REFSQ 18
Poster Slam!
1 minute each 😊
1. Tool Support for Value Modeling and Risk Analysis of e-Services
2. The Interactive Narrator Tool: Effective Requirements Exploration and Discussion through Visualization
3. Multiple Criteria Decision Support in Requirements Negotiation
4. Elicitation of SME Requirements for Cybersecurity Solutions by Studying Adherence to Recommendations
5. ORSIM: Integrating existing software components to detect similar natural language requirements
6. Managing Multi-Lingual User Feedback: the SUPERSEDE project experience
8. Back to Basics: Extracting Software Requirements with a Syntactic Approach
9. PACAS: A Gamified Platform for Participatory Change Management in Air Traffic Management Systems
Value engineering iterates over four activities.

- **Value network design**
  - Make a map your business network and explore what new services or products you could deliver with new technology, and what this would mean for your relation with partners in your network.
  - Quantify the value of services and products delivered, make assumptions about frequency of transactions, and estimate required investments.

- **Technology choice**
  - Choose technology to perform the commercial transactions in the value model. Update the value model with the commercial possibilities of the new technology.

- **Risk assessment**
  - Simulate different scenarios to compute profitability and assess sensitivity to your market assumptions. Revise the value model if necessary.
  - Automatically generate vulnerabilities to fraud, and rank them on severity. Revise the value model if necessary.

- **Process design**
  - Once you are satisfied with your peer-to-peer business model, map this to your business processes. Re-evaluate risk and profitability.

Value network design: e-value model of community radio journalism in rural Mali, with profitability sheet.

Risk assessment: Screenshot of a sensitivity analysis for market assumptions in a rights clearance model (left) and a fraud risk assessment generated for a flat-rate telecommunication subscription model (right).
The Interactive Narrator Tool
Effective Requirements Exploration and Discussion through Visualization

Slob, Dalpiaz, Lucassen & Brinkkemper

Context and Motivation
Natural language is the predominant notation for software requirements. In agile software development, requirements are often expressed as user stories:

- As a (role), I want (goal), so that (benefit).
- As a (actor), I want to choose an event, so that I can buy tickets for that event.

Problem:
It is hard to make accurate mental models of the system under development from a large quantity of user stories. This hinders the understanding of the system.

Background and Goal
Our previous Visual Narrator tool automatically extracts conceptual models from user stories by means of Natural Language Processing techniques. Its aim is to reduce cognitive overload through requirements visualization [2]. Our evaluation [3], however, proves these models quickly become too large for practitioners to effectively process.

Research goal:
To deliver conceptual models that are easy to read, so discussion is facilitated and software requirements are improved as a result.

Solution
Use requirements visualization with Slob’s value information seeking module “imagine first, zoom/flip details on demand” [4] to create a readable and interactive conceptual model. Support defect detection to improve requirement quality [5].

Core Functionality:
- Zoom in/out on specific areas
- Filter the most important entities
- Filter per sprint and/or role
- Turn relationships on/off
- View full user stories

Ambiguity resolution
Duplicate prevention
Conflict detection

Workflow & Architecture
- Interactive Narrator with Visual Narrator to canvas entities and relationships from the user stories.
- The entities represent nouns, for example: User, Event, Ticket.
- The relations represent verbs, for example: Check, Buy.
- The relationships are depicted as lines connecting between the entities.

Results from Ongoing Evaluations
Method
Explorative case studies with 5 participants from software companies were conducted. The participants used the tool on a 34-inch video wall touch-screen.

Findings
- It creates an overview, a bigger picture of a project
- Relationships help to identify connections between entities
- The automatic and quick generation of the visualization is attractive
- Models can be continuously presented easily after a short run-up time
- It creates a broader understanding of requirements, but not a deeper one
- It can help to identify inconsistencies, accidents and dependencies

Conclusions and Future Work
The Interactive Narrator is able to automatically and quickly create conceptual models from user stories. The filtering and zooming mechanisms increase the readability of the models to a suitable level. Because of the relationships, interactive capabilities and the animation of the data in user stories the tool facilitates discussion and can help to make requirements better.

Future work includes:
1. Improving support for the detection of inconsistencies, contradictions and dependencies.
2. Implementing semantics-based clustering to further reduce visual clutter.
3. Implementation capability to edit user stories from the visualization.

References

Contact
gj.dalpiaz@uu.nl
Can we create a visual representation of requirements that is comprehensible and helps to improve (understanding of) requirements?
Multiple Criteria Decision Support in Requirements Negotiation

Siamak Farshidi, Slinger Jansen, Rolf de Jong, Sjaak Brinkkemper
Multi-Criteria Decision-Making (MCDM) Problems

- Database Management System selection
- Cloud Service Provider selection
- Software Architecture Pattern selection
- Blockchain solution selection
Decision Support System

Decision-Makers:
- Software Architects
- Software Developers
- ...

Requirements & Priorities

DBMS Decision Model

CSP Decision Model

SAP Decision Model

DBMS Selection

CSP Selection

SAP Selection

(1) MySQL
(2) SQL Server
(1) Microsoft Azure
(2) Google Cloud Platform
...
Elicitation of SME Requirements for Cybersecurity Solutions by Studying Adherence to Recommendations

To mitigate SME cybersecurity problems, we aim to do requirements elicitation by studying how cybersecurity experts provide advice to SME.

SME characteristics regarding cybersecurity:
- Lack of investment and budget restrictions
- Lack of security awareness and skills
- Lack of cost-effective processes
- Concerning external threats and neglecting internal ones
- A lack of internal cyber-security policy

Some cyber threats specific to SME:
- Vulnerable Software
- Injection
- Malicious insiders
- Broken Authentication
- Denial of Service
- Insecure Direct Object References
- Sensitive Data Exposure
- Bad Configurations

Cybersecurity Solution for SME Requirements Elicitation

We aim to answer
- What are the hurdles and enablers of SME to adopt cybersecurity solutions?
- Can the study of adherence to cybersecurity practice be used as a method of requirements elicitation for improving cybersecurity solutions?
- Can requirements elicitation be automated by embedding the dialogue between the cybersecurity expert and the person in charge of the SME in the CYSEC tool?

Contact: Alireza Shojaifar, Alireza.Shojaifar@fhnw.ch | Samuel A. Fricker, Samuel.Fricker@fhnw.ch | Martin Gwerder, Martin.Gwerder@fhnw.ch | www.fhnw.ch/technik
ORSIM – OpenReq SIMilarity
Integrating existing software components to detect similar NL requirements
C.A. Furnari, C. Palomares, X. Franch

• Systematic evaluation of existing similarity detection components for NL texts

• No worries about components’ set up!

• Integrated components:
  • Cortical
  • Gensim
  • Paralleldots
  • Semilar

• Basics for other tasks: reuse, dependencies
Managing Multi-Lingual User Feedback: the SUPERSEDE project experience

An experience report

Fitsum Meshesha, Anna Perini*, Angelo Susi
*perini@fbk.eu

FBK, Center for Information and Communication Technology – ICT Trento (Italy) - Software Engineering Unit - http://se.fbk.eu
SUPERSEDE Objective
Enable a data-driven engineering process
Textual feedback analysis process

1. Dataset preparation: manual annotation is performed by a domain expert
2. Pre-processing: uninformative tokens are removed
3. Feature extraction: different linguistic properties and sentiment are extracted
4. Building a Feedback classifier: machine-learning techniques are employed to train a classifier on a (portion of the) dataset
5. Feedback classification: the Feedback classifier is applied to incoming feedback to classify it as Bug Report, Feature Request, Enhancement Request, and Other
What if you discover later, when project is already running, that user textual feedback is not only in English?
**Context description:** Early stages ideas and challenges in the application of Machine Learning techniques to defect detection in Natural Language requirements.

- All requirements depend on domain knowledge.
- Domain-specific concepts and terminology.
- Requirements need to be tailored for each customer.
Back to Basics: Extracting Software Requirements with a Syntactic Approach

Matthew Caron
Paderborn, University
Motivation

non-experts
crowdsourcing
custom
supervised
processing
inconsistency
non-expert
data
rule-based
research
software
complements
vagueness
developers
algorithm
natural language
technical
syntactic
software requirements
validation
Focus

Back to Basics: Extracting Software Requirements with a Syntactic Approach

Motivation
- Companies need, more than ever before, solutions tailored to their exact needs
- Custom software solutions are not always available and need to be developed anew
- Fundamental challenge: Extracting and understanding software requirements written in natural language

Vision
- Syntactic rule-based extraction tool for software requirements
- Main objectives:
  - Allow non-expert users to voice their needs in unfiltered natural language
  - Provide developers with comprehensive, structured, and complete information

Ongoing
Phase 1
- Identification and extraction of Subject-Verb-Object triples (SVO)
- Identification and extraction of complements
- Identification and extraction of negative words
- Sequential ordering (Lexicon-based)
- Disambiguation

Phase 2
- Refinement of the syntactic rule-based extraction algorithm based on crowdsourced data
- Development of a classification model for the validation of extracted requirements
- Improvement of the sequential ordering algorithm

Phase 3
- Detection and handling of inconsistency
- Detection and handling of incompleteness
- Detection and handling of vagueness

Architectural Phase 1
- Input + Tokenization (sentences) → Sequential Ordering
- Disambiguation API → Dependency Parser
- JSON Output

Summer 2018
Winter 2018
PACAS: A Gamified Platform for Participatory Change Management in ATM Systems

Elda Paja¹, Mauro Poggianella¹, Fatma Başak Aydemir², Paolo Giorgini¹
¹University of Trento and ²Utrecht University

Utrecht, March 20, 2018
A Gamified Platform for Participatory Change Management in ATM Systems

Elda Paja³, Mauro Poggianella³, Fatma Basak Aydemir³, and Paolo Giorgini³

³University of Trento, Italy; ³Utrecht University, The Netherlands

Contact: elda.paja@unitn.it

PACAS OVERVIEW

The main objective of PACAS is to better understand, model and analyze changes at different layers of the Air Traffic Management (ATM) system to support change management, while capturing how strategic and design choices influence the overall system. PACAS relies on three main pillars:

- **PILLAR #1**: Gamified experience
- **PILLAR #2**: Multi-view modelling
- **PILLAR #3**: Automated reasoning techniques

PACAS PROCESS & PLATFORM

Web-based
https://pacas.disi.unitn.it/pacas-review/

Modular
Extendable - add new languages

Reasoning as services
Easily integrate new techniques

VALIATION

The validation consists of an iterative process with the active participation of an external Advisory Board (ABB) of domain stakeholders throughout the whole duration of the project. The first two phases, “Requirements” and “Use case” definition have been validated through Workshops WS#1 and WS#3, along the first wireframe prototypes of the platform supporting multi-view modelling. Change impact propagation has been validated in WS#2, while the final version of the platform and the multi-criteria decision analysis were validated at WS#3.

Consortium

REFSQ’18 Posters & Demo
The PACAS platform

Web-based
Available and can be used freely

Automated reasoning as a services
Integrate easily new services

Modular
Extendable
Support for new languages