Architektur von Anwendungssystemen – Zusammenfassung

Sommersemester 2019

Lukas Heiland – last updated:

12. April 2019

Inhaltsverzeichnis

1 Definitions/Notions

1.1 Architecture

The architecture of an IT system is the structure or structures of the system which comprise software and hardware components, the externally visible properties of those components, and the relationships among them.

- Architecture isn't simply *good* or *bad*
- Architectue is fit or unfit *for a purpose*

1.1.1 Goals of an Architecture

- Producing a framework to support the development of software
- Creating an integration platform for future enhancements
- Producing the interface definitions for collaboration of components

1.1.2 Importance of Architecture

If the size and complexity of a software system increase, the global structure of the system becomes more important than the selection of specific algorithms and data structures.

1.1.3 Architecture vs. Design

An architecture provides a framework and a 'set of rules' for the act of designing a particular thing. So there can be many individually designed instances of each particular architectural style.

1.1.4 Architecture Levels

Conceptual Architecture

- direct attention at an appropriate decomposition of the system without delving into details
- provides a useful vehicle for communicating the architecture to non-technical audiences, such as management, marketing, and users
- consists of the Architecture Diagram (without interfaces) and an informal component specification for each component (see: ADL)

Logical Architecture

- adds precision, providing a detailed "blueprint"from which component developers and component users can work in relative independence
- incorporates the detailed Architecture Diagram (with interfaces)

Execution/Physical Architecture

- Shows the mapping of components onto the threads, processes, (virtual) machines, ...of the physical system
- created for distributed or concurrent systems

1.2 Architect

Some interesting definitions/quotations/etc. from various slides...

- The IT Architect defines (i.e. architects) solutions to client business problems through the reasoned application of information technology.
- The task of an architect is reduction of complexity to orders of magnitude that can be realistically handled.
- The definition of Vitruvius ($\approx 25B.C.$) adds, that an architect (of any kind) should have a lot of general knowledge.

The architect is the advocate of the client.

1.2.1 Where do Architects get ideas from?

Studies of work of other architects is key!
Importance of reference architectures, patterns, styles

1.2.2 Architectural thinking

Architectural thinking is based on basic architectural principles:

- Separation of concerns
- Information Hiding
- Design by interface
- Separation of interface and implementation
- Partitioning/distributing responsibilities

Architectural thinking involves

- Looking at the solution from the direction of requirements, not technology
- Understanding all aspects of the requirements (functional and non-functional)
- Understandign all aspects of the solution (functional and non-functional)
- Using reference architectures and patterns whenever appropriate
- Compromising and balancing; every solution to a requirement will cause other problems

1.3 System

Composition of parts into a new whole which represents via the collaboration of the parts more than the sum of its parts.

1.3.1 Emergence

This is a central aspect of Systems: Emergence is the appearance of properties of a system which none of its constituents has; i.e. a shelf: it is comprised of wooden planks and screws, and after you finished building it, you can put stuff on it. This is emergence: the planks and screws themselves did not offer the possibility to store things, it emerged from the system that is called *a shelf*.

1.4 Views

- Views = different models of a single system. Can be built by abstraction
- Architecture consists of multiple different model descriptions of a single building. Different model descriptions target different participants (stakeholders) of the project:

$$\text{Ground plan} \mapsto \text{Decorator}$$
$$\text{Wiring} \mapsto \text{Electrician}$$
$$\text{Plumbing} \mapsto \text{Plumber}$$
$$\ldots \mapsto \ldots$$

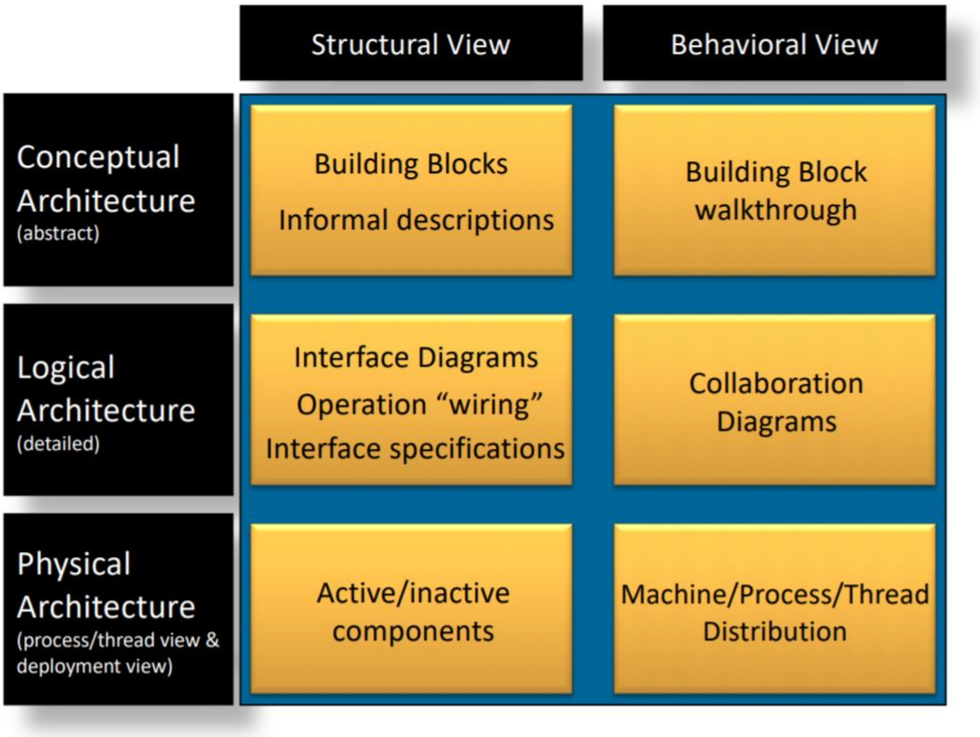
1.4.1 Structural and Behavioral Views

These are used to enhance understandability of the architecture's levels (see 1.1.4).

Structural Views consist of the Architecture Diagram, and Component and Interface Specifications

Behavioral Views Contain Component Collaboration or Sequence Diagrams; they answer the question '*How does this work?*'

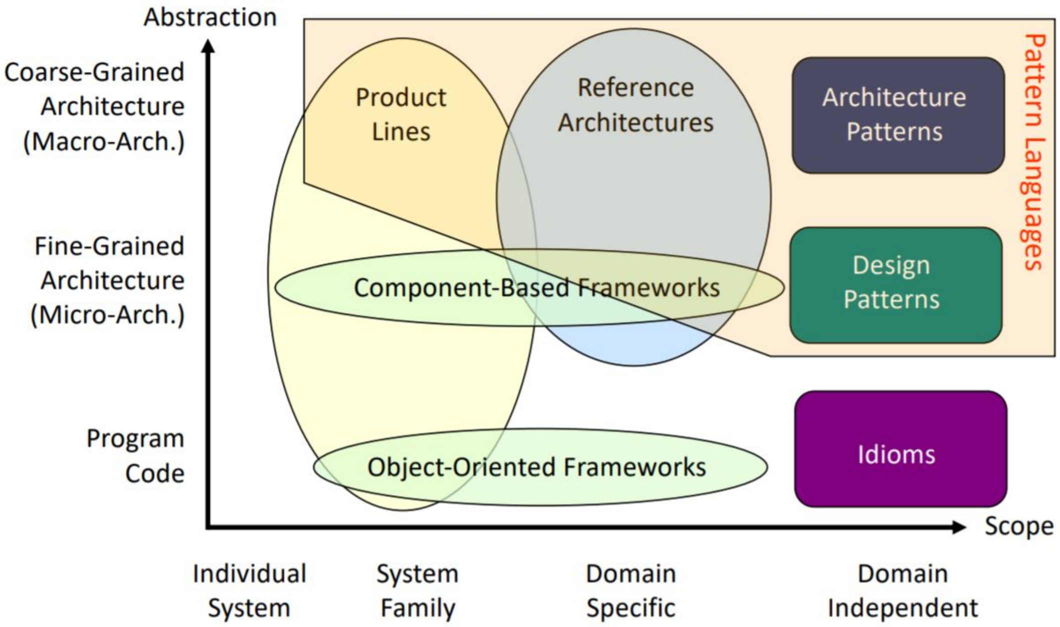
1.5 Architectural Levels and Views Together



	Structural View	Behavioral View
Conceptual Architecture (abstract)	Building Blocks Informal descriptions	Building Block walkthrough
Logical Architecture (detailed)	Interface Diagrams Operation "wiring" Interface specifications	Collaboration Diagrams
Physical Architecture (process/thread view & deployment view)	Active/inactive components	Machine/Process/Thread Distribution

1.6 Reuse

Classification



2 Diagrams and Styles

2.1 Basic elements of an architecture

2.1.1 Components

Components are the result of decomposition of a system.

2.1.2 Connectors

Connectors connect components.

2.1.3 Constraints

Components must be constrained to provide that

- the required functionality is achieved
- no functionality is duplicated
- the required performance is achieved
- the requirements are met
- modularity is realized (e.g. which modules interact with the operating system)

2.1.4 Rationales