

*Interdisciplinary Research Collaboration in Dependability of Computer-Based Systems*

<http://www.dirc.org.uk>



# Design for Dependability

**Massimo Felici**

*LFCS, Division of Informatics, The University of Edinburgh*

*19 July 2001, ITC-IRST/ARS, Trento, Italy*



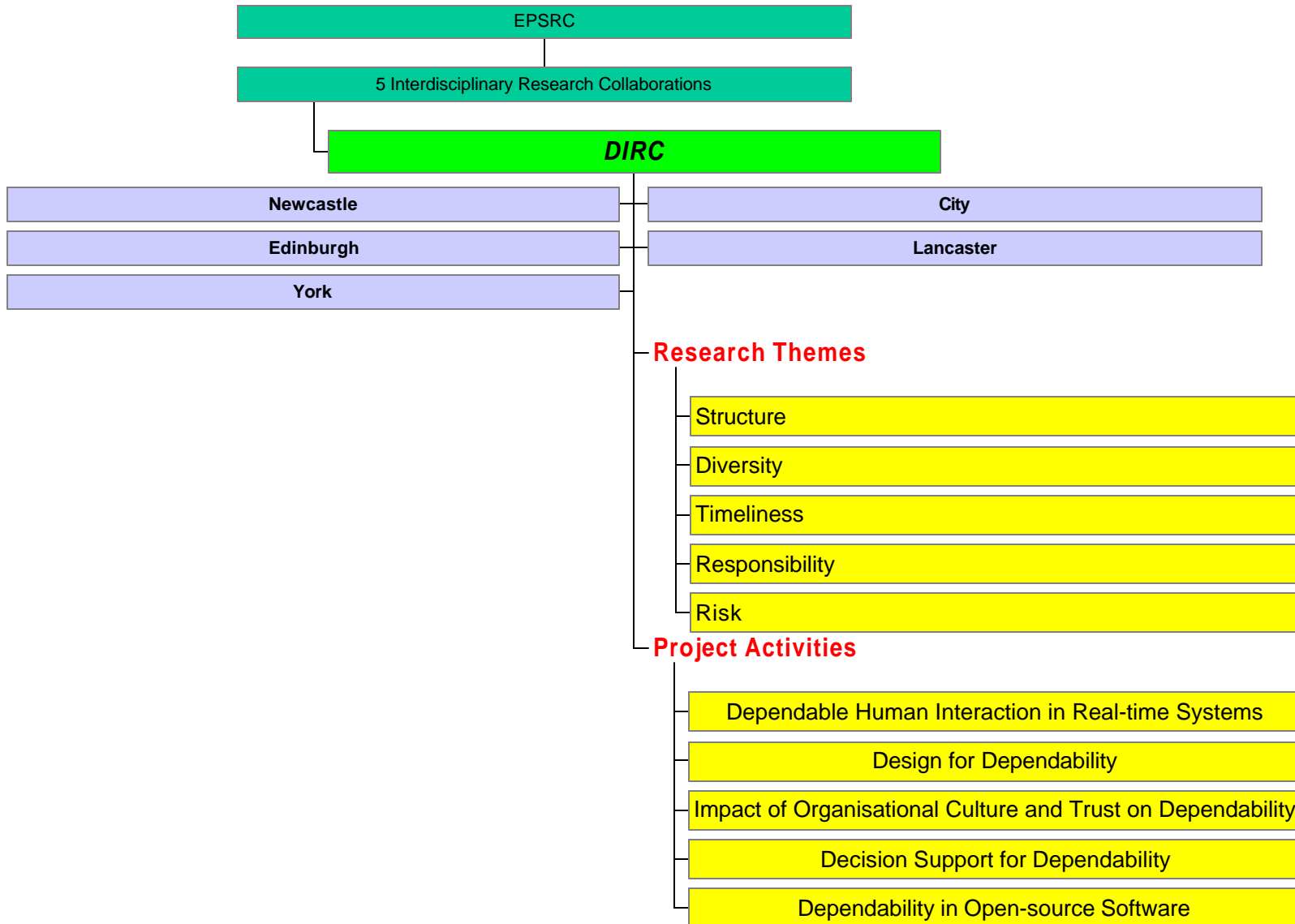
THE UNIVERSITY *of* York

*Massimo Felici*

# Interdisciplinary Research Collaboration in Dependability of Computer-Based Systems



<http://www.dirc.org.uk>



*Massimo Felici*

# Overview: What you will see...



*An invitation to discuss Design for complex organisational settings*

- ❖ ***Design for Dependability***: Aim & Work Items
- ❖ Competence of the team
- ❖ Core issues under investigation about ***Design***
- ❖ Overview of the work being carried out
- ❖ Questions raised

# What you will not see...

- o A tutorial about Design
- o A detailed presentation of work in *Design for Dependability*
- o A list of references

# Design for Dependability

## Aim & Work Items



To develop *design techniques* for *dependability* of *distributed, heterogeneous, computer-based* systems in *complex organisational* settings.

- ❖ Characterising dependability
- ❖ Evolution in complex organisations
- ❖ Structures, processes, components
- ❖ Evidence centred design

# Team Competence: 9 Researchers



Gillian Hardstone  
Massimo Felici  
Corin Gurr  
Alexander Voss

*Complex organisations*  
*Requirements Engineering & Software Metrics*  
*Cognitive and Semantic aspects of Representations*  
*Social Learning and systems development*

Denis Besnard

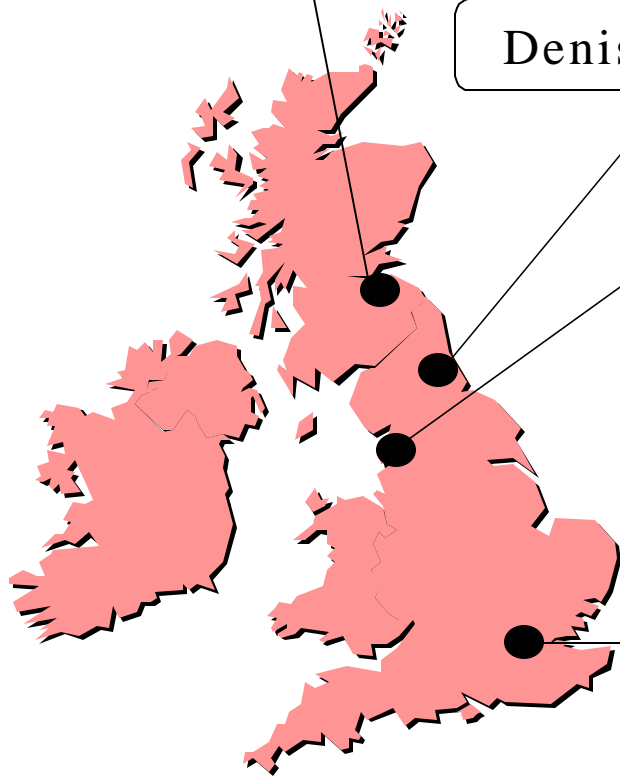
*Cognitive Ergonomics*

Jo Mackie  
David Martin  
Stephen Viller

*IT technologies*  
*Ethnography*  
*Requirements Engineering*

Andrey Povyakalo

*Formal safety analysis in  
dynamic systems*



*Massimo Felici*

# Characterising Dependability



- ❖ Based on **empirical** work (healthcare focus)
- ❖ Investigating sources of **undependability**
- ❖ Using **patterns** to capture beneficial forms of cooperation from empirical data

# Healthcare System Failures



*Three issues identified in the study of system failures*

*Designers aware of factors not included  
in requirements documentation*

1. What is meant by a **medical device**?
    - anything that is used in a medical situation (HW + SW)
  2. Different types of **system failures** to characterise:
    - non delivery of expected service
    - incorrect delivery of the expected service
    - delivery of incorrect service
  3. Different types of **healthcare system** considered:
    - resource allocation systems
    - medical devices
    - information systems
- Cases studied so far:
- London ambulances system failure
  - Therac 25
  - Various medical records

# Healthcare System Failures



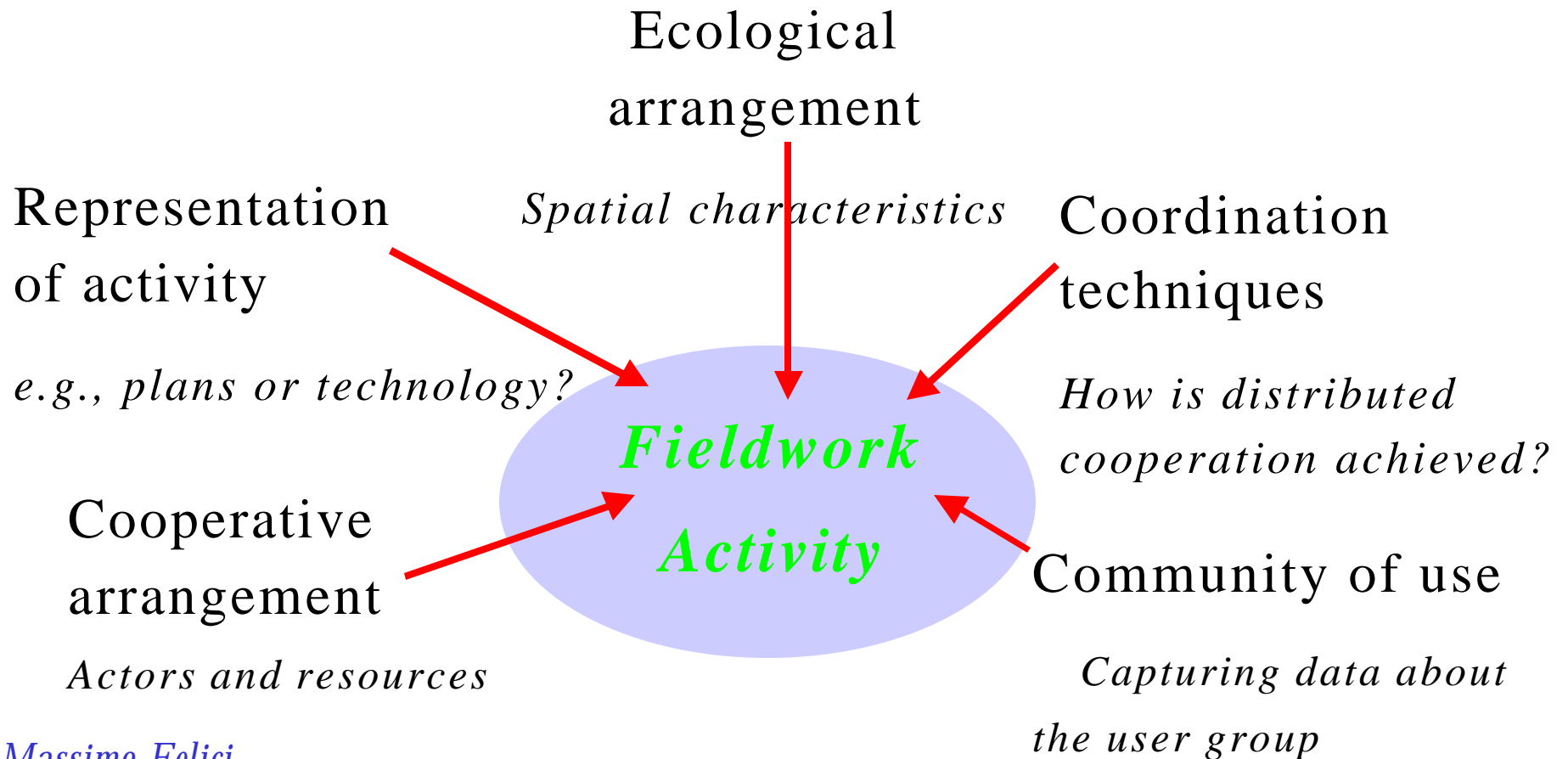
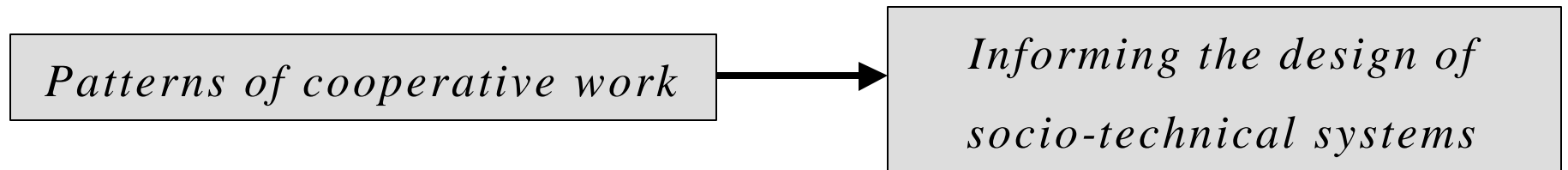
## *Recommendations*

- ❖ 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 Interaction for Dependability



*Patterns encapsulate salient features of work*

- ❖ (Re)examination of (previous) studies
- ❖ Patterns as *grossly observable* features
- ❖ *Descriptive* rather than design patterns
- ❖ **Within and across** studies
- ❖ Moving **from empirical studies to general design resource**

# 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 Organisations

- ❖ *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



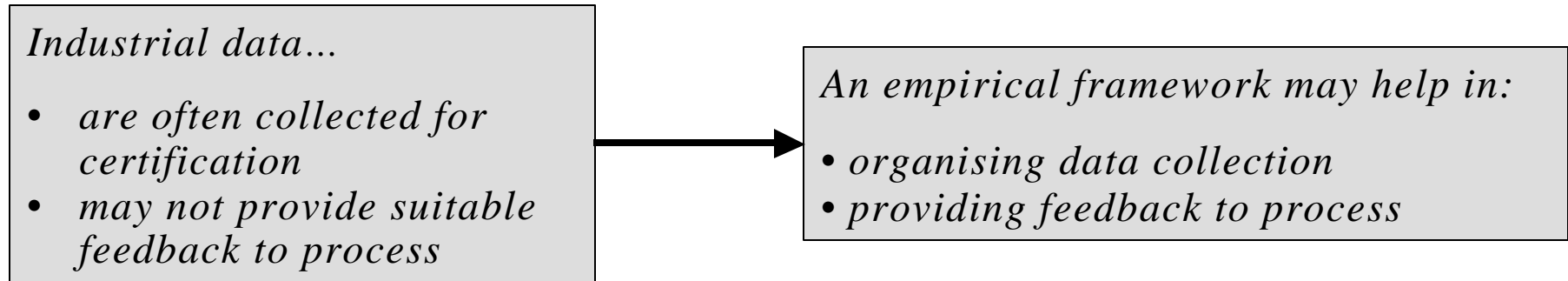
*Design for Dependability* + *Requirements Evolution* = *Dependable Requirements Evolution*

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

# Requirements Evolution



*Empirical framework for requirements evolution*



*Many models! Integration needs to be a dependable process*

## *Formal framework for requirements evolution*

A graphical model representing requirements evolution:

- ❖ Easy to understand
- ❖ Easy to analyse
- ❖ Permitting 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 characterise?
- ❖ Where is Evolutionary Information?
- ❖ From Process to Product Evolutionary Management?

# Structure, Processes and Components

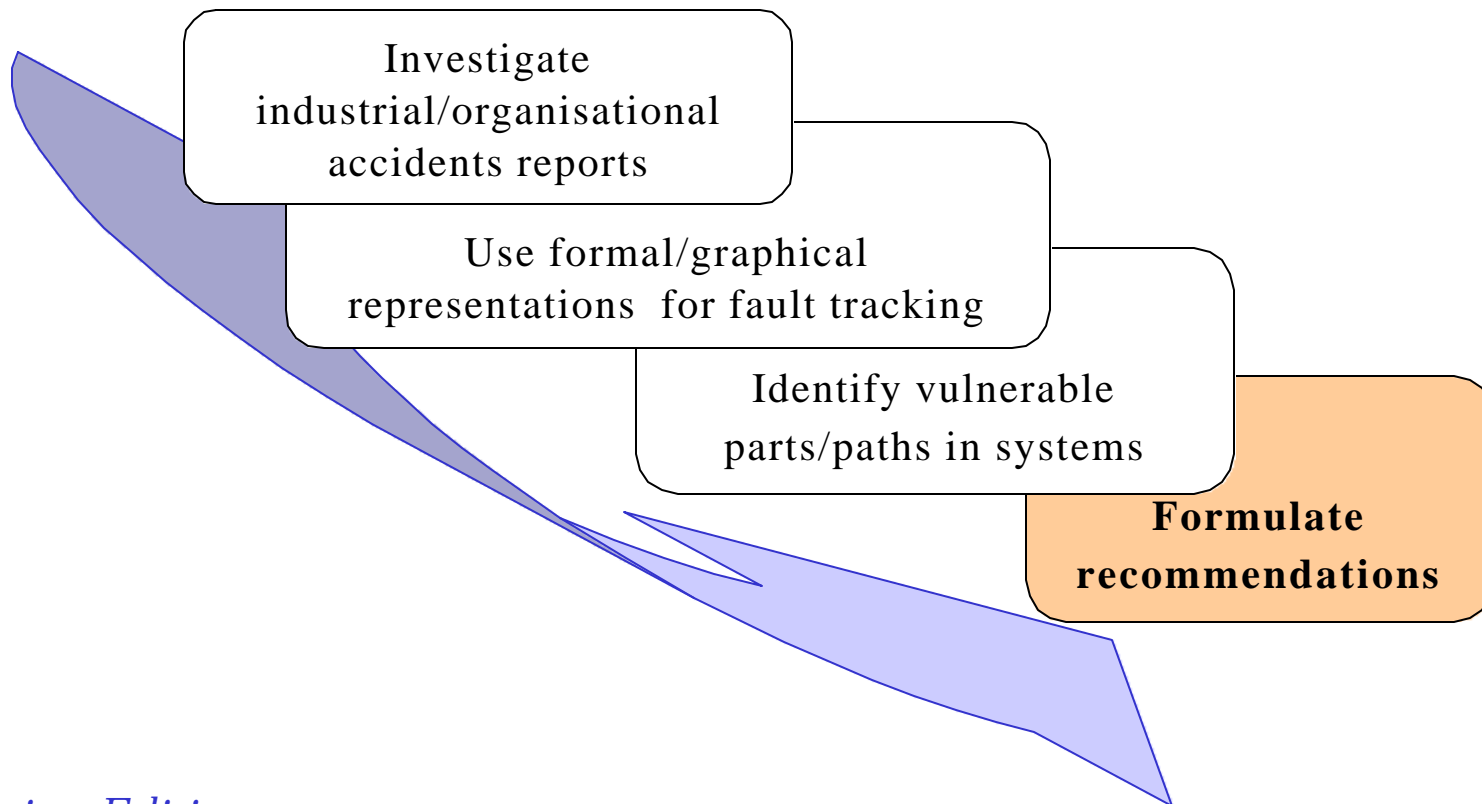
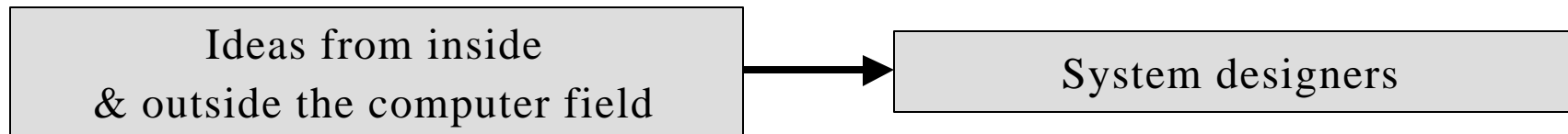


Elements of design:

- ❖ Survey of the **design process**
- ❖ **Configuration - Evolution**
- ❖ **Representations**
- ❖ **Reflection**
- ❖ Other areas: **compositionality/composability**

# Case Studies Survey for Design Recommendations

*Informing the design process*



# Case Studies Survey for Design 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**

# Case Studies Survey for Design Recommendations



## *Further work*

- ❖ Continue the investigation of case studies
- ❖ Find an integrative graphical representation (e.g., AND/OR gates, time, hierarchical levels)
- ❖ 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)

# 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 and
  - ways of justifying claims for good dependability
- ❖ a few examples of interesting questions ...

# Example 1: Issues of “Perfection”



- ❖ 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*

# Example 2: Diversity



- ❖ **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*

*Thank  
you!*

# **SAFECOMP 2002**

**The 21<sup>st</sup> International Conference on  
Computer Safety, Reliability and Security  
10-13 September 2002, Catania, Italy  
<http://www.safecomp.org>**