Business Process Modeling and Reengineering

Module 1
Business Process Modeling

Prof. Fabiana Pirola
Prof. Giuditta Pezzotta

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Lecture objectives

- Understanding the meaning of a process modeling

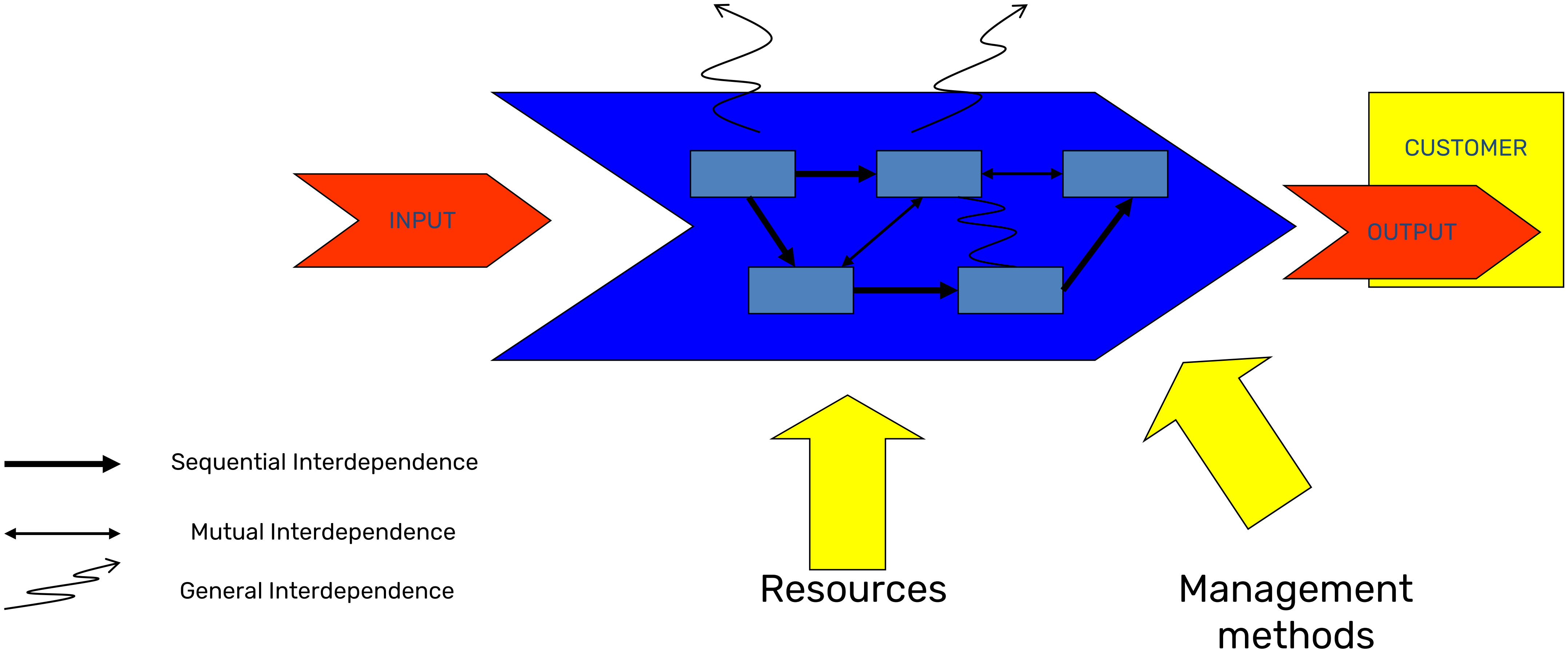
- Making an analysis of the possible modeling tools

- Getting to know and use ARIS and IDEF0

- Apply these tools to a case study

What is a business process

Set of organized and coordinated activities that use the available resources, to achieve an output with a value for the customer and to achieve the company's business target



Process elements

- **Activities**: aggregated operations (production or decision) that lead to the transformation of inputs into outputs through the use / consumption of resource.
- **Resources**: all the components necessary to carry out an activity. They can represent systems (machinery, plant, equipment, IT) or people.
- **Input**: physical factors or information acquired from outside or other business processes that are required to start the process
- **Output**: outcome of the process by which the customer (internal or external) assigns a specific value in its possession / use / exploitation.
- **Interdipendences**: logical and priority links between phases, activities and decisions
  - *Sequential*: the output of an asset is an input to a second
  - *Reciprocal:* the output of each activity is a required input for other activities
  - *General*: activities not linked by a logical point of view, but with bonds (eg. sharing the same resources).
- **Management Methods**: logical basis to coordinate activities, make decisions and to control the process progress.

Process Modeling

- A **model** is a simplified and reduced representation of reality. Simplified because reality is too complex to be exactly copied. Moreover much of the process complexity is irrelevant to a specific problem.

- **Process model** helps to clarify the steps (or phases) involved in a particular process.
- It is used to:
- understand the current processes
- clarify responsibilities
- identify process inefficiencies
- design new procedures considering the identified improvement (BPR)
- manage the company knowledge and training

Process Modeling

- What is a **model**?
- A representation of the real thing. It is built to a certain scale and certain level of detail. It is built for a purpose and to show a view point.

- Why should we use a **Business Modelling Method**?
- To have a consistent way of documenting and analysing a whole business.

- Why should we use a **Business Modelling Tool**?
- It encourages standardization and use of process vocabulary, improves quality, allows multiple view points, provides analysis tools and represents the starting point for development of software systems.

Process Modeling - Benefits

- The benefits of modeling are:
  - The cost of virtual experimentation is much lower than the cost of experimentation with a real system
  - Models allow for the simulated compression of time
  - Manipulating the model is much easier than manipulating the real system. The cost of mistakes are much lower in virtual experimentation
  - Modeling allows a manager to better deal with the uncertainty by introducing “what-ifs” and calculating the risks involved in specific actions
  - Mathematical models allow the analysis and comparison of a very large number of possible alternative solutions
  - Models enhance and reinforce learning and support training

How to produce a process map

1. Assemble and train the team
2. Identify the main objectives
3. Determine the boundaries of the process (where it starts and ends) and the level of detail desired
4. Consult with the experts: the people managing and working with the process
5. Brainstorm the major process tasks, and list them in order (sticky notes are often helpful here): identify activities, resources, decision points.
6. Generate an initial process map (also called a flowchart)
7. Draw the formal flowchart using standard symbols for process mapping
8. Check the formal flowchart for accuracy by all relevant personnel

Modeling tools

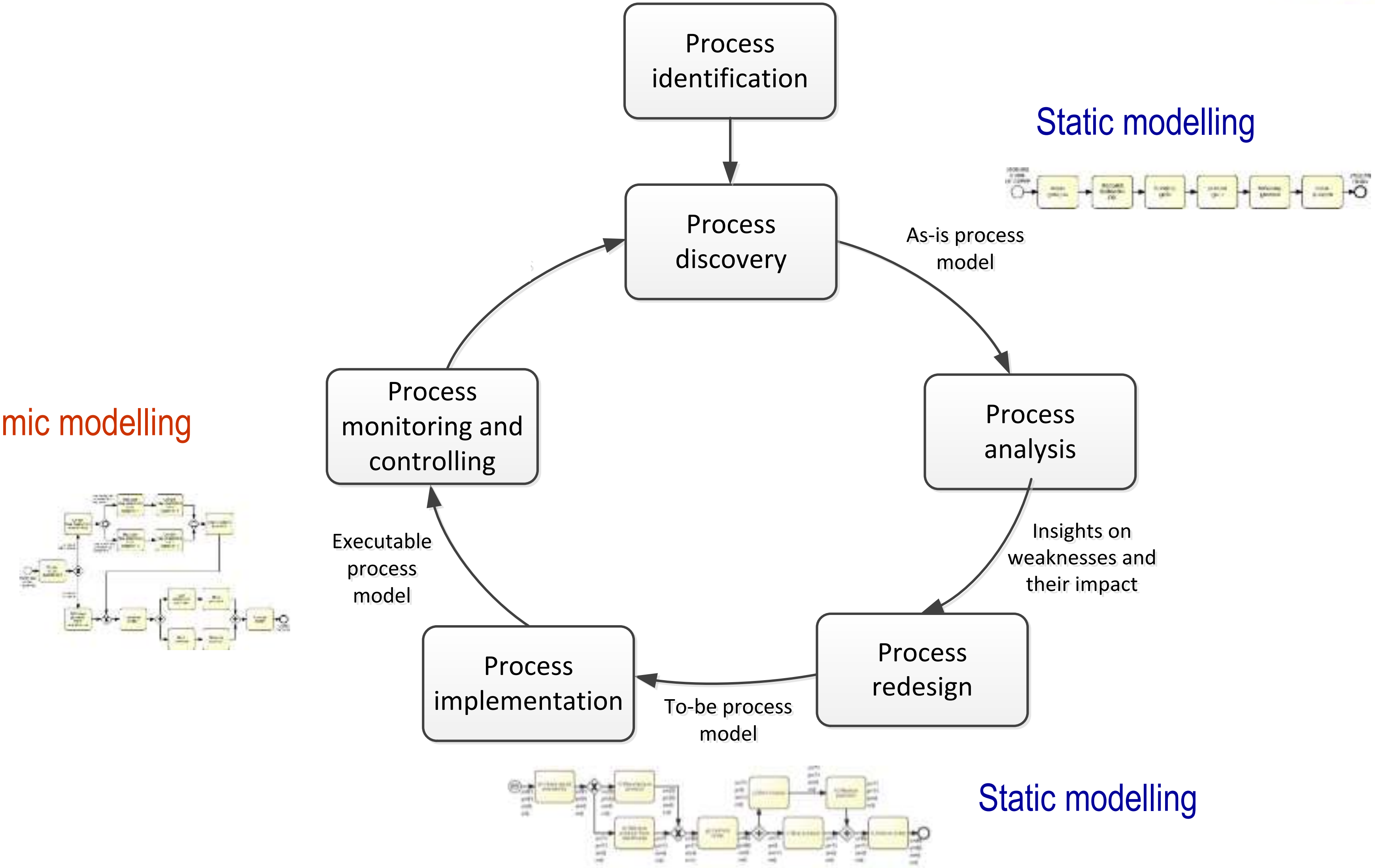
There are many tools, but be careful:

- A tool is not only a software is more correct to define it as methodology
- Different methodologies highlight different features of the process and support different analysis
- The choice of methodology depends on the context and objectives and not vice versa

Methodologies

	FUNCTIONS	TYPICAL METHODOLOGIES
STATIC TECHNIQUES OF BPM	• Show processes as interrelated activities • Show the flow of entities (control, information, materials, etc..) through activities • Represent structures that do not provide time dependency	• Process Flowcharting • Data Flow Diagramming • IDEF0 (Function Modelling) • IDEF 1 and 1x (Data/Information Modelling) • IDEF3 (Process Modeling) • Basic Petri Nets • RAD (Role Activity Diagramming)
DYNAMIC TECHNIQUES OF BPM	• Show process relevant information in a chronological, time dependent manner • Support analysis and experimentation with alternative process structures • Support communication through animation and visualization • Support user interaction with the process model	• Discrete-event Business Process Simulation • Systems Dynamics/continuous simulation • Timed Petri Nets • Hybrid simulation

Why and when to use process modelling


Process identification

Process discovery

As-is process model

Static modelling

Process analysis

Insights on weaknesses and their impact

Process redesign

To-be process model

Static modelling

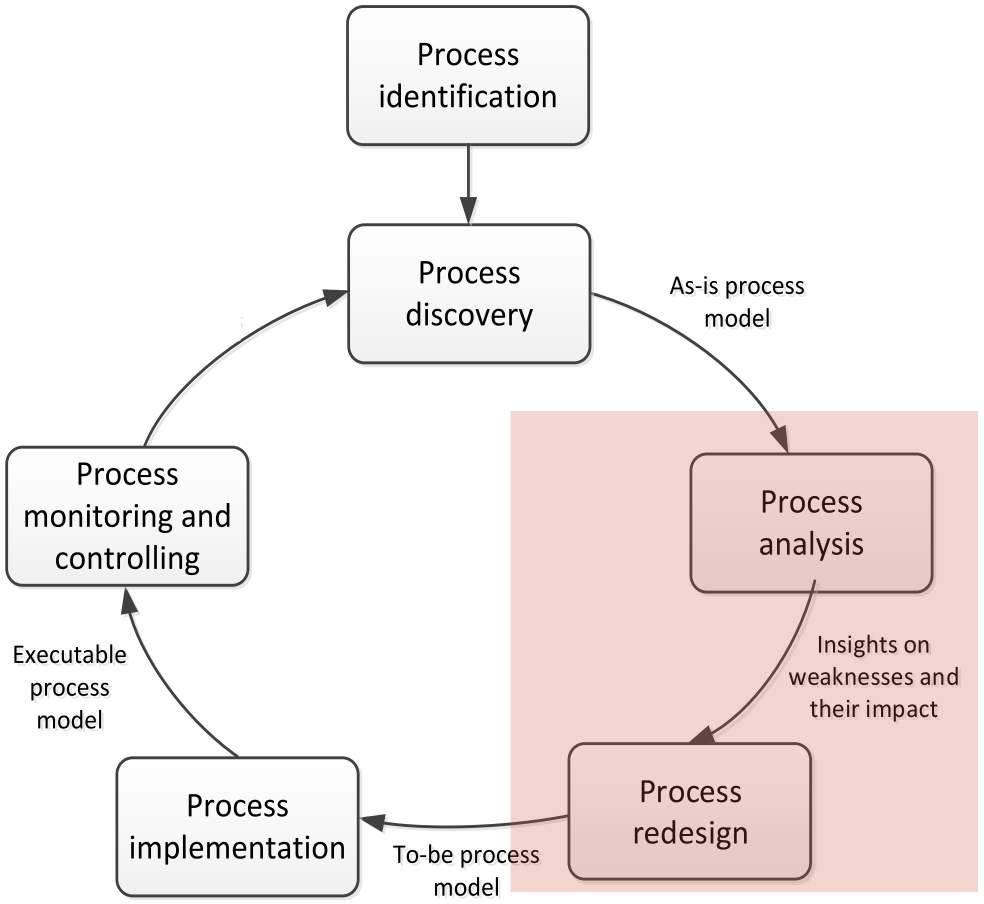
Process implementation

Executable process model

Process monitoring and controlling

Dynamic modelling

Process improvement



1. Identify a system's owner
2. Ensure that the correct problem is being addressed
3. Model the process
4. Measure the process
5. Improve the process

Measure the process

- **Capacity of a process**: maximum possible amount of output (goods or services) that a process or resource can produce or transform.
- **Capacity utilization**: proportion of capacity actually being used. Measured as actual output/maximum possible output
- **Throughput time**: average time a unit spends in the process. Includes both processing time and waiting time and is determined by the critical (longest) path through the process
- **Throughput rate**: average number of units that can be processed per unit of time
- **Service time or cycle time**: time to process one unit. Cycle time of a process is equal to the longest task cycle time in that process

Measure the process

- **Idle or wait time**: time a unit spends waiting to be processed
- **Arrival rate**: rate at which units arrive at the process
- **Work-in-process (WIP), things-in-process (TIP), patients-in-process (PIP), or inventory**: total number of units in the process
- **Setup time**: amount of time spent getting ready to process the next unit
- **Value-added time**: time a unit spends in the process where value is actually being added to the unit
- **Non-value-added time**: time a unit spends in the process where no value is being added. Wait time is non-value-added time
- **Number of defects or errors**

Process improvement

- Basic process re-design techniques:
- Eliminate non-value-added activities
- Eliminate duplicate activities (ex. asking customer repeatedly for their contact information)
- Combine related activities (ex: reducing movement among departments)
- Process in parallel (ex: raw material preparation)
- Balance workloads
- Develop alternative process flow paths and contingency plans (to respond well to unexpected event, ex: have alternative paths for when emergency occurs, a supplier is late, a failure happen in a machine)

Process improvement

- Establish the critical path (i.e. longest length of time it will take to carry out the process)
- Embed information feedback and real-time control (ex: machine status can be continuously monitored and if it reaches a certain level contingency plan can be initiated)
- Ensure quality at the source
- Match capacity to demand
- Let the customer do the work (ex: online tools to book appointment)
- Use technology (ex: using devices, tablet and smartphone, to improve productivity)
- Apply the theory of constraints (identify bottleneck and synchronize activities with it)
- Identify best practices and replicate

IDEF 0

What is IDEF?

- ***Definition*****:** IDEF is the common name referring to a family of modeling languages used to describe operations in an enterprise.
- ***Objective*****:** IDEF is used for modeling activities necessary to support system analysis, design, improvement or integration.
- Originally, IDEF was developed to enhance communication among people trying to understand the system. *Now*, IDEF is being used for **documentation, understanding, design, analysis, planning, and integration**.

IDEF history

- During the 1970s, the U.S. Air Force Program for Integrated Computer Aided Manufacturing (ICAM) sought to increase manufacturing productivity through systematic application of computer technology. The ICAM program identified the need for better analysis and communication techniques for people involved in improving manufacturing productivity.
- As a result, the ICAM program developed a series of techniques known as the IDEF (ICAM DEFinition, renamed in 1999 as Integration DEFinition) techniques which included the following:
- IDEF0, used to produce a "function model"
- IDEF1, used to produce an "information model"
- IDEF2, used to produce a "dynamic model"

IDEF history

- In 1983, the U.S. Air Force Integrated Information Support System program enhanced the IDEF1 information modeling technique to form IDEF1X (IDEF1 Extended), a semantic data modeling technique.

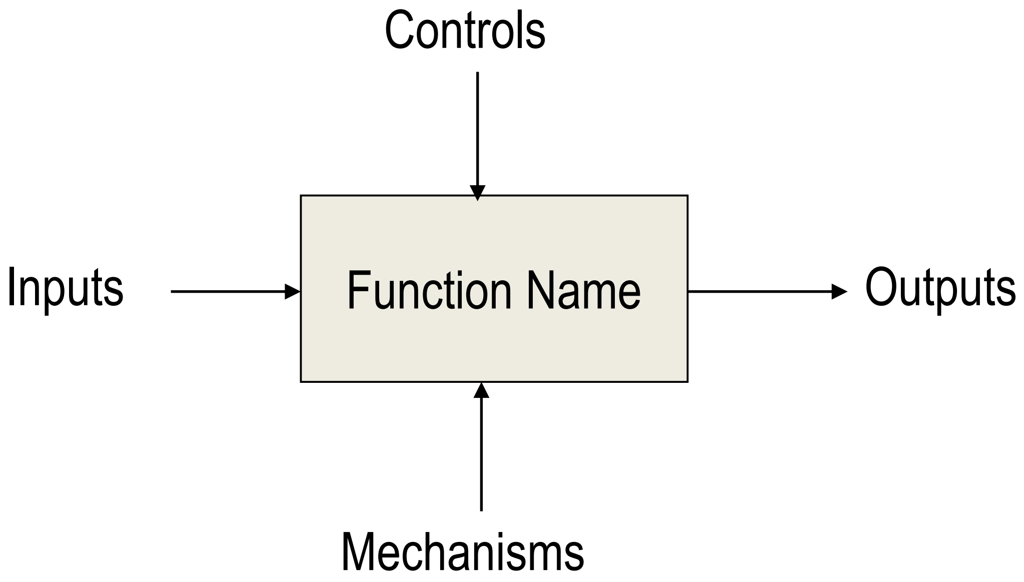
- Currently, IDEF0 and IDEF1X techniques are widely used in the government, industrial and commercial sectors, supporting modeling efforts for a wide range of enterprises and application domains.

IDEF Family

- IDEF Family of Methods:
  - IDEF0: for Function Modeling → structurally represents the functions, activities or processes (purpose: description)
  - IDEF1: for Information Modeling → represents the structure and semantics of information (purpose: description)
  - IDEF1x: for Data Modeling → designs relational DB (purpose: design)
  - IDEF3: for Process Modeling (purpose: description)
  - IDEF4: for Object-Oriented Design (purpose: design)
  - IDEF5: for Ontology Description Capture (purpose: description)

IDEF0- Function Modeling Method

- It is based on the «box» and «arrow» approach.

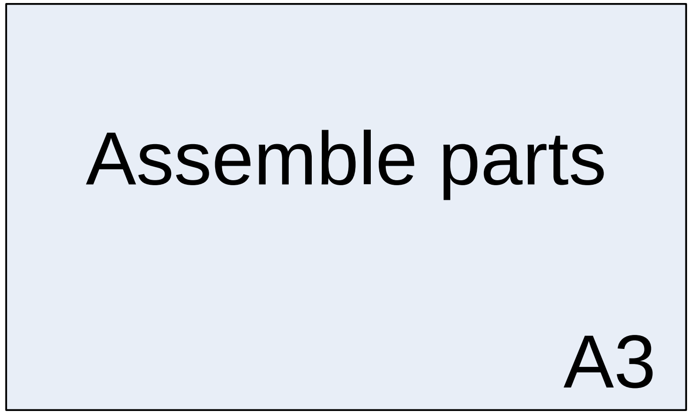


Arrows

- **Input**: Describe resources or data that are needed to perform the function and are transformed by the function into outputs.
- **Control**: Describe the conditions, rules, procedures, or circumstances that govern the execution of the function. An arrow is a control unless it obviously serves only as input. *Each function should have at least one control arrow*. Most of controls are in the form of data.
- **Output**: The data or objects produced when the function is performed.
- **Mechanism**: Define the supporting mechanisms that carry out the function. A mechanism may be a person, an organizational unit, a physical device, or a computer program.

Boxes

- A box provides a description of what happens in a designated function.
- It is important to use:
  - Solid lines
  - Verb or verb phrase
  - Box number



Box and Arrow Syntax Rules

Boxes

- Boxes shall be sufficient in size to insert box name.
- Boxes shall be rectangular in shape, with square corners.
- Boxes shall be drawn with solid lines.

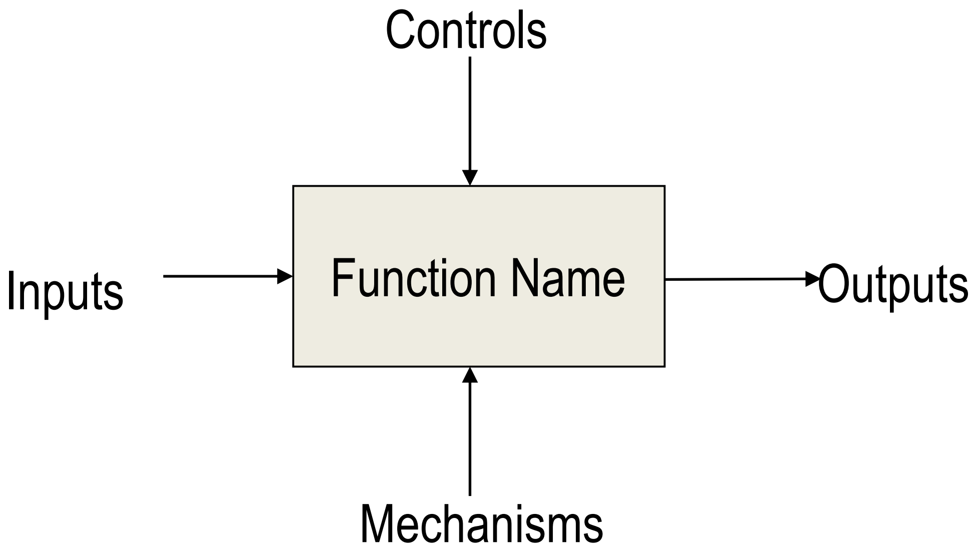
- Arrows
  - Arrows that bend shall be curved using only 90 degree arcs.
  - Arrows shall be drawn in solid line segments.
  - Arrows shall be drawn vertically or horizontally, not diagonally.
  - Arrow ends shall touch the outer perimeter of the function box and shall not cross into the box.
  - Arrows shall attach at box sides, not at corners.

More Box and Arrow Syntax Rules

- A box shall be named with an **active verb or verb phrase**.
- Each side of a function box shall have a standard box/arrow relationship:
  - Input arrows shall interface with the left side of a box.
  - Control arrows shall interface with the top side of a box.
  - Output arrows shall interface with the right side of the box.
  - Mechanism arrows shall point upward and shall connect to the bottom side of the box.
- Arrow segments shall be labeled with a **noun or noun phrase** unless a single arrow label clearly applies to the arrow as a whole.
- Arrow labels shall not consist solely of any of the following terms: function, input, control, output, mechanism

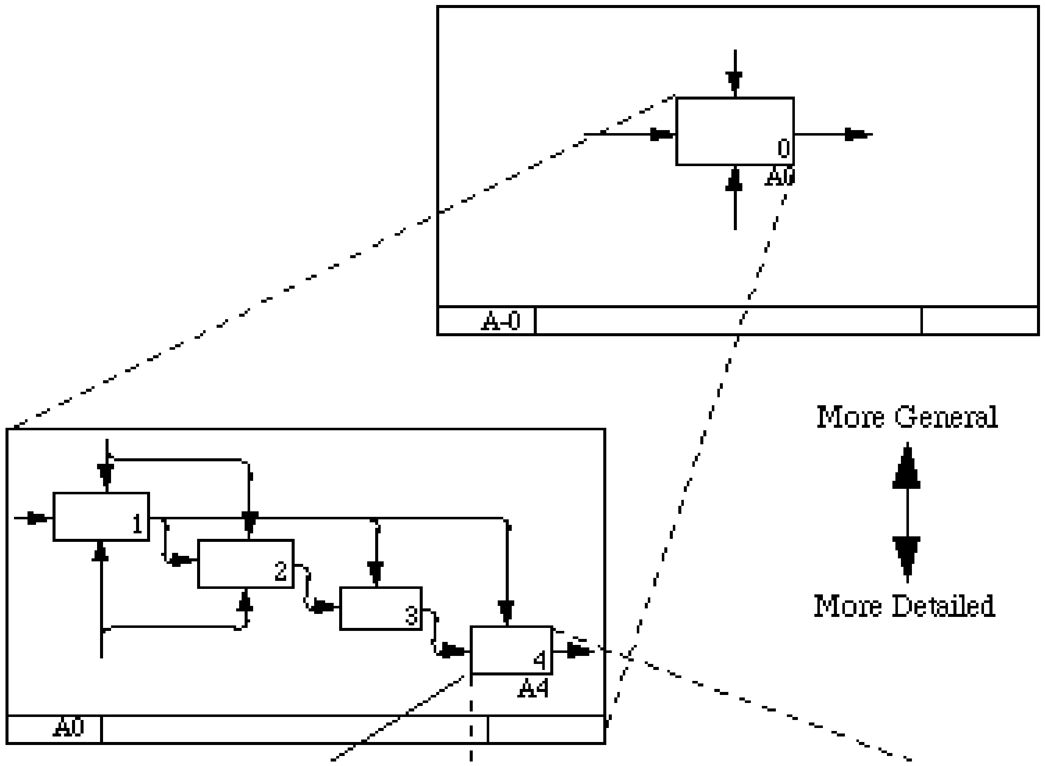
IDEF0- Context Diagram

Context Diagram (A-0 diagram): is a model of the function at the highest level of inputs, controls, outputs, and mechanisms.



- It establishes the model focus.
- Since a single box represents the whole subject, the descriptive name is general.
- The name of the arrows are general since they represent the complete set of external interfaces to the subject.

IDEF0- Decomposition Structure



- The single function represented on the top-level context diagram may be decomposed into its major sub-functions by creating its child diagram.
- In turn, each of these sub-functions may be decomposed, each creating another, lower-level child diagram

IDEF0 - Decomposition Structure

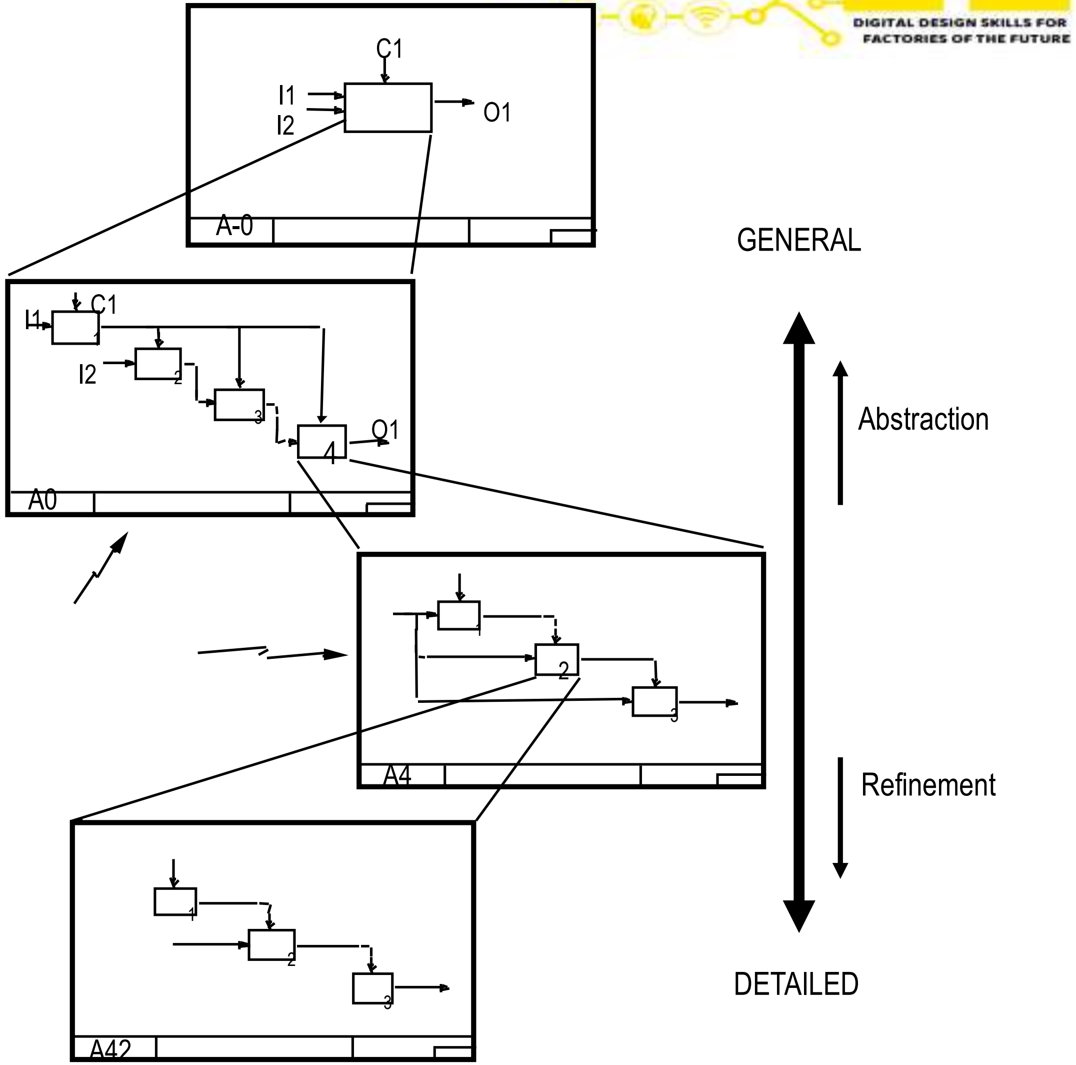
- **Parent Diagram:** A parent diagram is one that contains one or more parent boxes.
- **Child Diagram**: diagram that provides more detail about a function. On a given diagram, some of the functions, none of the functions or all of the functions may be decomposed.

- A diagram may be both a parent diagram (containing parent boxes) and a child diagram (detailing its own parent box)

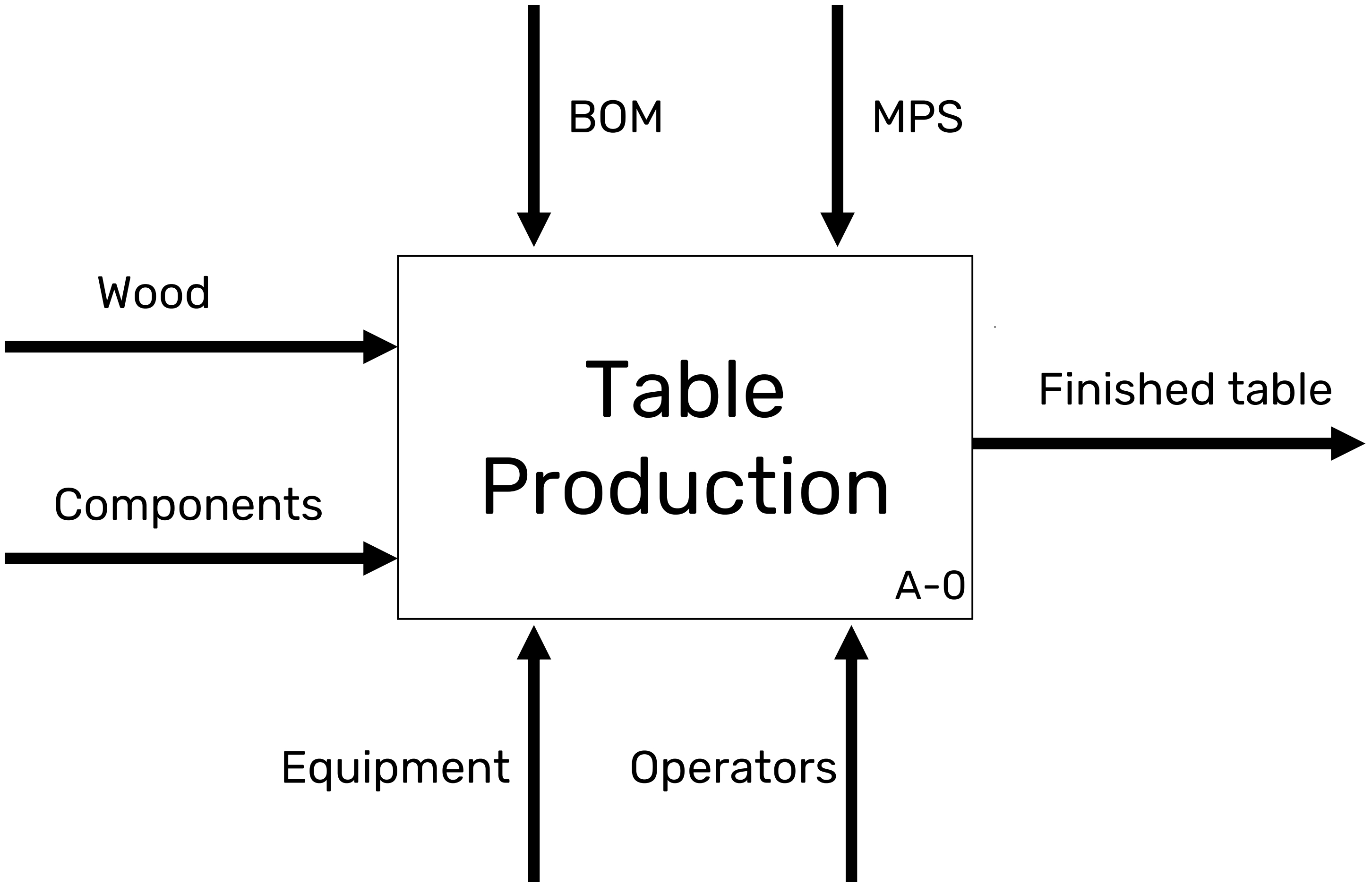
- In each diagram there must be **minimum 3 boxes and maximum 6 boxes**

IDEF0 Hierarchical Structure

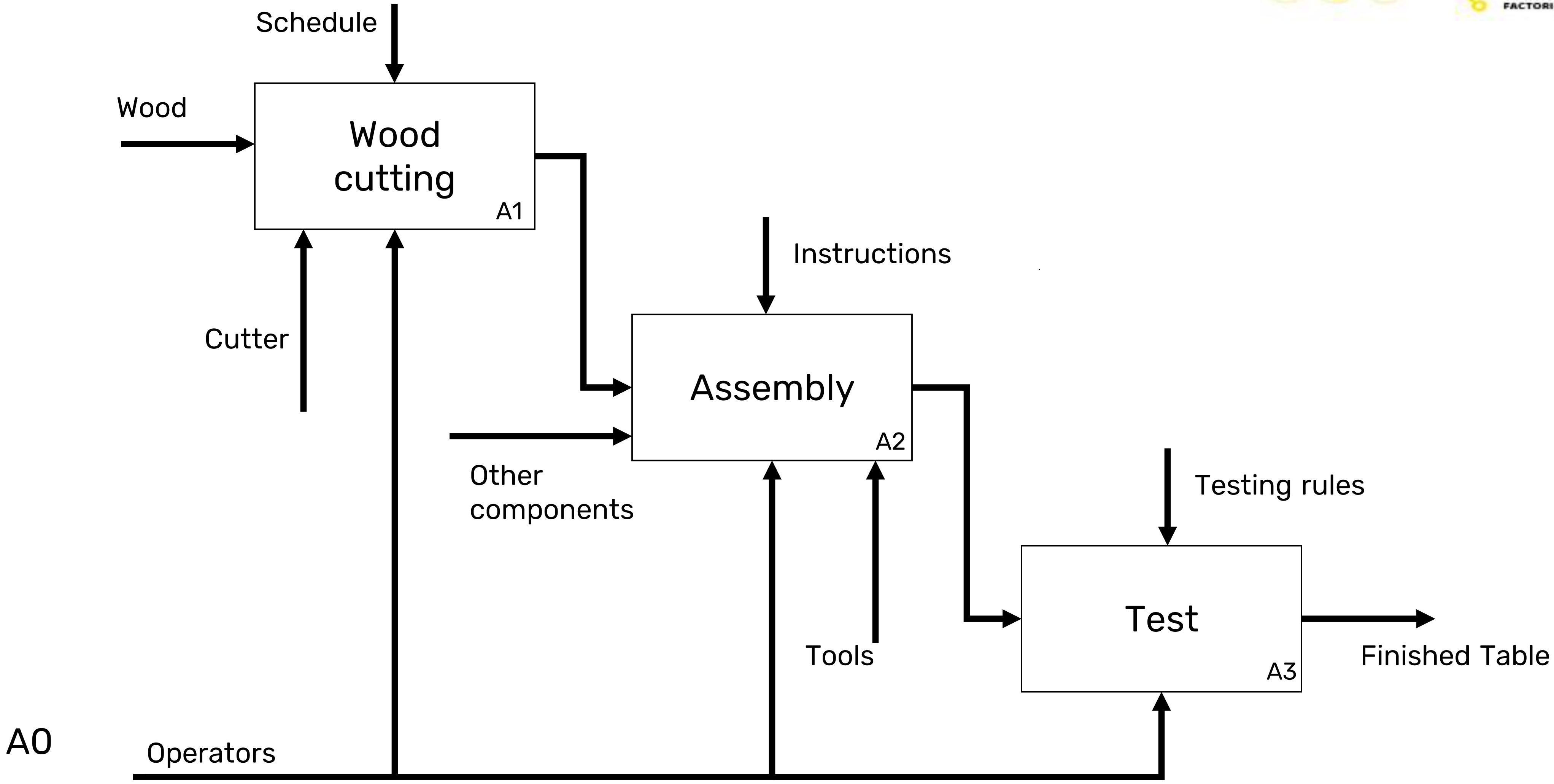
- The diagram A0 is the "parent" of the diagram A4 and it is the "child" of A-0 diagram.



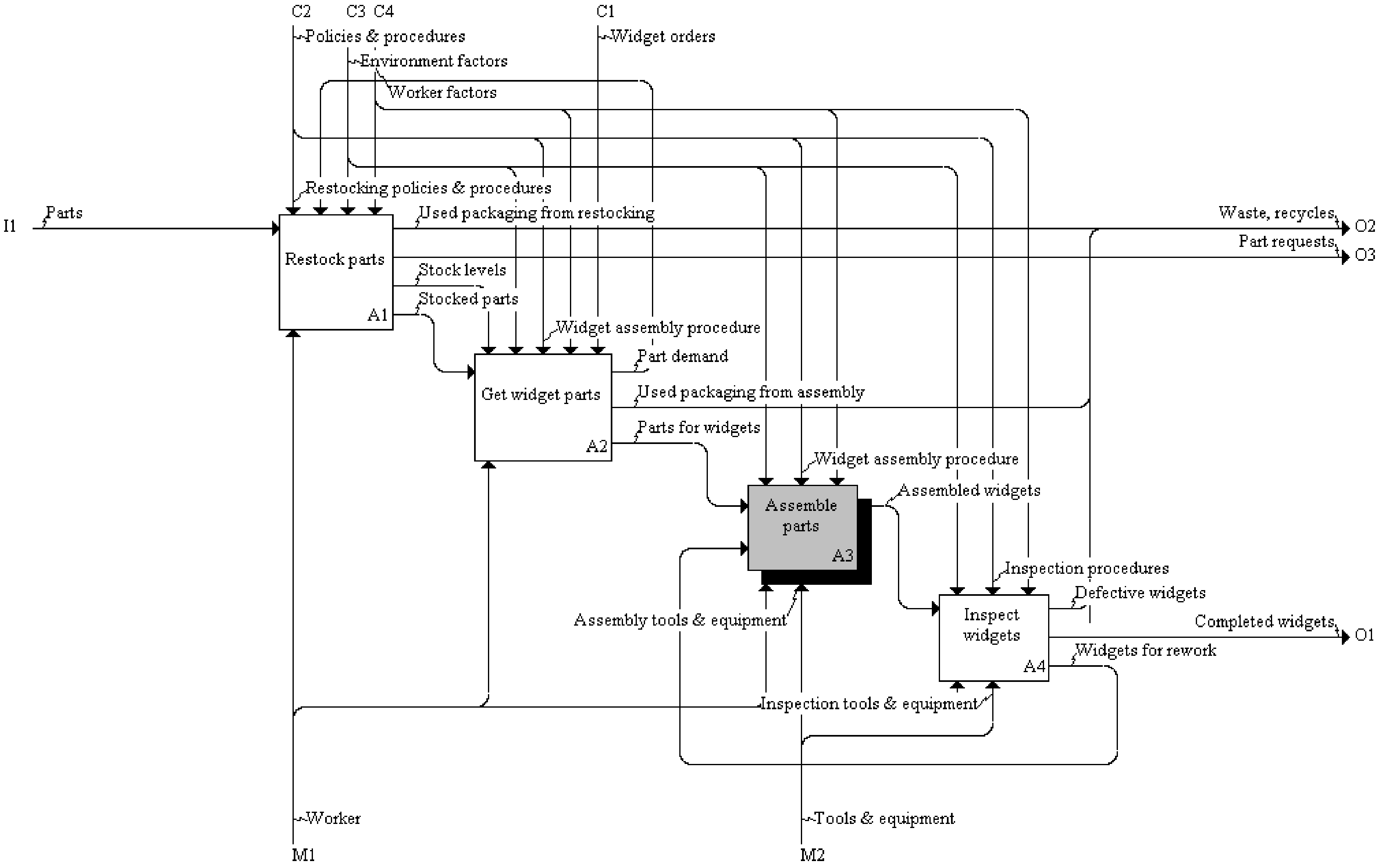
IDEF0 – Example



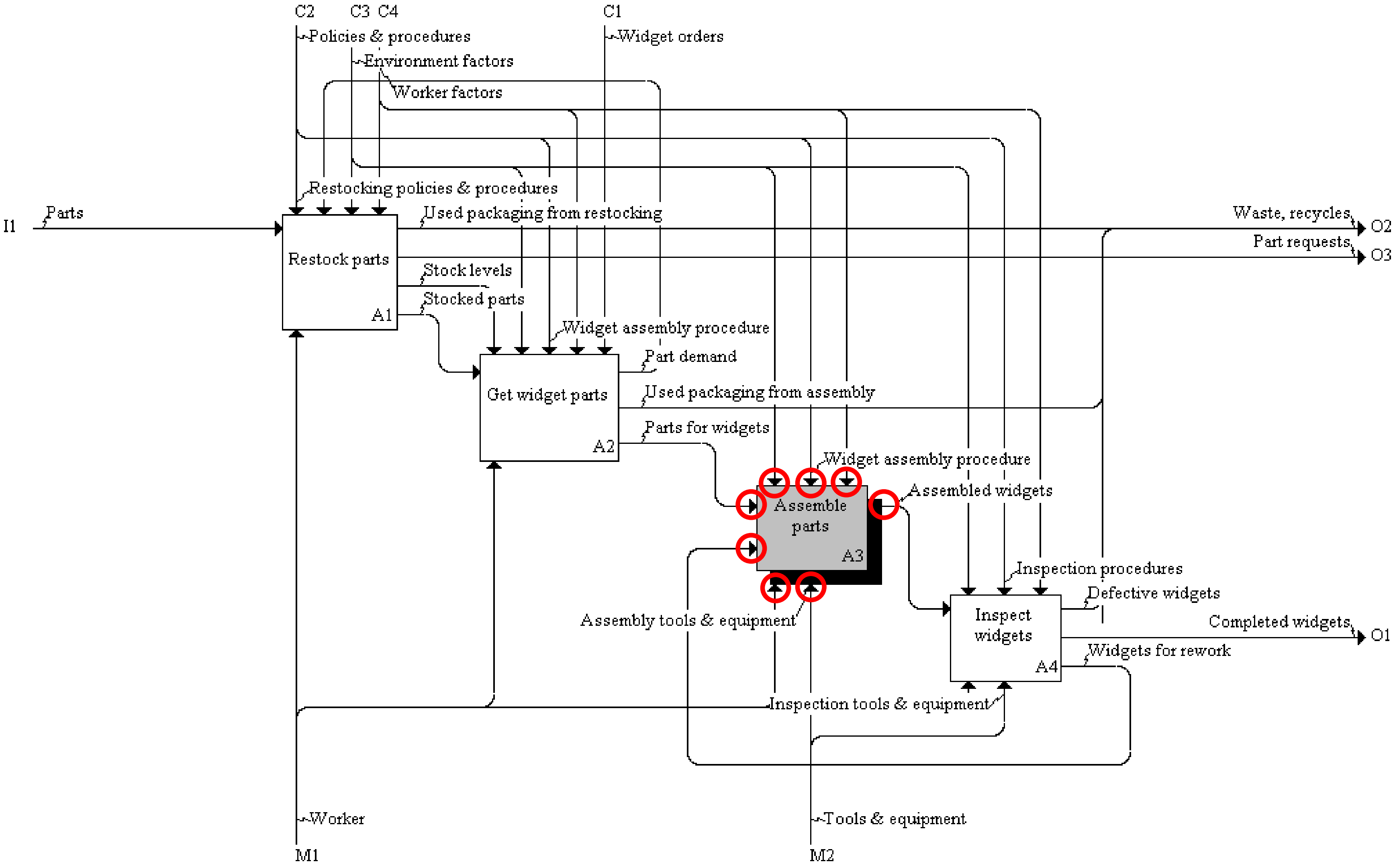
IDEF0 – Example



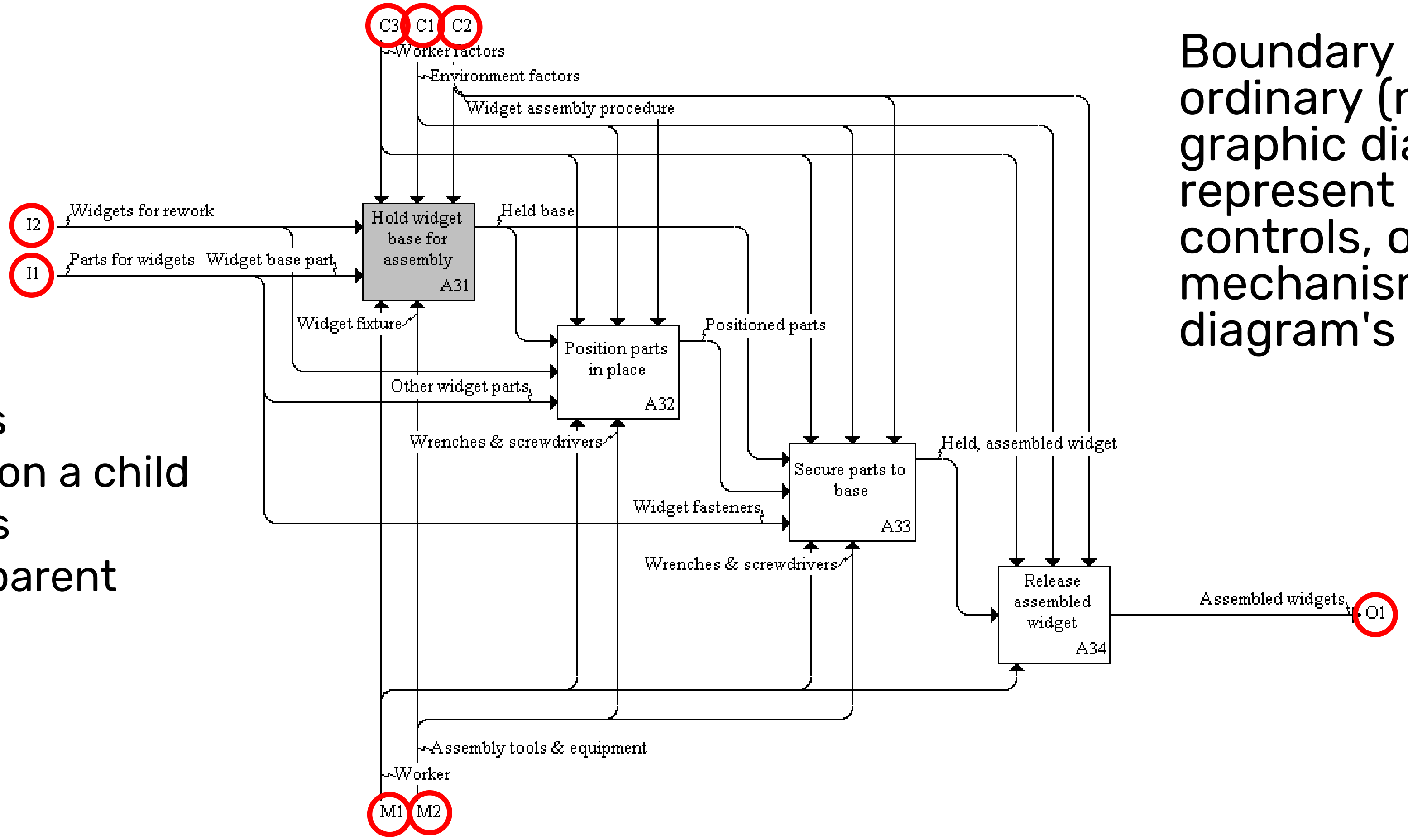
IDEF0 Example



Inter-Box Connections (arrows for child diagram)



Boundary Arrows
Arrows from parent box on parent diagram



Boundary arrows on an ordinary (non-context) graphic diagram represent the inputs, controls, outputs, or mechanisms of the diagram's parent box.

ICOM code relates boundary arrows on a child diagram to arrows connected to its parent box

IDEF0- Strengths

- The model has proven effective in detailing the system activities for function modeling.
- IDEF0 models provide an abstraction away from timing, sequencing and decision logic. However, it is easy to use IDEF0 for modeling activity sequences whenever needed.
- Provides a concise description of systems, by using the Inputs, Controls, Output, Mechanism.
- The hierarchical nature of IDEF0 allows the system to be easily refined into greater detail until the model is as descriptive as necessary for the decision-making task.

IDEF0- Weaknesses

- IDEF models might be so concise that only the domain experts can understand.
- IDEF models are sometimes misinterpreted as representing a sequence of activities.
- The abstraction away from timing, sequencing and decision logic leads to comprehension difficulties for the people outside the domain.

ARIS

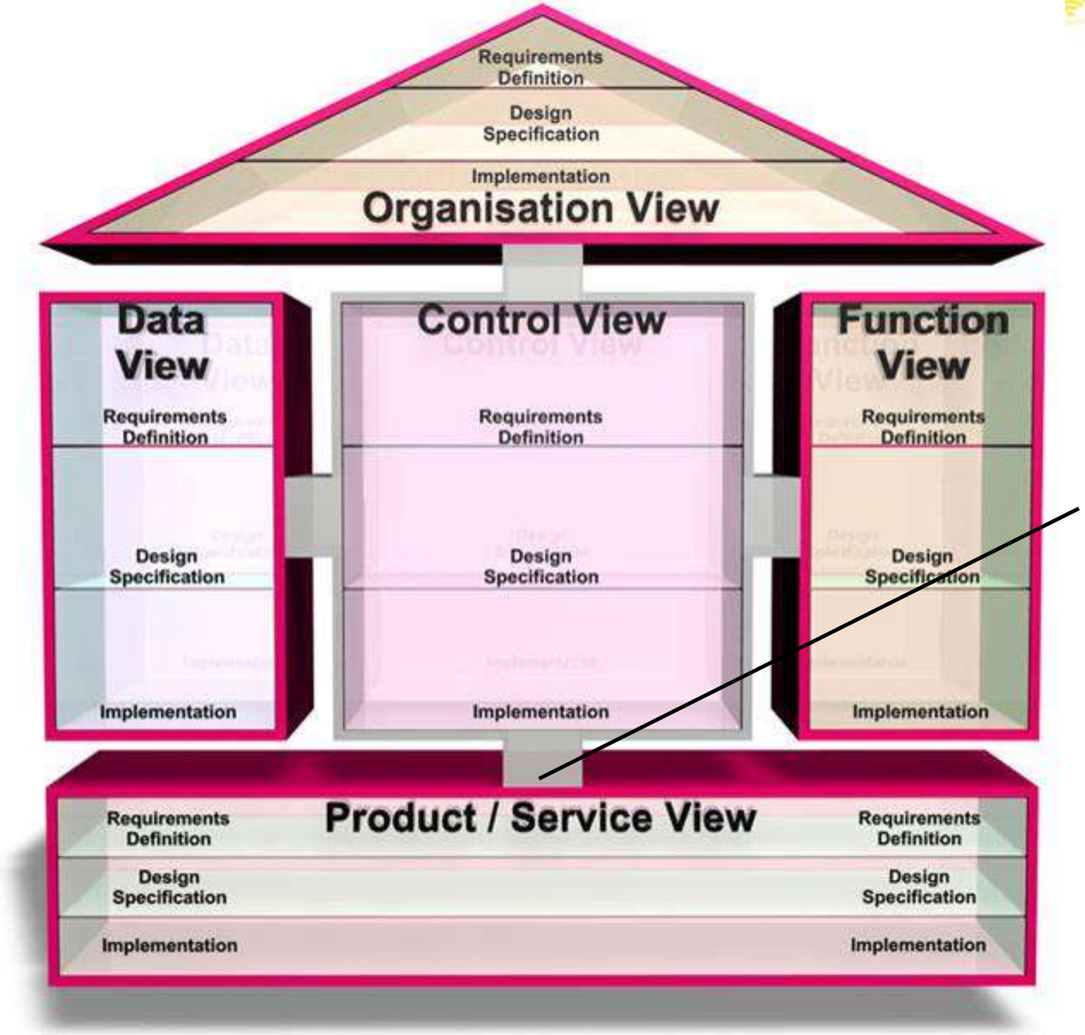
Introducing ARIS

- **ARchitecture** of **Integrated** Information **Systems**, developed by prof. Scheer in 1990s.

- ARIS is not a tool, but a concept. It can model:
  - processes
  - data
  - organisations
  - information
  - product
  - knowledge
  - business objectives

The Aris House

- ARIS includes:
  - an architecture for describing business processes
  - a set of modelling methods
  - the foundation of the ARIS Toolset software system
  - a concept for computer-aided business process management



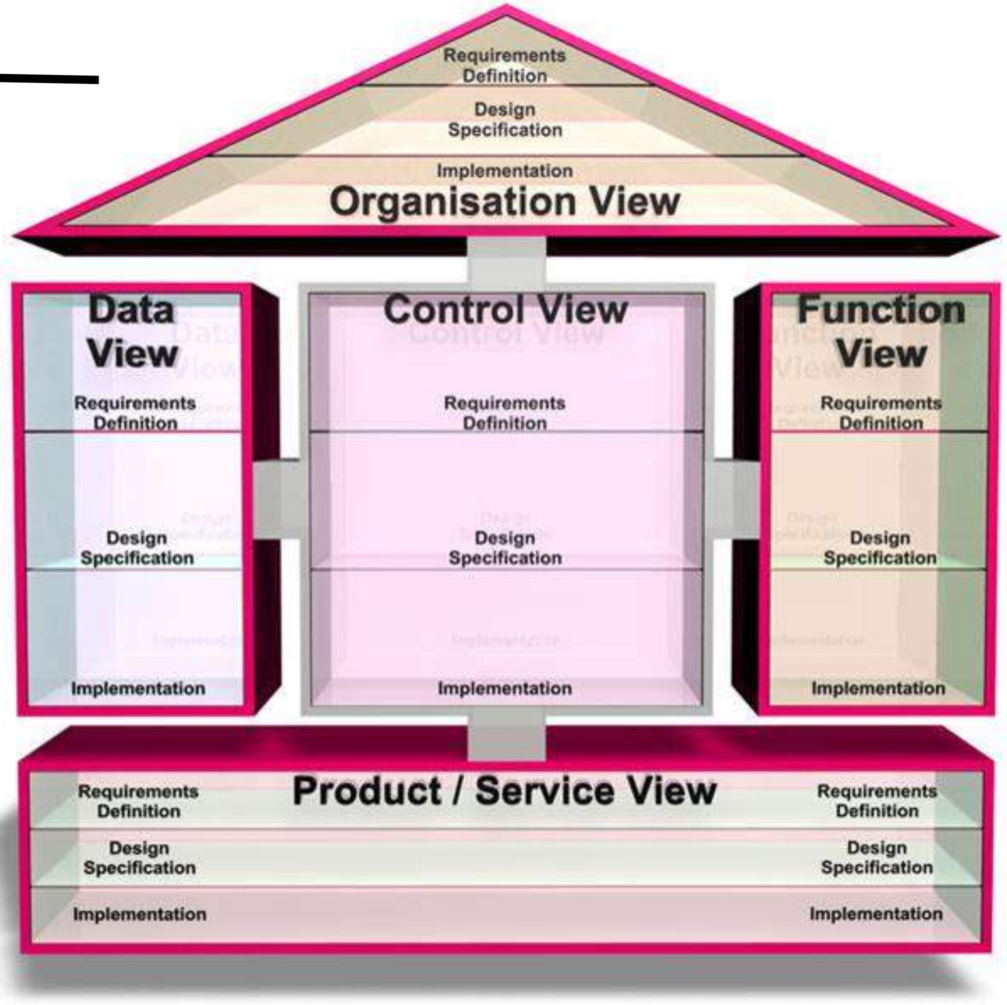
Relationship (Interaction)

The Aris House

It describes the organizational elements and their relationships, human resources, operating resources, and computer hardware.
Modeling method like: Organigram

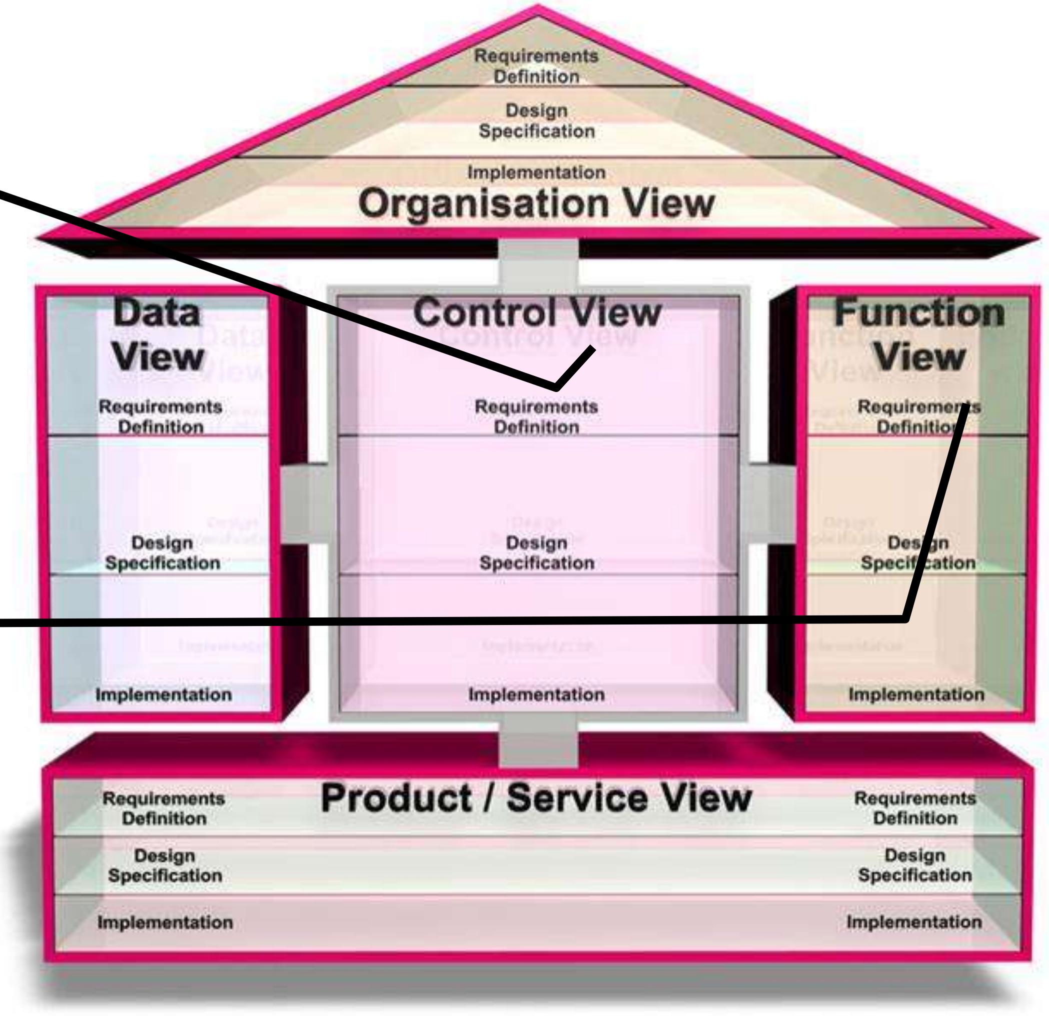
It describes the information objects and their attributes, the relationships between information objects.
Modeling method like: ERM (Entity Relationship Model) or UML (Unified Modeling Language)

It represents all material and non material inputs and outputs that are brought to, or performed by, the business process.
Modeling method like: product diagram



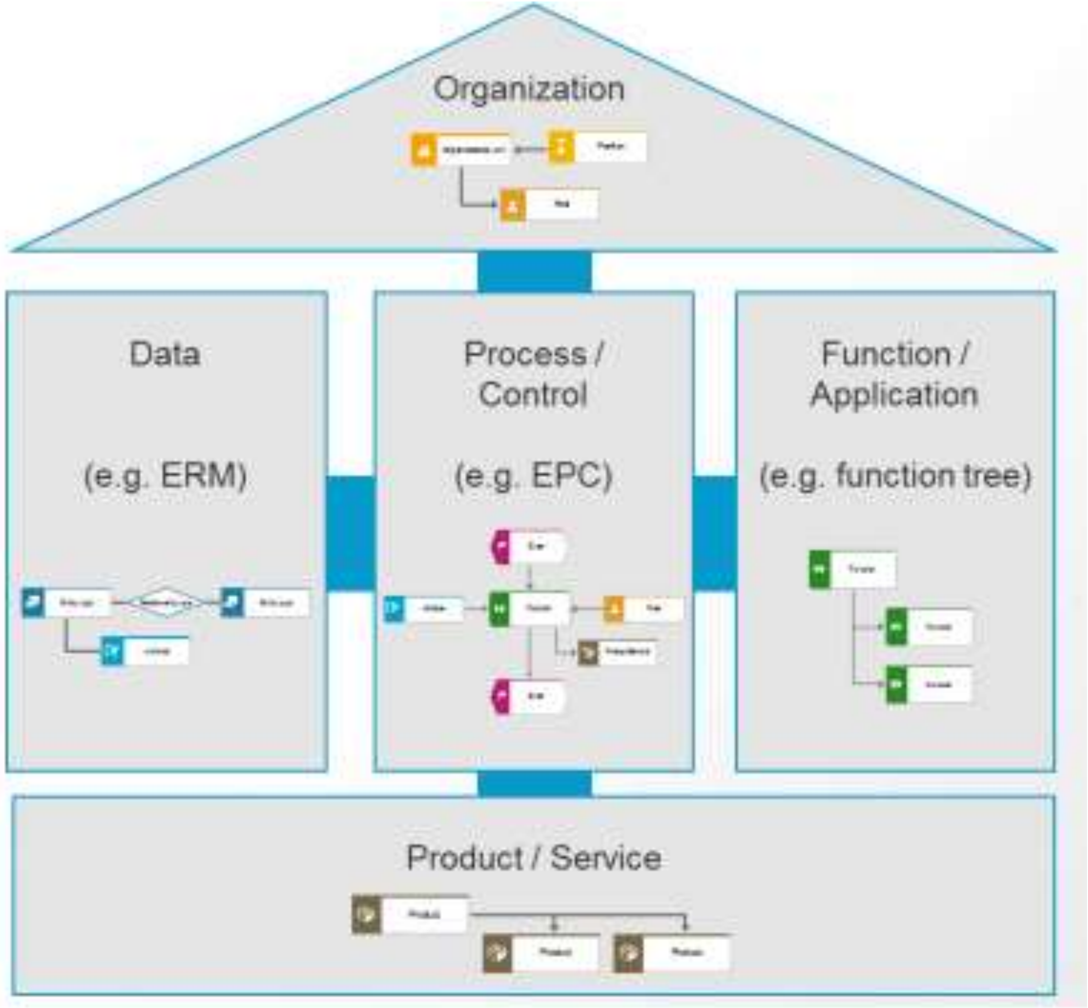
The Aris House

It displays the static connections between the objects of the data, function, performance and organization views, and the dynamic, chronological process flows. *Modeling method like: EPC (Event-Driven Process Chain) or eEPC (Extended Event-Driven Process Chain)*



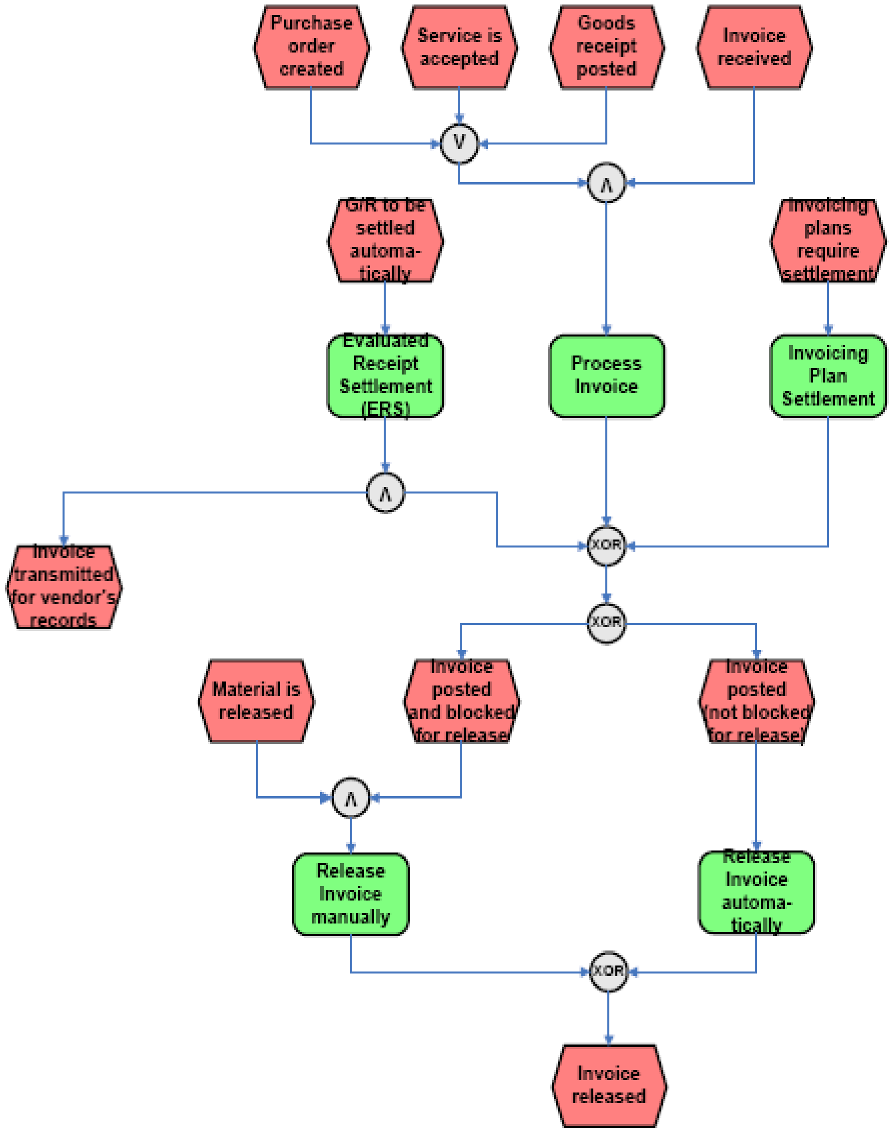
It describes the transactions that transform performance and the static relationships between them. Application systems are also included in the function view, because they determine the computer-supported processing rules for activities.
Modeling method: Function tree

Event Process Chain (EPC)

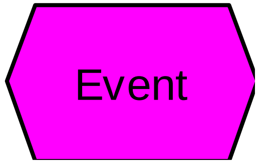


EPC (Example)

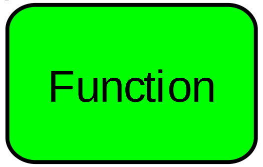
Note the multiple start and end events


Purchase order created
Service is accepted
Goods receipt posted
Invoice received
V
Λ
G/R to be settled automa-tically
Invoicing plans require settlement
Evaluated Receipt Settlement (ERS)
Process Invoice
Invoicing Plan Settlement
Λ
XOR
Invoice transmitted for vendor's records
XOR
Material is released
Invoice posted and blocked for release
Invoice posted (not blocked for release)
Λ
Release Invoice manually
Release Invoice automa-tically
XOR
Invoice released

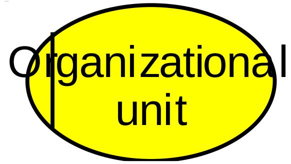
Basic elements of EPC



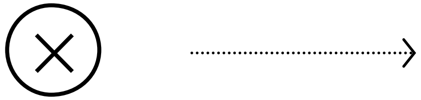
Event: start points and end points of processes. They can represent midway positions in a process. Events trigger functions or are the result of them



Activity: they model the elaborative units of the process. They are activated by events.

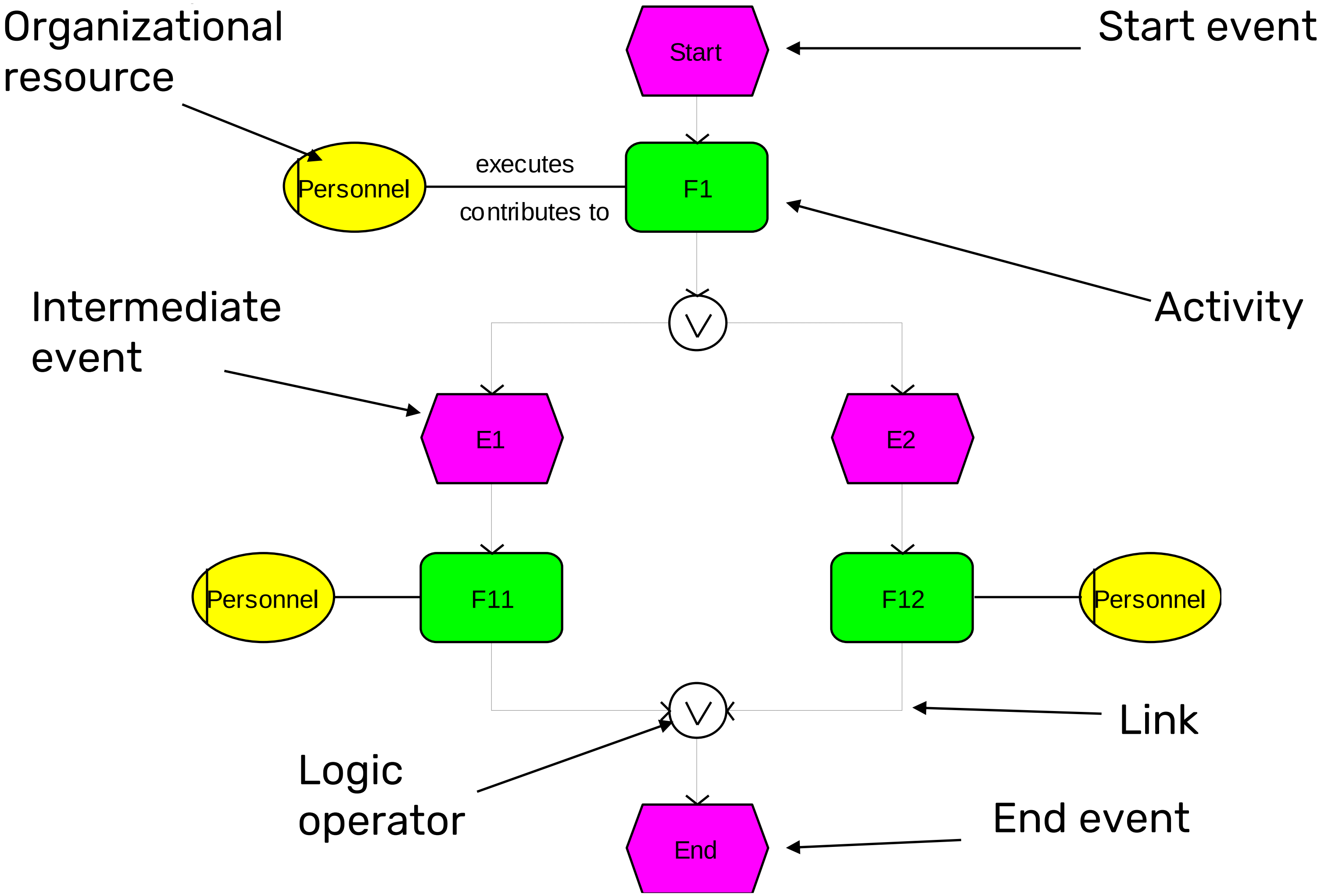


Organizational resource: they model the human resources available for the process

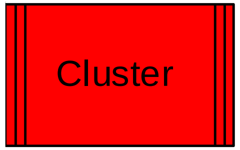


Logic connectors and links: they model the relation between events, activities and organizational resources

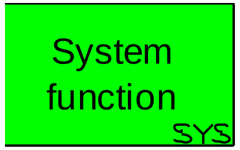
Event Process Chain – basic elements



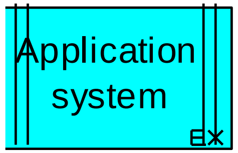
Basic elements of EPC



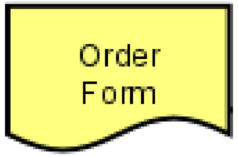
Cluster: It models “cluster” of information and data (eg: invoice, order…), which is not specified the structure.



System function: It represents activities automatically developed, without organizational resources.



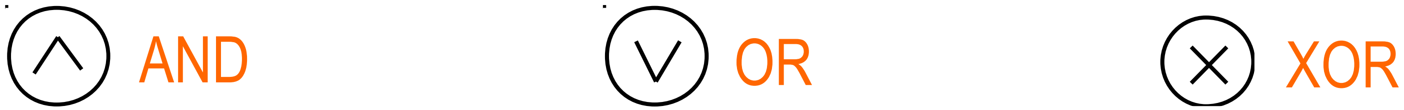
Application system: It represents IT resources to support the process.



Document (paper based)

Representation of process flow

- Logic operators have two main functionalities:
  - **Ramification** of process.
  - **Junction** of process flow



AND

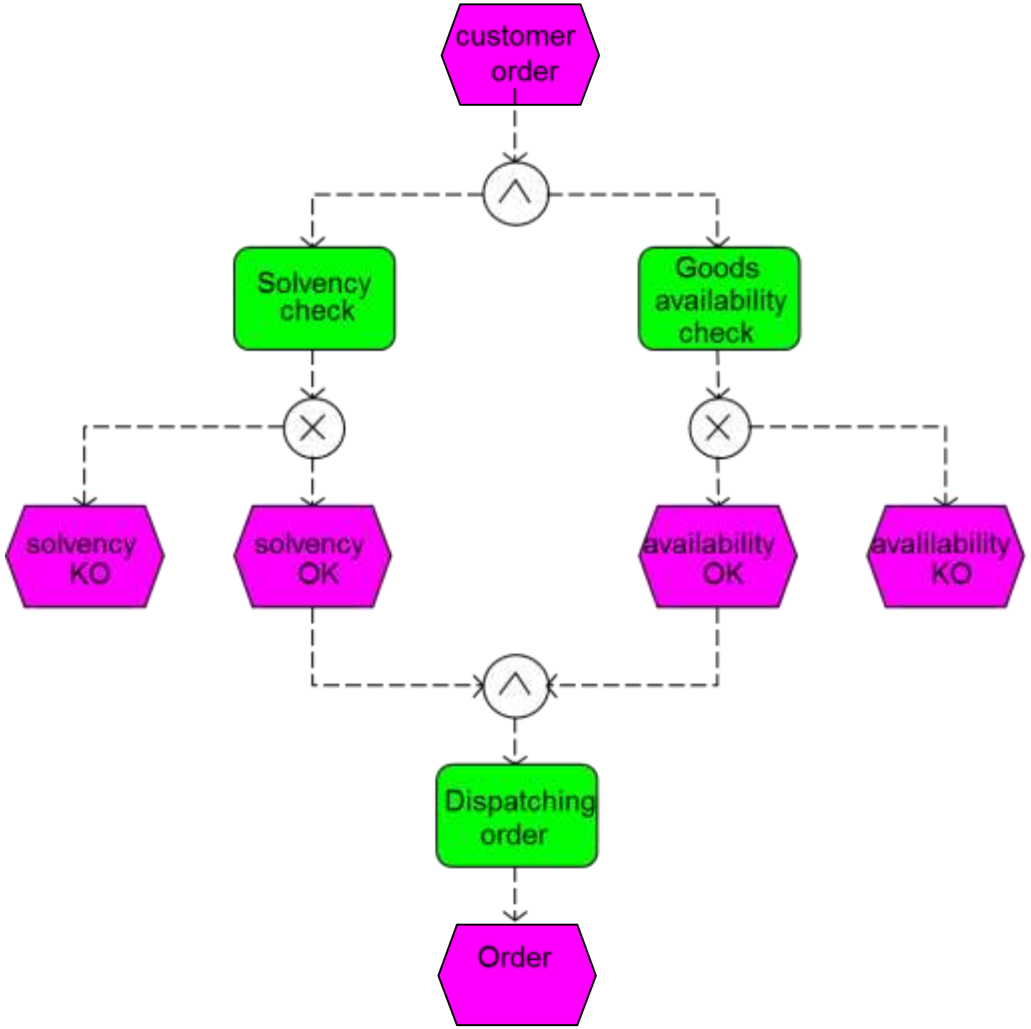
OR

XOR

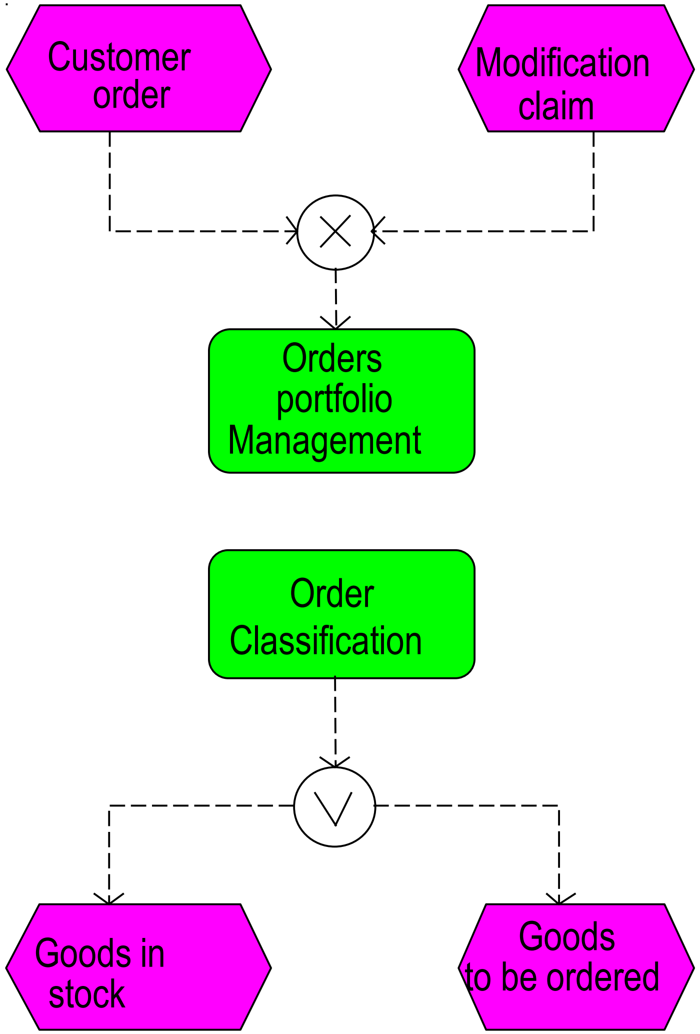
Logic operators

	Following a function	Preceding a function
OR	One or many possible paths will be followed as a result of the decision	Anyone Event, or combination of Events, will trigger the Function
XOR	One, but only one, of the possible paths will be followed	One, but only one, of the possible Events will be the trigger
AND	Process flow splits into two or more parallel paths	All Events must occur in order to trigger the following Function

Representation of a process flow: An example



Representation of a process flow: An example



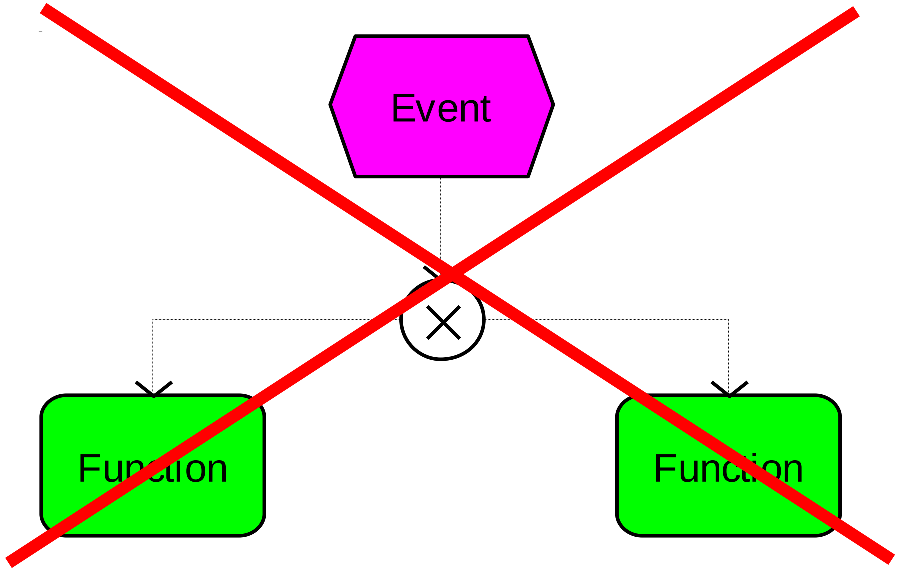
General rules to remember

- Each process begins and ends with one or more events.
- When the activity is identified, it is necessary to identify immediately the person who performs it.
- Any activity should be followed by an event.
- Avoid cycles.

- Beware of logical operators!

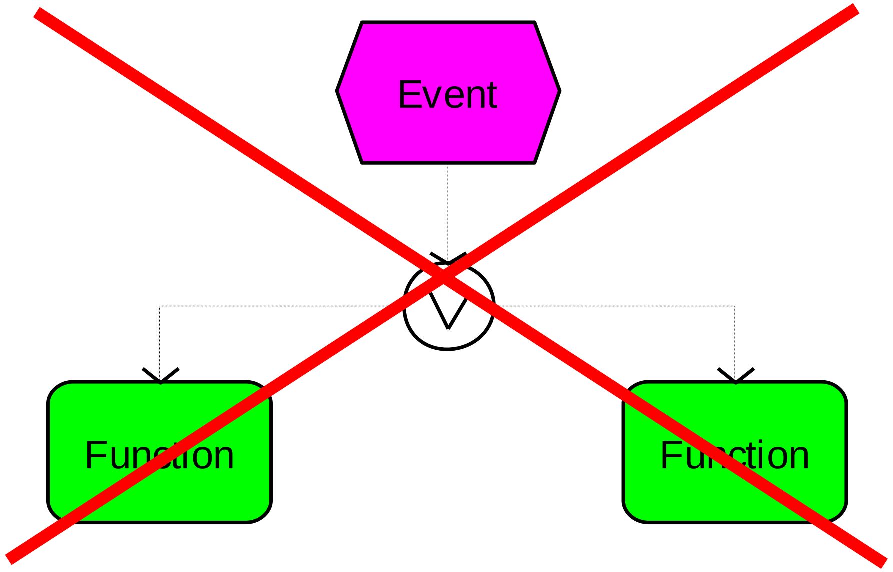
Forbidden connections

Events DON'T have decision-making power

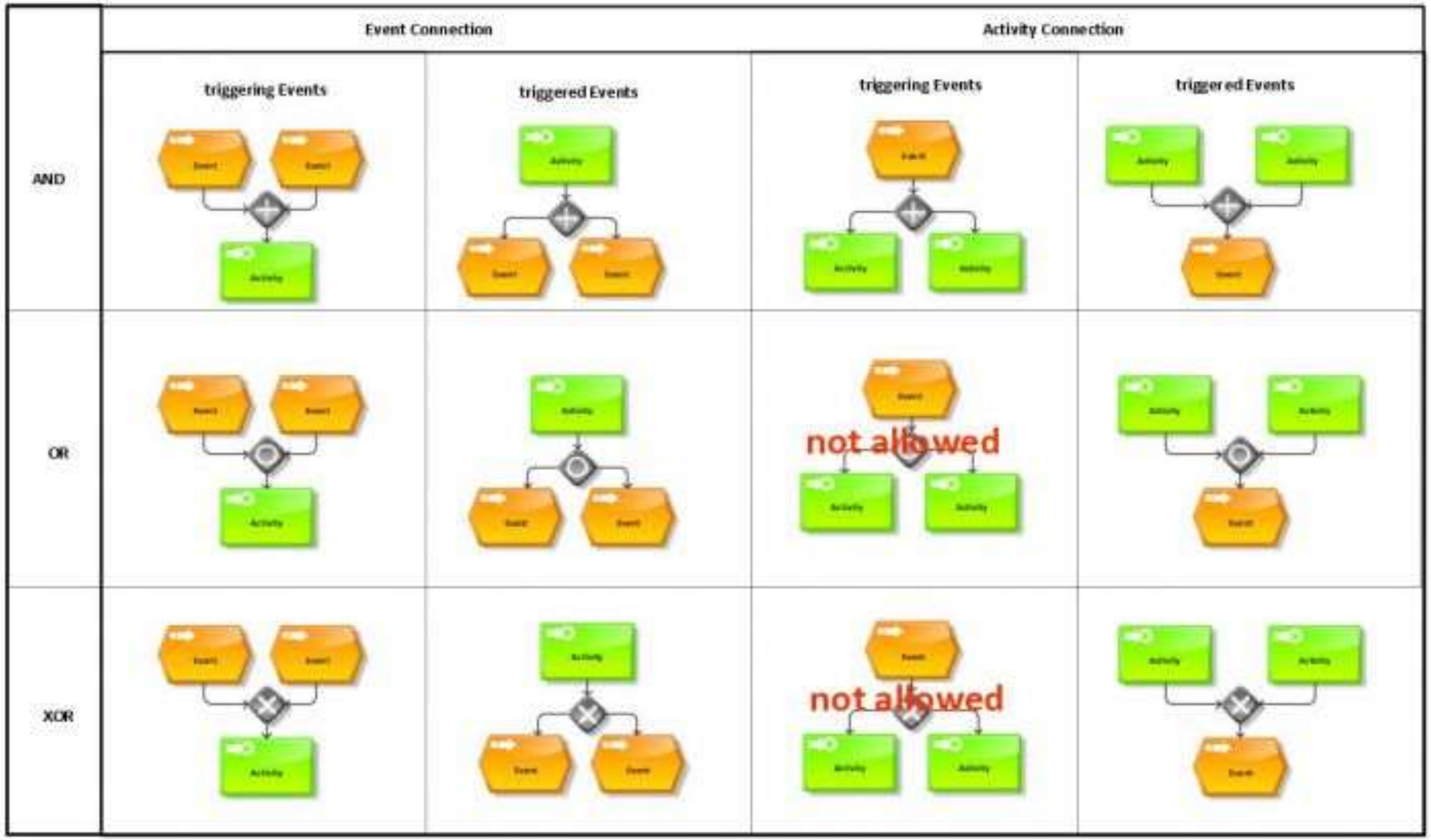


Link XOR forbidden

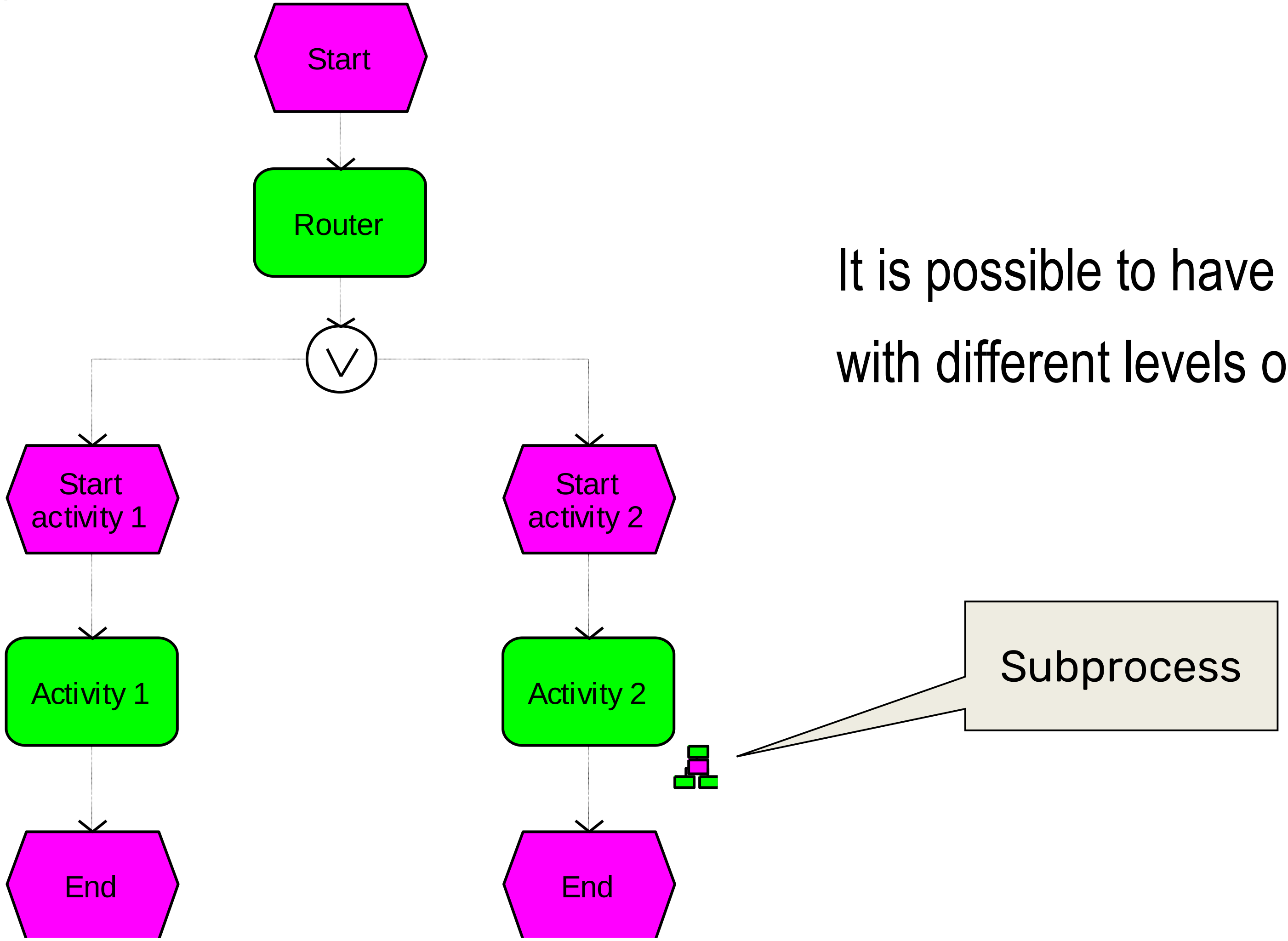
Link OR forbidden



Connection rules for Event-driven process chain

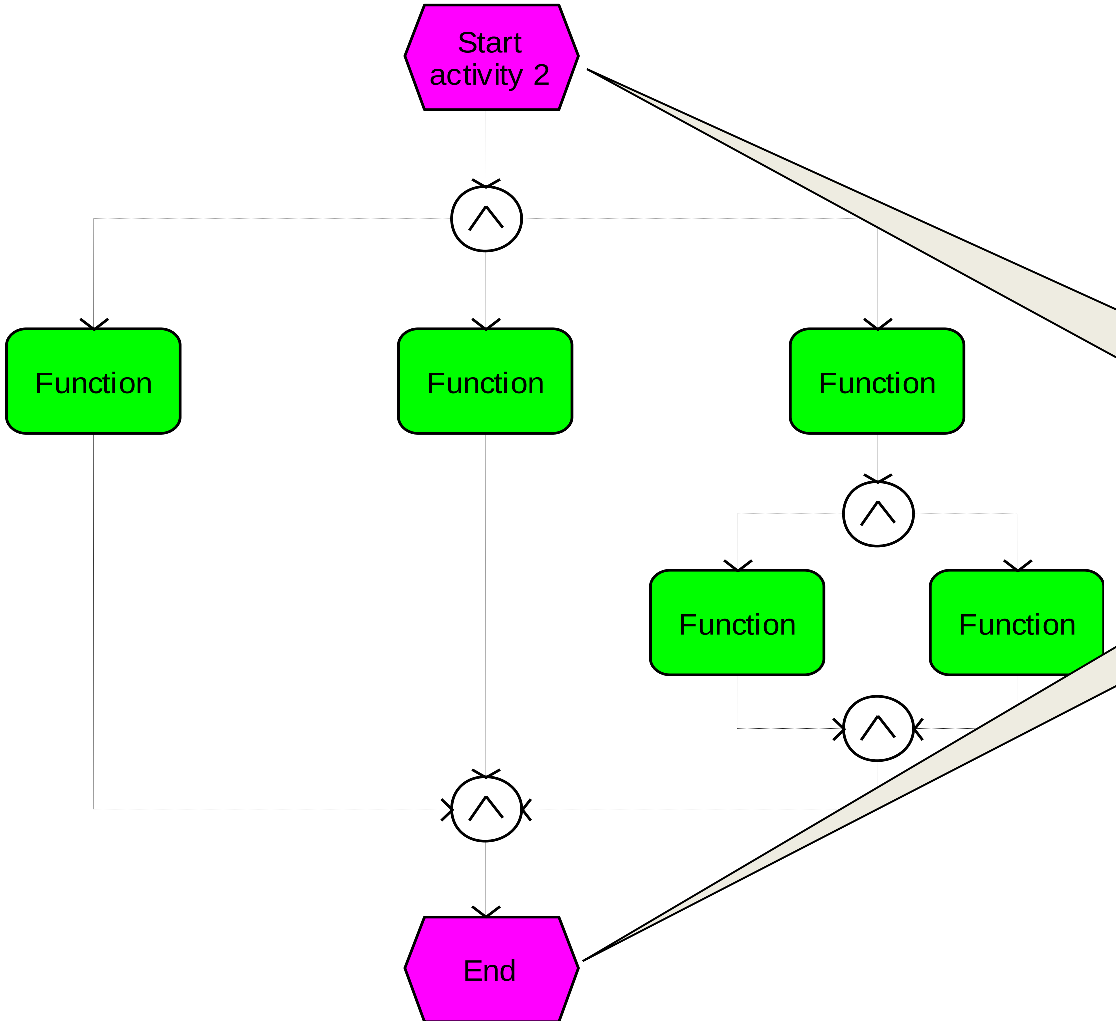


Subprocess



It is possible to have a hierarchical view with different levels of detail

Hierarchical view



The terminal events must be the same of the higher level

Strengths and weaknesses of the ARIS

- Set of models and methods to select considering the context under analysis
- Clear visual representation of processes
- The hierarchical approach allows to describe the process at different levels of detail
- Timelines are clearly identified
- **BUT…**
- The model can result very complex.
- A good experience is needed to determine which models should be used in each case