TDDE41 Software Architectures
Concepts and definitions

Lena Buffoni
lena.buffoni@liu.se

LINKÖPING UNIVERSITY

Lecture plan

- The importance of architecture
- What concepts do we need?
- The role of the architect

Reading: chapters 1,3,17

(Software) Architecture: Definitions

architecture *noun*

- *the art or practice of designing and constructing buildings. schools of architecture or design*
  the style in which a building is designed and constructed, especially with regard to a specific period, place, or culture: Georgian architecture
- *the complex or carefully designed structure of something: the chemical architecture of the human brain*
  the conceptual structure and logical organization of a computer or computer-based system

The *Oxford English Dictionary*

A software system architecture is the set of principal design decisions made about the system

Richard Taylor et. Software Architecture Foundations, Theory and Practice

More definitions

- The minimal set of properties that determines what programs will run and what results they will produce

  -Gerrit Blaaw & Frederick Brooks, Computer Architecture

- fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution

  – IEEE

Architecture as a discipline

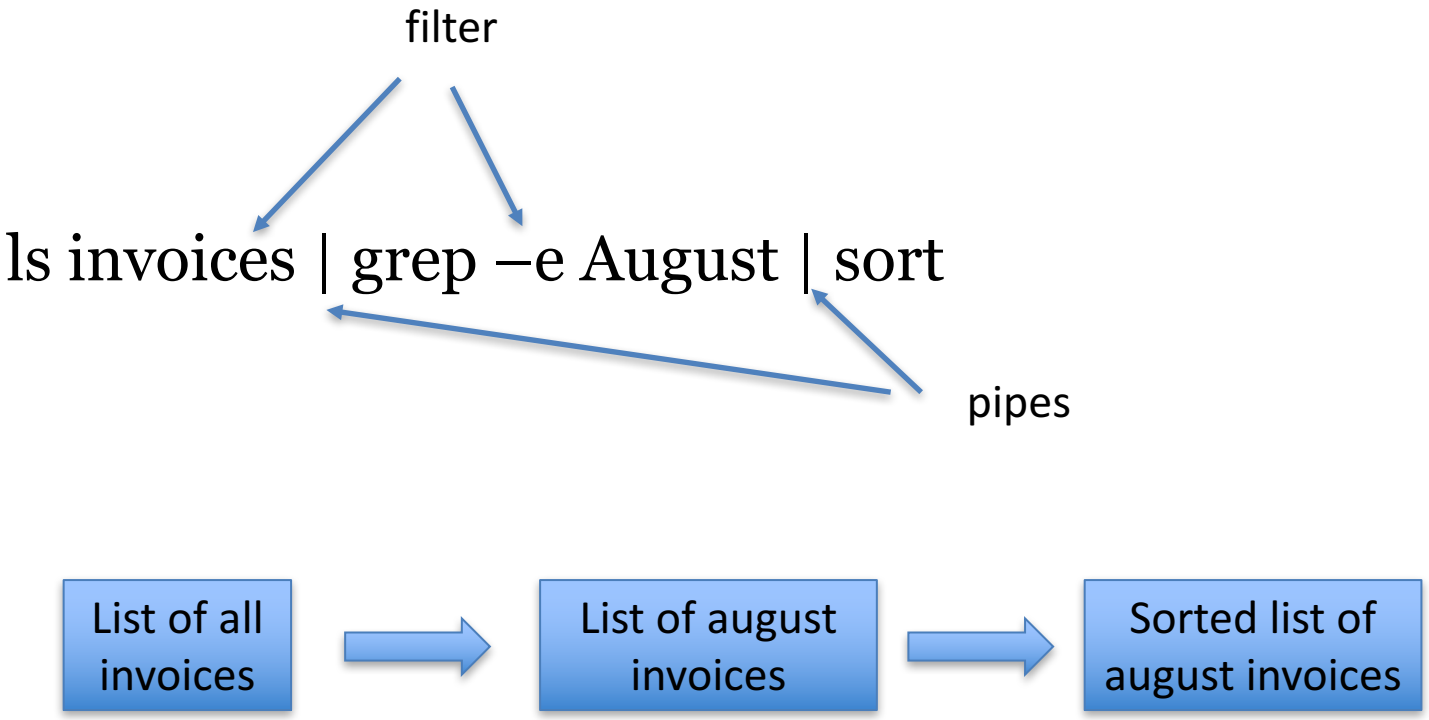
- Building software vs. building buildings
- The properties of the structure are induced by the design of it's architecture
- We know a lot of properties from previous experience

Jussieu campus, Paris: great for warm days, not so much for humid winters

The architecture centric process

- Determine architectural requirements
- Design architecture
- Implement
- Validate architecture
- Maintain and evolve architecture

Simple example: pipe-and-filter



Relies on a few simple rules

Pitfalls

- No process will guarantee a good result
- Software design is less clear cut than physical design
- Dynamic nature of software

Architecture and Requirements

- An iterative process
- Requirements cannot be written without awareness for architecture
- Identifying architecturally significant requirements
- Requirements and Constraints

Domain	Architecture requirement
Business	Technology must run as a plug-in for MS BizTalk
Development	The system must be developed in Java
Performance	Response time must be less than 4 seconds for 90% of requests
Scalability	Application must handle peak load of 500 concurrent users

Balancing design concerns

Buisness

Technical

Project life-cycle

Relying on previous experience

- Concepts from pre-existing architectures are usually used to reason about new architectures
- Green-field vs brown-field development
- Is previous experience always a +?

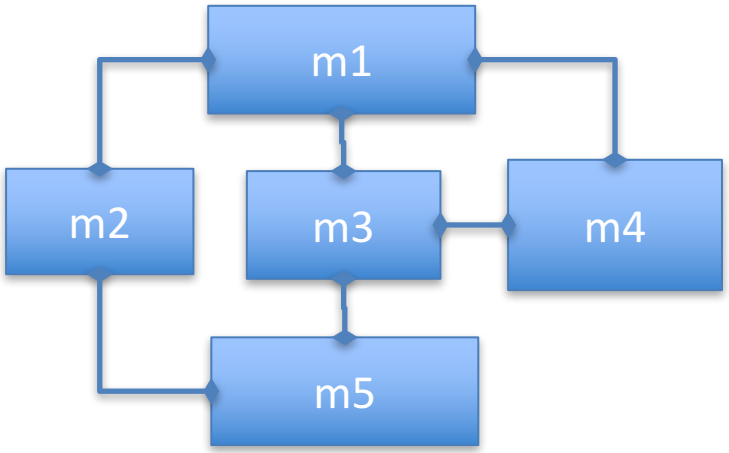
Prescriptive vs. descriptive architecture

- Prescriptive = set of design decisions "as-intended"
- Descriptive = set of design decisions as implemented in artefacts

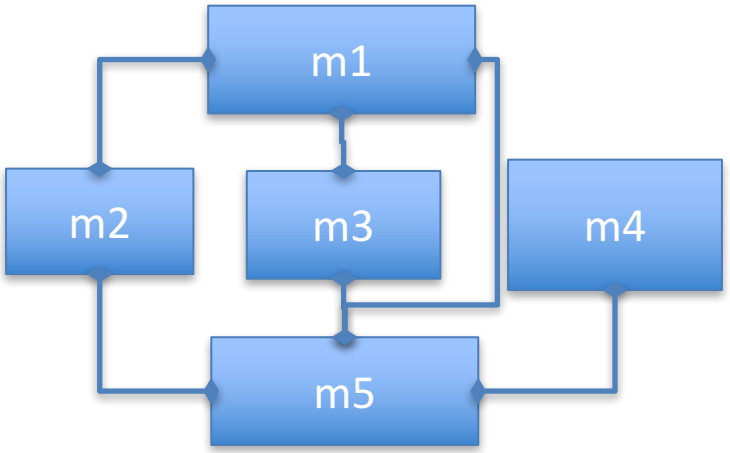
initial decision sets
for a legacy
architecture:
P0 = {}
D0 = {d1,….d100}

initial decision sets
for "greenfield"
development:
P0 = {p0,….p100}
D0 = {}

Prescriptive vs. descriptive: discrepancies



Prescriptive (P)



Descriptive (D)

Drift: D includes design decisions not included in P but not in conflict with P
Erosion: D includes design decisions not included in P and in conflict with P

Different kinds of structures

- Module
  - classes or layers
  - general overview of the structure
  - functional breakdown
- Component and connector
  - how the system is structured
  - runtime focus
- Allocation
  - non software structures (CPU, file systems…)

Architecture elements

- Component: an architectural entity that encapsulates a subset of the system's functionality or data; restricts access to this subset via explicit interfaces and has explicitly defined dependencies
- Connector: architectural elements regulating interactions between components
- Configuration: a set of specific associations between the components and connectors of a software system's architecture

Architectural style

A named collection of architectural design decisions that:

- Are applicable in a given development context
- Constrain architectural decisions specific to a particular system in that context
- Result in good qualities of the system

Eg: distributed systems and client-server style

Architectural Pattern

A named collection of architectural design decisions that

- Are applicable to a recurring design problem
- Parametrized to account for different contexts

Architectural style: “highly distributed”

Architectural pattern: “business logic must be separated from data management”

Discuss

- What defines a good software architect?

Architect role

Participate in product design meetings

Work with tool vendors

Familiar with common architectural styles in the domain

Knowledge of different design tools and techniques

Present and explain technical decisions

Engineering skills

Interpersonal skills

Sell the vision

Recognize problems and patterns

Consult

Listen

Develop project strategy

Plan

Negotiate

Modify and create design solutions

Organizational Skills

Maintain documentation for existing architecture

Evaluate commercial tools and software components from architectural perspective

Understanding of company business practices

Establish architecture-based testing perspectives

Facets of the role

- Software designer
- Domain expert
- Software technologist
- Standard compliance
- Risk management

Single architect or a team?

What makes a successful team?

- Good balance of skills
- Central to the project
- Involved over the course of the whole project (vs consultants for example)

Arcitect vs other stakeholdes

- Analysis: tend to be aligned in goals
- Software Engineers: validate technical design decisions, point out technical issues
- System Engineers: may be interested in local optimization, may not see the benefits of architecture design decision
- Managers: negotiation of budget, 3rd party contracts, infrastructures…
- Marketing : ensure customers needs and marketing goals are properly reflected in architecture, check claims are realistic

Becoming a better architect

- Gain experience by doing
- Improve non-technical skills
- Be prepared to compromise
- Solid technical background
- Avoid micro-managing
- …

Summary

- Architecture is a pivotal element of the software development process
- Consistency between intent and resulting architecture is important
- Non technical/ soft skills are very important

The End.

Questions?

www.liu.se

LINKÖPING UNIVERSITY