Software Engineering Group
Department of Computer Science
Nanjing University
http://seg.nju.edu.cn

Technical Report No. NJU-SEG-2012-IC-008

Verifying Aspect-Oriented Activity Diagrams Against Crosscutting Properties with Petri Net Analyzer

Zhanqi Cui, Linzhang Wang, Xi Liu, Lei Bu,
Jianhua Zhao, Xuandong Li

Postprint Version. Originally Published in:
International Conference on
Software Engineering & Knowledge Engineering
Knowledge Systems Institute Graduate School, 2012, pp. 369-374


SEKE
San Francisco Bay
July 1-3
2012
Program for the Twenty-Fourth
International Conference on
Software Engineering &
Knowledge Engineering

PROCEEDINGS

SEKE 2012

The 24th International Conference on Software Engineering & Knowledge Engineering

Sponsored by

Knowledge Systems Institute Graduate School, USA

Technical Program

July 1-3, 2012

Hotel Sofitel, Redwood City, San Francisco Bay, USA

Organized by

Knowledge Systems Institute Graduate School

ISBN-10: 1-891706-31-4 (paper)
ISBN-13: 978-1-891706-31-8

Additional Copies can be ordered from:
Knowledge Systems Institute Graduate School
3420 Main Street
Skokie, IL 60076, USA
Tel:+1-847-679-3135
Fax:+1-847-679-3166
Email:office@ksi.edu
http://www.ksi.edu

Proceedings preparation, editing and printing are sponsored by
Knowledge Systems Institute Graduate School

Printed by Knowledge Systems Institute Graduate School

Foreword

This year marks the 24th anniversary for the International Conference on Software Engineering and Knowledge Engineering (SEKE). For nearly a quarter of century, SEKE has established itself as a major international forum to foster, among academia, industry, and government agencies, discussion and exchange of ideas, research results and experience in software engineering and knowledge engineering. The SEKE community has grown to become a very important and influential source of ideas and innovations on the interplays between software engineering and knowledge engineering, and its impact on the knowledge economy has been felt worldwide. On behalf of the Program Committee Co-Chairs and the entire Program Committee, I would like to extend to you the warmest welcome to SEKE 2012.

We received 219 submissions from 30 countries this year. Through a rigorous review process where a majority (86 percent) of the submitted papers received three reviews, and the rest with two reviews, we were able to select 59 full papers for the general conference (27 percent), 18 full papers for three special tracks (8 percent), and 60 short papers (27 percent), for presentation in thirty nine sessions during the conference. In addition, the technical program includes excellent keynote speech and panel discussions, and three special tracks: Software Engineering with Computational Intelligence and Machine Learning, Petri Nets for SEKE, and Software Testing and Analysis with Intelligent Technologies.

The high quality of the SEKE 2012 technical program would not have been possible without the tireless effort and hard work of many individuals. First of all, I would like to express my sincere appreciation to all the authors whose technical contributions have made the final technical program possible. I am very grateful to all the Program Committee members whose expertise and dedication made my responsibility that much easier. My gratitude also goes to the keynote speaker and panelists who graciously agreed to share their insight on important research issues, to the conference organizing committee members for their superb work, and to the external reviewers for their contribution.

Personally, I owe a debt of gratitude to a number of people whose help and support with the technical program and the conference organization are unfailing and indispensable. I am deeply indebted to Dr. S. K. Chang, Chair of the Steering Committee, for his constant guidance and support that are essential to pull off SEKE 2012. My heartfelt appreciation goes to Dr. Masoud Sadjadi, the Conference Chair, for his help and experience, and to the Program Committee Co-Chairs, Dr. Marek Reformat of University of Alberta, Canada, Dr. Swapna Gokhale of University of Connecticut, USA, and Dr. Jose Carlos Maldonado of University of Sao Paulo, Brazil, for their outstanding team work. I am truly grateful to the special track organizers, Dr. Taghi Khoshgoftaar of Florida Atlantic University, Dr. Marek Reformat of University of Alberta, Canada, Dr. Dianxiang Xu of Dakota State University, South Dakota, Dr. Haiping Xu of University of Massachusetts Dartmouth, Dr. Zhenyu Chen of Nanjing University, China, and Dr. Zheng Li of Beijing University of Chemical Technology, China, for their excellent job in organizing the special sessions. I would like to express my great appreciation to all the Publicity Co-Chairs, Dr. Xiaoying Bai of Tsinghua University, China, Dr. Raul Garcia Castro of Universidad Politecnica de Madrid, Spain, Shihong Huang of Florida Atlantic University, and Dr. Haiping Xu of University of Massachusetts Dartmouth, for their important contributions, to the Asia, Europe, and South America liaisons, Dr. Hironori Washizaki of Waseda University, Japan, Dr. Raul Garcia Castro of Universidad Politecnica de Madrid, Spain, and Dr. Jose Carlos Maldonado of University of Sao Paulo, Brazil, for their great efforts in helping expand the SEKE community, and to the Poster/Demo session Co-Chairs, Dr. Farshad Samimi of Trilliant and Dr. Ming Zhao of Florida International University, for their work. Last but certainly not the least, I must acknowledge the important contributions the following KSI staff members have made: David Huang, Rachel Lu, Alice Wang, and Dennis Chi. Their timely and dependable support and assistance throughout the entire process have been truly remarkable. It has been a great pleasure to work with all of them.

Finally, I hope you will find your participation in the SEKE 2012 programs rewarding. Enjoy your stay in the San Francisco Bay area.

Du Zhang
SEKE 2012 Program Chair

The 24th International Conference on Software Engineering & Knowledge Engineering (SEKE 2012)

July 1-3, 2012
Hotel Sofitel, Redwood City, San Francisco Bay, USA

Conference Organization

Steering Committee Chair

Shi-Kuo Chang, *University of Pittsburgh, USA*

Steering Committee

Vic Basili, *University of Maryland, USA*
Bruce Buchanan, *University of Pittsburgh, USA*
C. V. Ramamoorthy, *University of California, Berkeley, USA*

Advisory Committee

Jerry Gao, *San Jose State University, USA*
Natalia Juristo, *Madrid Technological University, Spain*
Taghi Khoshgoftaar, *Florida Atlantic University, USA*
Guenther Ruhe, *University of Calgary, Canada*

Conference Chair

S. Masoud Sadjadi, *Florida International University, USA*

Program Chair

Du Zhang, *California State University Sacramento, USA*

Poster/Demo Sessions Co-Chairs

Farshad Samimi, *Trilliant, USA*
Ming Zhao, *Florida Int'l University, USA*

Publicity Co-Chairs

Xiaoying Bai, *Tsinghua University, China*
Raul Garcia Castro, *Universidad Politecnica de Madrid, Spain*
Shihong Huang, *Florida Atlantic University, USA*
Haiping Xu, *University of Massachusetts Dartmouth, USA*

Asia Liaison

Hironori Washizaki, *Waseda University, Japan*

South America Liasion

Jose Carlos Maldonado, *University of Sao Paulo, Brazil*

Proceedings Cover Design

Gabriel Smith, *Knowledge Systems Institute Graduate School, USA*

Conference Secretariat

Judy Pan, *Chair, Knowledge Systems Institute Graduate School, USA*
Noorjhan Ali, *Knowledge Systems Institute Graduate School, USA*
Dennis Chi, *Knowledge Systems Institute Graduate School, USA*
David Huang, *Knowledge Systems Institute Graduate School, USA*
Rachel Lu, *Knowledge Systems Institute Graduate School, USA*
Alice Wang, *Knowledge Systems Institute Graduate School, USA*

Table of Contents

Foreword iii

Conference Organization iv

Keynote: On the Naturalness of Software
Prem Devanbu xxv

Panel Discussion on Future Trends of Software Engineering and Knowledge Engineering
Moderator: Du Zhang
Panelists: Masoud Sadjadi, Taghi Khoshgoftaar, Eric Grégoire, Swapna S. Gokhale and Marek Reformat xxvi

Data Mining

Sparse Linear Influence Model for Hot User Selection on Mining a Social Network
Yingze Wang, Guang Xiang and Shi-Kuo Chang 1

Mining Call Graph for Change Impact Analysis
Qiandong Zhang, Bixin Li and Xiaobing Sun 7

A Mobile Application for Stock Market Prediction Using Sentiment Analysis
Kushal Jangid, Pratik Paul and Magdalini Eirinaki 13

Requirement Engineering

Using Semantic Relatedness and Locality for Requirements Elicitation Guidance
Stefan Farfeleder, Thomas Moser and Andreas Krall 19

Phases, Activities, and Techniques for a Requirements Conceptualization Process
Alejandro Hossian and Ramón Garcia-Martínez 25

Using Empirical Studies to Evaluate the REMO Requirement Elicitation Technique
Sérgio Roberto Costa Vieira, Davi Viana, Rogério do Nascimento and Tayana Conte 33

Consistency Checks of System Properties Using LTL and Büchi Automata
Salamah Salamah, Matthew Engskow and Omar Ochoa 39

Evaluating the Cost-Effectiveness of Inspecting the Requirement Documents: an Empirical Study
Narendar Mandala and Gursimran S. Walia **45**

Requirement Analysis and Automated Verification: a Semantic Approach (S)
Animesh Dutta, Prajna Devi Upadhyay and Sudipta Acharya **51**

Risk-Driven Non-Functional Requirement Analysis and Specification (S)
Yi Liu, Zhiyi Ma, Hui Liu and Weizhong Shao **55**

Eliciting Security Requirements in the Commanded Behavior Frame: an Ontology Based Approach (S)
Xiaohong Chen and Jing Liu **61**

An Overview of the RSLingo Approach (S)
David de Almeida Ferreira and Alberto Rodrigues da Silva **66**

Detecting Emergent Behavior in Distributed Systems Caused by Overgeneralization (S)
Seyedehmehrnaz Mireslami, Mohammad Moshirpour and Behrouz H. Far **70**

Special Session: Software Engineering with Comp. Intelligence & Machine Learning

Stability of Filter-Based Feature Selection Methods for Imbalanced Software Measurement Data
Kehan Gao, Taghi M. Khoshgoftaar and Amri Napolitano **74**

Semantic Interfaces Discovery Server
Laura Maria Chaves, José Renato Villela Dantas, Bruno de Azevedo Muniz, Júlio Cesar Campos Neto and Pedro Porfírio Muniz Farias **80**

Cloud Application Resource Mapping and Scaling Based on Monitoring of QoS Constraints
Xabriel J. Collazo-Mojica, S. Masoud Sadjadi, Jorge Ejarque and Rosa M. Badia **88**

An Empirical Study of Software Metric Selection Techniques for Defect Prediction
Huanjing Wang, Taghi M. Khoshgoftaar, Randall Wald and Amri Napolitano **94**

Progressive Clustering with Learned Seeds: an Event Categorization System for Power Grid
Boyi Xie, Rebecca J. Passonneau, Haimonti Dutta, Jing-Yeu Miaw, Axinia Radeva, Ashish Tomar and Cynthia Rudin **100**

Multi-Objective Optimization of Fuzzy Neural Networks for Software Modeling
Kuwen Li, Marek Z. Reformat, Witold Pedrycz and Jinfeng Yu **106**

Generating Performance Test Scripts and Scenarios Based on Abstract Intermediate Models
Leandro T. Costa, Ricardo M. Czekster, Flávio M. de Oliveira, Elder M. Rodrigues, Maicon B. da Silveira and Avelino F. Zorzo **112**

Case Study

A Catalog of Patterns for Concept Lattice Interpretation in Software Reengineering
Muhammad U.Bhatti, Nicolas Anquetil, Marianne Huchard, and Stéphane Ducasse **118**

Client-Side Rendering Mechanism: a Double-Edged Sword for Browser-Based Web Applications
Hao Han, Yinxing Xue and Keizo Oyama **124**

An Empirical Study on Improving Trust among GSD Teams Using KMR (S)
Mamoona Humayun and Cui Gang **131**

Modeling and Analysis of Switched Fuzzy Systems (S)
Zuohua Ding and Jiaying Ma **135**

An Empirical Study on Recommendation Methods for Vertical B2C E-Commerce (S)
Chengfeng Hui, Jia Liu, Zhenyu Chen, Xingzhong Du and Weiyun Ma **139**

Automated Approaches to Support Secondary Study Processes: a Systematic Review (S)
Jefferson Seide Molléri and Fabiane Barreto Vavassori Benitti **143**

Aspect-Oriented SE

Enforcing Contracts for Aspect-Oriented Programs with Annotations, Pointcuts and Advice
Henrique Rebêlo, Ricardo Lima, Alexandre Mota, César Oliveira and Márcio Ribeiro **148**

Towards More Generic Aspect-Oriented Programming: Rethinking the AOP Joinpoint Concept (S)
Jonathan Cook and Amjad Nusayr **154**

Aspect-Orientation in the Development of Embedded Systems: a Systematic Review (S)
Leonardo Simas Duarte and Elisa Yumi Nakagawa **158**

Program Understanding

Evaluating Open Source Reverse Engineering Tools for Teaching Software Engineering
Swapna S. Gokhale, Thérèse Smith and Robert McCartney **162**

Coordination Model to Support Visualization of Aspect-Oriented Programs
Álvaro F. d'Arce, Rogério E. Garcia, Ronaldo C. M. Correia and Danilo M. Eler **168**

Improving Program Comprehension in Operating System Kernels with Execution Trace Information (S)
Elder Vicente, Geycy Dyany, Rivalino Matias Jr. and Marcelo de Almeida Maia **174**

Component-based SE

An Approach for Software Component Reusing Based on Ontological Mapping
Shi-Kuo Chang, Francesco Colace, Massimo De Santo, Emilio Zegarra and YongJun Qie **180**

Online Anomaly Detection for Components in OSGi-Based Software
Tao Wang, Wenbo Zhang, Jun Wei, Jianhua Zhang and Hua Zhong **188**

An Exploratory Study of One-Use and Reusable Software Components (S)
Reghu Anguswamy and William B. Frakes **194**

Choosing Licenses In Free Open Source Software (S)
Ioannis E. Foukarakis, Georgia M. Kapitsaki and Nikolaos D. Tselikas **200**

Software Quality

A Unified Model for Server Usage and Operational Costs Based on User Profiles: an Industrial Evaluation
Johannes Pelto-Piri, Peter Molin and Richard Torkar **205**

A Model-Centric Approach for the Integration of Software Analysis Methods
Xiangping Chen, Jiaxi Chen, Zibin Zhao and Lingshuang Shao **211**

CATESR: Change-Aware Test Suite Reduction Based on Partial Coverage of Test Requirements
Lijiu Zhang, Xiang Chen, Qing Gu, Haigang Zhao, Xiaoyan Shi and Daoxu Chen **217**

A Process Model for Human Resources Management Focused on Increasing the Quality of Software Development
Flávio E. A. Horita, Jacques D. Brancher and Rodolfo M. de Barros 225

Verification of Cyber-Physical Systems Based on Differential-Algebraic Temporal Dynamic Logic (S)
Xiaoxiang Zhai, Bixin Li, Min Zhu, Jiakai Li, Qiaoqiao Chen and Shunhui Ji 231

HybridUML Based Verification of CPS Using Differential Dynamic Logic (S)
Min Zhu, Bixin Li, Jiakai Li, Qiaoqiao Chen, Xiaoxiang Zhai and Shunhui Ji 235

A HybridUML and QdL Based Verification Method for CPS Self-Adaptability (S)
Jiakai Li, Bixin Li, Qiaoqiao Chen, Min Zhu, Shunhui Ji and Xiaoxiang Zhai 239

Agent-based Learning

Disabling Subsumptions in a Logic-Based Component
Éric Grégoire and Sébastien Ramon 243

i2Learning: Perpetual Learning through Bias Shifting
Du Zhang 249

Evolutionary Learning and Fuzzy Logic Applied to a Load Balancer
Francisco Calaça Xavier, Max Gontijo de Oliveira and Cedric L. de Carvalho 256

Using Social Networks for Learning New Concepts in Multi-Agent Systems
Shimaa M. El-Sherif, Behrouz Far and Armin Eberlein 261

Special Session: Software Testing and Analysis with Intelligent Technology

Identifying Coincidental Correctness for Fault Localization by Clustering Test Cases
Yi Miao, Zhenyu Chen, Sihan Li, Zhihong Zhao and Yuming Zhou 267

Regression Testing Prioritization Based on Fuzzy Inference Systems
Pedro Santos Neto, Ricardo Britto, Thiago Soares, Werney Ayala, Jonathas Cruz and Ricardo Rabelo 273

Parallel Path Execution for Software Testing over Automated Test Cloud (S)
Wei Liu, Xiaoqiang Liu, Feng Li, Yulong Gu, Lizhi Cai, Genxing Yang and Zhenyu Liu 279

An Empirical Study of Execution-Data Classification Based on Machine Learning
Dan Hao, Xingxia Wu and Lu Zhang **283**

Identification of Design Patterns Using Dependence Analysis (S)
Wentao Ma, Xiaoyu Zhou, Xiaofang Qi, Ju Qian, Lei Xu and Rui Yang **289**

Slicing Concurrent Interprocedural Programs Based on Program Reachability Graphs (S)
Xiaofang Qi, Xiaojing Xu and Peng Wang **293**

Service-Centric SE

A Usage-Based Unified Resource Model
Yves Wautelet, Samedi Heng and Manuel Kolp **299**

Petri Net Modeling of Application Server Performance for Web Services
M. Rahmani, A. Azadmanesh and H. Siy **305**

Implementing Web Applications as Social Machines Composition: a Case Study (S)
Kellyton dos Santos Brito, Lenin Ernesto Abadie Otero, Patrícia Fontinele Muniz, Leandro Marques Nascimento, Vanilson André de Arruda Burégio, Vinicius Cardoso Garcia and Silvio Romero de Lemos Meira **311**

Interactive Business Rules Framework for Knowledge Based Service Oriented Architecture (S)
Debasis Chanda, Dwijesh Dutta Majumder and Swapan Bhattacharya **315**

Defining RESTful Web Services Test Cases from UML Models (S)
Alexandre Luis Correa, Thiago Silva-de-Souza, Eber Assis Schmitz and Antonio Juarez Alencar **319**

A Model Introducing Soas Quality Attributes Decomposition (S)
Riad Belkhatir, Mourad Oussalah and Arnaud Viguier **324**

Software as a Service: Undo (S)
Hernán Merlino, Oscar Dieste, Patricia Pesado and Ramon García-Martínez **328**

Petri Nets for SEKE

A Petri Net Model for Secure and Fault-Tolerant Cloud-Based Information Storage
Daniel F. Fitch and Haiping Xu **333**

Decidability of Minimal Supports of S-invariants and the Computation of their Supported S-Invariants of Petri Nets
Faming Lu, Qingtian Zeng, Hao Zhang, Yunxia Bao and Jiufang An **340**

Automated Generation of Concurrent Test Code from Function Nets
Dianxiang Xu and Janghwan Tae **346**

SAMAT - A Tool for Software Architecture Modeling and Analysis
Su Liu, Reng Zeng, Zhuo Sun and Xudong He **352**

Singular Formulas for Compound Siphons, Complementary Siphons and Characteristic Vectors for Deadlock Prevention in Cloud Computing (S)
Gaiyun Liu, D.Y.Chao and Yao-Nan Lien **359**

Model-Based Metamorphic Testing: A Case Study
Junhua Ding and Dianxiang Xu **363**

Verifying Aspect-Oriented Activity Diagrams Against Crosscutting Properties with Petri Net Analyzer
Zhanqi Cui, Linzhang Wang, Xi Liu, Lei Bu, Jianhua Zhao, and Xuandong Li **369**

Parametric Verification of TimeWorkflow Nets
Hanifa Boucheneb and Kamel Barkaoui **375**

Resource Modeling and Analysis for Workflows: a Petri Net Approach
Jiacun Wang and Demin Li **381**

Security and Privacy

ACADA: Access Control-Driven Architecture with Dynamic Adaptation
Óscar Mortágua Pereira, Rui L. Aguiar and Maribel Yasmina Santos **387**

Connectors for Secure Software Architectures
Michael E. Shin, Bhavya Malhotra, Hassan Gomaa and Taeghyun Kang **394**

How Social Network APIs Have Ended the Age of Privacy (S)
Derek Doran, Sean Curley and Swapna S. Gokhale **400**

Computer Forensics: Toward the Construction of Electronic Chain of Custody on the Semantic Web (S)
Tamer Fares Gayed, Hakim Lounis and Moncef Bari **406**

Ontologies and Architecture

A Holistic Approach to Software Traceability
Hazeline U. Asuncion and Richard N. Taylor **412**

Pointcut Design with AODL (S)
Saqib Iqbal and Gary Allen **418**

Feature modeling and Verification Based on Description Logics (S)
Guohua Shen, Zhiqiu Huang, Changbao Tian, Qiang Ge and Wei Zhang **422**

A Context Ontology Model for Pervasive Advertising: a Case Study on Pervasive Displays (S)
Frederico Moreira Bublitz, Hyggo Oliveira de Almeida and Angelo Perkusich **426**

Ontology-based Representation of Simulation Models (S)
Katarina Grolinger, Miriam A. M. Capretz, José R. Marti and Krishan D. Srivastava **432**

An Ontology-based Approach for Storing XML Data Into Relational Databases (S)
Francisco Tiago Machado de Avelar, Deise de Brum Saccol and Eduardo Kessler Piveta **438**

Automatic Generation of Architectural Models From Goals Models (S)
Monique Soares, João Pimentel, Jaelson Castro, Carla Silva, Cleice Talitha, Gabriela Guedes and Diego Dermeval **444**

Towards Architectural Evolution through Model Transformations (S)
João Pimentel, Emanuel Santos, Diego Dermeval, Jaelson Castro and Anthony Finkelstein **448**

Testing

Using FCA-Based Change Impact Analysis for Regression Testing
Xiaobing Sun, Bixin Li, Chuanqi Tao and Qiandong Zhang **452**

Forecasting Fault Events in Power Distribution Grids Using Machine Learning
Aldo Dagnino, Karen Smiley and Lakshmi Ramachandran **458**

Testing Interoperability Security Policies
Mazen EL Maarabani, César Andrés and Ana Cavalli **464**

A New Approach to Evaluate Path Feasibility and Coverage Ratio of EFSM Based on Multi-objective Optimization
Rui Yang, Zhenyu Chen, Baowen Xu, Zhiyi Zhang and Wujie Zhou **470**

Structural Testing for Multithreaded Programs: an Experimental Evaluation of the Cost, Strength and Effectiveness (S)
Silvana M. Melo, Simone R. S. Souza and Paulo S. L. Souza **476**

Programming Languages

Towards a Unified Source Code Measurement Framework Supporting Multiple Programming Languages (S)
Reisha Humaira, Kazunori Sakamoto, Akira Ohashi, Hironori Washizaki and Yoshiaki Fukazawa **480**

A Tiny Specification Metalanguage (S)
Walter Wilson and Yu Lei **486**

SciprovMiner: Provenance Capture Using the OPM Model (S)
Tatiane O. M. Alves, Wander Gaspar, Regina M. M. Braga, Fernanda Campos, Marco Antonio Machado and Wagner Arbex **491**

Engineering Graphical Domain Specific Languages to Develop Embedded Robot Applications (S)
Daniel B. F. Conrado and Valter V. de Camargo **495**

Patterns and Frameworks

Dynamically Recommending Design Patterns
S. Smith and D. R. Plante **499**

Towards a Novel Semantic Approach for Process Patterns' Capitalization and Reuse
Nahla JLAIEL and Mohamed BEN AHMED **505**

DC2AP: a Dublin Core Application Profile to Analysis Patterns (S)
Lucas Francisco da Matta Vegi, Jugurta Lisboa-Filho, Glauber Luis da Silva Costa, Alcione de Paiva Oliveira and José Luís Braga **511**

Modeling

Bridging KDM and ASTM for Model-Driven Software Modernization
Gaëtan Deltombe, Olivier Le Goaer and Franck Barbier **517**

Modal ZIA, Modal Refinement Relation and Logical Characterization
Zining Cao **525**

Towards Autonomic Business Process Models
Karolyne Oliveira, Jaelson Castro, Sergio España and Oscar Pastor **531**

Iinteroperable EMR Message Generation: a Model-Driven Software Product Line Approach (S)
Deepa Raka, Shih-Hsi Liu and Marjan Mernik **537**

A Data Collaboration Model for Collaborative Design Based on C-Net (S)
Xin Gao, Wenhui Hu, Wei Ye, ZHANG Shi-kun and Xuan Sun **541**

Tools and Environment

Working and Playing with SCRUM
Erick Passos, Danilo Medeiros, Wandresson Araújo and Pedro Santos Neto **545**

Follow-the-Sun Software Development: a Controlled Experiment to Evaluate the Benefits of Adaptive and Prescriptive Approaches
Josiane Kroll, Alan R. Santos, Rafael Prikladnicki, Estevão R. Hess, Rafael A. Glanzner, Afonso Sales, Jorge L. N. Audy and Paulo H. L. Fernandes **551**

Software Process Monitoring Using Statistical Process Control Integrated in Workflow Systems
Marília Aranha Freire, Daniel Alencar da Costa, Eduardo Aranha and Uirá Kulesza **557**

AI for SE

Model Transformation for Frameworks Using Logical Planning
Guilherme A. Marchetti and Edson S. Gomi **563**

Investigating the Use of Bayesian Networks as a Support Tool for Monitoring Software Projects (S)
Fábio Pittoli, Abraham L. R. de Sousa and Daltro J Nunes **570**

Reuse of Experiences Applied to Requirements Engineering: an Approach Based on Natural Language Processing (S)
Adriano Albuquerque, Vládia Pinheiro and Thiago Leite **574**

Specification of Safety Critical Systems with Intelligent Software Agent Method (S)
Vinitha Hannah Subburaj, Joseph E. Urban and Manan R. Shah **578**

Human-Computer Interaction

Using the Results from a Systematic Mapping Extension to Define a Usability Inspection Method for Web Applications
Luis Rivero and Tayana Conte **582**

Improving a Web Usability Inspection Technique through an Observational Study
Priscila Fernandes, Tayana Conte and Bruno Bonifácio **588**

Identification Guidelines for the Design of Interfaces in the Context of ECAs and ADHD (S)
Sandra Rodrigues Sarro Boarati and Cecília Sosa Arias Peixoto **594**

Measuring the Effect Of Usability Mechanisms On User Efficiency, Effectiveness and Satisfaction (S)
Marianella Aveledo M., Diego M. Curtino, Agustín De la Rosa H. and Ana M. Moreno S **599**

Automatic Generation of Web Interfaces from User Interaction Diagrams (S)
Filipe Bianchi Damiani and Patrícia Vilain **605**

Semantic Web

Semantic Technology Recommendation Based on the Analytic Network Process
Filip Radulovic and Raúl García-Castro **611**

P2P-Based Publication and Location of Web Ontology for Knowledge Sharing in Virtual Communities (S)
Huayou Si, Zhong Chen and Yong Deng **617**

Software Product Lines

Empirical Validation of Variability-based Complexity Metrics for Software Product Line Architecture
Edson A. Oliveira Junior, Itana M. S. Gimenes and José C. Maldonado **622**

A Mapping Study on Software Product Lines Testing Tools
Crescencio Rodrigues Lima Neto, Paulo Anselmo Mota Silveira Neto, Eduardo Santana de Almeida and Silvio Romero de Lemos Meira **628**

Optimal Variability Selection in Product Line Engineering
Rafael Pinto Medeiros, Uéverton dos Santos Souza , Fábio Protti and Leonardo Gresta Paulino Murta **635**

Synthesizing Evidence on Risk Management: a Narrative Synthesis of Two Mapping Studies (S)
Luanna Lopes Lobato, Ivan do Carmo Machado, Paulo Anselmo da Mota Silveira Neto, Eduardo Santana de Almeida and Silvio Romero de Lemos Meira **641**

PlugSPL: an Automated Environment for Supporting Plugin-Based Software Product Lines (S)
Elder M. Rodrigues, Avelino F. Zorzo, Edson A. Oliveira Junior, Itana M. S. Gimenes, José C. Maldonado and Anderson R. P. Domingues **647**

GS2SPL: Goals and Scenarios to Software Product Lines
Gabriela Guedes, Carla Silva, Jaelson Castro, Monique Soares, Diego Dermeval and Cleice Souza **651**

A Set of Inspection Techniques on Software Product Line Models
Rafael Cunha, Tayana Conte, Eduardo Santana de Almeida and José Carlos Maldonado **657**

Non-Functional Properties in Software Product Lines: a Taxonomy for Classification (S)
Mahdi Noorian, Ebrahim Bagheri and Weichang Du **663**

A Proposal of Reference Architecture for the Reconfigurable Software Development (S)
Frank José Affonso and Evandro Luis Linhari Rodrigues **668**

Dependability and Maintenance

A Variability Management Method for Software Configuration Files
Hiroaki Tanizaki, Toshiaki Aoki and Takuya Katayama **672**

Tool Support for Anomaly Detection in Scientific Sensor Data (S)
Irbis Gallegos and Ann Gates **678**

Reconfiguration of Robot Applications Using Data Dependency and Impact Analysis (S)
Michael E. Shin, Taeghyun Kang, Sunghoon Kim, Seungwook Jung and Myungchan Roh **684**

Automated Software Specification

Spacemaker: Practical Formal Synthesis of Tradeoff Spaces for Object-Relational Mapping
Hamid Bagheri, Kevin Sullivan and Sang H. Son **688**

A Formal Support for Incremental Behavior Specification in Agile Development
Anne-Lise Courbis, Thomas Lambolais, Hong-Viet Luong, Thanh-Liem Phan, Christelle Urtado and Sylvain Vauttier **694**

Knowledge Acquisition and Visualization

A Process-Based Approach to Improving Knowledge Sharing in Software Engineering
Sarah B. Lee and Kenneth Steward **700**

Automatic Acquisition of isA Relationships from Web Tables
Norah Alrayes and Wo-Shun Luk **706**

A Light Weight Alternative for OLAP
Hugo Cordeiro, Jackson Casimiro and Erick Passos **712**

A Tool for Visualization of a Knowledge Model (S)
Simon Suigen Guo, Christine W. Chan and Qing Zhou **718**

UML

Rendering UML Activity Diagrams as a Domain Specific Language— ADL
Charoensak Narkngam and Yachai Limpiyakorn **724**

umlTUowl - a Both Generic and Vendor-Specific Approach for UML to OWL Transformation
Andreas Grünwald and Thomas Moser **730**

A Framework for Class Diagram Retrieval Using Genetic Algorithm (S)
Hamza Onoruoiza Salami and Moataz A. Ahmed **737**

Measurement and Adaptive Systems

Managing Linear Hash in a Closed Space
Satoshi NARATA and Takao MIURA **741**

CLAT: Collaborative Learning Adaptive Tutor
Alaeddin M.H Alawawdeh, César Andrés and Luis Llana **747**

A proposal for the improvement of the Technique of Earned Value Management Utilizing the History of Performance Data (S)
Adler Diniz de Souza and Ana Regina Cavalcanti Rocha **753**

Agents and Mobile Systems

A Goal-Driven Method for Selecting Issues Used in Agent Negotiation (S)
Yen-Chieh Huang and Alan Liu **759**

Using Cell Phones for Mosquito Vector Surveillance and Control (S)
S. Lozano-Fuentes, S. Ghosh, J. M. Bieman, D. Sadhu, L. Eisen, F. Wedyan, E. Hernandez-Garcia, J. Garcia-Rejon and D. Tep-Chel **763**

Proactive Two Way Mobile Advertisement Using a Collaborative Client Server Architecture (S)
Weimin Ding and Xiao Su **768**

Poster/Demo

The COIN Platform: Supporting the Marine Shipping Industrial Sector (P)
Achilleas P. Achilleos, Georgia M. Kapitsaki, George Sielis, and George A. Papadopoulos **A-1**

A proposal for the Improvement of the Technique of EVM Utilizing the History of Performance Data (P)
Adler Diniz de Souza and Ana Regina Cavalcanti Rocha **A-3**

Checking Contracts for AOP Using XPIDRs (P)
Henrique Rebelo, Ricardo Lima, Alexandre Mota, César Oliveira, Márcio Ribeiro **A-5**

Author's Index **A-6**

Reviewer's Index **A-12**

Poster/Demo Presenter's Index **A-15**

Note: (S) indicates a short paper.
(P) indicates a poster or demo, which is not a refereed paper.

Keynote
On the Naturalness of Software

Professor Prem Devanbu
Department of Computer Science
University of California Davis

Abstract

Natural Language processing (NLP) has been revolutionized by statistical language models, which capture the high degree of regularity and repetition that exists in most human speech and writing. These models have revolutionized speech recognition and translation. We have found, surprisingly, that "natural software", viz., code written by people is also highly repetitive, and can be modeled effectively by language models borrowed from NLP. We present data supporting this claim, discuss some early applications showcasing the value of language models of code, and present a vision for future research in this area.

About the Speaker

Prem Devanbu received his B.Tech from the Indian Institute of Technology in Chennai, India, before you were born, and his PhD from Rutgers in 1994. After spending nearly 20 years at Bell Labs and its various offshoots, he escaped New Jersey to join the CS faculty at UC Davis in late 1997. He has published over 100 papers, and has won ACM SIGSOFT distinguished paper awards at ICSE 2004, ICSE 2009, and ASE 2011, and the conference best paper awards at MSR 2010 and ASE 2011. He has been program chair of ACM SIGSOFT FSE (in 2006) and ICSE (in 2010). He has served on the Editorial boards of both IEEE Transactions on Software Engineering and the ACM equivalent. He has worked in several different areas over a 25 year research career, including logic programming, knowledge representation, software tools, secure information storage in the cloud, and middleware. For the past years, he has been fascinated by the abundance of possibilities in the veritable ocean of data that is available from open-source software projects. He is funded by grants from the NSF, the AFOSR, Microsoft Research, and IBM.

Panel on Future Trends of Software Engineering and Knowledge Engineering

Du Zhang
California State University, USA
(Moderator)

The International Conference on Software Engineering and Knowledge Engineering (SEKE) is celebrating its 24th anniversary this year. For nearly a quarter of century, while SEKE has established itself as a major international forum to promote research and practice in software engineering and knowledge engineering, the computing fields have undergone profound changes. Today, our daily lives are intimately intertwined with artifacts that are the results of software engineering and knowledge engineering. What will the future hold for SEKE as a field of inquiry in the next ten years? What are the challenges that lie ahead? What can we do as a community to further our agenda on SEKE? Toward illuminating our path to the future, an excellent panel of experts has been assembled. Panelists will share their insight on the future trends of software engineering and knowledge engineering. We hope you will find the panel an inspiring impetus for the continued growth of SEKE in the years to come.

Software Engineering of Autonomic Clouds

Masoud Sadjadi
Florida International University, USA
(Panelist)

Autonomic or self-managing clouds are becoming prevalent software deployment environments for applications ranging from commerce (e.g., banking), to education (e.g., virtual labs), to research (e.g., high-performance computing). Unfortunately, traditional approaches to software engineering are not applicable to the specific characteristics of autonomic clouds, which are becoming a major part of every software application's solution domain. Therefore, there is a desperate need for a paradigm shift in how software applications are designed, developed, tested, deployed, hosted, and consumed in the clouds. One example of the specific characteristics of autonomic clouds is the concept of on-demand services leasing, which has major impacts on the growth of new businesses, from their inception to booming popularity. To respond to such needs, service providers face major challenges when trying to keep up with their promise of infinite capacity with unconditional elasticity.

Big Data in Software Engineering: Challenges and Opportunities

Taghi Khoshgoftaar
Florida Atlantic University, USA
(Panelist)

The field of software engineering has changed drastically in the past 20 years. Although traditional quality assurance approaches such as unit tests and change tracking remain essential tools, these approaches can be easily overwhelmed by the sheer volume of modules, bugs, programmers, and projects managed in large software development firms. To deal with this "Big Data," a new class of software engineering tools are needed: those from the fields of data mining and machine learning. By employing techniques specifically designed to sift through enormous datasets and identify the elements in need of human attention, data mining tools permit software practitioners to focus valuable human resources where they are needed most. I will discuss a number of topics concerning the use of data mining to manage Big Data in the context of software engineering, including software metric selection, data balance issues, and quality of data.

Knowledge Engineering, Operational Research and AI: the Time to Meet

Eric Grégoire
Université d'Artois, France
(Panelist)

Although they share many paradigms, the Operational Research and Artificial Intelligence fields have often evolved separately. This last decade, both domains have come ever closer, through new insights in constraint solving and SAT-related technologies, allowing problems to be solved that were long considered out of reach. This opens new perspectives for Knowledge Engineering as well.

Computational Issues in Social Networks

Swapna S. Gokhale
University of Connecticut, USA
(Panelist)

Online social networks (OSNs) have had an enormous impact on the way people communicate and share information. Today, the population of Facebook exceeds that of the United States and Lady Gaga has more Twitter followers than the entire population of Australia! OSNs not only provide social channels for communication, but they also offer critical marketing and customer profiling tools for businesses. This revolution has precipitated a deep desire to understand the structure of OSNs, identify the latent patterns that may exist within these networks, and leverage these structures and patterns to build novel applications and services. While sociologists have researched such social networks for decades, never before has such a vast quantity of structured social network data been available for analysis. Social network analysis is thus a rapidly emerging field that combines algorithmic, graph theoretical, and data mining techniques to map, measure, and find patterns in the relationships and communication flows in massive OSN datasets. This talk will summarize the recent, state-of-the-art research in OSN analysis on topics such as topology characterization, information and influence diffusion, community detection, inferring relationship strength, microblog analysis, friend and link prediction, data anonymity, workload characterization, and security and privacy, and outline avenues for further exploration.

Web Intelligence: Representation and Processing of Knowledge with Uncertainty

Marek Reformat
University of Alberta, Canada
(Panelist)

Uncertainty is an integral component of information and knowledge. Many concepts we deal with are without precise definitions, or with unknown facts, missing or inaccurate data. Such a situation is also present on the Internet where many sources of information could be corrupted, or partially and temporally inaccessible. Our dependence on the Internet is growing with every day. We relay on it doing research, learning new things, and finding what is happening in the world and in our neighborhood. But, how much imprecision and ambiguity is out there? How many sources of data are trustworthy? How much we can relay on the web to discover new things? Additionally, uncertainty is not only associated with data and information stored on the web – the users also bring ambiguity and imprecision. In many cases, the users' behavior and decisions depend on current circumstances, users' judgments, their understanding of situations, and their needs and requirements – things that are "equipped" with ambiguity. In order to make the web a user-friendly environment where the users can easily and quickly find things they are looking for, new web utilization tools have to be developed. They should be able to deal with numerous alternatives provided by the Internet, as well as with imprecision. The purpose of this topic is to provoke discussion how critical is to address the issue of imprecision and what methods, tools and approaches would be possible solutions.

Verifying Aspect-Oriented Activity Diagrams Against Crosscutting Properties with Petri Net Analyzer

Zhanqi Cui, Linzhang Wang, Xi Liu, Lei Bu, Jianhua Zhao, Xuandong Li
State Key Laboratory of Novel Software Technology
Department of Computer Science and Technology
Nanjing University, Nanjing, 210046, China
zqcui@seg.nju.edu.cn, lzwang@nju.edu.cn, liux@seg.nju.edu.cn, {bulei, zhaojh, lxd}@nju.edu.cn

***Abstract*—Aspect-oriented model-driven approaches are proposed to model and integrate crosscutting concerns at design phase. However, potential faults that violate desired properties of the software system might still be introduced during the process. Verification technique is well-known for its ability to assure the correctness of models and uncover design problems before implementation. This paper presents a framework to verify aspect-oriented UML activity diagrams based on Petri net verification techniques. For verification purpose, we transform the integrated activity diagrams into Petri nets. Then, the Petri nets are checked against formalized crosscutting requirements to detect potential faults. Furthermore, we implement a tool named Jasmine-AOV to support the verification process. Case studies are conducted to evaluate the effectiveness of our approach.**

Keywords: aspect-oriented modeling; verification; model checking; activity diagram; Petri net

I. INTRODUCTION

Dealing with crosscutting concerns has been a critical problem during software development life cycles. In our previous work [1], we proposed an aspect-oriented model-driven approach based on UML activity diagrams. The approach shifts aspect-oriented techniques [2] from a code-centric to a model-centric, which is employed to handle the crosscutting concerns during design phases. Thus, it alleviates software complexity in a more abstract level. The primary functional concerns are modeled with activity diagrams, and crosscutting concerns are modeled with aspectual activity diagrams, respectively. Then the overall system design model, which is also an activity diagram, is integrated by weaving aspect models into primary models.

Design models are widely used as a basis of subsequent implementation [3][4] and testing [5][6][7] processes. It is costly if defects in design models are discovered at later implementation and testing stages. Aspect-oriented modeling techniques cannot guarantee the correctness of produced design models. For instance, wrong weaving sequences may cause the integrated models violate system crosscutting requirements. Therefore, assuring the correctness of the aspect-oriented design models is vitally important. So far, the applicable approach is manual review. It is time consuming and dependent on reviewers' expertise. However, existing automatic verification tools cannot deal with UML diagrams directly.

As an ongoing work, in this paper, in order to ensure crosscutting concerns are correctly modeled, we propose a rigorous approach to automatically verify aspect-oriented models (activity diagrams) by using Petri net based verification techniques. Firstly, the integrated activity diagram is translated into a Petri net. Then, crosscutting concerns in system requirements are refined to properties in the form of CTL formulas. Finally, the Petri net is verified against the formalized properties.

The rest of this paper is organized as follows. Section 2 presents backgrounds of activity diagrams, Petri nets, and a running example. Section 3 discusses the verification of aspect-oriented activity diagrams. Section 4 presents 2 case studies and evaluations of our approach. Section 5 reviews the related work. Finally, section 6 concludes the paper and discusses future work.

II. BACKGROUND

In this section, we briefly introduce UML activity diagrams and Petri nets, and a running example that will be employed to demonstrate our approach in following sections.

A. Activity Diagrams and Petri nets

The UML activity diagram is a powerful tool to describe control flow based program logic at different levels of abstraction. Designers commonly use activity diagrams to describe the sequence of behaviors between classes in a software system. Nodes and edges are two kinds of elements in activity diagrams. Nodes in activity diagrams are connected by edges. We formally define activity diagrams as follows.

Definition 1. (Activity Diagram). An activity diagram *AD* is a 4-tuple (N, E, F), where:

- $N = \{n_1, n_2, \ldots, n_i\}$ is a finite set of **nodes**, which contains action, initial/final, decision/merge and fork/join nodes, $n_I \in N$ is the initial activity state, $N_F \subseteq N$ is a set of final activity states;
- $E = \{e_1, e_2, \ldots, e_j\}$ is a finite set of **edges**;
- $F \subset (N \times E) \cup (E \times N)$ is the flow relation between nodes and edges.

Due to the nature of UML is semi-formal and UML diagrams are design-oriented models, translating activity diagrams into formal verification-oriented models is necessary before verification. In this approach, we translate activity diagrams into Petri nets, because in UML 2, the semantics of activity diagrams is explained in terms of Petri net notations [9], like tokens, flows etc. Petri net is a formal specification language that is widely used to model software behaviors. A Petri net consists of places, transitions, and arcs. Like UML activity diagrams, Petri nets offer a graphical notation for stepwise processes that include choice, iteration, and concurrent execution. On the other hand, Petri nets have a precise mathematical definition of their execution semantics, with a well-developed mathematical theory for process analysis. A Petri net is formally defined as follows.

Definition 2. (Petri net) A Petri net [8] is a 4-tuple $PN = \{P, T, A, M_0\}$, where

- P is a finite set of **places** and T is a finite set of **transitions**, and P and T are disjoint.
- A is a finite set of **arcs** connect between places and transitions, where $A \subseteq (P \times T \cup T \times P)$.
- M_0 is the initial **marking**, $M_0(p)$ denotes the number of tokens at place p under initial marking M_0.

Places, transitions and arcs in A are drawn as circles, boxes and arrows, respectively. We do not consider weights of arcs in this paper for simplification.

B. Running Example

We adapt the order processing scenario from [9] as a running example to demonstrate our approach. There are 4 crosscutting concerns related to this scenario: authentication, validation, logging, and informing.

Figure 1 is the primary model of the order processing scenario, which consists of 3 main steps: fill order, ship order, and close order.

Figure 1. The primary model of the order processing scenario

Based on our previous aspect-oriented modeling approach [1], the crosscutting concerns of the running example are modeled in Figure 2.

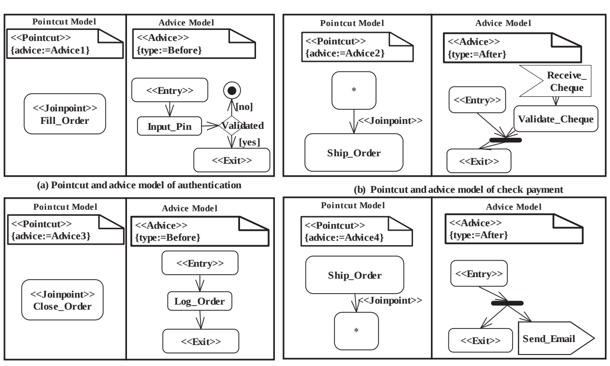

Figure 2. Pointcut and advice models of the order processing scenario

In order to understand how crosscutting concerns will affect primary functionalities, aspect models are integrated with primary models to generate an overall system design model. Different weaving sequences would produce different integrated models. For example, we add an authorization aspect in the running example, which describes the logged-in user need to be checked whether she/he has the permission to fill orders. If the authorization aspect is woven before authentication, then the result of integration is shown in Figure 3 (a). Otherwise, if the authentication aspect is woven before authentication, then the result of integration is shown in Figure 3 (b). As we know, the legal user has to be logged-in before being checked whether the corresponding permission is granted or not. As a result, the authentication aspect should be woven firstly, and Figure 3 (b) is the correct integration result we expected. Extensive explanations about the integration process can be also found in [1].

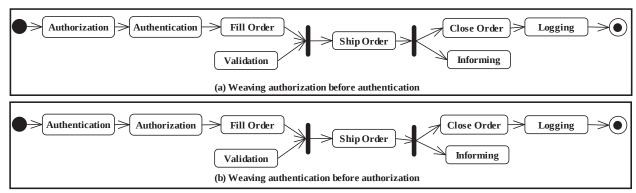

Figure 3. Two different integrated models of the order processing scenario

III. VERIFYING ASPECT-ORIENTED MODELS

In our previous work, aspect-oriented models, including primary models, aspect models, as well as integrated models, were all depicted with UML activity diagrams. Since the correctness of the integration process cannot be guaranteed, how to ensure the consistence between the integrated activity diagrams and crosscutting requirements becomes a critical research problem. In UML 2, the semantics of activity diagrams is explained in terms of Petri net. There are also various automatic tools, i.e., LoLA (a Low Level Petri Net Analyzer) [10], verifying Petri nets against specified properties. As a result, if we can translate activity diagrams into Petri nets, we could verify the activity diagram models by verifying corresponding Petri net models for specific properties. In this section, we first discuss transformation from activity diagrams to Petri nets, and then present the verification against crosscutting concerns.

A. Transforming from Activity Diagrams to Petri Nets

We adapt the mapping semantics of control-flows in UML 2 activities in [9] to convert activity diagrams into Petri nets. Basically, action nodes and fork/join nodes are translated to net transitions, control nodes (initial, final, decision, and merge nodes) become net places, and edges are transformed to net arcs. Auxiliary transitions or places are added when the ends of an arc both are transitions or both are places. For simplification, we restrict an activity diagram only consists of action nodes, control nodes, and control flows in this approach. The transformation of more complex activity diagrams (containing data flows, exceptions, and expansions etc.) is straightforward based on transformation rules in [11].

Based on the mapping rules in [12], we construct an algorithm to transform activity diagrams to Petri nets and implement in our verification tool to provide automatic transformation support. The algorithm is described in List 1. With the algorithm, the activity diagram of the running example in Figure 3 (b) is converted to the Petri net in Figure 4. The transformed Petri net is a bi-simulation of the activity diagram, which means they are semantically equal. So we can achieve the verification of the activity diagram by verifying the equivalent Petri net against same system properties.

List 1. Convert an activity diagram into a Petri net

```
Input: AD := an activity diagram
Output: PN{P, T, A, M0} := a Petri net
for each node N in AD
        if N is an initial node, final node, decision node, or merge node
                Generate a corresponding place in PN.P
        else // action node, fork node, or join node
                Generate a corresponding transition in PN.T
for each edge E in AD
        N1 := source node of E in AD
        N2 := target node of E in AD
        M1 := corresponding place or transition of N1 in PN.P
        M2 := corresponding place or transition of N2 in PN.P
        if both N1 and N2 ∈
            (initial nodes ∪ final node ∪ decision node ∪ merge node)
                Generate an auxiliary transition T1 in PN.T
                Generate an arc start from M1 to T1 in PN.A
                Generate an arc start from T1 to M2 in PN.A
        else if both N1 and N2 ∈ (action node ∪ fork node ∪ join node)
                Generate an auxiliary place P1 in PN.P
                Generate an arc start from M1 to P1 in PN.A
                Generate an arc start from P1 to M2 in PN.A
        else
                Generate an arc start from M1 to M2 in PN.A
for each place without an incoming arc
        Generate an initial token for that place in PN.M0
return PN
```

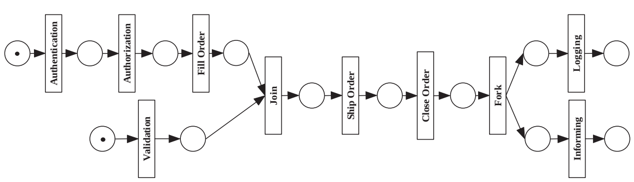

Figure 4. The Petri net transformed from the order processing scenario

B. *Verifying Petri Nets*

Crosscutting concerns describe the running sequences between advices and primary behaviors in all paths of integrated models. These properties can be described in the form of Computation Tree Logic (CTL) formulas [13] naturally. CTL formulas cannot be generated from aspect models by synthesizing conditions of join points specified by pointcut models and checking the corresponding advice models appears at right places. This is because that the context specified by a pointcut model would be changed after integration, and the join points matched by the pointcut model could no longer exist. In this approach, the properties to be checked are directly refined from crosscutting requirements.

1) *Properties specified from the requirement*

Based on the Petri net generated, we can easily analysis reachability, safety, liveness, and fairness properties [8]. In this approach, we only focus on checking properties that are closely related to crosscutting concerns. We categorize crosscutting concerns from two facets. Firstly, according to the execution sequence between action in advice models and join points, a crosscutting concern can be either executing before or after join points. Secondly, the execution of a crosscutting concern is either sequential or parallel with the primary behaviors. Sequential crosscutting concerns are synchronous features that their running positions are restricted by the join points. Parallel crosscutting concerns are asynchronous features that are running concurrently with primary actions and they are finished or started by the join points.

a) *Before-crosscutting concerns*

A before-crosscutting concern specifies some extra behaviors must be performed before matched join points. Actions specified by a sequential before aspect model are executed between the join point node and the predecessor node of the join point in the primary model. The key word of sequential before-crosscutting concerns in requirements level is "before". A parallel before aspect specifies crosscutting actions that must be finished by the join point edge. The key word of parallel before-crosscutting concerns in requirements is "be finished by". In the integrated model, the actions of the crosscutting concern are running concurrently with the primary behaviors, and then synchronized at the join node which replaced the join point edge.

In corresponding Petri nets, assume *jp* is the transition transformed from one of the join point, *ad* is the transition transformed from the structured activity node that represents the advice model. The requirement of a before aspect can be represented in the form of the CTL formula as: $\mathbf{AG}\neg((ad \wedge \mathbf{EX}(\neg ad \wedge \neg jp)) \vee ((\neg ad \wedge \neg jp) \wedge \mathbf{EX}\, jp))$.

b) *After-crosscutting concerns*

An after-crosscutting concern specifies some actions must be performed after matched join points. An after-crosscutting concern can also be either a sequential or a parallel aspect with respect to the flows of primary models. Actions specified by a sequential after aspect model are the actions executed between the join point node and the successor node of the join point in the primary model. The key word of sequential after-crosscutting concerns in requirements level is "after". A parallel after aspect specifies crosscutting actions must be started by the join point edge. The key word of parallel after-crosscutting concerns in requirements is "be started by". In the integrated model, the actions of the crosscutting concern are enabled by the fork node, which replaced the join point edge, and then running concurrently with primary behaviors.

In corresponding Petri nets, assume *jp* is the net transition transformed from the join point, *ad* is the net transition transformed from the structured activity node that represents the advice model. The requirement of a sequential after aspect can be represented in the form of the

CTL formula as: $\mathbf{AG}\neg((jp \wedge \mathbf{EX}(\neg jp \wedge \neg ad)) \vee ((\neg jp \wedge \neg ad) \wedge \mathbf{EX}\ ad))$.

2) Conflicts of Multiple Crosscutting Concerns

The CTL formula need to be adjusted if more than one crosscutting concerns (which are all "before" aspects or are all "after" aspects) match a same join point. Because the running sequence between one aspect and a join point can be affected by other aspects of the same before/after kind, which match the same join point. For instance, in the running example, the authentication and authorization concerns are conflicted because they both are be fore-crosscutting aspects and they have same join point, the "Fill_Order" action. The running sequence of authentication aspect and "Fill_Order" operation will be changed from "Authentication->Fill_Order" to "Authentication->Authorization->Fill_Order" after the weaving of authorization aspect.

a) Conflicts between two before-crosscutting concerns

For a before-crosscutting concern cc_1 with advice model ad_1 and join point jp_1, if any other before aspect, which matches the same join point jp_1 and weaves after cc_1, then some extra actions are performed after ad_1 and before jp_1. Assume it's a before-crosscutting concern cc_2 with advice ad_2 weaves after cc_1, then jp_1 should be replaced by ad_2 in the CTL formula of cc_1 as: $\mathbf{AG}\neg((ad_1 \wedge \mathbf{EX}(\neg ad_1 \wedge \neg ad_2)) \vee ((\neg ad_1 \wedge \neg ad_2) \wedge \mathbf{EX}\ ad_2))$.

b) Conflicts between two after-crosscutting concerns

For an after-crosscutting concern cc_1 with advice ad_1 and join point jp_1, if any other after aspect, which matches the same join point jp_1 and weaves after cc_1, then some extra actions are performed after jp_1 and before ad_1. Assume it's an after-crosscutting concern cc_2 with advice ad_2 weaves after cc_1, then jp_1 should be replaced by by ad_2 in the CTL formula of cc_1 as: $\mathbf{AG}\neg((ad_2 \wedge \mathbf{EX}(\neg ad_2 \wedge \neg ad_1)) \vee ((\neg ad_2 \wedge \neg ad_1) \wedge \mathbf{EX}\ ad_1))$.

3) Verification

After the system crosscutting properties are refined as a set of CTL formulas. We can verify the Petri net against specified CTL formulas generated. If the verification is passed, it means the model satisfies the corresponding crosscutting requirements. Otherwise, the model violates the corresponding crosscutting requirements to some extent, which means further revision about the model is needed.

In the running example, the integrated model in Figure 1 (a) and (b) are both verified against the crosscutting requirements of authentication, authorization, validation, logging, and informing. First, the integrated models are transformed to Petri nets. Then the 5 crosscutting requirements are refined to 5 CTL formulas. Finally, Petri net analyzer LoLA is employed to verify the two Petri nets against the formalized crosscutting requirements, respectively.

The Petri net transformed from the model in Figure 3 (b) passes the verification process and output "result: true" for all the 5 CTL formulas. While the Petri net transformed from the model in Figure 3 (a) fails when verifying against the 2 CTL formulas generated from authentication and authorization requirements, and passes the verification against the other 3 CTL formulas. This verification result shows that the crosscutting requirements of authentication and authorization do not hold in this aspect-oriented model. After correcting the weaving preference fault and integrating the aspect model again, the new integrated model passes the verification process.

C. Tool Implementation

We implemented a tool named Jasmine-AOV[1] based on Topcased[2] and LoLA[3]. As Figure 5 shows, this tool is composed of 4 main parts: Model Transformer, Crosscutting Concern Manager, CTL Generator, and Model Checker. The Model Transformer converts an activity diagram to a Petri net automatically. The inputs of Model Transformer are UML diagrams designed by Topcased in the form of XML file and the outputs of the tool are Petri net files which are readable for LoLA to perform verification tasks. The Crosscutting Concern Manager is used to manage mapping relations between crosscutting concerns in requirements and elements in corresponding activity diagrams. It provides an assistant for mapping textual crosscutting requirements to design activity diagrams. The CTL Generator can automatically generate CTL formulas from crosscutting requirements that are mapped to design models. The CTL Generator also supports users to input CTL formulas manually. Model Checker is implemented by directly wrapped an existing checker, LoLA. It can verify the Petri net against crosscutting properties in the format of CTL formulas and report the result.

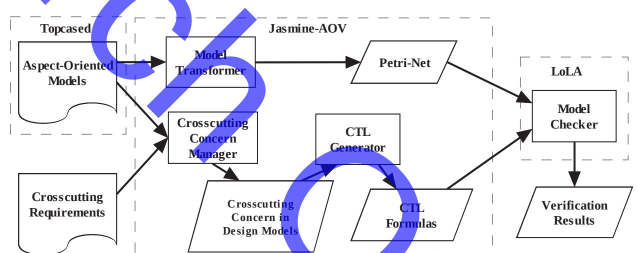

Figure 5. The framework of Jasmine-AOV

The screenshot of Jasmine-AOV is in Figure 6. The "Crosscutting concerns" area manages the crosscutting requirements which are mapped to design models. The "New Crosscutting Concern" dialog provides an assistant for mapping textual crosscutting requirements to design activity diagrams. The "Petri net" area displays the Petri net transformed from the corresponding activity diagram. The "CTL Formulas" area lists the formulas refined from crosscutting concerns in the "Crosscutting concerns" area automatically or wrote by users manually. The "Verification Results" area outputs the results of verifying the Petri net in

[1] Jasmine-AOV, http://seg.nju.edu.cn/~zqcui/Jasmine-AOV

[2] Topcased, http://www.topcased.org/

[3] LoLA, http://www.informatik.uni-rostock.de/tpp/lola/

the "Petri net" area against the CTL formulas in the "CTL Formulas" area by LoLA.

Writing complex CTL formulas is not easy for a software engineer without proper training about formal methods. To tackle this problem, we implemented the CTL Generator to assist generating CTL formulas automatically. As Figure 6 shows, the user only need to select actions which is the advice, the join points, and the relationship between the advice and the join points, based on the textual description of the crosscutting concern. After this information is inputted, the CTL Generator generates a CTL formula for the crosscutting concern and adjusts CTL formulas if there is more than one aspect of the same before/after type apply on a same join point.

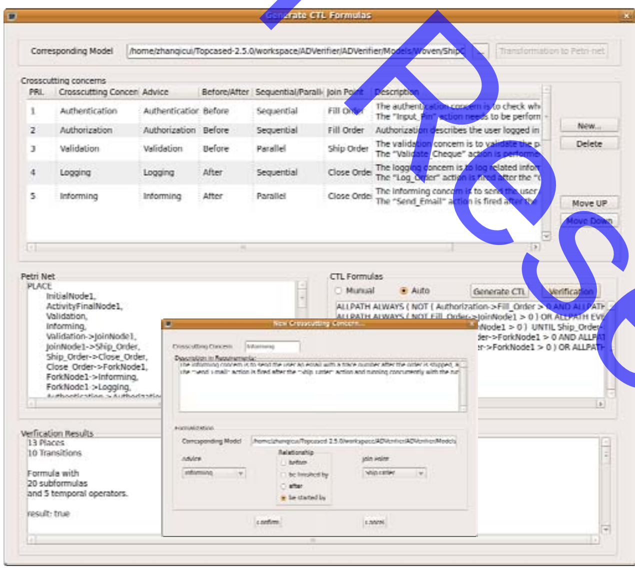

Figure 6. The screenshot of Jasmine-AOV

IV. EVALUATION AND CASE SUITES

To evaluate the effectiveness of our approach, we have applied our approach to the design models adapted from the Ship Order example in [9] and the Telecom System[4]. The Ship Order example contains 5 crosscutting concerns and the Telecom System contains 6 crosscutting concerns. For both of the 2 case studies, we transformed the integrated models to Petri nets, and mapped crosscutting requirements to the design models with the help of the tool. Then, corresponding CTL formulas of verification tasks are generated automatically. Finally, the Petri nets are checked against the CTL formulas generated.

The faults of aspect-oriented models, which can be caused by design defects or incorrect integration processes, are categorized as follows:

1. Aspect model faults
   a) Incorrect weaving preference. The priorities of aspect models are incorrectly assigned. This kind of fault will lead to match join pints faults or running sequence changed unexpected.
   b) Incorrect binding between pointcut model and advice model. The pointcut model is incorrectly mapped to an unrelated advice model. This kind of fault will result in improper advice models apply at some join points.
2. Pointcut model faults
   a) Overmatch/Mismatch join points. The pointcut model matches extra join points or miss some join points should be matched. The consequence of this kind of faults is that extra advices are performed at unexpected join points or desired advices are not going to be performed at join points.
   b) Incorrect position of join points. The element which serves as a join point in the pointcut model is incorrectly appointed. The phenomenon of this kind of faults is that advices are applied at incorrect points of the primary model.
3. Advice model faults
   a) Incorrect type of advice models. The type of the advice model is declared incorrectly. This kind of fault will cause the running sequence between the advice model and the primary model change unexpectedly.

To further evaluate the ability of our approach to detect the faults of aspect-oriented models, mutated models are created based on preceding category of aspect model faults. 26 and 28 model mutants are constructed for the 2 case studies, respectively. Table 1 classifies all these model mutants by their fault types. All of them are killed because they violate the crosscutting requirements from various ways and these violations are detected by the verification process. This result illustrates the ability of our approach to find the faults in aspect-oriented models and to improve the quality of design models.

TABLE I. MODEL MUTANTS OF THE 2 CASE STUDIES

Fault Types		Ship Order	Telecom System
Aspect model faults	Incorrect weaving preference	1	1
	Incorrect binding	5	3
Pointcut model faults	Overmatch join points	5	6
	Mismatch join points	5	6
	Incorrect position of join point	5	6
Advice model faults	In correct type of advice models	5	6
Number of model mutants in total		26	28
Mutants killed		26	28

V. RELATED WORK

There are many research projects on bringing aspect-oriented ideas to software requirement engineering from different perspectives. Whittle and Araujo [14] focus on scenario-based requirements and composing them with aspects to generate a set of state machines that represent the composed behaviors from both aspectual and non-aspectual scenarios. In contrast, our approach is carried out at the design level instead of requirement level. However, our approach can be enhanced with the aspect mining tool at

4 AJDT toolkit: http://www.eclipse.org/ajdt

requirements level, like EA-Miner [15], by inputting crosscutting concerns detected by these tools to our Jasmine-AOV tool for verification.

There is also a large body of research on aspect-oriented modeling. But most of them do not concern about the correctness of the integrated model and provides verification supports. In addition to support aspect-oriented modeling and integration, our approach also formally checks whether crosscutting concerns in requirements are correctly designed. Xu et al. proposed to model and compose aspects with finite state machines, and then transformed to FSP processes and cheeked by LTSA model checker against all system requirements [16]. Whereas our approach is carried out on activity diagrams and only focuses on checking crosscutting concerns. Furthermore, we categorize 4 kinds of crosscutting concerns and generate CTL formulas automatically from crosscutting concern specifications, which bridges the gaps between crosscutting requirements and aspect-oriented design models. We also provide a solution for the conflicts between crosscutting concerns.

Several model checking techniques have been presented for aspect-oriented programs. Denaro et al. first reported a preliminary experience on verifying deadlock freedom of a concurrent aspect [17]. They first derived PROMELA process templates from aspect-oriented units, and then analysis the aspect-oriented program with SPIN. Ubayashi and Tamai [18] proposed to apply model checking techniques to verify whether the result of weaving classes and aspects contained unexpected behaviors like deadlocks. The approach in this paper is different from these methods, because our approach is carried out at the model level other than the program level. In comparison, our approach can identify system faults at an earlier stage, and save costs to revise programs when detecting design faults at implementation or maintenance phase.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a framework to verify aspect-oriented UML activity diagrams by using Petri net based verification techniques. For verification purpose, we transform the integrated activity diagrams into Petri nets. Then, crosscutting properties of the system are refined as a set of CTL formulas. Last, the Petri net is verified against the refined CTL formulas. The verification result shows whether the Petri net satisfy the requirements or not. We can reason whether the integrated activity diagram meets the requirement since they are equivalent. In other words, we can claim that the aspect-oriented modeling is correct with respect to specified crosscutting requirements. Two case studies have been carried out to demonstrate the feasibility and effectiveness of our approach. Concerning the future work, we will focus on testing system implementations against aspect-oriented models have been verified.

ACKNOWLEDGMENT

We would like to thank Professor Karsten Wolf at University Bamberg, who is the author of LoLA, for his help in dealing with problems encountered when integrating LoLA into our tool Jasmine-AOV. This work is supported by the National Natural Science Foundation of China (No.61021062，No. 61170066), the National 863 High-Tech Program of China (No. 2012AA011205), and the National Grand Fundamental Research 973 Program of China (No. 2009CB320702).

REFERENCES

[1] Z. Cui, L. Wang, X. Li, and D. Xu. Modeling and integrating aspects with UML activity diagrams. In Proceedings of the ACM Symposium on Applied Computing, Honolulu, Hawaii, ACM, New York, NY, 2009, pp. 430-437.

[2] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. V. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-Oriented Programming. In the annual European Conference on Object-Oriented Programming, 1997, pp. 220–242.

[3] X. Li, Z. Liu, J. He, and Q. Long.Generating a Prototype From a UML Models of System Requirements. Distributed Computing and Internet Technology. Lecture Notes in Computer Science 3347, Berlin, Heidelberg: Springer, 2005, pp. 135-154.

[4] A. Fischer. Mapping UML designs to Java. In Proceedings of the 15th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, 2000, pp. 178-187.

[5] L. Wang, J. Yuan, X. Yu, J. Hu, X. Li, and G. Zheng. Generating Test Cases from UML Activity Diagram based on Gray-Box Method. In Proceedings of the 11th Asia-Pacific Software Engineering Conference, 2004, pp. 284-291.

[6] C. Nebut, F. Fleurey, Y. L. Traon, and J. Jezequel. Automatic Test Generation: a Use Case Driven Approach. IEEE Transactions on Software Engineering, Vol.32, No.3, 2006, pp. 140-155.

[7] M. Chen, X. Qiu, W. Xu, L. Wang, J. Zhao, and X. Li. UML Activity Diagram Based Automatic Test Case Generation for Java Programs. In The Computer Journal, Vol.52, No.5, Oxford Press, 2009, pp. 545-556.

[8] T. Murata, Petri nets: Properties, analysis and applications, Proceedings of the IEEE , Vol.77, No.4, Apr 1989, pp. 541-580.

[9] OMG, UML Superstructure v2.1, http ://www.omg.org/technology/documents/formal/uml.htm.

[10] K. Schmidt. LoLA A Low Level Analyser. In Proceedings of the Application and Theory of Petri Nets, 2000, pp. 465-474.

[11] H. Störrle, Structured nodes in UML 2.0 activities. Nordic J. of Computing, 11(3), 2004, pp. 279-302.

[12] H. Störrle. Semantics of Control-Flow in UML 2.0 Activities. In Proceedings of the 2004 IEEE Symposium on Visual Languages - Human Centric Computing, 2004, pp. 235-242.

[13] E. M. Clarke, E. A. Emerson, and A. P. Sistla, Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Trans. Program. Lang. Syst. 8, 2, pp. 244-263.

[14] J. Whittle, J. Araujo. Scenario Modelling with Aspects. In IEEE Software, Vol 151, Issue 4, Aug. 2004, pp. 157–172.

[15] A. Sampaio, A. Rashid, R. Chitchyan, and P. Rayson. EA-Miner: Towards Automation in Aspect-Oriented Requirements Engineering. Transactions on Aspect-Oriented Software Development III, Lecture Notes In Computer Science, Vol. 4620. Springer-Verlag, Berlin, Heidelberg, pp. 4-39.

[16] D. Xu, O. E. Ariss, W. Xu, and L. Wang, Aspect-Oriented Modeling and Verification with Finite State Machines, Journal of Computer Science and Technology, 24(5), Sept. 2009, pp. 949-961.

[17] G. Denaro, and M. Monga. An experience on verification of aspect properties. In Proceedings of the 4th international Workshop on Principles of Software Evolution, ACM, New York, NY, 2001, pp. 186-189.

[18] N. Ubayashi, and T. Tamai. Aspect-oriented programming with model checking. In Proceedings of the 1st international Conference on Aspect-Oriented Software Development, ACM, New York, NY, 2002, pp. 148-154.