**Problem structuring** (and: structuring the problem solving process)
Problem decomposition, “divide and conquer” method

1 Decomposition into solved and unsolved partial problems
1.1 Reference models and analogic transfer
1.2 Partially analogous reference models
1.3 Two sources of model construction
1.4 First, solution for standard cases and, then, for exceptional cases

2 Parallel decomposition: into simultaneous partial problems
2.1 Horizontal decomposition: same level of abstraction
2.2 Vertical decomposition: different levels of abstraction

3 Sequential decomposition: into a sequence of partial problems (steps)
3.1 Software (development) process

4 Inductive “decomposition”: into many individual cases
4.1 Cyclic inductive-deductive refinement

5 Problem structuring in a master’s thesis
5.1 Network modeling
5.2 Method matrix
A problem is a question of definition

Mathematician / physicist / chemist etc.
gets a fence
and the task:
“catch a lion (or a flock of sheep) in an enclosure made of the fence”
What happens?

(different versions on the web)
A problem is a question of definition multi-perspectivity of IS experts and organization / domain experts (ethnography)

Systems are relative to perspectives (adapted from Steinmüller 1993, 168)
1 Decomposition into solved and unsolved partial problems (parallel and sequential)

First, a mathematician / physicist gets an empty bucket and a tap and the task: “fill the bucket with water”

Then, the mathematician / physicist gets a bucket full of water and a tap and the same task: “fill the bucket with water”

What happens?

Reduction to an already solved problem
### 1.1 Reference models and analogic transfer
Here for data models; applies for process models as well

| creditor | debtor | umbrella terms 
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>supplier groups</strong></td>
<td><strong>customer groups</strong></td>
<td><strong>business partner gr.</strong></td>
</tr>
<tr>
<td>suppliers</td>
<td>customers</td>
<td>business partners</td>
</tr>
<tr>
<td>outgoing orders</td>
<td>incoming orders</td>
<td>orders/contracts</td>
</tr>
<tr>
<td>order lines</td>
<td>order lines</td>
<td>order lines</td>
</tr>
<tr>
<td>raw materials</td>
<td>products</td>
<td>items</td>
</tr>
<tr>
<td>material groups</td>
<td>product groups</td>
<td>item groups</td>
</tr>
</tbody>
</table>

→ one-to-many relationship

Cf. class models: generalization and specialization of models
1.2 Partially analogous reference models

Complete model analogies are rare. **Split a problem into a solved and an unsolved partial problem.**

**Example 1:** Number of order lines  
orders with only one order line vs. orders with more order lines

<table>
<thead>
<tr>
<th>customers library users</th>
<th>→</th>
<th>orders -</th>
<th>→</th>
<th>order lines borrow transactions</th>
<th>←</th>
<th>products books</th>
</tr>
</thead>
</table>

**Example 2:** Individual identifiability of items  
individually identifiable items (library books, cars) vs. not individually identifiable items

<table>
<thead>
<tr>
<th>borrow transactions order lines</th>
<th>←</th>
<th>books (copies) -</th>
<th>←</th>
<th>books (titles) products</th>
</tr>
</thead>
</table>
1.3 Two sources of model construction

Starting from reality: *empiristic* method/approach
observation and interviews of employees by a model designer
(contrary to natural sciences: only observation)
abstraction, inductive type construction

often used for individual parts of an organization

Starting from models, concepts, ideas: *rationalistic* method
reference models
activation in a model designer’s brain
analogic transfer

often used for standard parts of an organization, e.g. accounting

**Final step**: *integration* of individual and reference models
## 1.3 Two sources of model construction

<table>
<thead>
<tr>
<th>external world</th>
<th>phenomenon, individual experience</th>
<th>model, theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>World 1 objects of cognition</td>
<td>World 2 knowledge of an individual subject of cognition</td>
<td>World 3 common knowledge</td>
</tr>
</tbody>
</table>

- **World 1**
  - perception, cognitive processes (empiristic)
  - reconstruction of World 1

- **World 2**
  - memory

- **World 3**
  - learning rationalistic activations of World 3
  - creation, induction

- **Publication**
  - design, influence
  - new ideas, knowledge
1.4 First, solution for standard cases and, then, for exceptional cases

Lab problem:
Process description: acceptance of an order at the phone text, optionally with a diagram

1. describe your assumptions regarding the company
2. higher abstraction level: standard process (reference model)
3. lower abstraction level: details of the standard process
4. lower abstraction level: exceptions (observation)

Take care of exact correlations between 2, 3 and 4
Presentation as table with three columns for 2, 3 and 4
2 Parallel decomposition

Lab problem:
Normalization and integration (with overlaps) of partial data models
(assume that the invoice data is stored in a separate accounting IT system)

<table>
<thead>
<tr>
<th>Orders (sales department):</th>
<th>Sales representatives (marketing dept.):</th>
<th>Item statistics (controlling dept.):</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomerID</td>
<td>SalesRepID</td>
<td>ItemID</td>
</tr>
<tr>
<td>CustomerName</td>
<td>SalesRepProvision</td>
<td>StatisticYear</td>
</tr>
<tr>
<td>CustomerAddress</td>
<td>CustomerID</td>
<td>StatisticMonth</td>
</tr>
<tr>
<td>OrderNo</td>
<td>CustomerName</td>
<td>Turnover</td>
</tr>
<tr>
<td>OrderDate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ItemID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ItemDescription</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PricePerUnit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quantity</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 2.1 Horizontal decomposition: same level of abstraction

<table>
<thead>
<tr>
<th>Static models</th>
<th>Dynamic models</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data models</strong></td>
<td><strong>Information flow models</strong></td>
</tr>
<tr>
<td>Data (structure) models:</td>
<td>information / data flow charts / diagrams; Structured Analysis (SA); UML use case diagrams</td>
</tr>
<tr>
<td>data structure diagrams; entity-relationship models (ERM); UML class diagrams</td>
<td></td>
</tr>
<tr>
<td><strong>Function models</strong></td>
<td><strong>Behavior / process models</strong></td>
</tr>
<tr>
<td>Function structure models:</td>
<td>algorithms (functions); Nassi-Shneiderman diagrams; (control) flow charts; business process models; UML activity diagrams; (UML sequence diagrams)</td>
</tr>
<tr>
<td>compositional function trees; Jackson trees</td>
<td></td>
</tr>
</tbody>
</table>
Static function model: function structure model
irrespective of tests, iterations, sequences

Dynamic function model: control flow model

Dynamic data model: information flow model

Static data model: data structure model
2.2 Vertical decomposition: different levels of abstraction

- High abstraction level
  - Overview model
  - without details

- Low abstraction level
  - Partial models
  - with many details

Observer

Top-down

Bottom-up

Organization
2.2 Vertical decomposition: software process as a tree model
2.3 Summary: horizontal and vertical decomposition

vertical decomposition

↑

→ horizontal decomposition
3 Sequential decomposition

Geometric problem:
Transform an arbitrary quadrilateral into a square with equal area

Steps:
1. quadrilateral → triangle  
2. triangle → isosceles triangle (equal legs)  
3. isosceles triangle → rectangle  
4. rectangle → square (triangle altitude theorem)

Network modeling for project management
Example: see 5.1
### 3.1 Sequential decomposition: software (development) process

<table>
<thead>
<tr>
<th>Main phase</th>
<th>Subphase, model level</th>
<th>Methods (ex.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytic phase: problem analysis</td>
<td>Elicitation of the current state of the org</td>
<td>Systems analysis, Reverse engineering</td>
</tr>
<tr>
<td></td>
<td>Analysis of the current state of the organization</td>
<td>Requirements engineering, OOA</td>
</tr>
<tr>
<td></td>
<td>Design of the planned state of the org. (sociotechnical IS)</td>
<td>Requirements engineering, BPM</td>
</tr>
<tr>
<td></td>
<td>Design of the business concept of the IT system (technical IS)</td>
<td>Reference mod., test case description</td>
</tr>
<tr>
<td>Synthetic phase: IT system development</td>
<td>Design of the techn. concept of the IT system independ. of developm. tool</td>
<td>OO design, design patterns</td>
</tr>
<tr>
<td></td>
<td>... depending on development tool</td>
<td>Unit tests</td>
</tr>
<tr>
<td></td>
<td>Programming</td>
<td>Coding conv., agile programming</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>V model tests</td>
</tr>
</tbody>
</table>
3.1 Sequential decomposition: software (development) process

On its way through a software (development) process, a model of a technical IS has to be transferred in several steps via different models, each of which in turn is split vertically and horizontally, from an organization / enterprise model on the information level to a technical model on the implementation level. Every software process phase represents a certain perspective (diaphasic multi-perspectivity)
3.2 Summary: horizontal, vertical and sequential decomposition

vertical decomposition

↑
→ horizontal decomposition

↓ sequential decomposition
4 Inductive “decomposition”: into many individual cases

Find a general formula for the sum of natural squares.

1. Find solutions for individual cases (there are no standard cases)
2. Generalize these solutions

\[ 1^2 = 1 \]

\[ 1^2 + 2^2 = 5; \quad \text{is the sum equal to } \frac{n \cdot (n^2 + 1)}{2} ? \]

\[ 1^2 + 2^2 + 3^2 = 14 \]

\[ 1^2 + 2^2 + 3^2 + 4^2 = 30 \]
As there are $n$ summands, the biggest of which is $n^2$, the solution has the order $n^3$ and a good starting point is:

$$\sum_{j=1}^{n} j^2 = an^3 + bn^2 + cn + d$$

which leads to four cubic equations in four variables (difficult!).

Solution:

$$\sum_{j=1}^{n} j^2 = \frac{n \cdot (n + 1) \cdot (2n + 1)}{6}$$

Proof with mathematical induction.
4.1 Cyclic inductive-deductive refinement

(Holl / Paetzold / Breun, IS anti-aging, 2008; according to Holl, 1999, 175)
5.1 Problem structuring in a master’s thesis: network modeling
5.2 Problem structuring in a master’s thesis: method matrix

Description / motivation of the structure (from where? how? where?)

- **Starting point** of your thesis: what do we know? Where are the problems?
  Clear description of research questions / scientific issues

- **Exact definition of the objectives** (to answer the research questions),
  description of your task in detail;
  motivation: what is the purpose of your thesis? What are you aiming at?
  Objectives: reachable, pragmatic, not too optimistic / high / broad

- **Motivation of the selection of the methods used and the way how the methods are applied**
  (how shall the objectives be reached) and motivation of the use of particular methods.
## 5.2 Problem structuring in a master’s thesis: method matrix

<table>
<thead>
<tr>
<th>Where from?</th>
<th>Where?</th>
<th>How?</th>
<th>What?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting point scientific issues</td>
<td>Definition of the objectives</td>
<td>Methods (and their use)</td>
<td>(Form of the) Results</td>
</tr>
<tr>
<td><strong>Entire thesis</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 1</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 2</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 3</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>etc.</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ISSUES

Issue 1:
The big jumble of graphical BPM notations

Issue 2:
Frequent unstructured BPM modeling style

Issue 3:
Unmotivated BPM decomposition on several levels of abstraction

GOALS

Goal of Issue 1:
- Classification of graphical notations
- Evaluation of graphical notations
- Improvements of graphical notations

Goal of Issue 2:
- Demonstration of structured BPM

Goal of Issue 3:
- Demonstration of an approach for a motivated decomposition of BPM models on several levels of abstraction
- Minimization of complexity in decomposed BPM models

METHODS

Method 1 (Issue 1):
- Genealogic comparison
- Comparison using a structured BPM style

Method 2 (Issue 2):
- Analysis of the actual BPM modeling style
- Introduction and discussion of UML BPM

Method 3 (Issue 3):
- Discussion of theory of Gestalt
- Introduction and discussion of "Nature oriented EPCs"

RESULTS

Result of Issue 1:
- Genealogical tree of graphical notations
- Proposals for improvements of graphical notations
- Overview of analogies of graphical notations
- Evaluation with strengths and weaknesses profiles

Result of Issue 2:
- Proof of the necessity of structured BPM (SIPM) shown on an example
- Requirements to SIPM
- Demonstration of the defects caused by unstructured BPM style

Result of Issue 3:
- Proof of logic decomposition on several levels of abstraction (under the aspects of theory of Gestalt)
6 Appendix

6.1 Function point method: weighted aspects (pros, cons)

6.2 Decision table

6.3 Examination of the context / environment of a problem
(Ex.: improve the purchase processes …)

Etc. etc.