Evolution within the Interdisciplinary Research Collaboration on Dependability of Computer-based Systems

Massimo Felici
LFCS, School of Informatics
The University of Edinburgh
UK
massimo.felici@ed.ac.uk

Overview
- Design for Dependability (PA)
  - Issues under investigation about Design
  - Work overview
  - Questions raised
- Evolution... of Computer-based Systems
  - Taxonomy of Evolution
  - Dependable Evolution
  - Requirements Evolution
  - Quantitative aspects
  - Modelling evolutionary information about requirements

Design for Dependability
- Aim & Work Items
  - To develop design techniques for dependability of distributed, heterogeneous, computer-based systems in complex organisational settings.
    - Characterizing dependability
    - Evolution in complex organizations
    - Structures, processes, components
    - Evidence centred design

What you will not see...
- A tutorial about Design
- A detailed presentation of work in Design for Dependability
- A list of references

Characterizing Dependability
- Based on empirical work (various case studies)
- Investigating sources of undependability
- Using patterns to capture beneficial forms of cooperation from empirical data

DIRC – www.dirc.org.uk
- Interdisciplinary Research Collaboration on Dependability of Computer-based Systems (DIRC)
- 6-year project funded by the UK EPSRC
- 5 UK universities involved + industry
- various Project Activities (e.g., Dependable Human Machine Interaction in Real-time Systems, Design for Dependability, Impact of Organizational Culture and Trust on Dependability, etc.)
- Research Themes: Structure, Diversity, Timeliness, Responsibility, Risk
System Failures

Issues in Healthcare system failures

- What is meant by a medical device?
  - anything that is used in a medical situation (HW + SW)
- Different types of system failures to characterise:
  - non delivery of expected service
  - incorrect delivery of the expected service
  - delivery of incorrect service
- Different types of healthcare system considered:
  - resource allocation systems
  - medical devices
  - information systems

System Failures

Healthcare system failures

- Recommendations
  - Electronic and paper versions should be designed to co-exist
  - Data must be conceived with understanding of the situation of use
- Further Work
  - Addressing the problem of a lack of situation information available to engineers
  - Including situation information into the design process
  - Situation modelling

Patterns of Cooperative Interaction for Dependability

Patterns of cooperative work

Informing the design of socio-technical systems

Ecological arrangement

Spatial characteristics

Coordination techniques

How is distributed cooperation achieved?

Community of use

Activity

Fieldwork

Representation of activity

e.g., plans or technology?

Cooperative arrangement

Actors and resources

Patterns of Cooperative Interaction for Dependability

Further work

- Generating and validating more patterns
- Handling large amounts of patterns: generic descriptions, indexing, etc.
- Structures & taxonomies of patterns
- Patterns for dependability
  - Healthcare, control room studies
  - Configurations that work can illustrate good practice

Evolution in Complex Organizations

- Change is a fertile source of undependability
- Evolution seems highly domain specific
- Empirical work based on case studies
  - Avionics
  - Smart card
- Focus on Requirements Evolution
Requirements Evolution

- Process viewpoint
  - A dependable process supporting Requirements Evolution
- Product viewpoint
  - Requirements Evolution addressing system dependability

Modelling Requirements Evolution

- Industrial data:
  - Often collected for certification
  - May not provide suitable feedback to processes
- A modelling framework may help in:
  - Organizing data collection
  - Providing feedback to processes

- Many models! Integration needs to be a dependable process
- Framework for requirements evolution
- A graphical model representing requirements evolution:
  - Easy to understand
  - Easy to analyse
  - Reasoning on requirements evolution
  - Identifying requirements features (e.g., changeable or stable)

Requirements Evolution

- Issues with evolution
  - Incomplete Industrial data
  - Evidence for Dependability
  - Evolutionary Management: Process Oriented
- Evolutionary questions
  - Is Evolution too Complex?
  - How do we characterize Evolution?
  - Where is Evolutionary Information?
  - From Process to Product oriented Evolutionary Management?

Structures, Processes and Components

- Elements of design:
  - Survey of the design process
  - Configuration - Evolution
  - Representations
  - Reflection
- Other areas:
  - Compositionality / Composability

Design Recommendations

- Ideas from inside & outside the computer field
- System designers
  - Investigate industrial/organizational accidents reports
  - Use formal/graphical representations for fault tracking
  - Identify vulnerable parts/paths in systems
  - Formulate recommendations

- Examples of provisional recommendations:
  - Make small incremental steps
  - Integrate several organizational levels
  - Be aware of implications of ad-hoc reuse
  - Add redundancy/diversity
- But let’s keep in mind that...
  - The system must remain testable
  - You must be able to assess its reliability, availability, etc.
  - The system is likely to evolve
**Design Recommendations**

Further work

- Continue the investigation of case studies
- Find an integrative graphical representation
- Provide generic recommendations for design

**Configuration**

- Issues:
  - scale, complexity, enabling social learning, flexibility, cost, etc.
- Drivers:
  - Technical
    - Building the configuration system for EU DataGrid
  - Intervention in an ERP system in production planning
  - Access control design for a medical records system (conflicting dependability requirements)
  - Configuring a hospital information system (common system components, diverse user environments)

**Configuring PiMS: Classification in Healthcare Systems** (Very Preliminary)

**Outline**

- Classification (in health systems)
- Classification in PiMS
- Issues for PiMS work
- Classification as configuration
- Configuring for classification
- Hazards in classifying/of classification
- Supporting classification dependably
- Supporting dependable classification

**Classifications**

- Are
  - consistent, exclusive categories, complete.
  - Are
    - boundary objects
    - negotiated across social/work groups
  - Offer comparability across settings
  - Structure visibility (e.g., of activity)
  - Provide control over outcome

**Infrastructure**

- Classifications form part of an Information Infrastructure:
  - Embedded
  - Transparent
  - Reaches across contexts
  - Learned as part of group membership
  - Linked to conventions of practice
  - Multifunctional
Example

Iowa Intervention Project

- Created the Nursing Interventions Classification (NIC)
- Approx 300 categories:
  - Bleeding reduction - nasal
  - Humor
  - Hope Installation
  - Airway management
  - Spiritual Support
  - Cultural Brokerage
  - ...

NIC

- Structure
  - Name, Definition, Activities, Background Reading
  - Forgets earlier nursing practice
  - Nursing practice
    - clinical decision process + knowledge classification
  - Restructures education
  - NIC drives
    - clinical decision support, nurses work, hospital accounting
  - Nursing research stabilizes NIC

PiMS Classification

- Contact Purposes is a key classification
  - clearly problematic
  - completeness issues
  - structural issues
- Boundary object
  - Trust IS people
  - PiMS implementation team
  - Clinical Governance
  - National IS people
  - Clinicians
  - ...Other Agencies

PiMS Classification

- Comparability is a very big issue in the implementation
  - compare South Lanarkshire
  - Visibility issues about how the system makes the clinicians work visible
    - Difficulties:
      - patient centred/work centred
      - multifaceted work (taking blood)
  - Control
    - multifaceted control of patients
    - control of resources

PiMS Infrastructure

- PiMS classification may form part of an emergent infrastructure:
  - Embedded:
    - not yet, experience is quite patchy, differs greatly from context to context
  - Transparent:
    - no, discussion around aids to transparency
  - Reaches across contexts:
    - this is imposed and poses problems in managing across contexts and how the system manages it

PiMS Infrastructure

- Learned as part of group membership:
  - issues on how to get this to work, much more work needed to create the necessary conventions
- Linked to conventions of practice:
  - hope to get treatment planning tools into use, evidence based practice, Nurse Practitioners the centre of this
- Multifunctional:
  - yes but issues around agreement in the overlap
PiMS Concerns

- Aligning the interpretation across groups
- Is the system patient centred or work centred?
- Use both to allocate resource and as a basis for evidence-based medicine.
- Multi-faceted nature of work
  - Taking blood is a typical multi-faceted activity.
- Data centred vs process centred classification?

PiMS Issues

- Great deal of “misclassification”
- How to refine the classification to fit the needs of the users
- How to present it appropriately to users
- Agreement across contexts is difficult (both treatment contexts and IS departments)
- Dealing with multi-agency work
- In the NHS context, “evolution” is a big issue

Classification as Configuration

- Should the classification be part of the system configuration?
- Can we build a global classification from local classifications?
- How to represent multi-facted activities properly?
- How to fit classification to multiple purposes (resource allocation, evidence based medicine)?

Configuring for Classification

- PiMS is a socio-technical configuration
- Discussion on how to configure to accommodate an accurate classification
  - Combine context with category
  - Fits with system and the “affordance” of the menu system given training
  - Users understand the menus
  - Grouping and using context prefixes might do it
  - May afford communication across contexts

Hazards

- Only consider classification-related hazards
- Misdistribution of resources due to reporting biases (e.g., inability to handle multi-agency work)
- Incorrect treatment where classification is used for evidence-based treatment
- Misinterpretation across contexts?

Supporting Classification

- Are there ways to support the creation of classifications:
  - Around PiMS?
  - In a “generic” way?
- Are there generic ways of supporting the representation and deployment of agreed classifications in systems?
Dependable Classification

- Do we have ways of assessing goodness of a classification for its purpose?
- How do we evolve classifications:
  - How do we detect the need for evolution?
  - How do we implement change?

Conclusions

- Classification seems to play an important role in PiMS (and HIS and Mammography?)
- Classifications affect dependability
- There are detailed socio-technical questions
- Classifications strongly influence the acceptability of these systems

Credits

- Stuart Anderson, Gillian Hardstone, Rob Procter, Robin Williams
- Leigh Star and Geoff Bowker, “Sorting Things Out”.

Representations

- Key element in communicating between stakeholders
- Highly domain dependent
- Diagrams have limited expressiveness
  - Useful in communicating with users
  - Constraining design process, capturing domain assumptions
  - The basis for tools
- Empirical work on the use of representations
- Design guidelines for representations
- Elimination of some categories of communication errors

Reflection

- Most Human-Computer systems include incomplete self-models of some kind
- These are useful in adding flexibility and ease of description
- They have hazards:
  - Theoretically:
    - difficulties in reasoning about systems
  - Empirically:
    - can make systems unstable and hard to analyse / predict
- Empirical work on financial systems:
  - Long-Term Capital Management
  - Connections to other areas, e.g., Security

Design Driven by Dependability Modelling

- Basis:
  - probabilistic modelling of dependability
  - drive design decisions by evaluating their consequences
  - the fault tolerance viewpoint
  - design compromises that deliver:
    - good dependability
    - ways of justifying claims for good dependability
  - a few examples of interesting questions ...
Issues of “Perfection”

Example 1

- an important category of statements about dependability:
  - “this product has no defects [of this category]”
- open practical question:
  - integrate Boolean formal verification with probabilistic reasoning claims via simplicity of design
  - hence easier design/verification?
  - meaningful in terms of probability of no defects with a meaning in terms of probability of no failures
- e.g. 2-channel system with diverse claims for the two channels
  - probability of perfection
  - probability of failure per demand

Diversity

Example 2

- background:
  - modelling reliability of systems built with diverse redundancy
  - applicable to designing systems or processes
  - trade-offs between
    - seeking reliability of individual channel/stage
    - and seeking diversity between them
  - new applications in DIRC
    - human-machine systems
      - e.g., advisory computing systems + users
    - combination of methods in development stages
    - diversity of arguments in supporting decisions
      - e.g., safety case

Conclusions and Further work

- Design for Dependability
  - Multidisciplinary
  - Open theoretical and practical questions
  - Based on different case studies
- Further Work
  - Need to focus the results
  - Identify practical tools and guidelines

A Taxonomy of Evolution Dependability Perspectives

Massimo Felici

Overview

- Evolutionary Dependability
- Evolutionary Layering
- Evolutionary Modelling
- Dependable Evolution
- Taxonomy of Evolution
  - Practical issues
  - Practical Challenges
  - Evolution within DIRC
  - Evolutionary Work

Evolutionary Dependability?

- Laprie’s definition: “Dependability is that property of a computer system such that reliance can justifiably be placed on the service it delivers”
- Issue: Static definition of Dependability
- Issue: A computer system can just be to a certain extent Dependable or Undependable
**Dependability Basis**

- In the Dependability tree (Laprie’s definition)
  - Dependability consists of a set of attributes
  - Dependability means:
    - fault prevention, tolerance, removal and forecasting
  - Dependability impairments:
    - faults, errors and failures
  - Fenton, Pflegger and others use an equivalent reshuffled model based on errors, faults and failures

---

**Dependability Basis**

...faults...errors...failures...

---

**Evolutionary Layering**

- Software Evolution
  - M. Lehman defines E-type programs (software systems), which
    - continuously evolve in order to be satisfactory
    - need to accommodate environmental changes
    - have an increasing complexity
    - represent multi-level, multi-loop and multi-agent feedback systems

---

**Architecture Evolution**

- Architecture is (should be) a stable part of a system (Anderson and Felici, Safecomp 2000)
  - Issue: expensive to change
  - Issue: imply high risk
  - Product-line architecture represents the extent to which a product-line will be able to evolve
  - Trade-off: General vs. Specific
  - In practice: Identified variability points
  - Definition: unclear definition of architecture evolution. There are in general two (three) types of evolution:
    - the architecture evolves or components evolve (everything evolve).
Requirements Evolution

- It is impossible to freeze requirements. But we should (and could) identify the extent to which requirements evolve in order to identify the stable ones and the most likely to change. (Anderson and Felici, Safecomp 2000, Profes 2001, COMPSAC 2002)
- Requirements evolution has been considered mainly a management problem
  - little emphasis on product features
  - Research Hypothesis: Product features enhance our ability in understanding (requirements) evolution.
- Requirements evolution is unstructured
  - Research Hypothesis: There are structures for evolution (There are structures in Requirements, Changes and Evolution)

Computer-based System Evolution

- Social Learning:
  - Domestication and Innofusion
  - Emergent behaviour
- Distributed Cognition:
  - Human Cognition as result of interactions (knowledge distribution) between (among) individuals and artefacts
- Disappearing Computer (D. Norman):
  - The system (design) becomes less intrusive and evolves according to human needs

Organization Evolution

- (System) Evolution and (Organization) Co-evolution or vice versa
- System-Organization reflection (Conway's Law)
- Evolution as Corporate Knowledge

Evolutionary Modelling 1/2

- Software Evolution
  - E-type systems
- Code modification
- Architecture Evolution
- Product-line variability
- Requirements Evolution
  - Goal-structures framework based on first-order logic with Prolog resolution (Proteus Project)
- First-order logic model of Requirements Evolution (Zowghi)
- Traceability
- Management process
- Quantitative approaches:
  - (few) Metrics; (little) Probabilistic Distribution

Evolutionary Modelling 2/2

- Computer-based System Evolution
  - Human Evolution
    - Social Learning
    - Emergent Behaviour
- Organization Evolution
  - Business models ??? ;-)
  - Software Engineering Economics (e.g., COCOMO II Boehm)
Dependability Related Modelling
- Reliability Growth models
- Bayesian (probabilistic) models
- Fault Tree Analysis
- Event Tree Analysis
- Domino effect
- Cheese model
- ??? :-(

A Taxonomy of Evolution
- Practical Issues
  - There are many different assumptions about evolution embedded into methodologies
  - Unclear definition of evolution
  - Little coordination giving rise to undependability in place
  - Evolutionary data are difficult to capture and analyse
    - Incomplete, distributed, unrelated, unclear...

A Taxonomy of Evolution
- Practical Challenges
  - Classify environments according to evolutions in place
  - Model evolution
  - Link evolutionary layers
  - Link evolutionary models

Evolution
- faults...errors...failures...

Faults
Errors
Failures

Dependable Evolution
- faults...errors...failures...

Evolution

Ongoing Evolutionary Work
- Taxonomy of Evolution (Felici, ongoing)
- Requirements Evolution (Anderson and Felici, 2000, 2001)
  - Empirical Framework (Anderson and Felici, 2001)
  - Modelling and Representation (Felici, 2002)
  - Metrics (Anderson and Felici, 2002)
- Modelling and Design (Felici and Filipe, 2001)
- Evolutionary Diversity
  - can we have diversity by different evolutions over different releases? (Felici, ongoing)
Quantitative Aspects of Requirements Evolution

Massimo Felici

Informatics

Overview
- Requirements Evolution
- Rationale
- Measuring Requirements Evolution
  - First Experience (by means of case study)
  - Requirements Evolution
  - What can we say more…?
    - Modelling Requirements Evolution
  - Measuring Requirements Evolution
    - Second Experience (by means of case study)
- Conclusions & Further Work

Requirements Evolution Rationale
- Requirements Evolution
- Affects:
  - Cost (both in term of money and man-power)
  - Project risk
  - System dependability (and quality of service, system quality, etc.)
  - Process effectiveness
- So far… little investigated and supported
- Current practice: reactive (passive); non-proactive (e.g., traceability)
- Differently:
  - Requirements Evolution as system feature

How to investigate Requirements Evolution?
- Empirical Analysis I
- Case Study
- Empirical Analysis II
- Modelling

An Avionics Safety-Critical Case Study
- Safety Requirements
- Certification and Maintenance
- Functional and Operational Requirements
- Product Line Aspects and Standards
- Software Development Process
- Case Study
Measuring Requirements Evolution I

...so

Requirements evolve... over subsequent releases

There is a majority of added (new) requirements among changes

---

Measuring Requirements Evolution I

Maturity

...hence the RMI:

- is too sensitive to changes introduced in a single release
- does not take into account historical information about changes (e.g., age)

---

Measuring Requirements Evolution I

Functional Viewpoint

- The functional requirements have different evolutions...
  ...this may be due to requirements' dependencies
- Intuitive notion of stability based on the proportion between changes and requirements

---

Requirements Evolution Modelling

Taxonomy of Requirements Evolution

Add, Delete and Rename parameters / variables
Add, Delete and Modify Requirements
Range Modification
Hardware Modification
Hardware Modification

Requirements Evolution

Explanation
Rephrasing
Traceability
Non-compliance
Partial Compliance

---

Requirements Evolution Modelling

Structuring Requirements Evolution

F1 - Software Architecture
Add[n]
Del[n]
Mod[n]
Taking into account historical information

Being less sensitive to single release changes

Requirements Evolution Modelling

Measuring Requirements Evolution

Average Number of Requirements Changes

Cumulative Number of Requirements

Number of Software Releases

Total Number of Requirements

AR_C = \frac{CR_C}{n}

RSI = \frac{R_T - CR_C}{R_T}

HRMI = \frac{R_T - AR_C}{R_T}

Requirements Stability Index

Historical Requirements Maturity Index

Empirical Analysis I

Case Study

Empirical Analysis II

Modelling

Conclusions & Further Work...

o Conclusions
  o Experience in measuring Requirements Evolution
  o Models for Requirements Evolution
  o Empirical validation of the models
  o Drivers for Requirements Evolution (e.g., changes distribution)

o Further Work
  o Replicate the experiment on different case studies
  o Identify further drivers for requirements evolution
  o Devise a framework modelling requirements evolution supporting a pro-active approach to requirements evolution
Evolution within the Interdisciplinary Research Collaboration on Dependability of Computer-based Systems

Massimo Felici
massimo.felici@ed.ac.uk