4. Designing Components
Darmstadt University of Applied Sciences, Department of Computer Science
Dr. Markus Voß (Accso GmbH)
Today’s topic

1. Introduction
2. Business Architecture
3. Designing Domains
4. **Designing Components**
5. Designing Interfaces and Operations
6. Loose Coupling
7. Excursus: Business Information Systems
8. Web Services
9. Integration
10. Service Orchestration
11. Evolution of Application Landscapes
Agenda

- Overview
  - AL components
  - Components Categories
  - A method for designing components
  - Reference architecture for application landscapes
  - Literature
Designing components in the context of the enterprise architecture framework

<table>
<thead>
<tr>
<th>Business</th>
<th>IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contextual (why?)</td>
<td>Business Strategy</td>
</tr>
<tr>
<td>Conceptual (what?)</td>
<td>Business Architecture</td>
</tr>
<tr>
<td>Logical (how?)</td>
<td>(Business Services, Business Processes, Business Objects, Organisation)</td>
</tr>
<tr>
<td>Physical (with what?)</td>
<td>Physisch</td>
</tr>
</tbody>
</table>

AS IS

TO BE

IDEAL

Evolution

Integration

Technical Reference Architecture

Ideal Application Landscape

Technical Services

Physische AL-Komponenten und ihre Schnittstellen

Logische AL-Komponenten und ihre Schnittstellen

Physische Anwendungs- und Integrationsplattformen

Logische Anwendungs- und Integrationsplattformen

Domains und (Anwendungs-) Services

Technical Infrastructure (TI)

Information System (IS)
## Overview of all methods

<table>
<thead>
<tr>
<th>Business</th>
<th>IT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Modelling business services</strong></td>
<td><strong>2. Designing domains</strong></td>
</tr>
<tr>
<td><strong>Business services</strong></td>
<td><strong>Domains</strong></td>
</tr>
<tr>
<td><strong>3. Designing components</strong></td>
<td><strong>Components</strong></td>
</tr>
<tr>
<td><strong>4. Designing interfaces</strong></td>
<td><strong>Interfaces</strong></td>
</tr>
</tbody>
</table>

- Two weeks ago
- Last week
- Today
- Next week
So again: What does service-oriented architecture really mean?

An architecture of the application landscape which is oriented on the business services.
Agenda

- Overview
  - AL components
    - Components Categories
    - A method for designing components
    - Reference architecture for application landscapes
    - Literature
A component of the application landscape (AL component) is a unit of the application landscape with the following properties:

1. It implements business services
2. Is is extensive (coarse-grained)
3. It provides interfaces for operations it offers
4. It specifies interfaces for operations it needs
5. It can be coupled with other AL components
Components of the application landscape: The buildings in the software city
Agenda

- Overview
- AL components
  - Component categories
    - A method for designing components
    - Reference architecture for application landscapes
    - Literature
Service and component categories

- Business services (and application services) can be categorized
- Components that implement the services have the same category:
  - Interaction service → Interaction component
  - Business process service → business process component
  - Business function service → business function component
  - Business data service → business data component
Business data components: Storehouses in the software city

- Control business objects that are managed by applications
- Provide CRUD operations: Create, Read, Update, Delete
- Provide consistency rules
- Provide higher-level operations for writing data, e.g., data history
- Provide various views on data (reading)
- Examples: CustomerManagement, ProductManagement, ...

- Symbol:
Business function components: Factories in the software city

- Implement complex business logic
- Usually the largest components in an enterprise architecture
- Use business data components
- Examples: Pricing, Forecasting, CapacityControl, Ranking, ...

Symbol:
Business process components: Control centres in the software city

- Implement business processes: process control
- Do not implement complex business logic but use business function components instead
- Use business data components
- Are often implemented using workflow management systems
- BUT: A workflow management system is part of the technical infrastructure, NOT an enterprise component

- Symbol:
Interaction components:
Shopping malls in the software city

- Allow users to interact with enterprise components
- Often: Internet or Intranet portals
- Uniform layout
- Single sign on
- Examples: SalesPortal, EmployeePortal, ...

Symbol:
AL Component Categories are not Three-Layer-Architecture

- AL components of all categories (data, function, process, and interaction) may contain all three layers (data access, business logic, and presentation)
- Example interaction component: CMS stores pictures and text
- Example Data component: GUI for managing master data
Agenda

- Overview
- AL components
- Components Categories
- A method for designing components
- Reference architecture for application landscapes
- Literature
A Method for designing components

<table>
<thead>
<tr>
<th>Business</th>
<th>IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business services</td>
<td>Domains</td>
</tr>
<tr>
<td></td>
<td>3. Designing components</td>
</tr>
<tr>
<td></td>
<td>Components</td>
</tr>
</tbody>
</table>

Take business services of the service hierarchy. Disregard the business services which are performed by humans only and select the ones that are – at least partially – to be performed by IT (called application services). Assign those application services to the domains.

The application services are categorized according to the categories data, function, process, and interaction. For the application services of one domain and one category, one enterprise component is constructed each.

The enterprise components constructed so far are being split according to rules for designing components.

Check the components for completeness with respect to the services to be covered. Find meaningful names.
Step #1: Assign to domains
Step #2: Categorize Services

- Compose custom tour
- Transact payment
- Check plausibility
- Check availability
- Book travel
- Manage travel orders
- Manage customers
- Transact payment
Step #3 Refine components: Rules for designing components

- Business logic that changes at a different pace shall be separated
- Transaction data shall be separated from static data
- Components shall not have cyclic dependencies
- Components of different categories shall have dependencies according to a layering: 
  - interaction \(\rightarrow\) process \(\rightarrow\) function \(\rightarrow\) data
Step #4: Finalize
Agenda

- Overview
- AL components
- A method for designing components
- Components Categories
- Reference architecture for application landscapes
- Literature
The reference architecture for application landscapes

- Every component is assigned to exactly one domain
- Every component is categorized
- Dependency rules apply
An Example: The Register Factory® of the Federal Office of Administration

Siehe www.register-factory.de
Agenda

- Overview
- AL components
- A method for designing components
- Components Categories
- Reference architecture for application landscapes

- Literature

Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: *Quasar Enterprise – Anwendungslandschaften serviceorientiert* gestalten. dpunkt-Verlag 2008.