Functional Architecture

Session 9

Course Informatics Business

Prof.dr. Sjaak Brinkkemper
Contents

- What is architecture?
- Design structures in architecture
- Functional Architecture
  - Enterprise Function Diagrams
  - Decomposition
  - Positioning
- Reference Architectures
- Creating a Functional Architecture
Architecture

Definition:
An Architecture is a high-level decomposition of a system into major components together with a characterization of the interaction of the components (van Vliet 2000)

Purpose:
1. Communication among stakeholders
   Architects, Product Mgrs, Software Engineers, Testers, Sales&Marketing, Consultants, Customers
2. Capturing design decisions
   Rationale for later reference
3. Transferable abstraction for reuse
   Generalized constructs to be applied in other contexts
Types of architecture

- Functionele architecture: usage perspective
  - Enterprise architecture
  - Application architecture

- Technical architecture: development perspective
  - Software architecture
  - Component architecture
  - System architecture
  - Data architecture
  - Data flow architecture

- Examples of different architectures on the next slides
Functional architecture BaanERP

Enterprise Professional

ERP  Collaboration  Business Intelligence  Portal
High level technical architecture

- User Interface
- User Interface Driver
- Application Logic
- Data Access Layer
- DBMS Driver
- Database Management System

- Win-XP, Browser, X-Windows, even ASCII
- IBM DB-2, Oracle, MS SQL Server, Informix

Other Application

Middleware
Software architecture

- Business Object Repository
- Subscription Repository
- Audit Trail
- ERP Database
- Business Object
- Business Object Layer
- BOR Layer
- Subscription Layer
- Interface
- DAL
- DAL2
- Synchro-nization Server
- QCD
- DCD
- BOI implementation
- ERP Server Adapter
- ERP Client Adapter
- BOI proxy
- Client Adapter
- SCM /
  CRM /
  PLM /
  E-E /
  …
- OpenWorld
- ERP
- Public method
- Protected method
- BOI proxy
- BOI implementation
- James Server Adapter
Component architecture

- Function drivers
- Shared services
  - Data banker
  - Physical models
  - Filter behaviors
- Device interfaces
- Application data types
- Extended computer
- Software utilities
Component Architecture
Data Flow Architecture

Pilot, external world

Device interfaces
- sensor inputs

Data banker
- computed values
- stored values
- calculated real-world values

Shared services

Function drivers
- filtered values

Physical models
- sensor values

Filter behaviors
- values to display
- stored values
- computed values
Contents

- What is architecture?
- Design structures in architecture
- Functional Architecture
  - Enterprise Function Diagrams
  - Decomposition
  - Positioning
- Reference Architectures
- Creating a Functional Architecture
What makes a good architecture?
What makes a good architecture?

- Architecture is a premier key to the success of a product
  - Good architectures survive many releases
  - New functionality can easily be incorporated
  - Elegancy of a product is reflected in the elegancy of the architecture

What makes a good design?

- SAP R3, Google, MS Windows, Linux, ...
- No scientific evidence of good architectures
- Some evidence for other domains: electronics, building, civil engineering
Influences on architecture

- Requirements of product
  - Functionality
  - Technical context
- Development organization and history
  - Earlier versions
  - Other systems
- Background and expertise of architect
- Technical environment
  - Design tools available
  - Development platform
Design structures

Design structures for the functional architecture

A. Modularity: structure and flexibility
   • What are the modules?
   • For now and for in the future?

B. Variability: multiple platforms and products
   • What variability is to be distinguished?
   • Both technical and functional
   • In which modules resides the variability?

C. Interoperability: interfacing externally
   • What interfaces are needed?
   • Standards or dominant players available?
   • Positioning of interfaces in architecture?

Discuss these on a beamer
Example: Technical modularity

A. Single UI platform

B. Multiple UI platforms
Example: Functional modularity

A. Single Warehousing scenario

- Warehousing Module
- Manufacturing Module

B. Multiple Warehousing scenarios

- Retail
- Factory outlet
- Distributor
- WH scenario Module
- Generic WH Module
- Manufacturing Module
Contents

- What is architecture?
- Design structures in architecture
  - Functional Architecture
    - Enterprise Function Diagrams
    - Decomposition
    - Positioning
- Reference Architectures
- Creating a Functional Architecture
Product Software Architecture

1. Functional Architecture
   - Enterprise Function Diagram
   - Feature Model

2. Technical Architecture
   - use techniques from other courses
   - UML
   - software architecture
A functional architecture is an architectural model from a usage perspective.

So the functional architecture should resemble the enterprise functions of the customer organisation or customer context:
- Names of modules should resemble the names of enterprise functions
- Flows in functional architecture resemble the interactions in the customer domain.

Standard functional architecture is called Reference Architecture.

More theory on enterprise models is presented in the course Enterprise Architecture.
Enterprise Function Model

An Enterprise Function Model is the representation of the primary process of an enterprise, consisting of its physical and administrative functions.

The Enterprise Function Model also helps to:
- identify the main enterprise functions
- show the interaction of the enterprise functions
- create a clear overview of the enterprise as a whole

EFD: Enterprise Function Diagram
Example of EFD

Production planning in a manufacturing company

Enterprise function

Purchase

Requirements Planning

Production

Sales

Receipt & Goods

Manufacturing

Packing & Shipping

Interaction
Functional Architecture

Simple ERP for manufacturing industry

Module

Purchase
- purchase order
- progress
- received goods

Requirements Planning
- material plan
- production plan

Production
- production
- picking list
- mfg schedules

Sales
- invoice
- customer request
- shipment order
- sign off
- sales order for planning

Information flow

Product scope
Procurement Management

Purchase Order Management

Receipt & Inspection

- Suppliers
- Planned purchase orders
- Purchase volume
- Purchase contracts and inquiries
- Purchase orders
- Purchase order confirmation
- Rejected items
- Purchase items
Enterprise Functions

- What is an Enterprise function?
  
  **Definition:** An enterprise function is a collection of **coherent processes**, **continuously performed** within an enterprise and **supporting its mission**

- Examples:
  
  - Purchase Order Management
  - Requirements Planning
  - Supplier Contract Management
  - Shop Floor Control

- The supportive analogue of an Enterprise Function in a software product is a **module**

- Naming standards:
  
  - Use substantivised nouns: Planning in stead of Plan
  - Precise, determining terms known in the business domain
  - Name is Capitalized
Identifying Functions (2)

Function Typology

- Life cycle stadium of a product or service:
  Specification, Design, Manufacturing, Distribution
- Supporting resources:
  Human Resource Management, Facility Management
- Control and planning of the company core activities:
  Administration, Finance, Credit Control

Usually two levels are enough for Functional Architectures
Notation for Functional Architectures

- **Module or Sub-module**
  - Color is used for categorization
  - Capitalized

- **Information flow**
  - All lower case

- **Product/Module scope**
  - Module name in corner

- **Scenario**
  - As overlay on FA

---

- **Sales**

- **sales order**

- **Sales**
Sales Order Management

Assembly Control

Pack & Ship Control

Final Assembly

Packing & Shipping

- New sales order
- Sales order executed
- Assembly order
- Assembly order executed
- Packing & shipping transfer
- Packing & shipping control executed
- Shop floor order
- Shop floor order executed
- Shipping order
- Shipping executed
- Finished goods
- Shipment
Identification of interfaces

Assembly Scheduling

Production Order Control

- Pick Control (Prod.)
- Production Operation Control
- Receipt Control (Prod.)
- Assembly

Sales Order Request Definition

Sales Order Fulfillment

- Pick Control (Sales)
- Packing Control
- Shipping Control
- Packing
- Shipping

Sales order

fulfilled sales order
Contents

- What is architecture?
- Design structures in architecture
- Functional Architecture
  - Enterprise Function Diagrams
  - Decomposition
  - Positioning
- Reference Architectures
- Creating a Functional Architecture
Decomposition

- What is a process?
  
  “A business process is an activity of which the execution can be described in terms of needed and delivered data and of which the start and end can be determined.”

- A process is the WHAT a company does
- Not HOW it is being done.
Functions versus Processes

- **Examples of Processes:**
  - accept order
  - create delivery schedule
  - receive payment
  - pay employee
  - verify customer credit rating

- Lower level of detail compared to a function
- No realization (how) information
- The supportive analogue of a process in a software product is a feature
- Processes are modeled in the Feature Model
The modules in a functional architecture are modelled in 2 or 3 layers. At the lowest level the module provides support for processes. The lowest level modules are elaborated in a feature model.
Example: Authoring tool

- Editing
- Checking
- Document Mgmt
- Printing

- Author Management
- Conversion and recovery
- File Mgmt
- Permission Handling

- Printer handling
- Paper setting
- Previewing
- Select Printer
- Find Printer
- Set Default Printer
- ...

- Open file
- Save file
- Save file as
- Convert file
- Send file to
Features

Feature models are a modeling tool for the process functionality in a software product.

Feature models

Author Management

Conversion and recovery

File Mgmt

Permission Handling

Open file

Saving

File exchange

Save file

Save file as

Convert file

Send file to
Contents

- What is architecture?
- Design structures in architecture
- Functional Architecture
  - Enterprise Function Diagrams
  - Decomposition
  - Positioning
- Reference Architectures
- Creating a Functional Architecture
Positioning

Control

Execution progress

Purchase

Requirements Planning

Production

Sales

Customer request
Invoice
Sales order for planning
Sign off
Shipment order
Mfg schedules
Picking list
Production
Material plan
Purchase order
Progress
Received goods
Hierarchical positioning

- **Strategic modules (management)**
- **Tactical modules (control)**
- **Operational modules (execution)**
- **Supportive modules (platform)**
Sequential positioning

3rd Party Products

Input modules

Processing modules

Output modules

3rd Party Products
Superpositioning

Input modules

Processing modules

Output modules

Strategic modules (management)

Plan

Source

Tactical modules (control)

Control

Deliver

Operation (execution)

Processing

Supportive modules (platform)

Supportive

3rd Party Products
Module positioning

Input functions

Processing functions

Output functions

Strategic functions (management)

3rd Party Products

Tactical functions (control)

Operational functions (execution)

Supportive functions (platform)
Contents

- What is architecture?
- Design structures in architecture
- Functional Architecture
  - Enterprise Function Diagrams
  - Decomposition
  - Positioning
- Reference Architectures
- Creating a Functional Architecture
Reference Architectures

A Reference Architecture is a generic functional architecture for software products in a particular line of business

Purposes:
• Expressing complete functional coverage in line of business
• Indication of the variability of a product
• Indication of the technical quality of a module
• Identifying market growth options and roadmap
• Positioning with 3rd party products

Sources on the web, e.g.:
• Supply Chain Operations Reference (SCOR), see www.supply-chain.org
Control Objectives for Information and related Technologies (COBIT)
Quality Indication

Technical Quality

Key:
- Differentiator
- Good
- Weak
- Opportunity
### Variability Indication

**Business Typology**

<table>
<thead>
<tr>
<th>Real Estate</th>
<th>Facility Mgmt</th>
<th>Both</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="red" alt="Real Estate Symbol" /></td>
<td><img src="blue" alt="Facility Mgmt Symbol" /></td>
<td><img src="red" alt="Both Symbol" /></td>
</tr>
<tr>
<td><img src="red" alt="Real Estate Symbol" /></td>
<td><img src="blue" alt="Facility Mgmt Symbol" /></td>
<td><img src="red" alt="Both Symbol" /></td>
</tr>
<tr>
<td><img src="red" alt="Real Estate Symbol" /></td>
<td><img src="blue" alt="Facility Mgmt Symbol" /></td>
<td><img src="red" alt="Both Symbol" /></td>
</tr>
<tr>
<td><img src="red" alt="Real Estate Symbol" /></td>
<td><img src="blue" alt="Facility Mgmt Symbol" /></td>
<td><img src="red" alt="Both Symbol" /></td>
</tr>
<tr>
<td><img src="red" alt="Real Estate Symbol" /></td>
<td><img src="blue" alt="Facility Mgmt Symbol" /></td>
<td><img src="red" alt="Both Symbol" /></td>
</tr>
</tbody>
</table>
Healthcare EFD

- Insurers
- Regulators
- Payers

Health Interest Groups

Client

Supplier

External Care Provider

- Materials Management
- Human Resource Mngt / Payroll
- Supplier Management
- Reporting / Analyzing

Facilitating Functions
- Service Management
- Customer Request Mngt.
- Project Management

Healthcare EFD

- Care Business Management
- Quality
- Planning & Budgeting
- Client Order Management
- Client Care Process Control

- Intake
- Observation
- Interpretation
- Intervention
- End of Episode of Care

- Resource Allocation
- Financial Administration

Point of Service

Public Health

Educational Organization

Care Customer

Customer
Care Planning & Budgeting

Care Business Management

Contracts

Demand Plan Definition

Demand Plan

Resource Planning

Resource Plan Definition

Resource Plan

Internal Budgets

Internal Budget Creation

Internal Budgets

Internal Budgets to be defined

Budget & Plan Monitoring

Internal Budgets to be defined

Demand Forecast Definition

Demand Plan to be defined

Financial Administration

Financial Business Calculation

Investment information

Internal Budgets to be Changed

CCPC

Care Business Management
Contents

- What is architecture?
- Design structures in architecture
- Functional Architecture
  - Enterprise Function Diagrams
  - Decomposition
  - Positioning
- Reference Architectures
- Creating a Functional Architecture
Creating a Functional Architecture

1. Determine scope
2. Define request - feedback flows
3. Model the operational module flow
4. Add control and monitoring modules
5. Specify external to/ from internal interactions
Step 1: Determine scope

- Identify all external products in the usage environment
- Take the Product Context diagram as a starting point.
- Include future products to be interfaced with
Suppose we are architecting a **Sales** product that will be interfaced with a **CRM** system and possibly with a **Transportation** product.
Step 2: Define request - feedback flows

Define all request-feedback loops

a. with the external products
b. between modules and external products

Note: focus just on essential interactions; those interactions related to the main functionality of the product
Flows usually appear in a pair:
- a combination of a Request and
- a return arrow that represents the Feedback of the result.

This construction is called a Request Feedback loop.
Example

- Describe interaction between product and external products:
- E.g. with CRM

sales contract
agreed sales contract
sales order
delivered sales order
sales quotation
provided sales quotation
Step 3: Model the operational module flow

- The operational module flow (aka. primary process) is modeled by identifying the steps as module boxes. These modules are separated by flows that are usually information flows or waiting queues.

[Diagram showing the flow from Sales Order Request to Change Mgmt to Order Fulfillment]
Example
Model operational module flow
Step 4: Add control and planning modules

- The operational module flow is controlled by one or more modules.
- The relation between every operational module and every control module is usually a request-feedback loop.
- On top of these modules the appropriate planning modules are added.
Example
Add control and planning

Sales

Sales Order Request

Sales Planning

Change Mgmt

Sales Control

Order Fulfillment

Customer Relationship Management

plans

margins

targets

request

realisation

final sales order

delivered sales order
Step 5: Specify interactions

- Specifying the information represented by the request flow from the external product e.g. CRM requests a sales order to be delivered.

- Follow the subsequent steps through the scope until one or more feedback flows; splitting and joining (bifurcations) are possible.

- Follow the process from the outside to inside the product.
Example
Specify interactions

Sales

- Sales Planning
  - Sales Order Request
    - request
    - plans
  - Sales Control
    - targets
    - margins
    - realisation
  - Order Fulfillment
    - final sales order
    - delivered sales order
- Quotation Control
  - Quotation Registration
    - quotation

CRM
Concluding

- **Enterprise Function Diagrams** are a powerful modeling tool for a functional architecture of software products.
- Assists in the *determination of modules*.
- *Interactions* between the modules and the environment are shown.
- FA Models are recognized by non-specialists without formal training.
- Can be applied in *any type of business*:
  - Public sector: health care, governmental
  - Private sector: manufacturing, financial, services, food and beverage, project industries