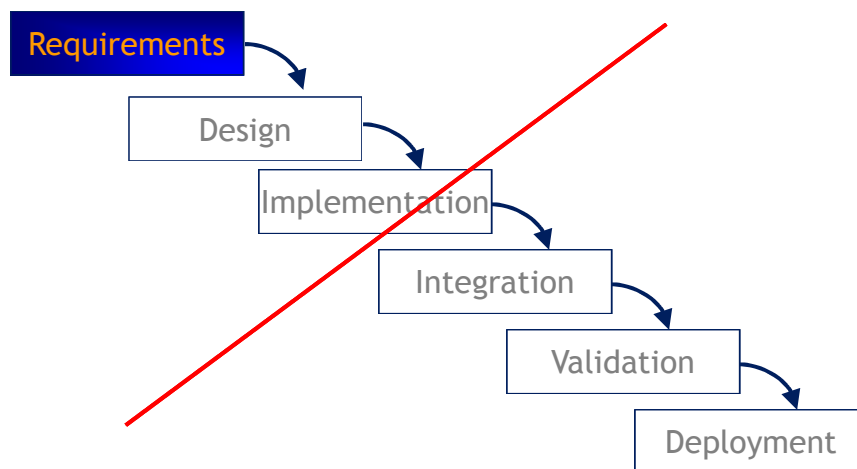


Requirements and Architecture during System Evolution

Emmanuel Letier
Department of Computer Science
University College London
<http://www.cs.ucl.ac.uk/people/E.Letier>

1

Requirements engineering is *not a stage of development*



2

Requirements Engineering

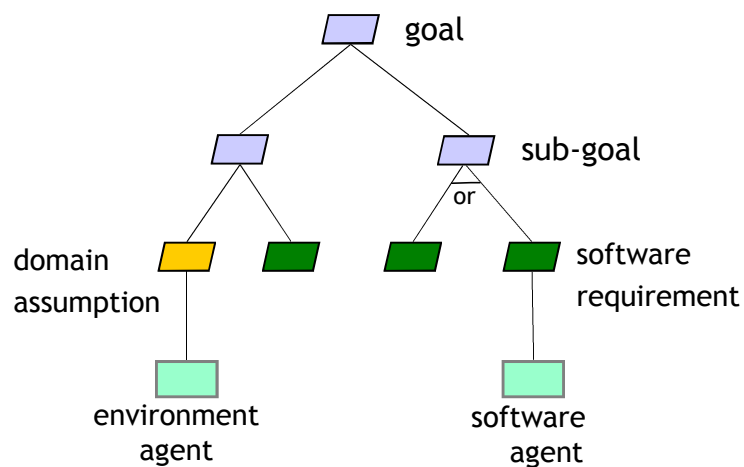
Requirements Engineering (RE) is the branch of software engineering concerned with the **real-world goals** for, functions of, and **constraints** on software systems.

It is also concerned with the relationship of these factors to **precise specifications** of software behaviour, and to their **evolution** over time and across software families.

(Pamela Zave, 1995)

3

Goal-Oriented Requirements Models (E.g. KAOS)



4

System Architecture

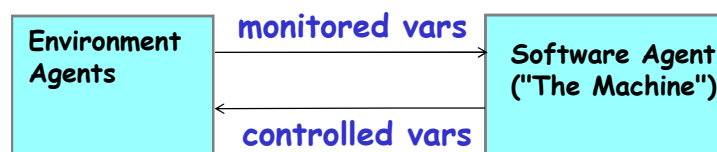
A **system architecture** is a description of a system in terms of its components, connectors and their configurations.

Composite system: human + devices + software

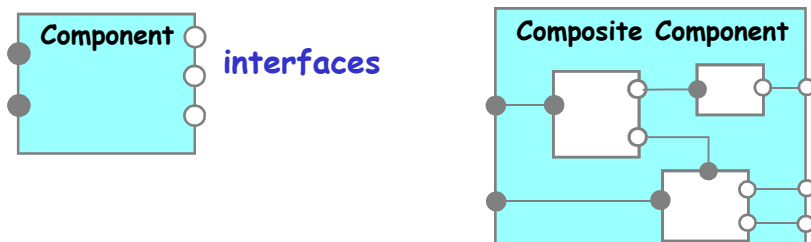
5

System Architecture Modelling

System context diagram

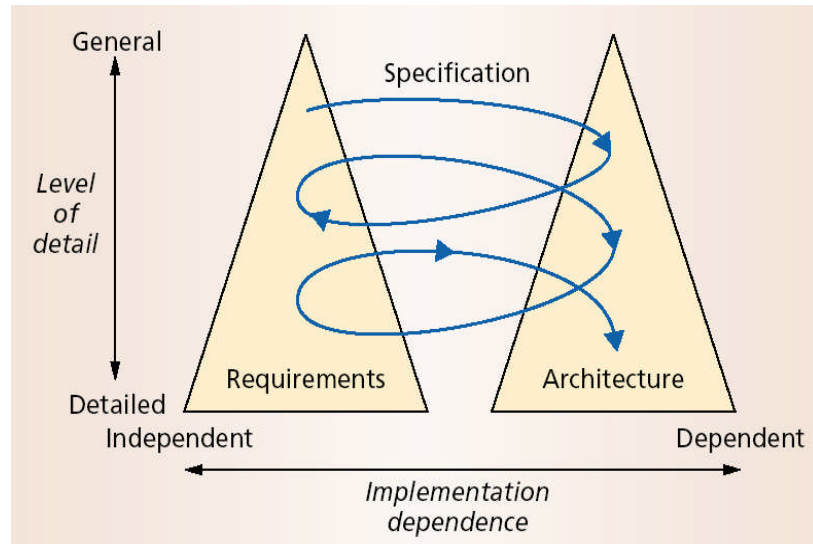


Software architecture model (E.g. DARWIN)



6

Intertwined Elaboration



(Swartout & Balzer 1982, Ward & Mellor 1985, Nuseibeh 2001)

7

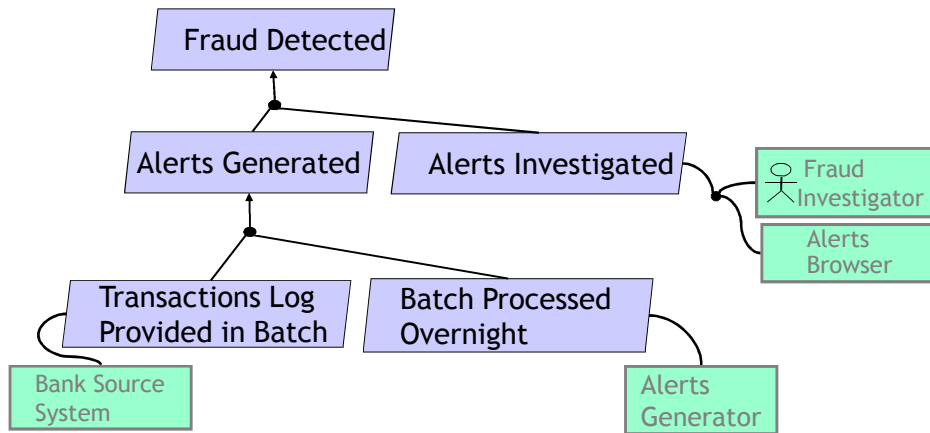
Example - A financial fraud detection system

(based on Leticia Duboc's Phd thesis)

- purpose: detect when plastic card accounts have been compromised by fraudsters, who are using account to steal funds
- software performs analysis of transactional data to identify unusual and possibly fraudulent patterns of card usage
- important **scalability** concerns due to growing number of accounts, transactions, and frauds

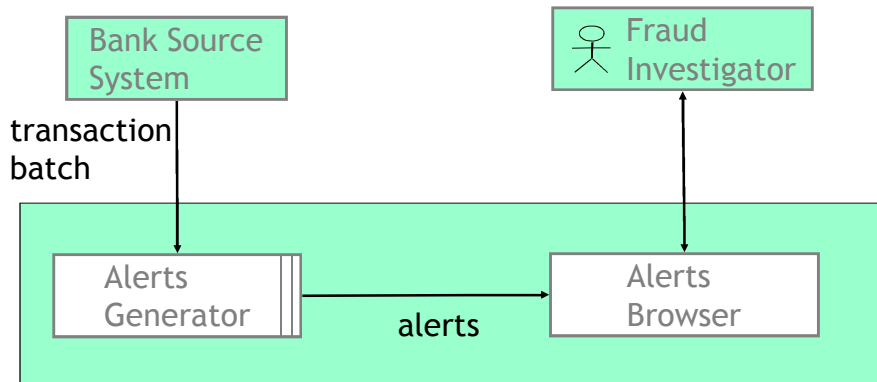
8

first sketch for goal model



9

related architecture model

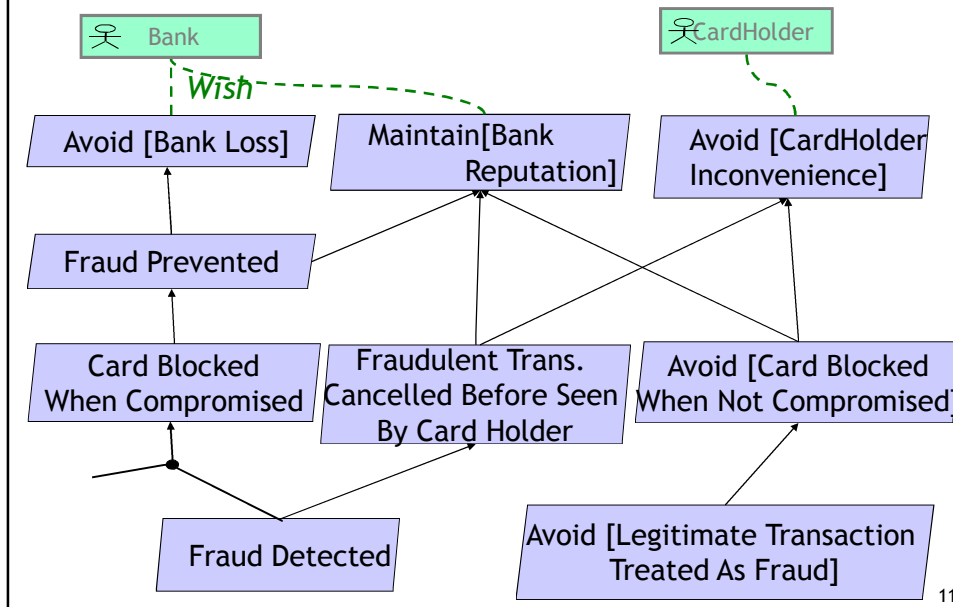


- Requirements specification and architecture decisions are linked:
 - If we assume a different architecture (e.g real-time processing), we will elicit and specify different goals

10

The same is true for higher-level goals

(e.g. goals would be different if we didn't have card payments at all)



11

Observations

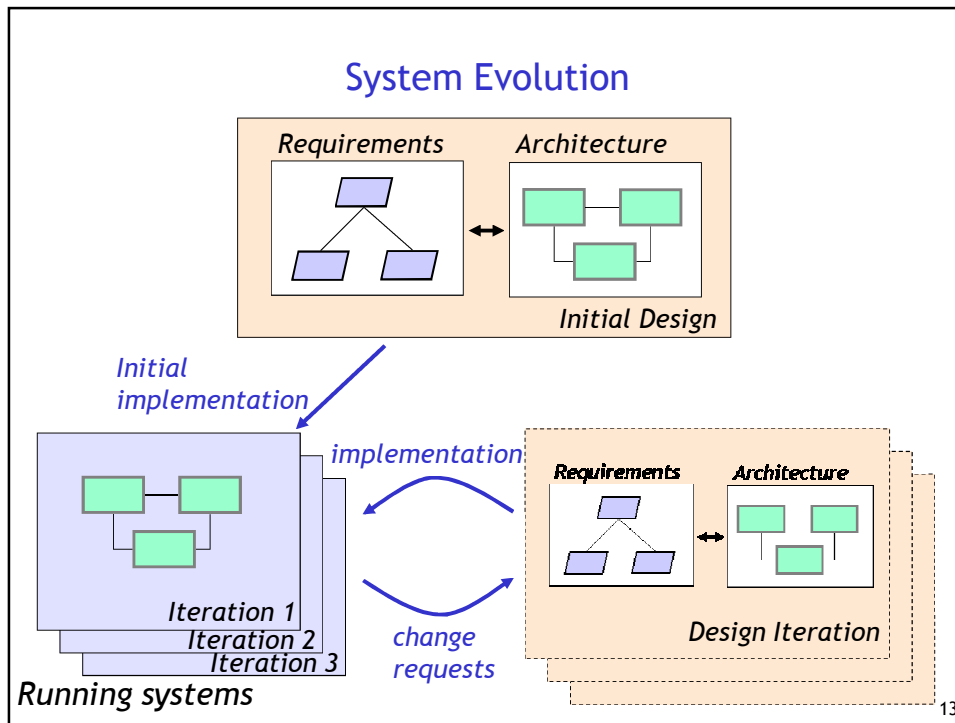
- The specifications of goals (the problem) and architecture (the solution) are intertwined. One is not derived from the other.
- The process of specifying goals and architecture is iterative



Image: Jeff Patton

- Goals, requirements and architecture are often not specified in full details before development begins

12



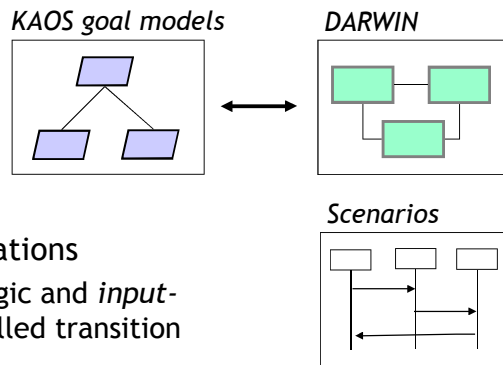
Research questions

- How to elaborate requirements and architecture during the initial design phase?
- When to *stop* the initial design phase ?
- How to co-evolve the requirements and architecture?

How to elaborate requirements and architecture during initial design?

Goals and architecture behaviour models

(with J. Kramer, J. Magee, S. Uchitel, W. Heaven)



- Formal foundations
 - temporal logic and *input-output* labelled transition systems
- Automated techniques
 - model checking
 - model synthesis from scenarios
 - model synthesis from goals

- **benefits: abstraction and analysis**

15

How to elaborate requirements and architecture during initial design?

Goals and design decisions

- **Quality goals**
 - performance, safety, security, ...
 - criteria for evaluating alternative design decisions
 - > **quantitative goal models** for reasoning about partial satisfaction defined in terms of **measurable criteria**
 - = probabilistic extension to KAOS goal models
- **Architecture stability**
 - architecture must be stable in face of requirements changes
 - > how to identify and document **likely changes?**
 - > how to choose architecture that is stable for likely changes?

16

When to stop the initial design phase?

Risk-based requirements engineering

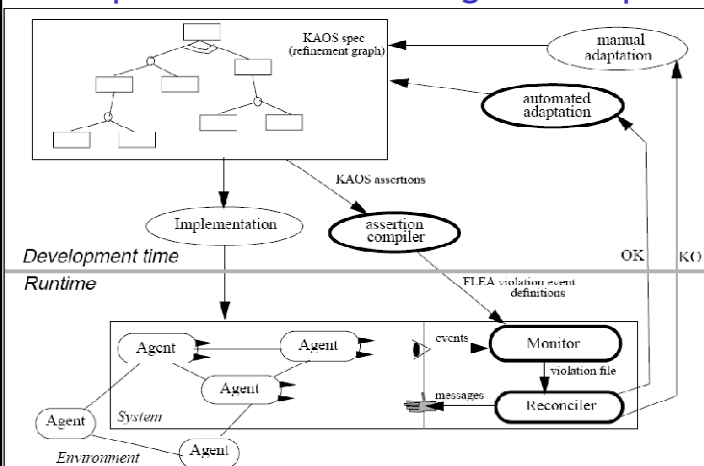
- Many projects fail due to premature leap to coding
 - “faults found late in development process are exponentially more costly to fix than if found and corrected earlier”
 - Yes, but
 - schedule is sometimes more important than quality
 - some requirements can not be fully understood before the system is built and used
 - a fully complete and flawless specification is unachievable

- How complete should the requirements be before coding?
 - approach: requirements defects entail **risks** to the development process
 - Need techniques to identify and reason about these risks

17

How to co-evolve requirements and architecture?

Requirements monitoring and adaptation



M. Feather et al.,
Reconciling System
Requirements and
Runtime Behavior,
IWSSD'98

- Lots of work on requirements monitoring
- Fewer work on requirements-level adaptation (Varsha Veerappa)
- A promising approach: **Requirements Reflection**

18

How to co-evolve requirements and architecture?

Enhancing stakeholders' involvement

through web-based participation and social networking

(an old idea: **participatory design**)



- Stakenet: recommend (and rate) stakeholders
<http://www.cs.ucl.ac.uk/research/StakeNet/> (Soo Ling Lim)
- StakeRare: submit and rate requirements (Soo Ling Lim)
- ARET: submit, review, modify requirements in goal model
(Camilo Fitzgerald)

19

Conclusion

- Goal-oriented requirements engineering is now a well-defined discipline in software engineering
- Current research aims at extending its scope
 - not limited to first phase of development
 - not limited to the problem definition (requirements)
independently from solutions (architecture)
- Happy to hear from you!
e.letier@cs.ucl.ac.uk

20