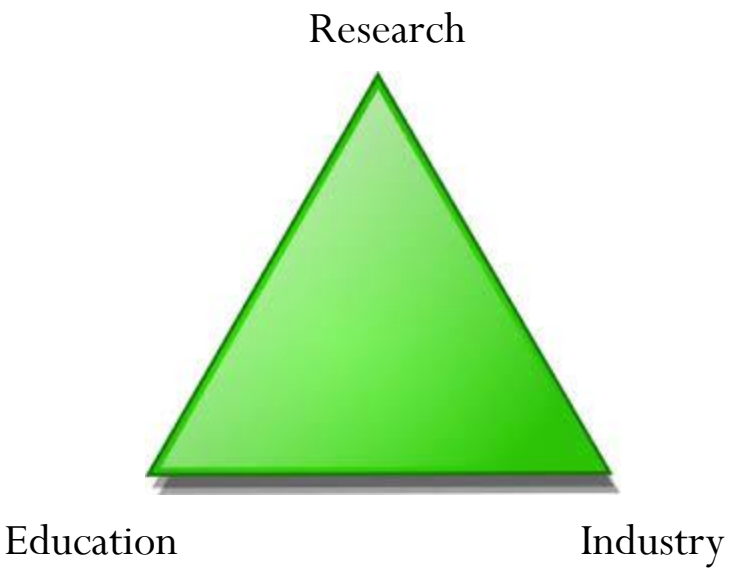
Software Engineering and Information Processing Lab in 2014

Tihana Galinac Grbac

UNIVERSITY OF RIJEKA
Faculty of Engineering

Activities

- Projects
- Collaorations
- Organization of events
- Papers and conferences
- Other



UNIVERSITY OF RIJEKA
Faculty of Engineering

Projects

- We applied for 9 projects:
  - Horizon 2020, Structural funds, Croatian Scientific Foundation
  - For some projects we still wait results
- We are currently preparing 3 new applications

UNIVERSITY OF RIJEKA
Faculty of Engineering

Analysis and inovative approaches to software development, management and application of complex software systems

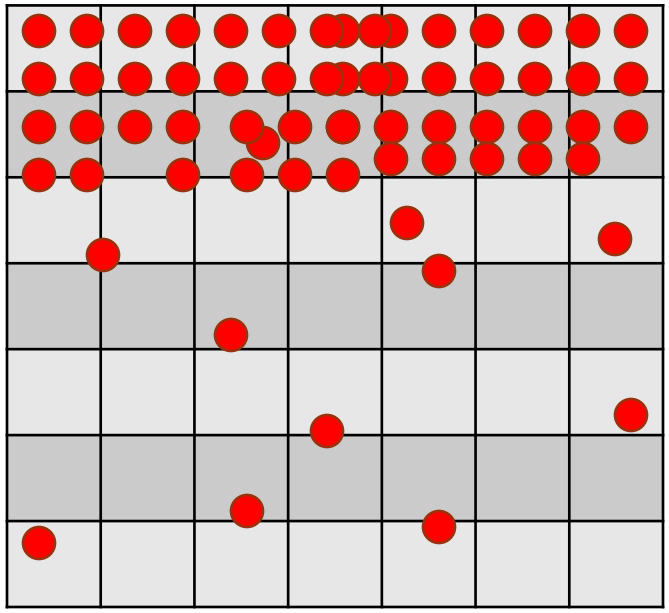
UNIVERSITY OF RIJEKA
Faculty of Engineering

Projects

Complex systems

- Number of levels of abstraction
- Global properties of system and local properties describing component behaviour
- Imposible to derive simple rules from local properties towards global properties

System and system components



Source: Complex software systems : Formalization and Applications -
Work done in EU project GENNETTEC: GENetic NeTworks: Emergence and Complexity

How to secure quality of complex software systems?

- Software Quality Assurance
  - a planned and systematic pattern of all actions necessary to provide adequate confidence that the software item conforms to its established requirements [IEEEStdGlos]
- Software testing is the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item
- Number of possible test cases is infinite that is esspecially the case with complex systems
- One result of testing process is software failure

Software Fault and failure

```
public double getCurrentRate(String fromCurrency, Strin
    if (fromCurrency == null || toCurrency == null) {
        return Double.NaN;
    }
    double answer = 0;
    if (fromCurrency == null) {
        return answer;
    }
    if (toCurrency.equals("USD") && fromCurrency.equals
        answer = USD_CDN;
    }
    return answer;
}
```

Condition 'fromCurrency == null' is always 'false' more... (Ctrl+F1)

Fault execution leads to system failure

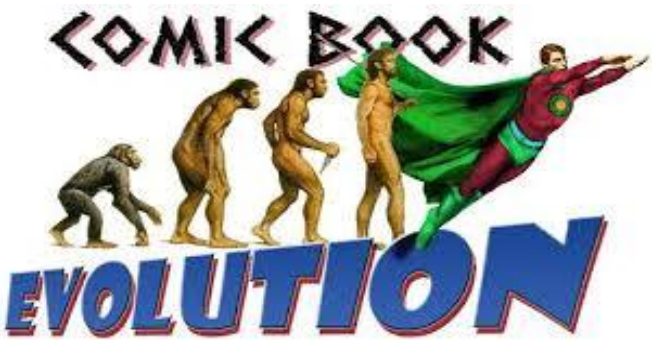
One system failure may be result of several software faults

One software fault may cause several system failures

Source: http://vapresspass.com/2013/04/24/failure-is-not-fatal-by-marcia-zidle/

Fault costs and complex systems

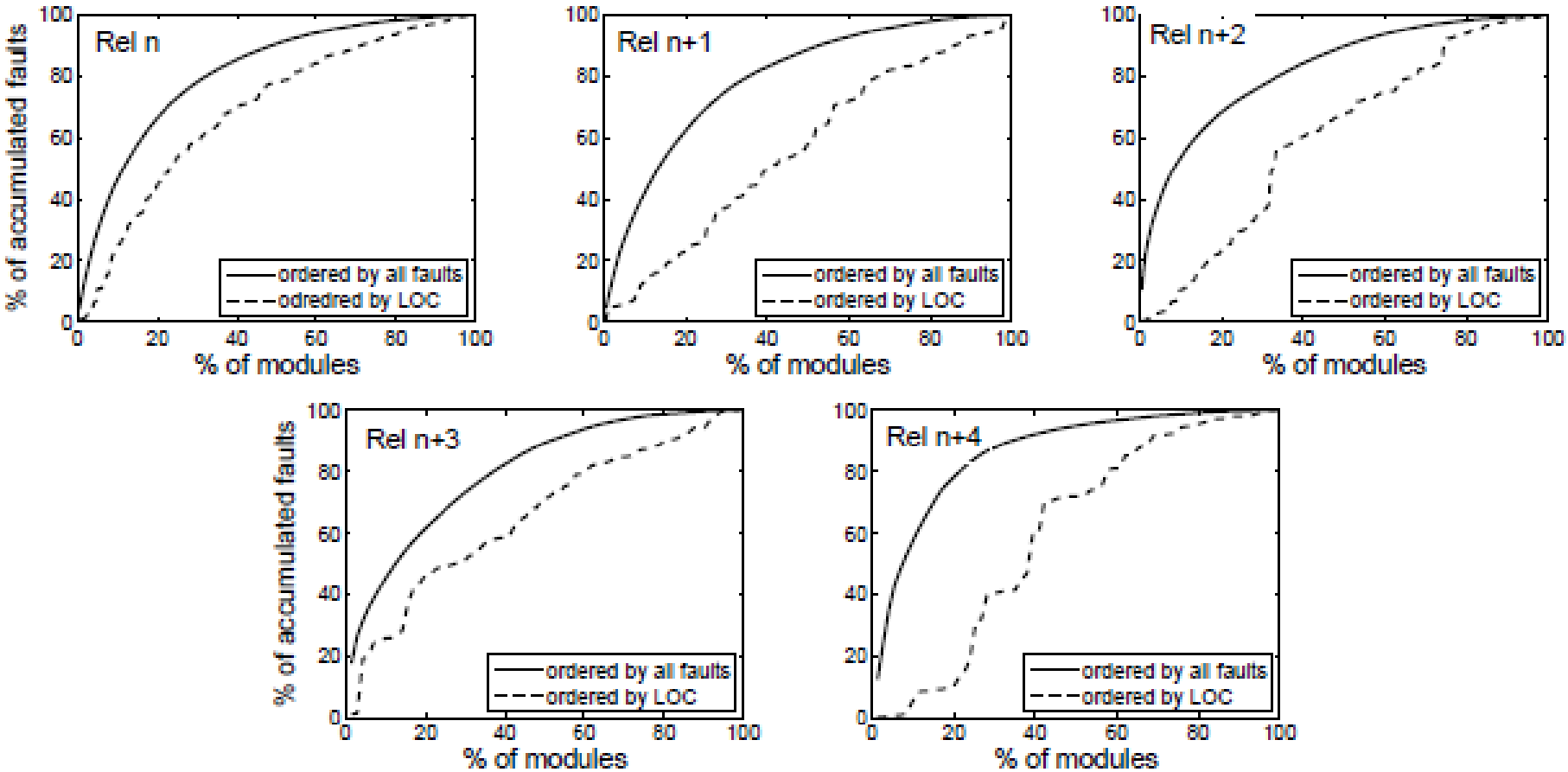
- Evolving system demands reusability
- Usually impacts thousands of end users
- Number of system versions may coexist at the same time
- Consequences of faults are impossible to predict
- Problem is not only effect of one fault, but effect of its repairment on the system as a whole



Fault distributions

- 2000: First systematic study on fault distributions by N.E. Fenton and N. Ohlsson in [ORIG] based on Basili and Pericone [BASPER]
- 2007: 1st systematic replication by C. Andersson and P. Runeson in [1REPL],
- 2013: 2nd systematic replication by T. Galinac Grbac, P. Runeson and D. Huljenić in [2REPL]

Effects of module size and complexity on fault proneness



Accumulated percentage of number of faults when modules are ordered with respect to LOC

Analytical fault distributions

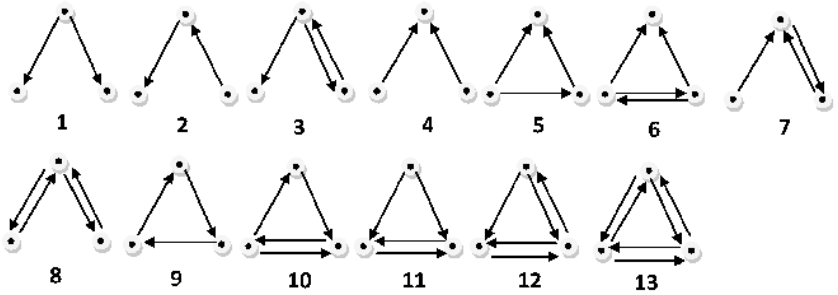
- All previous principles ultimately depend on the underlying probability distribution
- the fulfillment of a certain empirical principle does not determine the probability distribution uniquely
- The distibutions like double Pareto, Weibull, lognormal, Pareto, and Yule-Simon with power-law in the tail are confirmed

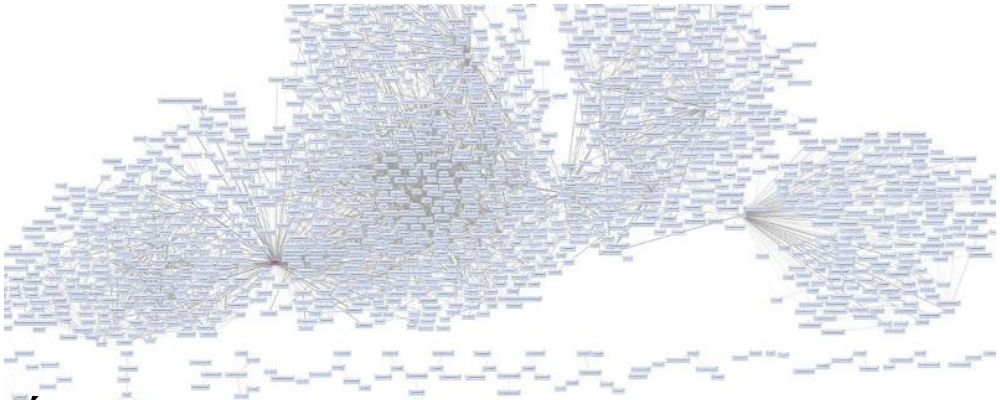
2009. Les Hatton. Power-Law Distributions of Component Size in General Software Systems. IEEE Trans. Software Eng. 35(4): 566-572

2014 Tihana Galinac Grbac, Darko Huljenić. On the Probability Distribution of Faults in Complex Software Systems, Information and Software Technology, published online first.

Software structure

- Software structure represented as dependency graph between structural components show promisable results in modeling fault behaviour
- Some subgraphs and motifs are dominant in faulty software structures





Petrić, Jean; Galinac Grbac, Tihana. ***Software structure evolution and relation to system defectiveness*** *// Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. ACM New York, NY, USA : ACM, 2014.*

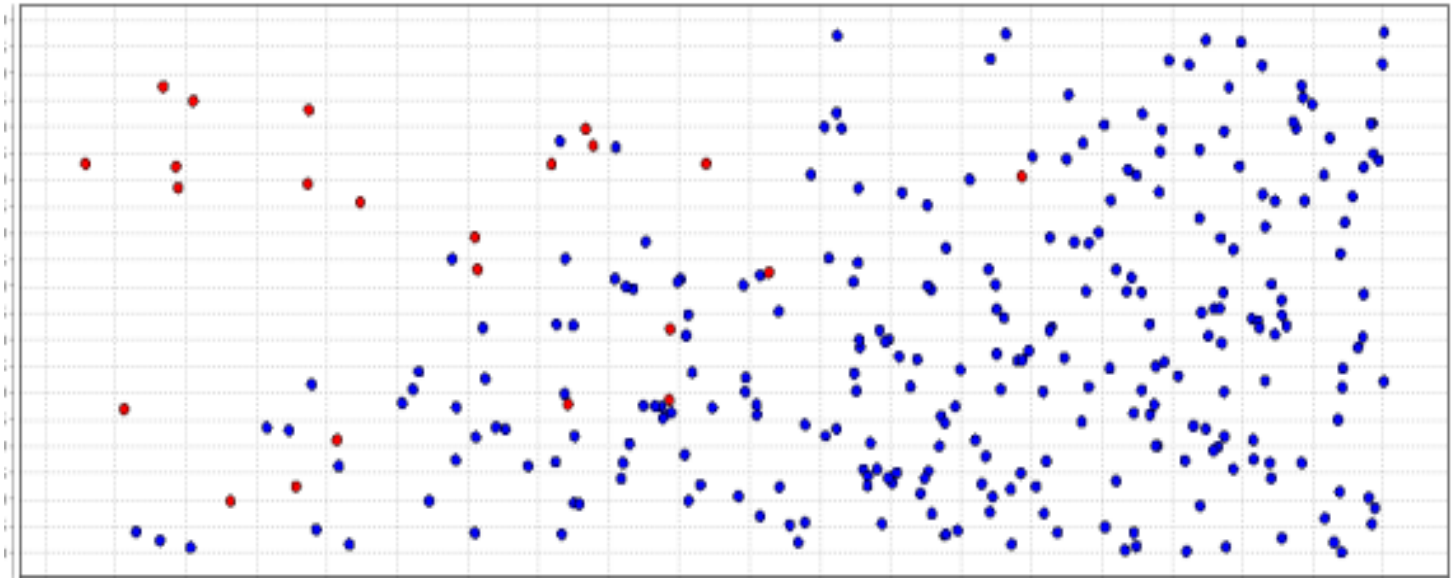
Fault Prediction using Classification modelling

1) Statistic classificators
   - Linear Discriminant Analysis
   - Quadratic Discriminant Analysis
   - **Logistic Regression**
   - Naive Bayes
   - Bayesian Networks
   - Least-Angle Regression
   - Relevance Vector Machine
3) Nearest neighbor methods
   - k-Nearest Neighbor
   - K-Star
5) Neural networks
   - Multi-Layer Perceptron
   - Radial Basis Function Network

2) Support vector machine
   - Support Vector Machine
   - Lagrangian SVM
   - Least Squares SVM
   - Linear Programming
   - Voted Perceptron
4) Decision trees
   - C 4.5 Decision Tree
   - Classification and Regression Tree
   - Alternating Decision Tree
6) Ensemble methods
   - Random Forest
   - Rotation Forest
   - Logistic Model Tree

Genetic approach

- Problems:
  - Unbalanced datasets
  - Soft computing approaches did not come to common solution
  - Data are very sensitive on linking bias



Predicting and simulating complex software systems behaviour

- Software become central part of the modern network, allow reconfiguration, flexibiliy
- It should run on any hardware, serve to many users, satisfy their complex communication needs and deliver proper ICT service, effectively and efficiently
- Modern software has to be flexible on network context, information context, communication context, ...
- Modern network should provide **reliable and robust** ICT services (resistant against system failures, cyber-attacks, high-load and overload situations, flash crowds, etc.)

Software in 'Internet of Service'

- In service oriented architecture software is provided '**as a service'**
- In that concept 'of service' is referring to a technical understanding of software functions provided as Web service
- IoS combine that services and integrate functionalities that led to **complex service chains**
- Usually these service chains are developed by number of providers and offered to number of users
- Service chain composition is happening at layers above network layer
- Problem is how to secure quality of these service chains
- We need algorithms for autonomous control for a reliable IoS

Key problems with software evolution

More and more software systems tend to evolve towards complex software systems (e.g. IoS)

Interconnection of peripheral systems over distributed network into system of systems (IoT)

- Key problems become:
  - Can we develop foundations on software behavior?
  - How can we measure software behaviour in network?
  - Can we predict and simulate software behaviour in network?
  - How to manage complex software system?
  - Are we able just by observing properties of system parts to predict its overall behaviour?

EU project ACROSS

1. Softwarization of networks (Javier Sainz Guerra, Kurt Tutschku)
2. Cross-layer QoE-management (Peter Pocta)
3. QoS-aware service composition (Rob van der Mei)
4. Cloud federations (Wojtek Burakowski)
5. Energy/cost-aware management control (Javier Sainz Guerra)
6. New transport protocols (Ake Arvidsson)
7. Quality/cost management (Stoyan Poryazov)
8. QoS/QoE modelling (Tobias Hossfeld, Thomas Bohnert)
9. QoS/QoE monitoring (Petteri Mannersalo, Phillip Leitner)
10. Human factors (Katarzyrna Wac)
11. End-user monitoring (Petteri Mannersalo, Toni Maki)
12. Effective monioting of encrypted 'grey' traffic (Ake Arvidsson)
13. Prediction and modelling of service composition performance, quality and reliability (Tihana Galinac Grbac)
14. Provisioning of context-dependent pricing, charging and billing (Peter Key)

IC1201: Behavioural Types for Reliable Large-Scale Software Systems (BETTY)

- **Behavioural type theory** encompasses concepts such as interfaces, communication protocols, contracts, and choreography.
- As stuctural principle for building reliable software systems
- Idea:
  - to codify the structure of communication to support the development of reliable communication-oriented software.
  - to encode as types the communication structure of modern computer systems and statically verify behavioural properties about them

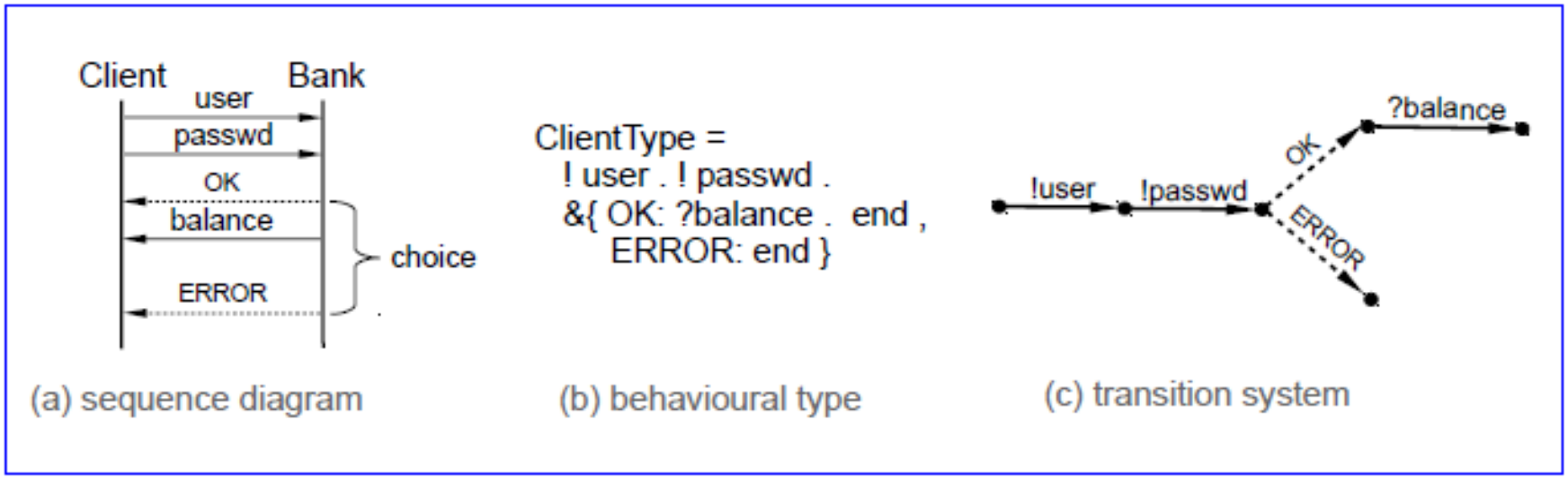
UNIVERSITY OF RIJEKA
Faculty of Engineering

cost
European Cooperation in Science and Technology

BETTY

Example – Session types

- Aim: to develop programming languages, tools for development of certified software solutions for global services

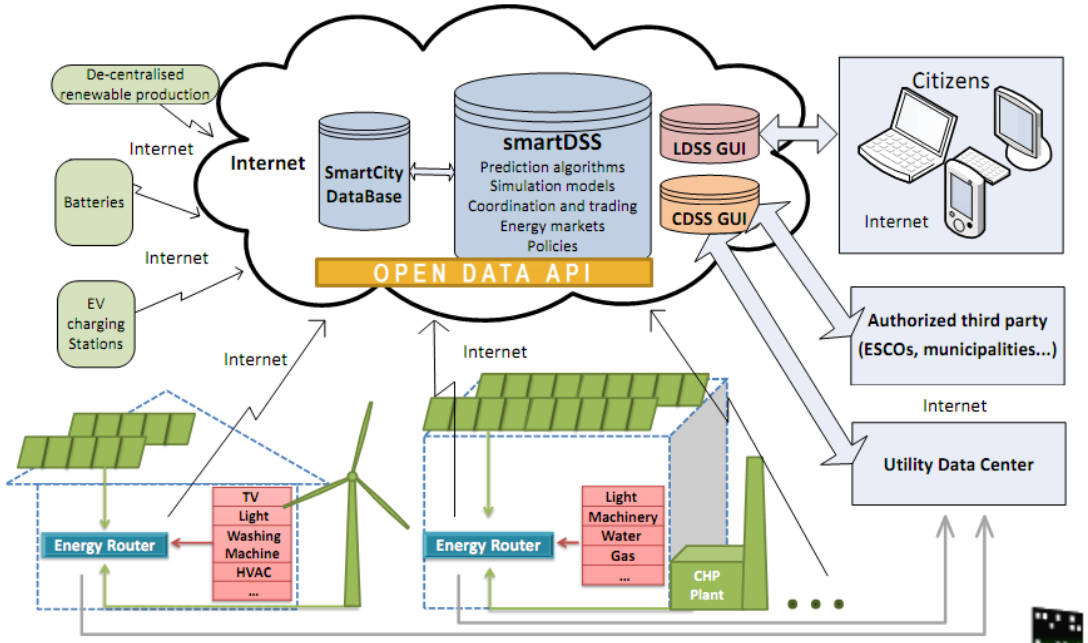


- Developed language: e.g. Scribble

UNIVERSITY OF RIJEKA
Faculty of Engineering

EU project iURBAN

- The iURBAN project will develop and validate a software platform that will integrate ICT energy management systems in two European cities – one is Rijeka.



UNIVERSITY OF RIJEKA
Faculty of Engineering

iURBAN

Projects

Software Engineering: Computer Science Education and Research Cooperation

- The aim of this project is to define a common curriculum for the undergraduate/graduate study programme on software engineering that would be used (partly or as a whole) in participating institutions.

UNIVERSITY OF RIJEKA
Faculty of Engineering

Stabilitätspakt für Südosteuropa
Gefördert durch Deutschland
Stability Pact for South Eastern Europe
Sponsored by Germany

Projects

Collaborations with other research institutions

- Sweden, Lund
- Sweden, Maraldalen
- Austria, Technishe Universitat Wien
- Hungary, Faculty of Informatics, Eötvös Loránd University
- Srbija, Odjel za matematiku i informatiku, Sveučilište u Novom Sadu
- Slovenija, Institut za informatiku, Fakultet za elektrotehniko računalništvo i informatiko
- Hrvatska, FER, ETF-Osijek, FESB, Politehnički fakultet u Puli

UNIVERSITY OF RIJEKA
Faculty of Engineering

Colleborations

Industry collaborations

- Ericsson Nikola Tesla, Zagreb
- Superius, Pula
- HT, Rijeka, Zegreb

UNIVERSITY OF RIJEKA
Faculty of Engineering

Colleboratations

Summer school - industry collaboration in SEIP Lab

Projects:

Mario Blažević, Diego Sušanj: Multiplatform Visualcom Client and Recording Server

Iva Petrović, Paolo Perković: Shadow Based Spatial Augmented Reality Concept for eHealth

Teo Manojlović: Smart M2M Application for Building and People Management

Government collaborations

- City of Rijeka – Smart City
- Primorsko-Goranska County – ICT in rural areas

iURBAN

EGOV4U
connecting everyone

UNIVERSITY OF RIJEKA
Faculty of Engineering

2020
Rijeka Healthy City

Colleborations

BETTY Summer School June 2014
Lovran, Croatia

UNIVERSITY OF RIJEKA
Faculty of Engineering

Events

SQAMIA Workshop October 2014, Lovran, Croatia

UNIVERSITY OF RIJEKA
Faculty of Engineering

Events

Invited talk – Prof. Per Runeson, Lund University, Sweden

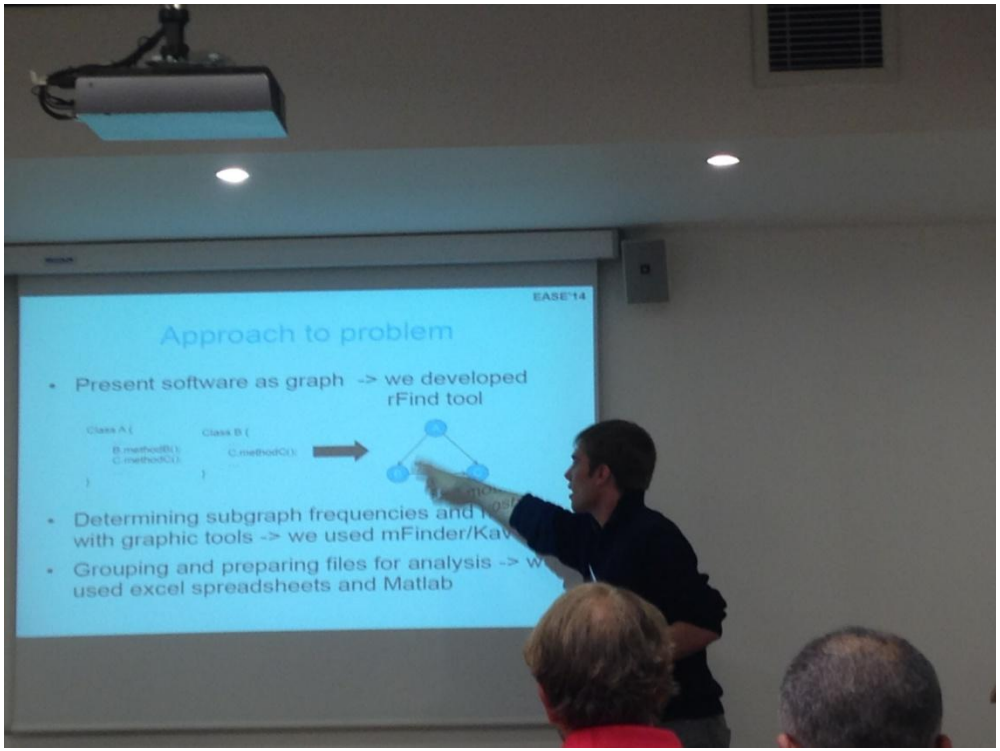
Events

MIPRO 2014

Events

Publications and conferences

- International Conference on Evaluation and Assessment in Software Engineering, May 2014, London



UNIVERSITY OF RIJEKA
Faculty of Engineering

Publications

Publications and conferences

- 3rd Workshop on Software Quality Analysis, Monitoring, Improvement, and Applications SQAMIA 2014, September, 2014, Lovran, Croatia

UNIVERSITY OF RIJEKA
Faculty of Engineering

Events

Publications and conferences

- 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM

UNIVERSITY OF RIJEKA
Faculty of Engineering

Events

Publications and conferences

Journal papers:

- Galinac Grbac, T., Huljenić, D., On the Probability Distribution of Faults in Complex Software Systems, *Information and Software Technology*, Vol.58, 2015, pp. 250-258.
- Galinac Grbac, Car, Ž., Huljenić, D., A Quality Cost Reduction Model for Large Scale Software Development, *Software Quality Journal*, accepted, published online first.
- Grguric A., Brestovac G., Marin D., Oroz T., Vidović A., Bozóki S., Mošmondor M., Galinac Grbac T., *Ambient orchestration in assisted environment, Engineering Review*, May 30, 2014

Conferences:

- Tanković N., Galinac Grbac T., Žagar M., Experiences from Building an EUID Business Portal, Proc. of MIPRO, CTI2014, pp. 635-640.
- Petric J., Galinac Grbac T., *Software structure evolution and relation to system defectiveness*, Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering *EASE2014*
- Petrić, Jean; Galinac Grbac, Tihana; Dubravac, Mario. Processing and data collection of program structures in open source repositories, *Proceedings of SQAMIA 2014*, 57-66
- Mauša, Goran; Perković, Paolo; Galinac Grbac, Tihana; Štajduhar, Ivan, Techniques for Bug–Code Linking, *Proceedings of SQAMIA 2014*
- Mauša, Goran; Galinac Grbac, Tihana; Dalbelo Bašić, Bojana **Software Defect Prediction with Bug-Code Analyzer - a Data Collection Tool Demo**, *Proc. of Softcom 2014*

Other Journals

- InfoTrend – Service Oriented Architecture

Ericsson_... | centralized... | Questionnaire ... | software s... | Uniri | InfoTrend / | Student P... | Photos - ... | SEIP L...

www.seiplab.riteh.uniri.hr

seip lab

Most Visited | Getting Started | Latest Headlines | Korisno | Apple | Yahoo!

SEIP Lab

Software Engineering and Information Processing Laboratory

Home

Welcome to the web-site of the Software Engineering and Information Processing Lab (SEIP Lab). We are group of interdisciplinary researchers with diverse backgrounds within Department of Computer Engineering at Faculty of Engineering, University of Rijeka.

Software Engineering and Information Processing Lab (SEIP Lab) is established in 2011. Its mission is the synergy of research and education in the field of software engineering and information processing. The research is motivated by the collaboration with other research groups and the real problems and needs through the collaboration projects with industry partners. Providing practical experience and challenges, these collaboration projects with industry strongly support the education of young scientists and professionals in the field. Through research, education and technology transfer, the SEIP Lab aims to become the regional excellence center that produces scientists able to compete in the European Research Area, and software professionals that would contribute to the regional software development capability.

Language
Hrvatski

Start | HR | 100% | 15:51 7.1.2015.