Observational Models of Requirements Evolution - Theory and Practice

Massimo Felici
LFCS, School of Informatics, The University of Edinburgh
massimo.felici@ed.ac.uk
http://homepages.inf.ed.ac.uk/mfelici/

The Critical System Conference
29th-30th September 2004
Overview

0 On Requirements Evolution...
   o Socio-technical grounds

0 Empirical analyses
   o Capturing Requirements Evolution

0 Observational Models of Requirements Evolution
   o A bit of theory

0 Evolution as Dependability
On Requirements Evolution...

Requirements Evolution is one of the main issues that affect development activities as well as system features (e.g., system dependability). Although researchers and practitioners recognise the importance of requirements evolution, research results and experience are still patchy. This points out a lack of methodologies that address requirements evolution.
On Requirements Evolution

- Do requirements evolve? Yes!!!
- Can requirements be frozen? No
  - you might try, but it wouldn’t work…
  - The completeness and correctness panaceas… are just utopias
- Why do requirements evolve?
  - It is possible further to understand requirements evolution from a **socio-technical** viewpoint
    - Systems Approach, Heterogeneous Engineering, Social Shaping of Technology, etc.
- **Empirical analyses** of requirements evolution are still patchy
  - Lack of support for analyzing evolutionary data
- **Organizational** culture/strategy
  - Requirements evolution (analysis) as work practice
  - Requirements evolution captures organizational knowledge
Capturing Requirements Evolution...

Everyone talks about Requirements Evolution, but just few know how it looks like.

No one talks about Requirements Evolution.
Capturing Requirements Evolution...

0 Requirements evolution as trends of requirements changes
   o added, deleted and modified requirements

0 An increasing number of requirements is due to
   o Stakeholders interaction, technology issues/constraints,
     further understanding of system requirements,
     Requirements refinements, new arising requirements,
     Requirements dependencies, operational usage, etc.
Capturing Requirements Evolution...

- **Requirements (changes) classification**
  - “A classification is a spatial, temporal or spatio-temporal segmentation of the world. ... In an abstract, ideal sense, a classification system exhibits the following properties: There are consistent, unique classificatory principles in operation; The categories are mutually exclusive; The system is complete.”
  - [Bowker and Star, 1999]

- **Classifications and Standards** are closely related, but not identical.
  - A standard:
    - agreed-upon rules, multi-community (users) of practice; distant and heterogeneous collaborations; legal bodies, lack of laws (or criteria) of definition (e.g., politics)
  - Classifications:
    - capture work practice; are ubiquitous; represent information infrastructures; etc.; evolve.
Capturing Requirements Evolution...

- Example of Requirements (changes) classification
  - **Type of Requirement:**
    - Stable or Changing (Mutable, Emergent, Consequential, Adaptive, Migration)
  - **Type of change:**
    - Add, Delete Modify; Partial compliance; Traceability; etc.
  - **Change Rationale:**
    - requirements errors, conflicts and inconsistency; evolving customer system knowledge; technical, schedule or cost issues; changing customer priorities; environmental changes; organizational changes
Capturing Requirements Evolution...

- Requirements (changes) classifications allow the analysis of requirements features, in particular, requirements evolution.

Classified Requirements
Changes per Software Release

Histograms of the size of the sets of requirements changes allocated to a single software release.
Requirements (changes) classifications allow the analysis of requirements features, in particular, requirements evolution.

**HRMI - Historical Requirements Maturity Index**

\[
HRMI = \frac{R\hat{O} - AR_C}{R\hat{O}}
\]

**RMI - Requirements Maturity Index**

\[
RMI = \frac{R\hat{O} - R_C}{R\hat{O}}
\]
Stable and Changing Requirements - A functional viewpoint points out how diverse parts of requirements evolve. Moreover, it supports the analysis of requirements dependencies.

A Functional Viewpoint
Requirements Evolution Processes - A functional viewpoint points out how diverse parts of requirements evolve. Moreover, it supports the analysis of requirements dependencies.

Requirements dependencies

Requirements evolution processes
gamma maps from sequence analysis
Capturing Requirements Evolution...

- It is actually possible to identify two main distinct processes:
  - the software process
  - the requirements process

- Empirical evidence should mutually control the two processes

- A product viewpoint – identifying reusable software functions
  - Repeatable processes to allocate software functions to high level system requirements
  - Refining requirements - information flow expansion

- The gap between these processes represents the extent to which an organization is able to identify an optimal and effective set of (reusable) software functions

- Issues: communications across multilevel organizations, cost, visibility, controllability, etc.
The Humans...

- Viewpoints Analysis - different professional profiles (in terms of responsibility and experience) point out different understandings

- Viewpoint Analysis - Conflicting viewpoints highlight issues in requirements process
Capturing Requirements Evolution...

- **Requirements Evolution Practice**
  - **Data Collection**: difficult; integrated in work practice; otherwise, issues - increased workload, frustration, etc.
  - Tailored **data organizations and goals**

- **Requirements Evolution Features**
  - **Quantitative Requirements Evolution**
    - Requirements maturity (e.g., RMI, HRMI, etc.)
  - **Taxonomy** of requirements (changes)
  - Functional requirements evolution
    - Requirements evolution processes
  - Requirements evolution viewpoints
    - Require viewpoint management support
    - Identify requirements issues
  - Requirements evolution affects project visibility
Heterogeneous Engineering: “People had to be engineered, too – persuaded to suspend their doubts, induced to provide resources, trained and motivated to play their parts in a production process unprecedented in its demand. Successfully inventing the technology, turned out to be heterogeneous engineering, the engineering of social as well as the physical world.”

[MacKenzie, 1990]
Modeling Requirements Evolution...

- **Heterogeneous Engineering**
  - Provides a comprehensive account of system requirements
  - Stresses a holistic viewpoint that allows us to understand the underlying mechanisms of evolution of socio-technical systems

- Requirements, as mappings between socio-technical solutions and problems, represent an account of the history of socio-technical issues arising and being solved within industrials settings

- Requirements are **socially shaped** (i.e., constructed and negotiated) through sequences of mappings between solution and problem spaces.
  - The Functional Ecology Model [Bergman et al, 2002]
Modeling Requirements Evolution...

- The Functional Ecology Model describes solution-problem iterations
  
  Solution → Problem → Solution

- The formal (i.e., formal logic) extension of the functional ecology model provides a framework to model and capture requirements evolution
Example – A Reparable Clock Design

- The clock should be repairable – None state transaction is terminal.
Example – A Tick-tock Clock Design

The clock should Tick-tock
The formal framework captures **Requirements Evolution** in terms of solution space transformations.

Requirements evolution is a co-evolutionary process. Requirements evolution consists of the Requirements Specification Evolution and the Changes Evolution.

Requirements evolution represents **organizational knowledge**. It is part of organizational learning process.

Solution and problem spaces are socially shaped.

- **Issue**: the social-shaping of solution and problem spaces is error-prone
- Any solution requires the organization commitment in terms of resources
Other requirements evolution examples

- **Cascade effect** - Changes propagate through the mappings between solution and problem spaces
- Allocation of **Safety Requirements** in subsequent releases
- **Hybrid Solution Spaces** - System Models and Human Cognition models
Evolutionary dependency identifies how changes eventually propagate through emergent requirements dependencies

The formal extension of the functional ecology model captures requirements evolutionary dependencies (in terms of solution space transformation)

- Solutions are requirements dependencies
- The formal framework used at an higher level of abstraction than design
Capturing Evolutionary Dependency

- **F1** Software Architecture
- **F2** Interface Subsystems
- **F8** System Maintenance

Too expensive and risky!!!
A Taxonomy of Evolution...
...Evolution as Dependability
A Taxonomy of Evolution...

- It is possible to identify an evolutionary space for socio-technical systems
  - **Temporal Dimension**
    - “Evolution in Design”
    - “Evolution in Use”
  - **Physical Dimension**
    - “Hard Evolution”
    - “Soft Evolution”
A Taxonomy of Evolution...

- The identification of a broad spectrum of evolutions in socio-technical systems points out strong contingencies between evolution and dependability.

- The better our understanding of socio-technical evolution, the better system dependability.

**Diagram:**
- Soft Evolution
  - Organisation Evolution
  - Socio-technical System Evolution
  - Requirements Evolution
  - Architecture (Design) Evolution
  - Software Evolution
- Hard Evolution
  - Evolution in Design
  - Evolution in Use
Requirements Evolution...
Requirements Evolution...

- On Requirements Evolution...
  - Socio-technical grounds of Requirements Evolution

- Empirical analyses
  - Capturing Requirements Evolution...

- Observational Models of Requirements Evolution
  - A bit of theory...

- Evolution as Dependability
1. **Heterogeneous engineering** stresses a holistic viewpoint that allows us to understand the underlying mechanisms of evolution of socio-technical systems.

2. **Requirements**, as mappings between socio-technical solutions and problems, represent an account of the history of socio-technical issues arising and being solved within industrial settings.

3. The formal extension of the functional ecology model provides a framework to model and capture requirements evolution.

4. The application of the proposed framework provides further evidence that it is possible to capture and model evolutionary information about requirements (e.g., evolutionary dependency).

5. The discussion of scenarios of use stresses practical necessities for methodologies addressing requirements evolution.
6. The identification of a broad spectrum of evolutions in socio-technical systems points out strong contingencies between system evolution and dependability.

7. This presentation argues that the better our understanding of socio-technical evolution, the better system dependability.

8. This presentation is concerned with software requirements evolution in industrial settings. It introduces methodologies to empirically investigate and model requirements evolution, hence Observational Models of Requirements Evolution.

9. The results provide new insights in requirements engineering and identify a foundation for Requirements Evolution Engineering.

10. This presentation addresses the problem of empirically understanding and modelling requirements evolution.
Observational Models of Requirements Evolution - Theory and Practice

Massimo Felici
LFCS, School of Informatics, The University of Edinburgh
massimo.felici@ed.ac.uk
http://homepages.inf.ed.ac.uk/mfelici/

The Critical System Conference
29th-30th September 2004

© Massimo Felici 2004
Background Slides
On Requirements Evolution

- The socio-technical grounds of requirements (evolution)
  - Systems Approach: “...system builders include heterogeneous components, such as, mechanical, electrical, and organizational parts, in a single system.
    [Hughes and Hughes, 2000]
  - Heterogeneous Engineering: “People had to be engineered, too - persuaded to suspend their doubts, induced to provide resources, trained and motivated to play their parts in a production process unprecedented in its demand. Successfully inventing the technology, turned out to be heterogeneous engineering, the engineering of social as well as the physical world.”
    [MacKenzie, 1990]
  - Social Shaping of Technology: The mechanisms underlying the social design and implementation of technology systems are referred to as the Social Shaping of Technology (SST).
    [MacKenzie, 1999]
On Requirements Evolution

- Requirements Evolution vs. Reuse and Maintenance
  - Current requirements engineering methodologies provide limited support to understand requirements evolution. Reuse and maintenance tend to focus on classifying and managing requirements changes, rather than analyzing or even anticipating changes.

  [Lutz and Mikulski, 2003]

  - Requirements evolution involves the practice on analyzing requirements changes in order to identify strategies dealing with environmental changes

- It is possible to classify requirements as stable or changing

  [PROTEUS Project]

  - It is possible to identify the stable or volatile requirements

- Empirical analyses of requirements evolution are still patchy
  - Lack of support for analyzing evolutionary data
  - Requirements evolution (analysis) as work practice
  - Organizational culture
On Requirements Evolution

- **Modeling** methodologies and languages (e.g., SCR, Intent Specifications, etc.) advocate different design strategies. Although these strategies support different aspects of (software) system development, they originate in a common **System Approach**.

- **Systems Approach**: “Practitioners and proponents embrace a holistic vision. They focus on the interconnections among subsystems and components, taking special note of the interfaces among various parts.”

  [Hughes and Hughes, 2000]

- **Requirements change**: “About four year later the original TCAS specification was written, experts discovered that it did not cover requirements involving the case where the pilot of an intruder aircraft does not follow his or her TCAS advisory and thus TCAS must change the advisory to its own pilot. This change in basic requirements caused extensive changes in the TCAS design, some of which introduced additional subtle problems and errors that took years to discover and rectify.”

  [Leveson, 2000]