day Workshop Proposal

Chair and Point of Contact

DR. MOHAMED E. FAYAD
Professor of Computer Engineering
Computer Engineering Department, 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

EDUARDO M. SEGURA
vrlSoft, Inc.
2065 Martin Ave., Suite 103
Santa Clara, CA 95050-2707
Phone/Fax: (408) 654-8972
E-mail: esegura@vrlsoft.com
http://www.vrlsoft.com

OBJECTIVE AND MOTIVATION

Please refer to:
http://www.engr.sjsu.edu/fayad/workshops/AICCSA-05/

There is little doubt that software engineering, like all other engineering fields, has helped to make life what it is today. With software controlling more equipment and becoming an integral part of our lives, the field of software engineering is quickly becoming more and more important. Unlike many other engineering fields, however, the products produced through software engineering are largely intangible. Also, unlike the products of other engineering fields, software products are unlikely to remain stable over a long period of time.

In hardware areas, the failure rates of products often start high, then drop low, and then go high again. Early in a hardware product's lifecycle, some problems with the system exist. When these problems are fixed, the failure rate of the hardware product drops. However, as hardware gets old, physical deterioration causes the hardware to fail. In other words, the hardware wears out and the failure rate rises again.

Software, on the other hand, is not subject to the same wear and tear that hardware is. No environmental factors cause software to break. Software is a set of instructions, or a recipe, for a piece of hardware to follow. There are no moving parts in software. There is nothing that can physically deteriorate. Software should not wear out. Unfortunately, it does. Countless authors in the field of software engineering have identified this problem. However, the software engineering techniques
Towards Stable and Adaptable Software Architectures – Workshop Proposal

Outlined by many software-engineering authors have not achieved a large amount of stability in software projects.

This problem is more than just an inconvenience for software engineers and software users. The reengineering that is required for these software products does not come without a price. Not uncommonly, these reengineering projects cost hundreds of thousands to millions of dollars. This does not take into account the time that is wasted by this continual reengineering process.

Software defects and "deterioration" are caused by changes in software. Many of these changes cannot be avoided, but they can be minimized. Currently, when a change must be made to a software program, the entire program is often reengineered. It does not matter if the change required is due to new technology or to a change in clientele. This reengineering process is ridiculous. The core purpose of the software product has not changed. Why, then, must the entire project be reengineered to incorporate a change?

This workshop will examine software stability with respect to four central themes: "How can we engineer software systems that are stable overtime?"; "How does one make software systems stable over time?"; "What is a stable software pattern?" and “What is the impact of software stability on new technologies, such as aspect-oriented architecture and programming, multi-agent technology, constraints-oriented software development, component-based software development, application and enterprise frameworks developments, and many more?

The workshop will debate several issues related to stability, such as how to build stable software systems and how to generate stable model-based architectures. We want researchers, framework developers, and application developers to answer the following questions:

1. What are the relationships between software architecture and software that have been stable over time?
2. What are the relationships between management workflow and software that have been stable over time?
3. How can we achieve software stability over time and extend the lifespan of software products?
4. What are the relationships between software that has been stable over time and business objects?
5. What are the roles of object-oriented techniques and technologies of making software stable over time?
6. What are the approaches to making software stable over time?
7. What is the relationship between software stability and various new technologies, such as aspect-oriented architecture and programming, constraints programming, multi-agent-oriented software developments, component-based software developments, and others?
8. What is the relationship between application frameworks and software stability?
9. What are the impacts of software stability on understanding the customers’ needs?
10. What is the impact of software stability on scalability, customizability, extensibility, integratability, and configurability?

In addition to the above themes, we invite papers on both theoretical and practical aspects relevant to software stability. Topics include (but are not restricted to):

- Theories of software stability
- Stable software architectures
Towards Stable and Adaptable Software Architectures – Workshop Proposal

- Model-based software reuse
- Impact of stability on reuse
- Case studies of the building stable software
- Stability patterns

PREVIOUS WORKSHOPS
Previous workshop on the same topic:

PARTICIPANTS
The intended audience is software engineering researchers, software analysts, software architects, software designers, software developers, Object-Oriented technologists, and software methodologists. Others who may be interested are welcome to participate. Participants are expected to have some adequate background on software modeling, object-oriented technology, and software patterns. Familiarity with UML is also preferred. The expected number of participants is 40+

In the era of rapid technology changes, it is more necessarily than ever to accomplish stability when building software systems. Stable systems can ensure long-term functionality in the sense that they can be adapted and extended without rebuilding the whole system from scratch. As a result, cost, time, and effort can be reduced. There is little doubt that building stable systems is one of the main concerns not only for information systems developers, but for any software system development. Therefore, we expect this workshop to attract a large number of participants who would like to investigate the state of the art in software development.

SOLICITATION, SUBMISSION AND SELECTION PROCESS
We will have the following:
1. General invitation: We will have a call for papers that will be publicized to over 10,000+ and will be sent to the patterns groups.
2. Special invitation: We will have a large number of people in the pattern community receiving the call for papers.
3. Very special invitation: We will send an invitation to some people to participate and submit to the workshop.

Each submission will be reviewed by at least three reviewers. Based on the received reviews, the organizers will select the accepted papers.

**WORKSHOP FORMAT**

The one-day workshop will consist of invited speakers and single-track presentation sessions. The plan is to have half the time devoted to presentations and the other half to discussion. The theme of the sessions will be determined based on the position statements. While not all accepted position statements are expected to be presented, they will be available on the web before the conference. A summary report on the workshop will also be posted on the web.

(please refer to the tentative agenda of the workshop in the preliminary call for papers at the end of this proposal)

**REQUESTED EQUIPMENT**

There are no special requirements. Standard equipment for PowerPoint presentations and an overhead projector is sufficient.

**LIST OF REFERENCES:**

**BOOKS (Reviewed):**


**COLUMNS:**

**Software Stability Sequence:**

Stable Software Patterns Sequence:


THEME ISSUES (GUEST EDITOR)

1. "Object-Oriented Experiences" with W.T. Tsai, CACM, October 95.
PRELIMINARY CALL FOR PAPERS

COMPSAC '07 Third Workshop on
Towards Stable and Adaptable Software Architectures

Call for Papers
COMPSAC '07 Full day workshop
31st Annual International Computer Software and Applications Conference
(COMPSAC 2007)
Beijing, July 24-27, 2007

There is little doubt that software engineering, like all other engineering fields, has helped to make life what it is today. With software controlling more equipment and becoming an integral part of more of our lives, the field of software engineering is quickly increasing in importance. Unlike many other engineering fields, however, the products produced through software engineering are largely intangible. Also, unlike the products of other engineering fields, software products are unlikely to remain stable over a long period of time.

In hardware areas, the failure rates of products often start high, then drop low, and then go high again. Early in a hardware product's lifecycle, some problems with the system exist. When these problems are fixed, the failure rate of the hardware product drops. However, as hardware gets old, physical deterioration causes the hardware to fail. In other words, the hardware wears out and the failure rate rises.

Software, on the other hand, is not subject to the same wear and tear that hardware is. No environmental factors cause software to break. Software is a set of instructions, or a recipe, for a piece of hardware to follow. There are no moving parts in software. There is nothing that can physically deteriorate. Software should never wear out, but often it does. Countless authors in the field of software engineering have identified this problem. However, the software engineering techniques outlined by many software-engineering authors have not achieved an acceptable level of stability in software projects.

This problem is more than just an inconvenience for software engineers and software users. The reengineering that is required for those software products does not come without a price. Not uncommonly, these reengineering projects cost hundreds of thousands to millions of dollars. This does not take into account the time that is wasted by this continual reengineering process. Software defects and "deterioration" are caused by changes in the software. Many of these changes cannot be avoided. These changes can be minimized, however. Currently, when a change must be made to a software program, the entire program is often reengineered. It does not matter if the change required is due to new technology or due to a change in clientele. This reengineering process is ridiculous. The core purpose of the software product has not changed. Why, then, must the entire project be reengineered to incorporate a change?
Towards Stable and Adaptable Software Architectures – Workshop Proposal

The workshop will debate several issues related to stability, such as how to build stable software systems and how to generate stable model-based architectures. We want researchers, framework developers, and application developers to answer the following questions:

11. What are the relationships between software architecture and software that have been stable over time?
12. What are the relationships between management workflow and software that have been stable over time?
13. How can we achieve software stability over time and extend the lifespan of software products?
14. What are the relationships between software that has been stable over time and business objects?
15. What are the roles of object-oriented techniques and technologies of making software stable over time?
16. What are the approaches to making software stable over time?
17. What is the relationship between software stability and various new technologies, such as aspect-oriented architecture and programming, constraints programming, multi-agent-oriented software developments, component-based software developments, and others?
18. What is the relationship between application frameworks and software stability?
19. What are the impacts of software stability on understanding the customers’ needs?
20. What is the impact of software stability on scalability, customizability, extensibility, integratability, and configurability?

In addition to the above themes, we invite papers on both theoretical and practical aspects relevant to software stability. Topics include (but are not restricted to):

- Theories of software stability
- Stable software architectures
- Model-based software reuse
- Impact of stability on reuse
- Case studies of the building stable software
- Stability patterns

Check my website for all columns and accepted position papers (under construction):

www.engr.sjsu.edu/~fayad/workshops/compsac07/ws1

**PAPER FORMAT AND SUBMISSION**

People interested in participating in the workshop are requested to submit a short position paper (3-5 pages) or regular workshop paper (limited to 10 pages, double space, including figures) representing the views and experiences relevant to the discussion topic. The title page should include a maximum 150-word abstract, five keywords, full mailing address, e-mail address, phone number, fax number, and a designated contact author. Papers will be selected depending on the originality, quality and relevance to the workshop. Interesting and relevant papers will be selected by the organizers for presentation at the workshop. To foster lively discussions, each author is encouraged to present open questions and one or two main statements that will be discussed at the workshop. Submissions must be either MS-Word or RTF
Towards Stable and Adaptable Software Architectures – Workshop Proposal

formats (please, DO NOT compress files). Initial submission can be made by emailing a URL pointing to an HTML version of the paper.

Depending on the number and spread of contributions, the scope may be narrowed to ensure effective communication and information sharing. Accepted position papers will be published in the workshop proceedings to be distributed to the participants before the workshop, and made generally available through the WWW and FTP. A workshop report will be published in the addendum proceedings of the conference.

People interested in participating in the workshop without submission are requested to fill out the participation form and e-mail to the workshop co-chair esegura@vrlsoft.com or to the workshop chair at m.fayad@sjsu.edu.

PARTICIPATION FORM:
Name and affiliation:
Position:
Address:
E-mail:
URL:
Areas of interest:
Why would you like to participate?

Please note that registration is required in order to participate in the workshop. Early registration discount is available. An overhead projector and a flipchart will be available.

For more information please visit (under construction)
http://www.engr.sjsu.edu/~fayad/workshops/compsac07/ws1/cfps

You may also contact the organizers.

PROPOSED AGENDA

1. Welcome and introduction of participants. The organizers will first give a short overview of open issues and of the main arguments arising out of the position papers. (Estimated time: 20-30 minutes)
2. Selected authors (representing the main trends) will be given about 20 minutes to explain how their position relates to other positions and what each sees as the three major issues. We expect between about 5-10 position papers. (Estimated time: 120-130 minutes)
3. The organizers will propose an identification of the major issues, and the participants will then discuss and select what they hold to be the hottest issues to be examined. (Estimated time: 10-15 minutes)
Towards Stable and Adaptable Software Architectures – Workshop Proposal

4. The participants will work for 70-95 minutes in small groups, with a designated moderator in each group. The groups will each deal with two different hot issues identified, and will produce a summary in the form of points and counterpoints, showing either how several views are irreducibly opposed or how they are complementary. The number of groups will depend on the number of participants and number of issues selected; ideally there should be 3-5 persons in each group. (Estimated time: 60-70 minutes)

5. Each group will be given 10-15 minutes to present its findings to the workshop. A closing discussion will follow. The workshop report will be written on the basis of these findings and will include an agenda for future exploration and cooperation; it will be made available through the WWW and FTP. (Estimated time: 50-60 minutes for five teams)

(Total estimated time: 285-315 minutes, i.e. about five hours +/- 15 minutes; lunch and breaks are not included.)

IMPORTANT DATES -- Will be updated based on acceptance process
Submission deadline May 10, 2007
Acceptance notification May 30, 2007
Camera-ready paper due June 15, 2007
Workshop date: TBD

ORGANIZERS

Chair and Point of Contact:

DR. M.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

Co-Chair:

EDUARDO M. SEGURA
vrlSoft, Inc.
2065 Martin Ave., Suite 103
Santa Clara, CA 95050-2707
Phone/Fax: (408) 654-8972
E-mail: esegura@vrlsoft.com
http://www.vrlsoft.com

ORGANIZERS BIOGRAPHY

DR. M.E. FAYAD is a Full Professor of Computer Engineering at San Jose State University from 2002 to present. He was a J.D. Edwards Professor, Computer Science & Engineering, at the University of
Towards Stable and Adaptable Software Architectures – Workshop Proposal

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, and a Columnist for The Communications of the ACM and his column is Thinking Objectively, and 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 he is 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, IEEE Computer's Managing OO Software Development Projects, Sept. 1996, CACM's Software Patterns, Oct. 1996, CACM's OO Application Frameworks, Oct. 1997, ACM Computing Surveys – OO Application Frameworks, March 2000, IEEE Software - Software Engineering in-the-small, Sept./Oct. 2000, and 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 Annals of Software Engineering Journal – OO Web-Based Software Engineering, October 2002. He has published 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 he has presented various seminars in 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 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: Transition to OO Software Development, August 1998, Building Application Frameworks, Sept., 1999, Implementing Application Frameworks, Sept., 1999, Domain-Specific Application Frameworks, Oct., 1999, and 3 new books in Progress: Stable Analysis Patterns, Stable Design Patterns, Stable Patterns in Action.

Workshop Co-Chair:

EDUARDO M. SEGURA
Lead Architect and Project Manager - vrlSoft, Inc.
2065 Martin Avenue, Suite 103, Santa Clara CA 95050
Ph: (408) 654-8972 (office)
E-mail: esegura@vrlsoft.com

Eduardo Segura will get his MS in software engineering, from the University of San Jose, California. His BE research topic was Alternatives Methods for Congestion Control in Windows®, which applied ideas from TCP-Vegas to a new algorithm for congestion control for Windows. He collaborated with Dr. M. E. Fayad for a chapter on a book about Model-Driven
Architectures (to be published). He is also co-author of several papers on stable analysis and design patterns, to be published in the next months.