



BPM WITH ARIS THE ARIS METHOD

Lecture Slides – Part II



Software AG
University Relations



THE ARIS METHOD

QUICK FACTS

ARIS is a meta-model for systematic process modelling.



Name: Architecture of Integrated Information Systems (ARIS)



Founder: A.-W. Scheer (Saarland University)



Idea: Meta-model for process modelling (guidelines to model processes systematically)

THE ARIS METHOD

ARIS HOUSE

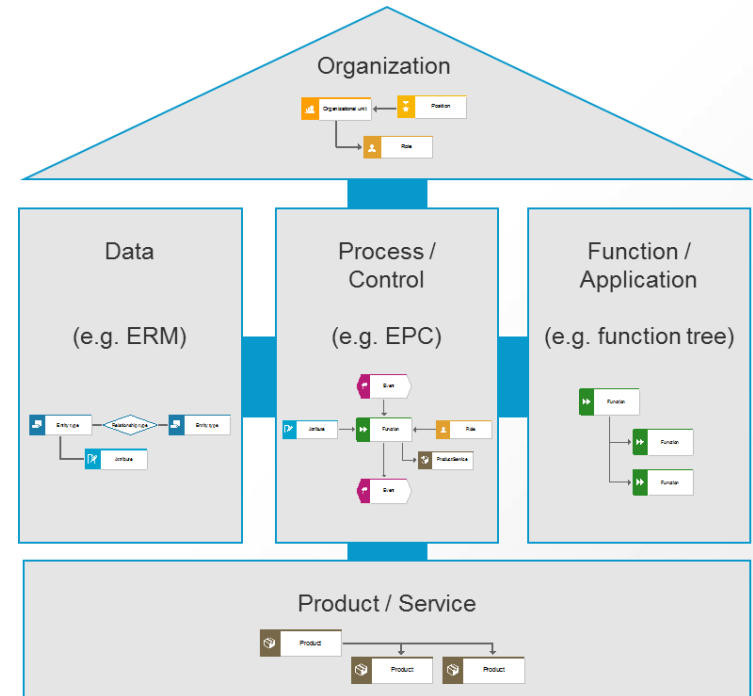
ARIS divides processes in 5 different views.

Each **process** (process view)

- Is performed by an **actor** (organization view),
- Requires or generates **data** (data view),
- Is composed by several **actions** (function view)
- And has some kind of **output** (product/service view).

In other words: the model answers the following questions

- **What** is done? → Function view
- **Who** does it? → Organization view
- **Why** is it done? → Product/Service view
- **Which information** is needed/provided? → Data view
- **How** is it done? → Process view

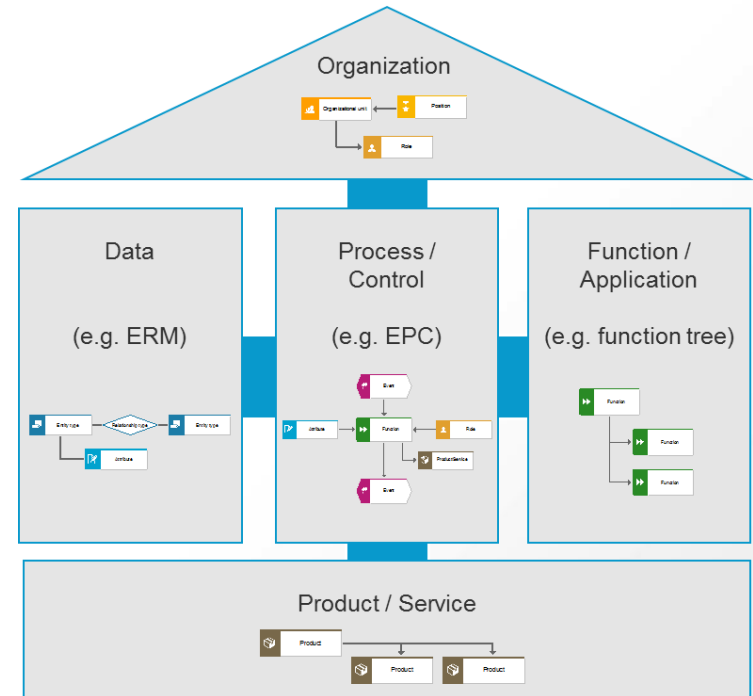


THE ARIS METHOD

ARIS HOUSE

The split in views **reduces complexity** and ensures a **systematic approach**.

- The process view connects all other views
- Every view is illustrated in a certain modelling notation



THE ARIS METHOD

CONTENT

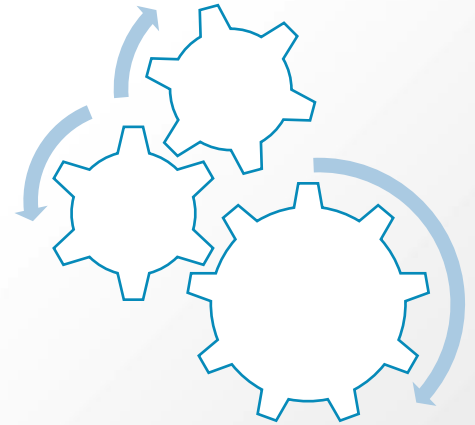
1 Product / Service View

2 Function View

3 Organization View

4 Data View

5 Control / Process View

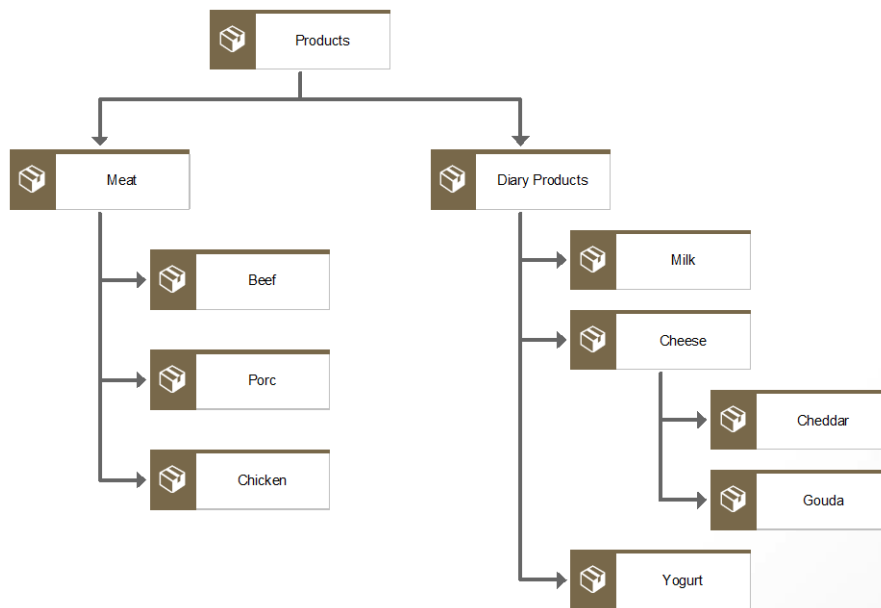


PRODUCT/SERVICE VIEW

PRODUCT/SERVICE TREE

The product/service tree is a **hierarchical list** of all products/services.

→ It provides an **overview** of the entire product/service **portfolio**.



THE ARIS METHOD

CONTENT

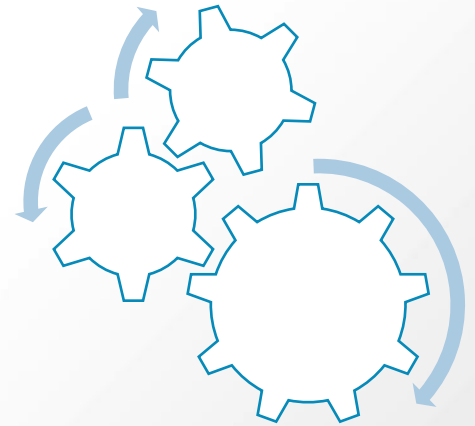
1 Product / Service View

2 **Function View**

3 Organization View

4 Data View

5 Control / Process View

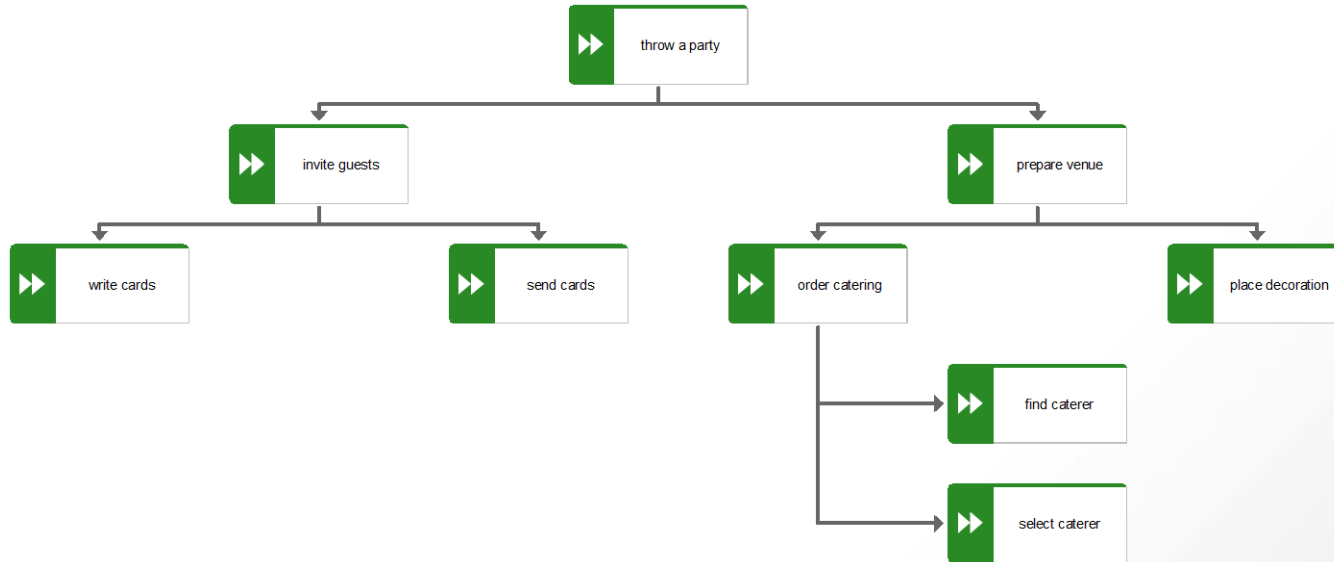


FUNCTION/APPLICATION VIEW

FUNCTION TREE

The function tree is a [hierarchical list](#) of all functions.

→ It provides an [overview](#) of the [activities](#) of a company.



THE ARIS METHOD

CONTENT

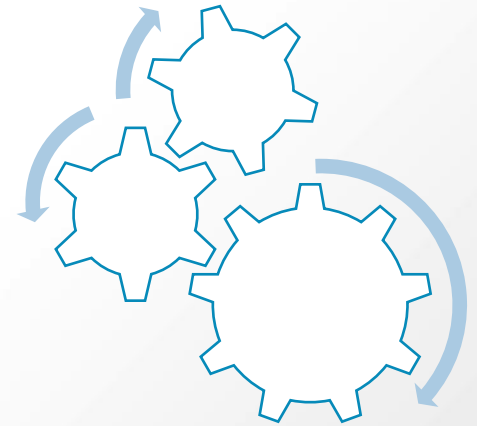
1 Product / Service View

2 Function View

3 Organization View

4 Data View

5 Control / Process View

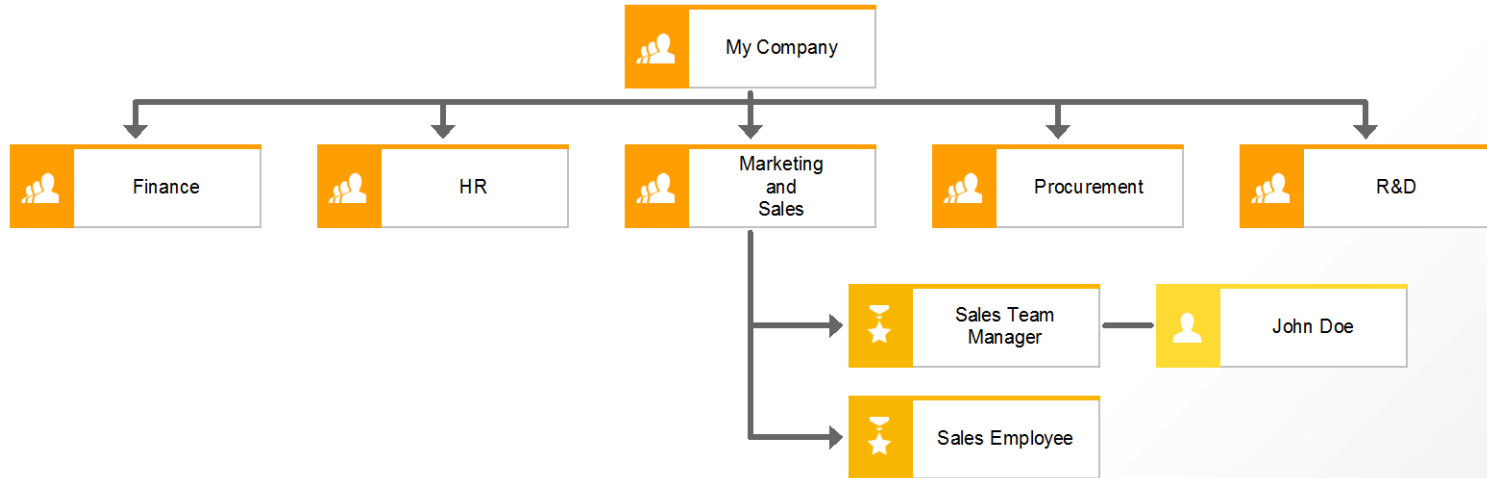


ORGANIZATION VIEW

ORGANIZATIONAL CHART

The organizational chart is a **hierarchical list** of all organizational units/elements of a company.

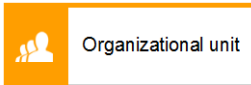
→ It provides an **overview** of the **organizational structure** of a company.



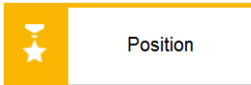
ORGANIZATION VIEW

ORGANIZATIONAL CHART

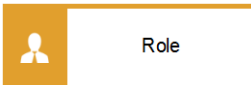
Most used are “organizational unit” and “position”, but there are also other elements.



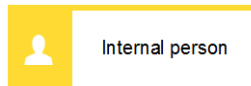
- Represents a department or sub-department (e.g. HR)



- Represents a job description (e.g. HR manager)
- Each employee occupies exactly one



- Represents a task description (e.g. ARIS Trainer)
- One employee can occupy none, one, or several (switch roles as necessary)



- Represents a specific person (e.g. John Doe)

THE ARIS METHOD

CONTENT

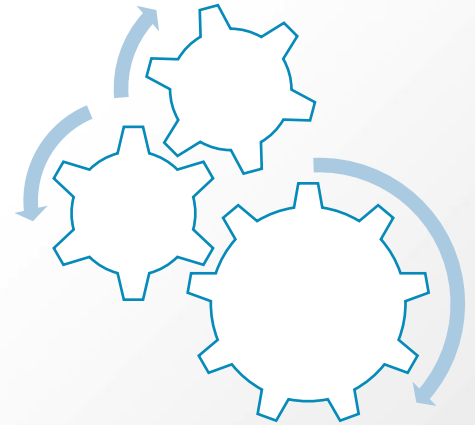
1 Product / Service View

2 Function View

3 Organization View

4 Data View

5 Control / Process View



DATA VIEW

ERM – QUICK FACTS

An ERM is used to model data.



Name: Entity Relationship Model (ERM)



Founder: Chen



Idea: Modelling notation for (relational) databases



DATA VIEW

ERM – RELATIONAL DATABASES

(Relational) databases are nothing more than (connected) tables.

Data File

↓

Student			
mat.no.	first name	name	birthday
12345	John	Doe	04/26/1963
12356	Jane	Doe	07/13/1965
12367	Johnnie	Doe	11/05/1996
12378	Janie	Doe	01/19/1998
...

Data Record →

↑

Data Field

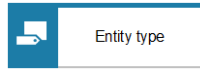
Book		
ISBN	name	author
151430	Economics 101	Smith
151431	Adv. Economics	Smith
151540	Economics 101	Marx
151550	Economics 101	Keynes
151560	Economics 101	Friedman
151570	Economics 101	Nash
...

→ ERM is used to model such a system of tables.

DATA VIEW

ERM – BASICS

The model is composed of **three basic elements** which are **linked** to each other.



Entity type

Entity type

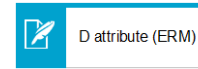
Type of a real world object



Relationship type

Relationship type

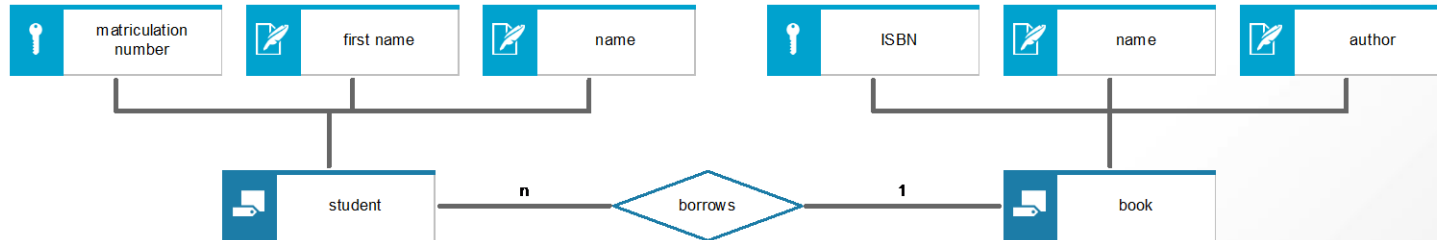
Logical link between two entity types



D attribute (ERM)

Attribute

Properties of entity types or relationship types

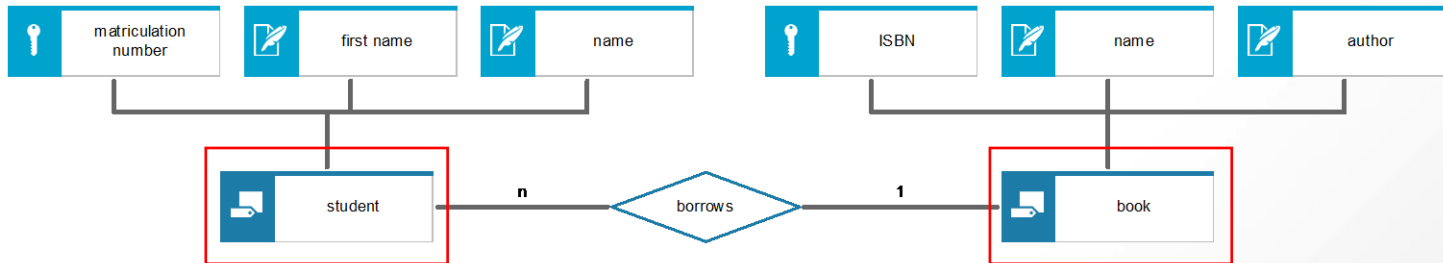


DATA VIEW

ERM – TYPES AND INSTANCES

Modeling is always done **at type level**.

- A type is an abstraction of a real world object (e.g. „student“)
- The concrete object is called an **instance** (e.g. „John Doe“)
- An instance of the previous example would be: “**John Doe** borrows **Economics 101**”
- However: modeling in ERM is always done at type level (!)

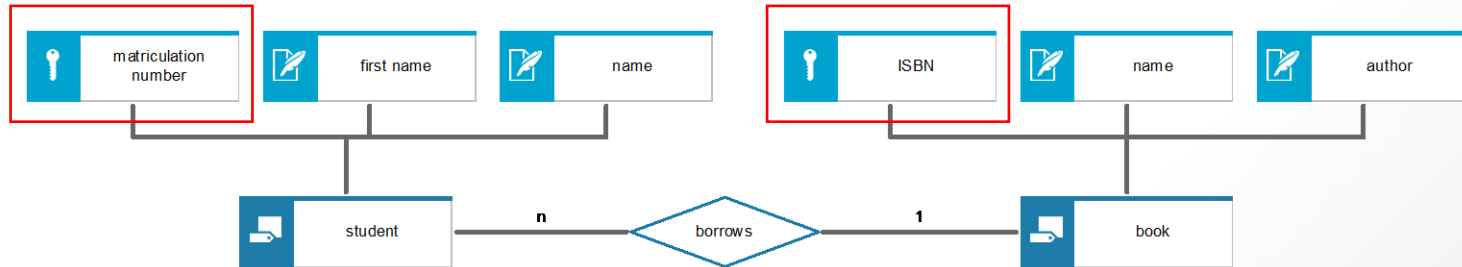


DATA VIEW

ERM – KEY ATTRIBUTE

A key attribute identifies one entity (instance) exclusively.

- E.g. an ISBN identifies one book exclusively, whereas the name of the author most likely would indicate several books.
- E.g. “151430” identifies “Economics 101” whereas “Smith” would identify „Economics 101“ and “Advanced Economics”.



DATA VIEW

ERM – MULTIPLICITY

Any relationship type can be specified by one of the **three** multiplicities.

1:1

one-to-one (1:1)

E.g.: student owns mobile phone

→ Every mobile phone belongs to exactly one student

1:N

one-to-many (1:N)

E.g.: student borrows book

→ One student can borrow several books, but one book can only be borrowed by one student (at a time)

N:M

many-to-many (N:M)

E.g.: student enrolls in class

→ One student can enroll in several classes, and one class is enrolled by several students

DATA VIEW

ERM – MULTIPLICITY – NOTATION

There are **three ways** to write those multiplicities down.

Chen (1976)

Question: How many objects can be assigned to one object of the other side?

- How many students can be assigned to one book? → 1
- How many books can be assigned to one student? → n



Schlageter/Stucky (1977)

Question: With how many other objects can one object (of this side) be related to?

- With how many books can one student be related to? → n
- With how many students can one book be related to? → 1



Min-Max-Notation (ISO-Notation, 1982)

- States a range (minimum, maximum) of possible relations
- Instead of an 'n', a * is widely used



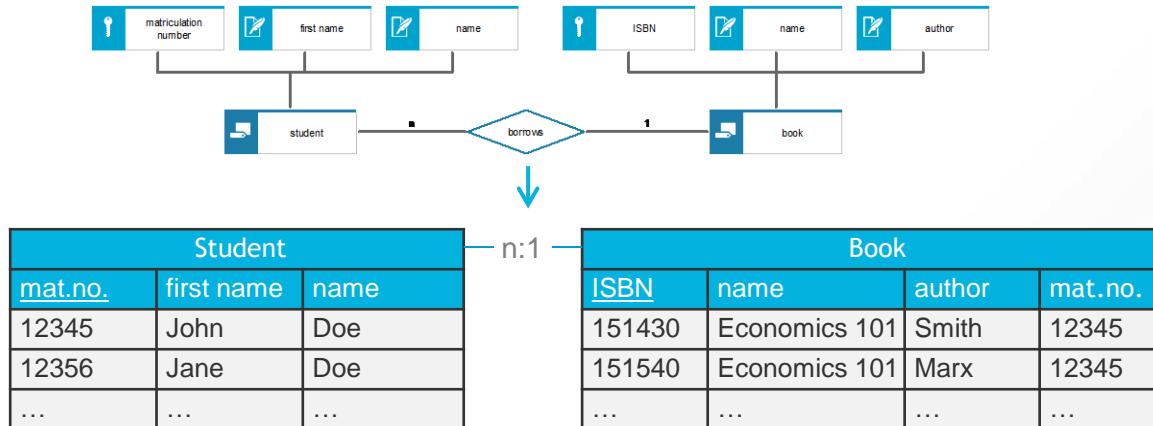
! Be careful: the UML class diagram uses the “old” Chen notation.

DATA VIEW

ERM – FROM MODEL TO DATABASE

1:n example

- Entity types represent tables, attributes represent data fields
- A data record can be identified by the primary key
- 1:n-relationships can be implemented using only two tables (one for each entity type)
- In the books-table “mat.no.“ is called a **foreign key**

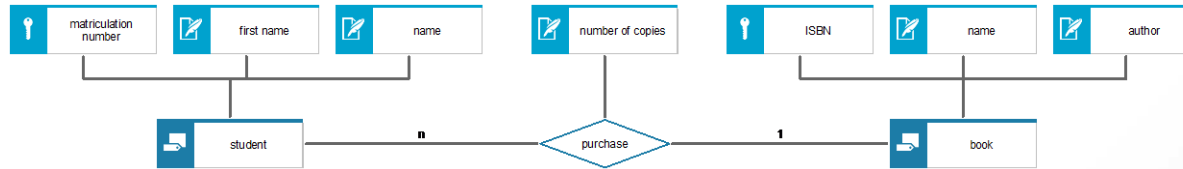


DATA VIEW

ERM – FROM MODEL TO DATABASE

n:m example

- If the student buys the books, the relationship changes into n:m
- An additional attribute (number of copies) can be added to the relationship type
- n:m-relationships are modelled using 3 tables (one for each entity type and relationship type)
- The primary key in the relationship-table is a **composed key** of the two foreign keys



Student		
<u>mat.no.</u>	first name	name
12345	John	Doe
12356	Jane	Roe
...

n:1

Purchase		
<u>mat.no.</u>	<u>ISBN</u>	Number
12345	151430	3
12345	151540	1
...

1:n

Book			
<u>ISBN</u>	name	author	<u>mat.no.</u>
151430	Economics 101	Smith	12345
151540	Economics 101	Marx	12345
...

THE ARIS METHOD

CONTENT

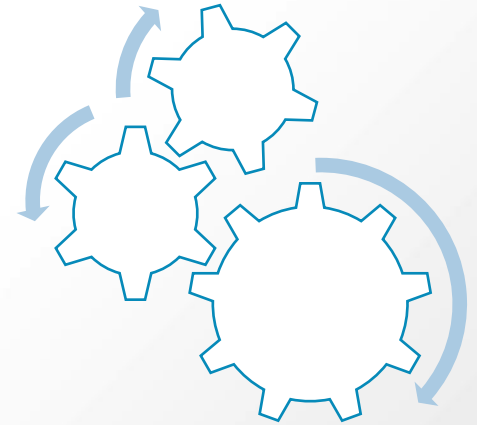
1 Product / Service View

2 Function View

3 Organization View

4 Data View

5 Control / Process View



PROCESS VIEW

EPC – QUICK FACTS

An EPC is used to model processes.



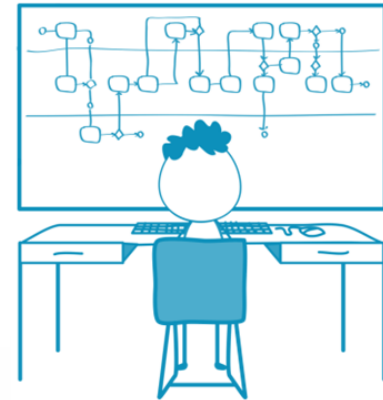
Name: Event-driven Process Chain (EPC)



Founder: A.-W. Scheer (Saarland University)



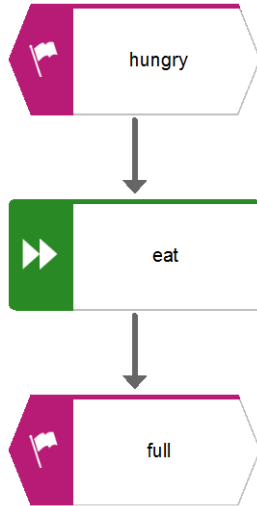
Idea: Modelling notation for processes



PROCESS VIEW

EPC – BASICS

There are **three** basic elements.



- **Functions** are **activities** carried out by a real person
- **Events** **trigger** these activities or are the **result** of them
- The control flow represents the **sequence** of activities

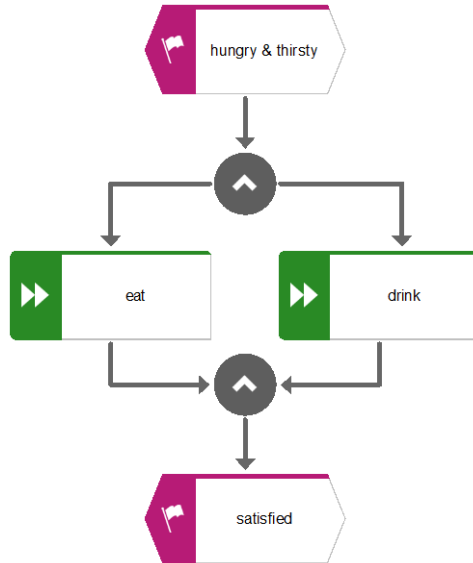
Rules:

- Every EPC **starts** and **ends** with an event.
- Events and functions **alternate**
(a function cannot be followed by another function and an event cannot be followed by another event)
- Exception: **trivial events** must not be modelled
(events who only state the termination of a function without adding any useful information, e.g. “check invoice” → “invoice checked”)

PROCESS VIEW

EPC – OPERATORS

Operators **split** and **merge** the control flow.



There are **three** operators:

- ⬆ AND – all succeeding paths will be executed (simultaneously)
- ⬇ OR – one or several of the succeeding paths will be executed (simultaneously)
- ⊗ XOR – only one of the succeeding elements will be executed (exclusive OR)

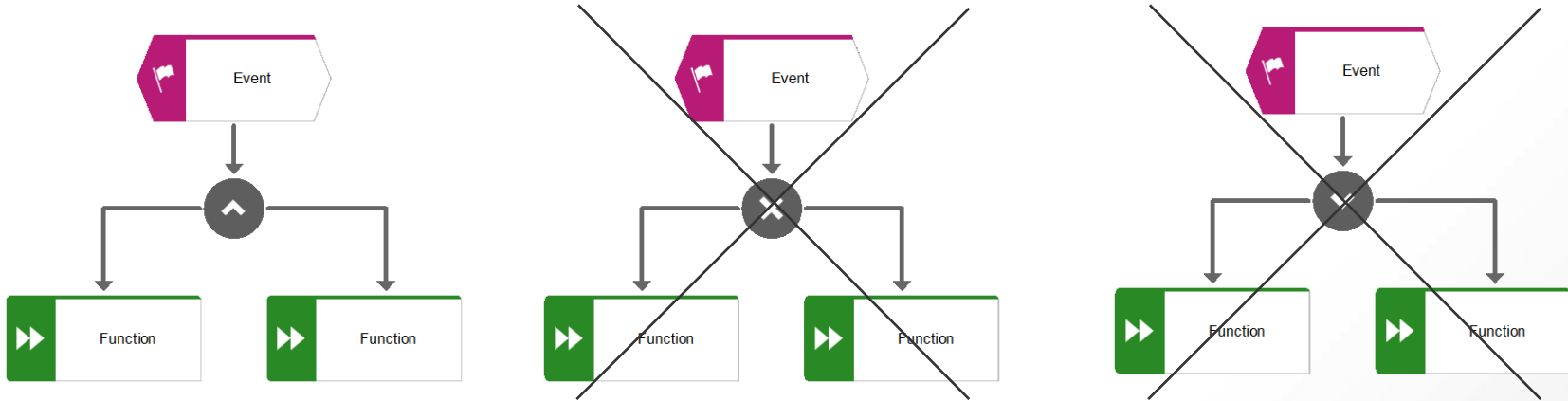
Rules:

- splitting operator = joining operator

PROCESS VIEW

EPC – OPERATORS

Events cannot 'decide'.

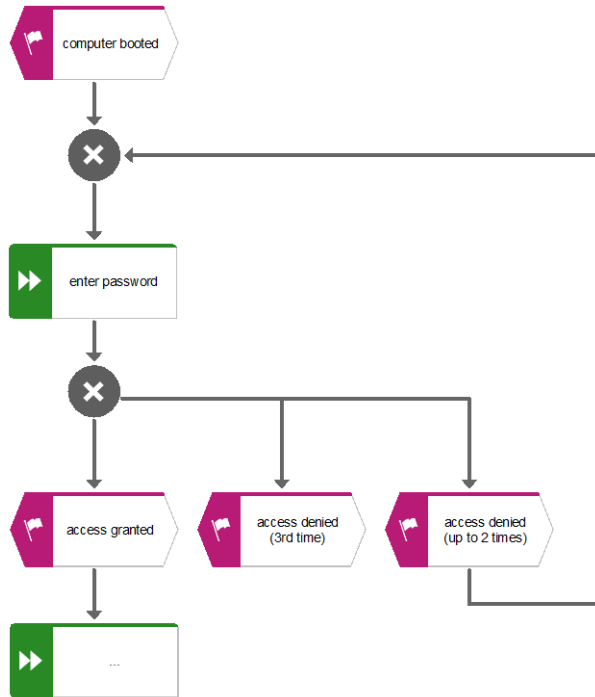


- Only use the AND-Operator to split the control flow after an event
- Apart from these two, there are no restrictions on how to use operators
 - e.g. use one operator to split in four different paths
 - e.g. use an OR-Operator to merge two events

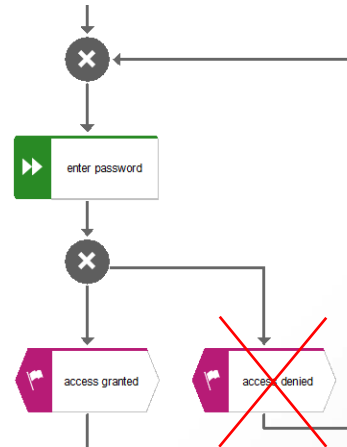
PROCESS VIEW

EPC – LOOPS

Loops are also possible.



- Introduced by XOR
- Joining occurs before Splitting
- Avoid 'infinite loops'!

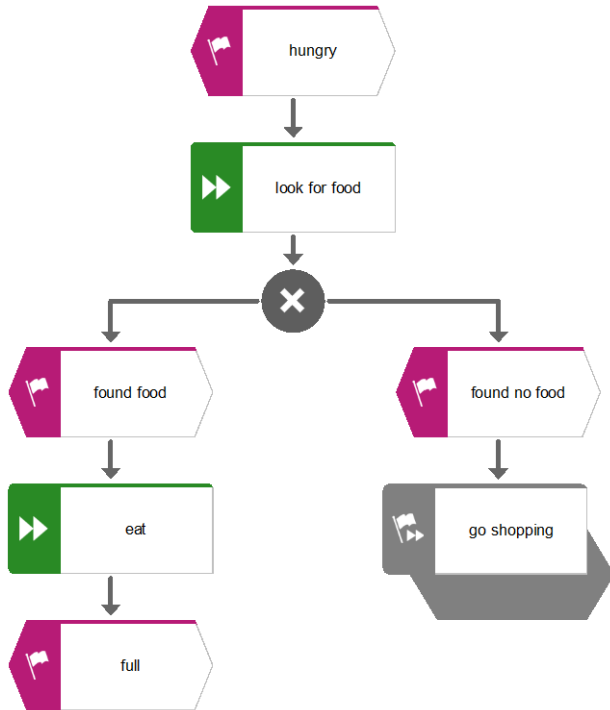


If you don't know the password, you might be stuck forever!

PROCESS VIEW

EPC – INTERFACES

Process Interfaces are used to **link independent processes**.

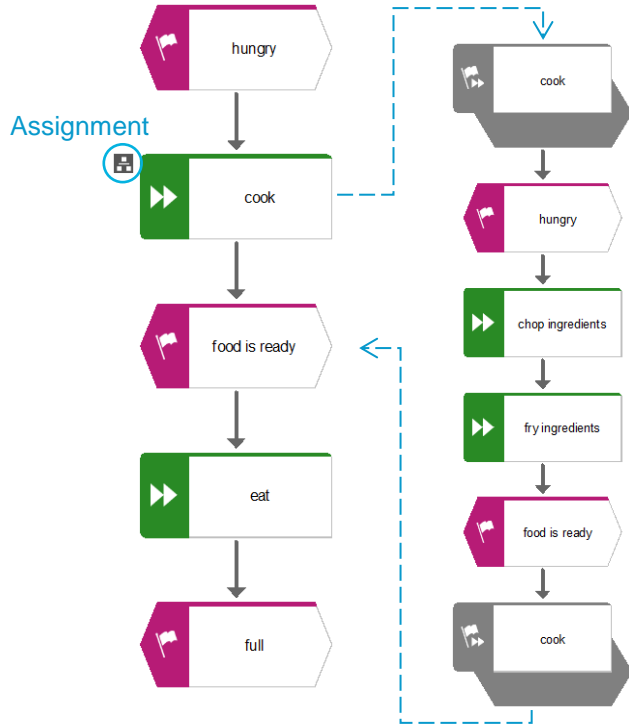


- Process Interfaces are used to signal
 - that the process **triggers** the execution of another process (like in the example)
 - that the process is **triggered by** another process
- Process Interfaces refer to a process that is **not part** (logically) of the original process

PROCESS VIEW

EPC – ASSIGNMENTS

Assignments are used to **outsource details**.



- With assignments, functions can be described in more detail in another EPC
 - **Outsource** detail in another EPC
 - **Reuse** sub-processes in different EPCs

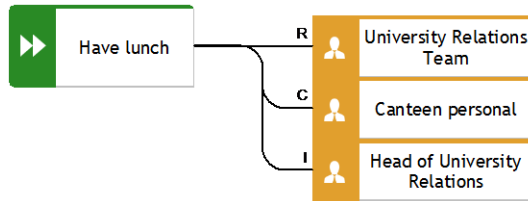
Rules:

- The **same events** are used both in the high-level and the low-level process (“hungry” / “food is ready”)
- Has to link to an **EPC with the same name** (“cook”)
- Starts and ends with a **process interface**, to indicate that the process does not stand alone

PROCESS VIEW

EPC – RACI

To each function, different **roles** (according to RACI*) can be assigned.



Responsible: Role is responsible for the execution of the function; execution may be handled by that person also, or delegated to somebody else (still the responsibility remains with the role)

Accountable: Role to whom “R” is accountable; who must sign off and approve; commercial responsibility

Consulted: Role has information and/or capability necessary to execute the function

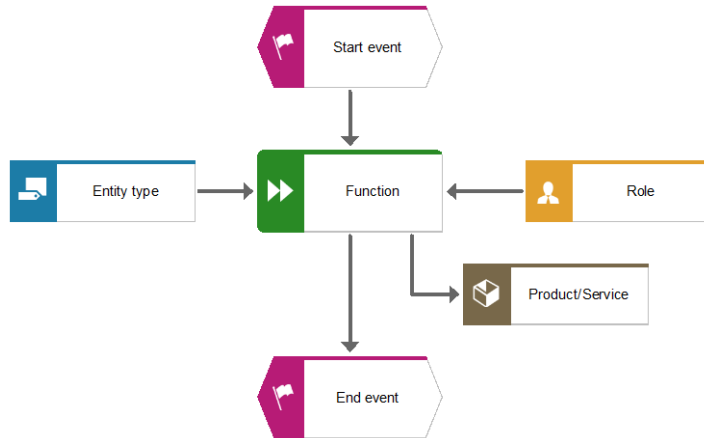
Informed: Role must be notified of results; has the right to ask for information; but does not need to be consulted during execution

*The RACI Matrix is the most prevalent tool for clarifying roles and responsibilities in cross-functional projects and processes.

PROCESS VIEW

EEPC

eEPC – Extended Event-driven Process Chain

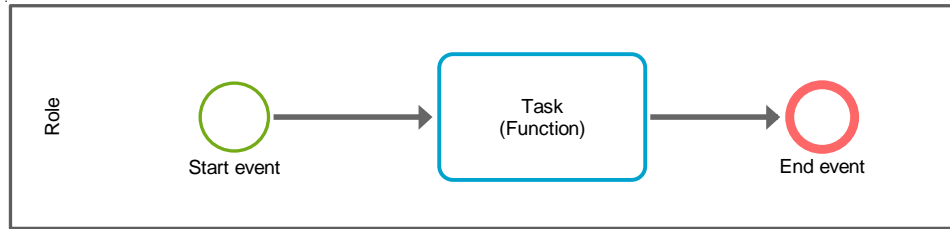


- An eEPC includes all relevant information to the process.
- Integrates all the other views:
 - Roles (Organization)
 - Product/Service
 - Data Input / Output

PROCESS VIEW

BPMN

BPMN – Business Process Model and Notation



ARIS is able to **automatically transfer** any EPC into a BPMN.

LEARN MORE

Students / Universities

→ University Relations
www.softwareag.com/university

☰ World-wide mail account
university@softwareag.com

Customers / Partners

→ Global Education Services
www.softwareag.com/education

☰ Regional mail accounts
Please refer to “[contact us](#)” section on the webpage mentioned above.

