4. Basic concepts for the description of software development

- a) Description of results of software development phases - software development documents
- b) Definition of basic concepts for the description of software development and their classification
- c) Basic concepts and software development phases
- d) Classification of basic concepts according to their notation
- e) Basic concepts and their application areas
- f) Basic concepts and their use in analysis methods
Description techniques of software documents

Problem: How to describe?

Customer's requests  Specification  Design  Code

Customer  Analyst  Designer  Programmer

Description of software documents by ...

classification?

UML  state automaton  sequence diagr.

use case diagr.  entity relationship diagr.

data flow diagr.  class diagr.

JML  pseudo code

Source: Schneider, SEUH 43, p. 123
Basic concepts of software development
Balzert vol. 1, 2nd edition 2001

Concepts and Views

Alternative Notations

Functional View

Use Case Diagram 1987

Data Flow diagram 1966

Data-Dictionary 1979

Entity Relationship Model 1976

Class diagram 1980/1990

Pseudo-code 1957

Rules 1960/1990

State Automation 1954

Petri Net 1962

Collaboration diagram 1987

Functional hierarchy

Business Process

Information Flow

Data Structures

Entity types and relations

Class structures

Control structures

If-Then structures

Finite State Automation

Concurrent structures

Interaction structures

Data-Oriented View

Object-Oriented View

Object-Oriented View

Algorithmic View

Rule-Based View

State-Oriented View

Scenario-Based View

Concepts and Views

elementary formalized description mechanisms

UML

classification in three levels:

Which view on the software?

Which actual notational forms?

Which aspect of this view?

basic concept

Box diagram 1973

Program flowchart 1966

Decision tables 1957

Activity diagram 1997

Collaboration diagram 1987

Box diagram 1973

Program flowchart 1966

Decision tables 1957

Activity diagram 1997

Collaboration diagram 1987

DAAD project "Joint Course on Software Engineering" © 2001
### Basic concepts of software development

**Balzert vol. 1, 1st edition 1996**

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternative Notations</td>
<td>Jackson Diagram 1975</td>
</tr>
<tr>
<td>Often used</td>
<td>Syntax Diagram 1964</td>
</tr>
<tr>
<td>Rarely used</td>
<td>Program Flowchart 1966</td>
</tr>
<tr>
<td>Function tree</td>
<td>Decision Tables 1957</td>
</tr>
<tr>
<td>Data dictionary 1979</td>
<td>State Automaton 1954</td>
</tr>
<tr>
<td>Class structures</td>
<td>Interaction Diagram 1987</td>
</tr>
<tr>
<td>Control structures</td>
<td>Interaction Structures</td>
</tr>
<tr>
<td>If-Then structures</td>
<td>Finite State Automata</td>
</tr>
<tr>
<td>Rule-Based View</td>
<td>Concurrent Structures</td>
</tr>
<tr>
<td>State-Oriented View</td>
<td>Scenario-Based View</td>
</tr>
<tr>
<td>Data-Oriented View</td>
<td>Object-Oriented View</td>
</tr>
<tr>
<td>Algorithmic View</td>
<td>Class Structures</td>
</tr>
<tr>
<td>Rule-Based View</td>
<td>Heterogeneity</td>
</tr>
<tr>
<td>Business Process</td>
<td>Diversity of approaches in basic concepts: different views on software products must be possible (different aspects of a software product)</td>
</tr>
<tr>
<td>Information Flow</td>
<td>Business Processes</td>
</tr>
<tr>
<td>Data Structures</td>
<td>Data Structures</td>
</tr>
<tr>
<td>Entity types and relations</td>
<td>Business Processes</td>
</tr>
<tr>
<td>Class diagrams</td>
<td>Data Structures</td>
</tr>
<tr>
<td>Pseudo-code</td>
<td>Class Diagram 1980/1990</td>
</tr>
<tr>
<td>Rules</td>
<td>Pseudo-code</td>
</tr>
<tr>
<td>State diagrams</td>
<td>Rule-Based View</td>
</tr>
<tr>
<td>Control structures</td>
<td>Finite State Automata</td>
</tr>
<tr>
<td>Concurrent structures</td>
<td>Interaction Structures</td>
</tr>
<tr>
<td>Decision Tables</td>
<td>Scenario-Based View</td>
</tr>
<tr>
<td>Interaction Diagram 1987</td>
<td></td>
</tr>
</tbody>
</table>

---

### Software Development Remains a Mixture of Methods

Software development as science and as a practical activity have something in common: heterogeneity.

> Diversity of approaches in basic concepts: different views on software products must be possible (different aspects of a software product)
Basic concepts: principles

- **phase-independent**
  - usable in different phases

- **application-independent**
  - kind of application determines the choice of a convenient basic concept

- **method-independent**
  - basic concepts can be combined to general methods (e.g. structured analysis, OOA with UML ...)

---

Basic concepts and software development phases

![Diagram of basic concepts and software development phases]

Legend: A → B: A is used in B

<table>
<thead>
<tr>
<th>Definition phase</th>
<th>Use case diag.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>data flow diagram</td>
</tr>
<tr>
<td></td>
<td>Petri net</td>
</tr>
<tr>
<td></td>
<td>ER (entity relationship)</td>
</tr>
<tr>
<td></td>
<td>DD (data dictionary)</td>
</tr>
<tr>
<td></td>
<td>class diagram</td>
</tr>
<tr>
<td></td>
<td>box diagram</td>
</tr>
<tr>
<td></td>
<td>program flow chart</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Design phase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Implementation phase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Classification of basic concepts according to their notation

Basic concepts and their application areas

<table>
<thead>
<tr>
<th>application areas</th>
<th>complexity...</th>
<th>basic concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>data dictionary, ER (Entity Relationship), (state automaton), class diagram,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>use case diagram, data flow diagram, function tree</td>
</tr>
<tr>
<td>technical-scientific</td>
<td>...of data</td>
<td>data dictionary, ER (Entity Relationship), (state automaton), class diagram,</td>
</tr>
<tr>
<td></td>
<td>...of functions</td>
<td>use case diagram, data flow diagram, function tree</td>
</tr>
<tr>
<td></td>
<td>...of algorithms</td>
<td>pseudo code, program flow chart, box diagram, decision table, rules</td>
</tr>
<tr>
<td></td>
<td>...of system environment</td>
<td>data flow diagram</td>
</tr>
<tr>
<td></td>
<td>...of time-dependent behavior</td>
<td>Petri nets, state automaton, activity diagram, sequence diagram, collaboration diagram</td>
</tr>
<tr>
<td></td>
<td>...of user interface</td>
<td>graphical specification, (Petri nets), (state automaton), control structures, rules</td>
</tr>
</tbody>
</table>
### Basic concepts of ... software development:

**Balzert vol. 1, 2nd edition 2001**

<table>
<thead>
<tr>
<th>Functional View</th>
<th>Data-Oriented View</th>
<th>Object-Oriented View</th>
<th>Algorithmic View</th>
<th>Rule-Based View</th>
<th>State-Oriented View</th>
<th>Scenario-Based View</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional hierarchy</td>
<td>Business Process</td>
<td>Information Flow</td>
<td>Data Structures</td>
<td>Entity types and relations</td>
<td>Class structures</td>
<td>Control structures</td>
</tr>
<tr>
<td>Functional View</td>
<td>Data-Oriented View</td>
<td>Object-Oriented View</td>
<td>Algorithmic View</td>
<td>Rule-Based View</td>
<td>State-Oriented View</td>
<td>Scenario-Based View</td>
</tr>
</tbody>
</table>

#### Basic concepts:
- complete?
- additional forms of notations?

#### 3 level classification scheme flexible

- Box diagram 1973
- Program flowchart 1966
- Decision tables 1957
- Activity diagram 1997
- Petri Net 1962
- Sequence diagram 1987

#### Basic concepts: classification of enhancements

<table>
<thead>
<tr>
<th>Functional hierarchy</th>
<th>Business Process</th>
<th>Information Flow</th>
<th>Data Structures</th>
<th>Entity types and relations</th>
<th>Class structures</th>
<th>Control structures</th>
<th>If-Then structures</th>
<th>Finite State Automation</th>
<th>Concurrent structures</th>
<th>Interaction structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional View</td>
<td>Data-Oriented View</td>
<td>Object-Oriented View</td>
<td>Algorithmic View</td>
<td>Rule-Based View</td>
<td>State-Oriented View</td>
<td>Scenario-Based View</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DAAD project „Joint Course on Software Engineering“ © 2001

---

**Basic concepts of ... software development:**

**Balzert vol. 1, 2nd edition 2001**

- Algebraic Specification 1972
- Hoare Logic 1969
- Z
- Structure Chart
- Grammars
- EBNF
- Package Diagram
- Component Diagram
- State Chart 1987
- SDL

**Function tree**

- Use Case Diagram 1987
- Data flow diagram 1966
- Data Dictionary 1979
- Entity Relationship Model 1976
- Class diagram 1980/1990
- Pseudo code
- Rules
- State automaton 1954
- Petri Net 1962
- Sequence diagram 1987

**Basic concepts:**
- Functional View
- Data-Oriented View
- Object-Oriented View
- Algorithmic View
- Rule-Based View
- State-Oriented View
- Scenario-Based View

**DAAD project „Joint Course on Software Engineering“ © 2001**
Basic concepts: detailed classification of enhancements

Combination of basic concepts in object-oriented analysis

Legend:
- A → B: A is contained in B
- A −−→ B: A is implicitly contained in B
Combination of basic concepts in structured analysis

SA 1979

function tree
data flow diagram
data dictionary

decision table
decision tree

pseudo code

functional hierarchy
information flow
data structures
control structures

legend: A B: A is contained in B
A ~ B: A is implicitly contained in B

Combination of basic concepts in structured analysis and real-time analysis

RT 1987

SA 1979

data flow diagram
data dictionary

decision table
decision tree

pseudo code
state automaton

ER (entity relationship)

information flow data structure entities & relations control structures finite automaton