

Focus Group on Patterns Languages: Addressing Challenges (PLAC 2010)

2-Hour Focus Group Proposal

VIJAY KUMER ERANTI

Google Inc.,
Mountain View, CA, USA
evijaykumar@yahoo.com

SHIVANSHU K. SINGH (PRIMARY CONTACT)

Department of Computer Engineering,
Charles W. Davidson College of Engineering,
San Jose State University,
One Washington Square, San José, CA 95192-0180.
Ph: (408) 701-8294
E-mail: shivanshukumar@gmail.com, shivanshu@vrlsoft.com
URL: <http://www.shivanshusingh.com>

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
URL: <http://www.engr.sjsu.edu/fayad>
pattern.ijop.org – **Pattern Blog**
fayadsblog.vrlsoft.com – Fayad's Report

INTRODUCTION

A pattern language consists of a cascade or *hierarchy of parts*, linked closely together by *patterns*, which solve generically recurring problems that are associated with the parts. Each pattern has a title, and collectively the titles form a language for design [1] Pattern Languages are in life, simply a collection of interrelated patterns [2]. These interrelated patterns are combined in any way and combination to create new environments, where practitioners can solve context-specific problems. Precisely, the concept of pattern languages has invaded over into the software engineering field, to describe prior

experiences and the processes that stem from them, in a very simple language, where patterns are tactfully woven as a whole, and can be combined in any manner to solve a particular and complex problem. Yet, this process is still done in an ad-hoc manner and is not straightforward enough, to ease and speed up the software development process.

Thus, this focus group subject is driven forward by three main questions. First, how can we classify, develop, and utilize analysis and design patterns together towards the path of a problem resolution? Second, what is the “behind-the-“language that guides the sewing of patterns together as a whole? And third, how can we overcome and face challenges, other than patterns composition problems (patterns traceability, etc.) that can hinder the development of a system of patterns? The inherent inability to answer these questions detrimentally impacts the understanding of how to put patterns in real practice, and will therefore make software patterns’ use more complex than it should.

OBJECTIVE AND MOTIVATION

Building high quality systems is not an easy exercise, specifically when several factors can undermine their quality success, such as cost, time, and lack of systematic approaches. The potential promise of using software patterns in software development to deal with these aforementioned obstacles, has led software practitioners to steadfastly believe in the power of pattern languages, as the means for constructing complex systems in a constrained environment.

Software Patterns, along with Pattern Languages, have recently attracted software practitioners for more than a decade. They have seen software patterns and pattern languages as really promising techniques that ease and speed up their software development [2, 3, 4, and 5]. However, developing robust software patterns and pattern languages has not reached the expected ease and flexibility it should have been, when dealing with determined problems; instead, they construct models that specifically lack some essential qualities that diminish the overall quality of the system rather than improving it [6].

The concept of Pattern Languages [3, 7, 8, and 9] is spilling over into the software engineering field, to highlight software development’s prior experiences or best practices, using a coherent language that can be used for both discussing about a particular problem and also in creating new environments from the patterns it conveys. This language works by connecting a collection of patterns, as if they were in a detailed, narrated story. Each of the patterns in the collection is an insightful and a novel way to manage or solve a set of recurrent problems in a particular context [2, 3, and 5]. As a whole, they make clearly visible both the knowledge that is pertinent to a particular domain, and the solutions for a set of recurrent problems.

Pattern languages have recently emerged as a promising classification technique and in providing ways to build frameworks. However, there are problems [10], such as:

1. Context's missing indicators/guidelines for in-context patterns selection within the pattern language.
2. Classifications of patterns' rationale within the pattern language structure is also missing.
3. Traceability is also lost, especially when dealing with deeper levels of pattern language implementation.
4. No systematic way for compositing patterns, similar or different, to build software architectures
5. There is a loss of generality in traditional pattern languages.
6. Pattern language's struggle in providing full software maintainability and stability.
7. How pattern languages deal with the problem they address is neither straightforward nor easy.
8. There is no classification in pattern languages
9. Pattern languages don't distinguish between associate and remote knowledge.

The focus group will address pattern languages' challenges and debate several issues related to the following questions. We also want researchers, framework developers, and application developers to discuss and debate the following questions related to:

I. Pattern Languages Creation and Development

- a. Leaving experience claim on the side, can you show how to create and develop pattern languages?
- b. What are the bases of creating pattern Languages?
- c. Are there any guidelines, methodologies, and/or processes for pattern language creation and development?
- d. Would you show an example or two of systematic and non-systematic pattern languages?
- e. What is the starting point of any pattern language?
- f. What are the components of any pattern language?
- g. What kind of patterns that appear in pattern languages?

II. Pattern Languages Selection Process:

- a. How does one select analysis and design patterns to create a pattern language?
- b. What is the main basis for selecting these patterns into the pattern language?
- c. If someone would like to build a system from patterns, how does he/she select patterns from the pattern language?

- d. What kind of patterns should one select to build a system from patterns?
- e. Is there a guideline for the selection process from a pattern language?

III. Patterns Languages Composition

- a. How does one integrate the selected pattern languages to build any system? Or how does one compose any system from one or more pattern languages?
- b. What are the various claims related to pattern languages composition? Are they really true?
- c. Are there set guidelines or techniques for pattern languages composition? Would you illustrate the ways to use them?

IV. System of Patterns and General Reuse

- a. What do we mean, when we say “systems of patterns”?
- b. Are the various claims related to building any system from pattern languages reasonable?
- c. How to develop pattern repositories and catalogs, from which pattern languages can be retrieved and reused?
- d. Are there any automated approaches for patterns using languages mining and integration?
- e. What other concepts will help build any system from pattern languages?
- f. Can patterns within a given pattern language appear in other remote pattern languages?
- g. Is it possible to create many architectures from a given pattern languages? If so, how many architectures?
- h. Can we measure the ROI from the pattern language of a given domain?
- i. Is it possible to measure or perform cost estimation using pattern languages?
- j. It is possible to insert the quality factors with the pattern languages? How?

V. Impacts

- a. What is the overall impact of software stability on the above challenges?

PARTICIPANTS

The intended audiences for this focus group are software engineering researchers, software analysts, software architects, software designers, software developers, object-oriented technologists, and software methodologists. All others who are interested are welcome to participate. Participants are expected to possess adequate background and experience in software modeling, object-oriented technology, and software patterns. Familiarity with UML is also preferred. The expected number of participants is 20+.

SOLICITATION, SUBMISSION AND SELECTION PROCESS

We will have the following:

1. General invitation: We will have a call for papers that will be publicized over electronically and will be sent to all the patterns groups, using SEWORLD and other appropriate mailing-lists.
2. Special invitation: We will invite a large number of people in the pattern community with a call for papers.
3. Very special invitation: We will send a special invitation to selected people to participate and submit their papers to the focus group.

Each submission will be reviewed by at least three reviewers. Based on the received reviews and opinions, the organizers will choose the accepted papers for presentation.

FOCUS GROUP FORMAT

The two-hour focus group will consist of invited speakers and single-track presentation sessions. The overall action plan is to have an open session for presentations of position papers and discussion. The main theme of the sessions will be determined based on the position statements. All accepted position statements are expected to be presented in the focus group. A summary report on the focus group will also be posted on the web. (Please refer to the tentative agenda of the focus group in the preliminary call for papers given at the end of this proposal)

PREVIOUS WORKSHOPS

Previous workshop on the same topic:

1. "Accomplishing Software Stability Workshop" with M. Laitinen, OOPSLA '99, Denver, Colorado, Nov 1999. The number of participants was 27.
2. "Software Stability: Timeless Architectures, Systems of Patterns, and Model-Based for Reuse" Mohamed E. Fayad and Haitham S. Hamza, IEEE IRI 03, Las Vegas, Nevada, October 2003.
3. **M.E. Fayad** and H.S. Hamza. AICCSA '05 First Workshop on Software Stability: Timeless Architectures and Systems of Patterns. AICCSA '05 Full day workshop, The 3rd ACS/IEEE International Conference on Computer Science Systems and Applications, January 3, 2005, Cairo, Egypt. <http://www.engr.sjsu.edu/fayad/workshops/AICCSA-05/>
4. **M.E. Fayad** and H.S. Hamza. IEEE '05 Second Workshop on Software Stability @ Work. IEEE IRI '05 Full day workshop, The 2005 IEEE International Conference on Information Reuse and Integration (IEEE IRI '05), August 15-17, 2005, Las Vegas, NV, USA.
5. **M.E. Fayad** and H.S. Hamza. ECOOP 2005 First Workshop on Building Systems Using Patterns: Examine the Illustrious Claim. Full day workshop, 2005 19th European Conference on Object-Oriented Programming (ECOOP 2005), July 25-29, 2005, Glasgow, Scotland. <http://2005.ecoop.org/workshops.html>, and <http://www.engr.sjsu.edu/fayad/workshops/ECOOP05/>
6. **M.E. Fayad, I.A. Zualkernan.** and H.S. Hamza. AICCSA06 5th International Workshop on Software Stability: Methodologies, Applications and Tools. Full day workshop, 2006

Focus Group on Pattern Languages: Addressing Challenges (PLAC 2010)
Focus Group Proposal

4th ACS/IEEE International Conference on Computer Science Systems and Applications, March 8-11, 2006, Sharjah, United Arab Emirates <http://www.cs.utk.edu/aiccsa06/>

7. **M.E. Fayad, H.S. Hamza, and E. Segura.** The First IEEE International Workshop on Software Pattern: Addressing Challenges (SPAC 07). In conjunction with COMPSAC 2007 -- Full day workshop, Beijing, July 24-27, 2007. <http://conferences.computer.org/compsac/2007/workshops/SPAC>
8. **Licia Capra, Rami Bahsoon, Wolfgang Emmerich, M.E. Fayad:** The international workshop on software architectures and mobility (SAM 2008). *ICSE Companion 2008*: 1033-1034 <http://www.cs.bham.ac.uk/~rzb/sam.htm>

REQUESTED EQUIPMENT

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

ORGANIZERS' BIOGRAPHIES

SHIVANSHU K. SINGH is a Co-Founder and Senior Researcher at vrlSoft, Inc., Palo Alto, California, USA. His research is focused on the areas of Unified Software Engines, Software Architecture, Architectural and Stable patterns, Knowledge Maps, Spatiotemporal databases, Software Engineering processes, Requirements Engineering, Collaborative Systems and more. Shivanshu received his Bachelor of Technology degree in Information and Communication Technology from Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India in 2007 and has 3+ years of professional experience in software engineering, research and development and teaching. He is involved in the development of some major journals in the field of software engineering and multiple new business developments. He has multiple journal, conference and column publications in his name. He is a columnist in the International Journal of Software Architecture (IJSA) and the Editor in Chief for the International Journal of Unified Software Engines (IJUSE). He is involved in writing several successful research and business proposals to various government agencies, institutions, public and private organizations, NSF, SBIR and industrial proposals etc. and is also in the process of writing two books, to be titled 'Knowledge Maps' and 'The Unified Software Engine'. Shivanshu is the Lead - Research and Development, of the research team, with more than 10 graduate students, working on several areas related to Software Engineering research, at San Jose State University. He is an invited member of the Phi Kappa Phi Honor Society, for academic excellence; He is currently working towards his Master of Science degree in Software Engineering at San Jose State University, San Jose, California and towards a PhD degree thereafter.

DR. M.E. FAYAD is a Full Professor of Computer Engineering at San Jose State University from 2002 to present. He is one of the **founders and president** of *Arab Computer Society (ACS)* from April 04 to April 2007. Dr. Fayad is *a known and well recognized authority* in the domain of theory and the applications of software engineering. Fayad's publications are in the very core, archival journals and conferences in the software engineering field. Dr. Fayad has published more than **218** high quality papers, that includes more than 40 profound reports in reputed journals, and 90 advanced articles in refereed conferences, more than 20 journal columns, 16

*Focus Group on Pattern Languages: Addressing Challenges (PLAC 2010)
Focus Group Proposal*

blogged columns; 9 well-cited theme issues in prestigious journals and flagship magazines, 24 different workshops in very respected conferences, over 125 tutorials, seminars, and short presentations in 20+ different countries, a founder of 5 new online journals, NASA Red Team Review of QRAS and NSF-USA Research Delegations' Workshops to Argentina and Chili and four authoritative books, of which three of them are translated into different languages such as Chinese. **Dr. Fayad** received an MS and a Ph.D. in computer science from the University of Minnesota at Minneapolis. He is the lead author of several classic Wiley books.

References:

- [1] <http://www.designmatrix.com/pl/anatomy.html>
- [2] D. C. Schmidt, M.E. Fayad, R.E. Johnson October 1996, Software Patterns, Communications of the ACM, Volume 39 Issue 10
- [3] B. Appleton, Patterns and Software: Essential Concepts and Terminology, <http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html>
- [4] J.O. Coplien, "Software Patterns", BellSouth Laboratories, The Hillside Group.
- [5] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995.
- [6] S. Wu, H. Hamza, and M.E. Fayad. "Implementing Pattern Languages Using Stability Concepts," ChiliPLoP 03', Carefree, Arizona, USA, March 2003.
- [7] N.A. Salingaros, The Structure of Pattern Languages, <http://www.math.utsa.edu/sphere/salingar/StructurePattern.html>.
- [8] S. Fincher, What is a Pattern Language?, Chi'99, ACM SIGCHI Conference on Human Factors in Computing Systems May 15-20, 1999 David Lawrence Convention Center Pittsburgh, Pennsylvania, USA
- [9] F. Buschmann, et, al., Pattern-oriented Software Architectures: A System of Patterns. Chichester ; New York : Wiley, ©1996.
- [10] M.E. Fayad, H. Sanchez "The Knowledge Map", In progress, 2007.

=====

CALL FOR PAPERS

**Focus Group on Patterns Languages:
Addressing Challenges
(PLAC 2010)**

*Reno/Tahoe, Nevada, USA, October 17 - 21, 2010
(in conjunction with SPLASH 2010)*

INTRODUCTION

Focus Group on Pattern Languages: Addressing Challenges (PLAC 2010)
Focus Group Proposal

A pattern language consists of a cascade or *hierarchy of parts*, linked closely together by *patterns*, which solve generically recurring problems that are associated with the parts. Each pattern has a title, and collectively the titles form a language for design [1]. Pattern Languages are simply a collection of interrelated patterns [2]. These interrelated patterns are combined in any way and combination to create new environments, where practitioners can solve context-specific problems. Precisely, the concept of pattern languages has invaded over into the software engineering field, to describe prior experiences and the processes that stem from them, in a very simple language, where patterns are tactfully woven as a whole, and can be combined in any manner to solve a particular and complex problem. Yet, this process is still done in an ad-hoc manner and is not straightforward enough, to ease and speed up the software development process.

Thus, this focus group subject is driven forward by three main questions. First, how can we classify, develop, and utilize analysis and design patterns together towards the path of a problem resolution? Second, what is the “behind-the-” language that guides the sewing of patterns together as a whole? And third, how can we overcome and face challenges, other than patterns composition problems (patterns traceability, etc.) that can hinder the development of a system of patterns? The inherent inability to answer these questions detrimentally impacts the understanding of how to put patterns in real practice, and will therefore make software patterns’ use more complex than it should.

OBJECTIVE AND MOTIVATION

Building high quality systems is not an easy exercise, specifically when several factors can undermine their quality success, such as cost, time, and lack of systematic approaches. The potential promise of using software patterns in software development to deal with these aforementioned obstacles, has led software practitioners to steadfastly believe in the power of pattern languages, as the means for constructing complex systems in a constrained environment.

Software Patterns, along with Pattern Languages, have recently attracted software practitioners for more than a decade. They have seen software patterns and pattern languages as really promising techniques that ease and speed up their software development [2, 3, 4, and 5]. However, developing robust software patterns and pattern languages has not reached the expected ease and flexibility it should have been, when dealing with determined problems; instead, they construct models that specifically lack some essential qualities that diminish the overall quality of the system rather than improving it [6].

The concept of Pattern Languages [3, 7, 8, and 9] is spilling over into the software engineering field, to highlight software development’s prior experiences or best practices, using a coherent language that can be used for both discussing about a particular problem and also in creating new environments from the patterns it conveys. This language works by connecting a collection of patterns, as if they were in a detailed, narrated story. Each of the patterns in the collection is an insightful and a novel way to manage or solve a set of recurrent problems in a particular context [2, 3, and 5]. As a whole, they make clearly visible both the knowledge that is pertinent to a particular domain, and the solutions for a set of recurrent problems.

Focus Group on Pattern Languages: Addressing Challenges (PLAC 2010)
Focus Group Proposal

Pattern languages have emerged as a promising classification technique and in providing ways to build frameworks. However, there are a number of problems [10], such as:

1. Context's missing indicators/guidelines for in-context patterns selection within the pattern language.
2. Classifications of patterns' rationale within the pattern language structure is also missing.
3. Traceability is lost, especially when dealing with deeper levels of pattern language implementation.
4. No systematic way for compositing these patterns, similar or different, to build software architectures
5. There is a loss of generality in traditional pattern languages.
6. Pattern languages struggle and conflict in providing full software maintainability and stability.
7. How pattern languages deal with the problem they address is neither straightforward nor easy.
8. There is no set classification in pattern languages.
9. Pattern languages don't distinguish between associated and remote knowledge.

The focus group will address pattern languages' challenges and debate several issues related to the following questions. We want researchers, framework developers, and application developers to discuss and debate the following questions related to:

VI. Pattern Languages Creation and Development

- a. Leaving career experience claim on the side, can you show how to create and develop pattern languages?
- b. What are the bases of creating pattern Languages?
- c. Are there guidelines, methodologies, and/or processes for pattern language creations and developments?
- d. Would you show an example or two of systematic and non-systematic pattern languages?
- e. What is the starting point of any pattern language?
- f. What are the components of any pattern language?
- g. What kind of patterns that appear in pattern languages?

VII. Pattern Languages Selection Process:

- a. How does one select analysis and design patterns to create a pattern language?
- b. What is the basis for selecting these patterns into the pattern language?
- c. If someone would like to build a system from patterns, how does he/she select patterns from the pattern language?
- d. What kind of patterns should one select to build a system from patterns?
- e. Is there a guideline for the selection process from a pattern language?

VIII. Patterns Languages Composition

Focus Group on Pattern Languages: Addressing Challenges (PLAC 2010)
Focus Group Proposal

- a. How does one integrate the selected pattern languages to build any given system? Or how does one compose any system from one or more pattern languages?
- b. What are the various claims related to pattern languages composition? Are they really true?
- c. Are there any guidelines or techniques for pattern languages composition? Would you illustrate how to use them?

IX. System of Patterns and General Reuse

- a. What do we mean, when we say “systems of patterns”?
- b. Are the various claims related to building any system from pattern languages reasonable?
- c. How to develop pattern repositories and catalogs, from which pattern languages can be retrieved and reused?
- d. Are there any automated approaches for patterns using languages mining and integration?
- e. What other concepts will help assist build any system from pattern languages?
- f. Can patterns within a given pattern language appear in other remote pattern languages?
- g. Is it possible to create many architectures from a given pattern languages? How many architectures?
- h. Can we measure the ROI from the pattern language of a given domain?
- i. Is it possible to measure or perform cost estimation using pattern languages?
- j. It is possible to insert the quality factors with the pattern languages? How?

X. Impacts

- a. What is the impact of software stability on the above mentioned challenges and software quality factors?

PAPER FORMAT: SUBMISSION & PARTICIPATION

Developers and programmers, who are interested in participating in the focus group, are requested to submit a short position paper (**2-3 pages**), by representing views and experiences that are relevant to the given discussion topic. Please include full mailing address, e-mail address, phone number, fax number, and a designated contact author. Focus group papers will be selected depending on their relevance to the focus group theme. Papers should be submitted electronically by e-mailing it to the lead organizer of the focus group: [Shivanshu K. Singh](mailto:Shivanshu.K.Singh@shivanshukumar@gmail.com) <shivanshukumar@gmail.com>. We also encourage authors to present novel and fresh ideas, critiques of existing work, and practical studies.

Each accepted focus group paper must be presented in the person, either by the author or by one of the co-authors. To foster and promote lively discussions, authors are encouraged to present open-ended questions and one or two main statements for the purpose of discussion at the focus group. Submissions must be made either in MS-Word or RTF formats (please, DO NOT compress files).

*Focus Group on Pattern Languages: Addressing Challenges (PLAC 2010)
Focus Group Proposal*

People who are interested in participating in the focus group, without making any submissions are requested to fill out the participation form and e-mail to any of the focus group organizers.

PARTICIPATION FORM:

Name and Affiliation:

Position:

Address:

E-mail:

URL:

Areas of interest:

Reasons for Attending?

For more information please visit any of the following websites:

<http://www.hillside.net/plop/2010/>

You may also contact the organizers, either by e mail or by phone.

PROPOSED AGENDA

1. Welcome and introduction of participants. The organizers will first provide a short overview of all open issues, and also of the main arguments arising out of the position papers. (Estimated time: 10-15 minutes)
2. Selected authors (who'll be representing the main trends) will be allotted 10-15 minutes, to explain and discuss their position paper with the audience. We are expecting about 5-10 position papers in this session. (Estimated time: 90 minutes)
3. The organizers will also propose an identification process of the major issues, and the participants will then discuss, choose and select what they perceive are the hottest issues to be examined and analyzed. (Estimated time: 10-15 minutes)

(Total estimated time: 120 minutes, i.e. about two hours +/- 5 minutes)

IMPORTANT DATES -- Will be decided based on acceptance process.

ORGANIZERS: Point of Contacts:

VIJAY KUMER ERANTI

Google Inc.,

Mountain View, CA, USA

evijaykumar@yahoo.com

SHIVANSHU K. SINGH (PRIMARY CONTACT)

Department of Computer Engineering,

Focus Group on Pattern Languages: Addressing Challenges (PLAC 2010)
Focus Group Proposal

Charles W. Davidson College of Engineering,
San Jose State University,
One Washington Square, San José, CA 95192-0180.
Ph: (408) 701-8294
E-mail: shivanshukumar@gmail.com, shivanshu@vrlsoft.com
URL: <http://www.shivanshusingh.com>

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, mefayad@gmail.com
URL: <http://www.engr.sjsu.edu/fayad>
pattern.ijop.org - Pattern Blog
fayadsblog.vrlsoft.com - Fayad's Report