Welcome to the IEEE International Conference on Software Architecture (ICSA)

Eduardo Almeida
General Chair
Conferences in Brazil

• 2011 – 7th International Conference on Open Source Systems (OSS)

• 2011 – 10th International Conference on Aspect-Oriented Software Development (AOSD)

• 2012 – 7th IEEE International Conference on Global Software Engineering (ICGSE)

• 2012 – 26th International Symposium on Distributed Computing (DISC)
Conferences in Brazil – cont.

- 2012 – 16th International Conference on Software Product Line (SPLC)
- 2013 – 17th International Conference on Evaluation and Assessment in Software Engineering (EASE)
- 2013 – 21st International Requirements Engineering Conference (RE)
- 2014 – 10th IEEE International Conference on e-Science (e-Science)
- 2016 – 10th International Workshop on Variability Modeling of Software-Intensive-Systems (VaMoS)
Conferences in Brazil – cont.

- 2017 – 16th International Conference on Software Reuse (ICSR)
- 2021 - IEEE International Conference on Software Testing, Verification and Validation (ICST)
About ICSA...

- It is the premier international conference in the field of software architecture, the first being held in 2017*

- Travelling around the world (Germany, USA, Sweden)

- First time in Brazil

- Premier forum for Software Architecture
  - Research
  - Industry

We hope you enjoy this year as well
Program Structure

• Research track
• Software Architecture in Practice (SAIP) track
• New and Emerging Ideas (NEMI) track
• Tool Demonstrations track
• Early Career Researchers Forum (ECRF) track

• **Journal first**
• 3 Tutorials
• 3 Workshops
Conference Organization

Eduardo Almeida  
General Chair  
Federal University of Bahia, Brazil

Sam Malek  
Program Co-Chair  
University of California, Irvine

Alessandro Garcia  
Program Co-Chair  
PUC-Rio, Brazil

Paulo Merson  
Program Co-Chair  
Brazilian Federal Court of Accounts (TCU)

Crescencio Lima  
Finance Chair  
Federal Institute of Bahia, Brazil

Ivan Machado  
Local Chair  
Federal University of Bahia, Brazil
Conference Organization

Nelly Bencomo
New and Emerging Ideas Co-Chair
Aston University, England

Thais Batista
New and Emerging Ideas Co-Chair
Federal University of Rio Grande do Norte, Brazil

Joshua Garcia
Early Career Researchers Forum Co-Chair
University of California, Irvine

Neil Ernst
Early Career Researchers Forum Co-Chair
University of Victoria, Canada

Matthias Naab
Software Architecture in Practice Chair
Fraunhofer Institute, Germany

Danny Weyns
Tool Demonstrations Chair
Katholieke Universiteit Leuven, Belgium
Conference Organization

Ali Babar  
Workshop Co-Chair  
University of Adelaide, Australia

Elisa Nakagawa  
Workshop Co-Chair  
University of Sao Paulo, Brazil

Henry Muccini  
Tutorial Chair  
University of L'Aquila, Italy

Iuri Souza  
Publicity Chair  
Federal University of Bahia, Brazil

Leandro Oliveira  
Publicity Chair  
Federal University of Bahia, Brazil

Paulo Silveira  
Proceedings Chair  
Federal Rural University of Pernambuco (UFRPE), Brazil
Conference Organization

Bastian Tenbergen
Media Co-Chair
State University of New York, Oswego, US

Roselane Silva
Media Co-Chair
Federal University of Bahia, Brazil

Tassio Vale
Media Co-Chair
Federal University of Recôncavo da Bahia (UFRB), Brazil

Larissa Rocha Soares
Student Volunteer Chair
Federal University of Bahia, Brazil
104 participants

- Conference only: 41 participants
- Workshops: 33 participants
- Full Conference: 30 participants
104 participants

- Brazil: 35 participants
- Germany: 20 participants
- USA: 8 participants
- Italy: 4 participants
- Finland: 3 participants
- Others: 34 participants
104 participants
Communication

- Use #icsa2020 in your Tweets
Keynotes

- Dr. Andre van der Hoek,
- UC Irvine, USA

Title: What Makes Expert Software Designers Successful?

Nov 4, 2020
Keynotes

- Dr. Tim Menzies,
- NC State University, USA

Title: SE for AI: ethics is a choice and not choosing is unethical (or, If Deep Learning is the answer, what was the question?)

Nov 5, 2020
Keynotes

Alexandre Freire and Henrique Alves,
Nubank, Brazil

Title: Building a bank outside-in: The challenges evolving Nubank to millions of customers and multiple products

Nov 6, 2020
Tutorial 1: Enabling Industry 4.0 with Eclipse BaSyx
  Thomas Kuhn, Frank Schnicke and Pablo Oliveira Antonino

Tutorial 2: Modeling Microservices with DDD
  Paulo Merson and Joe Yoder

Tutorial 3: Challenges and Approaches for the Assessment of Micro-Service Architecture Deployment Alternatives in DevOps
  Alberto Avritzer
Workshops

- **BlockArch**
  - Mohamad Kassab - Penn State University

- **SESoS/WDES**
  - Thais Batista (Federal University of Rio Grande do Norte, Brazil)

- **SEH**
  - Pablo Oliveira Antonino (Fraunhofer IESE, Germany)
Review Process

- 102 registered abstracts
- 87 valid submissions
- Minimum of 3 reviews per submission
- Virtual PC discussion
- 16 accepts + 3 conditional accepts
- 19 eventual accepts (22% acceptance rate)
Awards
Enforcing Architectural Security Decisions

Stefanie Jasser
Department of Informatics
University of Hamburg
Hamburg, Germany
jasser@informatik.uni-hamburg.de

Abstract—Software architects should specify security measures for a software system on an architectural level. However, the implementation often diverges from this intended architecture including its security measures. This may lead to severe vulnerabilities that have a wide impact on the system and are hard to fix afterwards. In this paper, we propose an approach for checking the implementation’s conformance with the defined security measures using architectural security rules: We extend a controlled natural language approach to formalize these rules and use dynamic analysis techniques to extract information on the actual system behavior for the conformance check. We evaluate our approach by an industrial case study to show the applicability and flexibility of our conformance checking approach.


I. INTRODUCTION

It is a highly complex task to design and implement a secure software system: Architecture design decisions have wide impact on the system [1], [2]. They should respect its required quality attributes. An important quality attribute is a system’s security. Hence, taking security into account during architecture design is crucial for developing a secure software system. However, many software engineers only have basic knowledge on systematically build a secure software system, today [3]. This makes secure design and development not only time-consuming but also challenging to many software engineers.

Usually, only security flaws and security bugs are con-

Checking the implementation’s conformance with the architectural security rules is an important activity to avoid vulnerabilities that are caused by architecture violations. Several approaches and tools exists for rule-based architecture conformance checking, such as [6]–[11]. Each approach defines a domain specific language to describe the intended software architecture and as well as for mapping the intended architecture’s concepts to the source code in order to conduct a conformance check. We identified three issues when using these approaches for architectural security rules: Most approaches concentrate on structural rules but do not consider system behavior. As behavioral rules are essential for security this may cause undetected software vulnerabilities (Issue 1). Most conformance checking techniques base on static analysis of source code. They do not consider dynamically loaded sources and content which are important for a security analysis (Issue 2). Existing conformance checking approaches do rarely consider artifact beyond source code. However, considering additional source artifact such as configuration settings and their impact on the system behavior is essential when conducting a security analysis (Issue 3).

We propose a conformance checking approach to monitor and, thereby, enforce architectural security rules. We base our work on the ontology-based approach of Schröder et al. [6] that allows to flexibly define an architecture concept language and its rules. We extend the proposed formalization language with additional language constructs in order to support the formalization of architectural security rules. In Section IV we further present an approach to dynamically analyze the
Most Influential Paper Award

• All CBSE (2010) and QoSA (2010) papers

• Top 5 papers from CBSE/QoSA based on num. citations

• Voting process
  – PC Co-chairs (3)
  – Industry members (2)
    • Jens Knodel
    • Matthias Naab
Evaluating Maintainability with Code Metrics for Model-to-Model Transformations

Lucia Kapová, Thomas Goldschmidt, Steffen Becker, and Jörg Henss
Chair for Software Design and Quality, Universität Karlsruhe (TH), 76131 Karlsruhe, Germany
{kapova, henss}@ipd.uka.de
FZI Forschungszentrum Informatik, 76131 Karlsruhe, Germany
{goldschmidt, sbecker}@fzi.de

Abstract. Using model-to-model transformations to generate analysis models or code from architecture models is sought to promote compliance and reuse of components. The maintainability of transformations is influenced by various characteristics - as with every programming language artifact. Code metrics are often used to estimate code maintainability. However, most of the established metrics do not apply to declarative transformation languages (such as QVT Relations) since they focus on imperative (e.g. object-oriented) coding styles. One way to characterize the maintainability of programs are code metrics. However, the vast majority of these metrics focus on imperative (e.g., object-oriented) coding styles and thus cannot be reused as-is for transformations written in declarative languages. In this paper we propose an initial set of quality metrics to evaluate transformations written in the declarative QVT Relations language. We apply the presented set of metrics to several reference transformations to demonstrate how to judge transformation maintainability based on our metrics.
Thank you all
Authors, Organization Committee,
PC members and All

Enjoy ICSA 2020