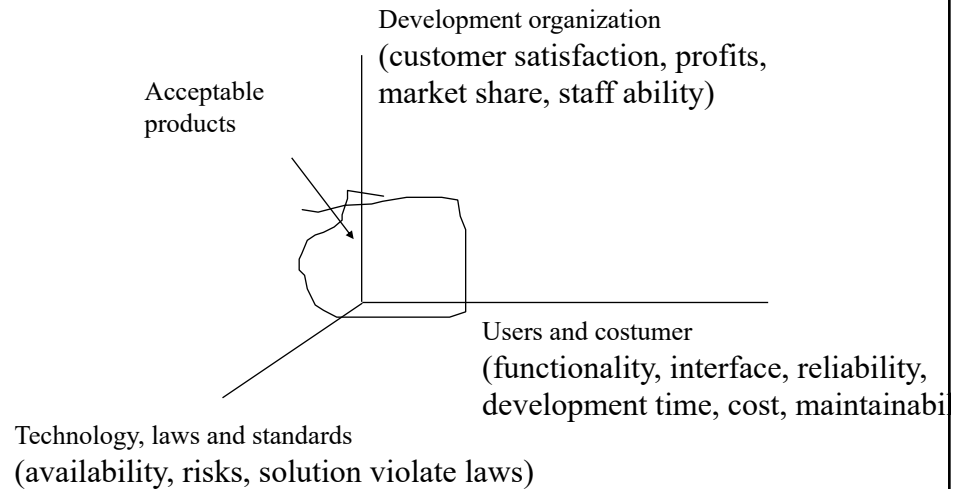


CENG 323 Project Management

Lecture 2 Problem Analysis

1

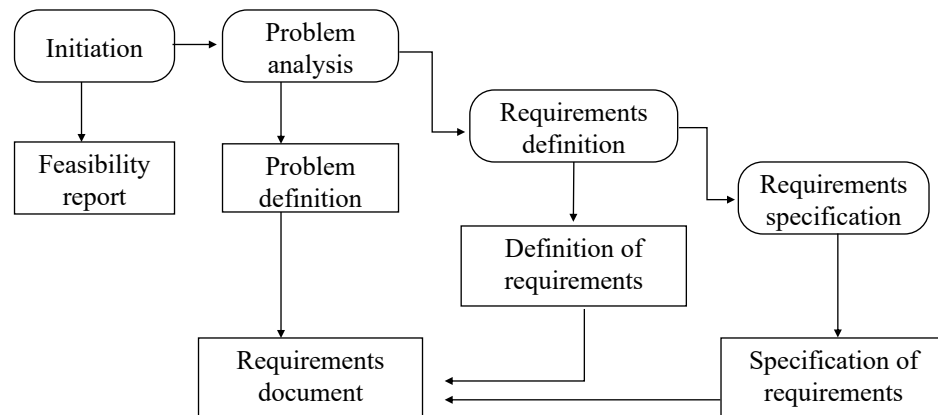
Product Space



2

2

The Contexts of problem Analysis: Requirements Engineering



3

3

Problem Analysis

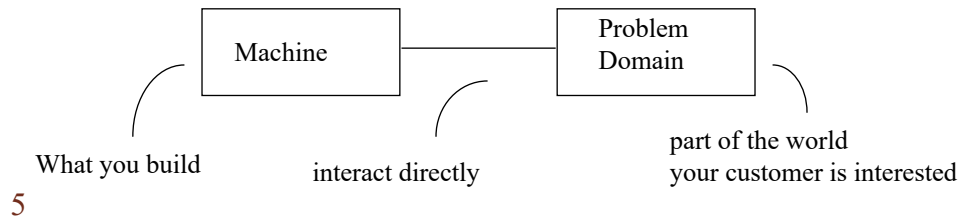
- Primary goal - understanding
 - Learning about the problem domains
 - Find out the actual users
 - Understanding the needs of the users
 - Understanding the constraints on the solution
- Overview
 - Jackson context diagrams
 - extended Event Process Chains

4

4

Problem Context

- You are an engineer planning to build a bridge across a river.
 - Visit the site.
 - How hard the wind is blowing.
 - Picture the bridge yourself.
- You are examining the problem context.



5

Initially - Problem Domain

- To understand the problem, understand the problem domains
 - What parts of the world are in it?
 - How are they interconnected?
 - What are their significant properties?
 - What processes exists?
 - What properties can you exploit?
 - What interactions can they have with the machine?

6

6

Problem domain vs Solution domain

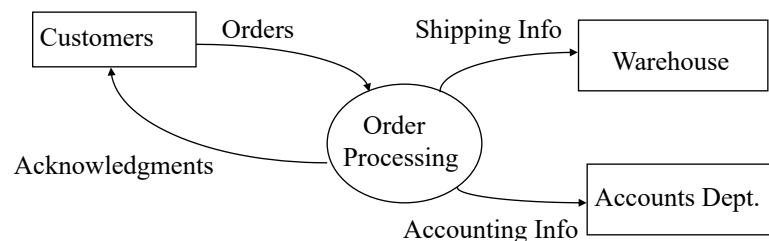
- Think in problem domain:
 - What properties it has right now?
 - What are the AS-IS processes?
 - What properties your customer wants you to bring about?
- Think in solution domain:
 - Programming languages, database management systems, module decomposition, design patterns, software architecture ...
 - As the work progress keep asking your customer: will this machine solve your problem?

7

7

Context Diagrams in SADT

- The level 0 data flow diagram:



8

8

Difficulties of Data-flow Diagrams

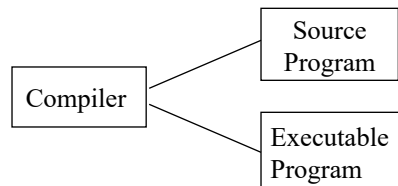
- Interactions among problem domains are not always require a data flow.
- The domains that do not interact with the system but have significance to describe requirements are not represented.
- The concerns that should be focused at this level such as specification of the machine, problem domain characteristics that should be considered, and requirements that should be satisfied can not be easily separated.

9

9

Jackson's Context Diagram

- Show all the domains relevant to the problem requirement (not just directly those providing input to the machine).
- The connections are just a line (all communication is not necessarily of dataflow kind).
- Machine is just one of the domains.
- Simplified context diagram of the compiler construction problem:



10

10

Patient Monitoring Problem

- Patients are in intensive-care unit
- Analog devices monitor their pulse rate, temperature etc.
- The system is to check each patient's specified factors, at specified frequencies against specified safe ranges.
- The factors are to be stored in a database.
- An out of the range factor will be reported to the nurse's station.
- The system should also check that the analog devices working properly and report to nurse's station.

11

11

Problem Domains

- The principle of domain relevance
 - Everything that's relevant to the requirements must appear in some part of the problem domain.
- Patients - nurse's station - analog devices
 - The requirements will be expressed in terms of them.
 - The patients shall be monitored according to ...
 - The nurse's station shall be notified ...
 - The analog devices shall be monitored and any ...
- Nurses
 - Not included we do not concern with nurses actions in response to the alarm messages.

12

12

Two More Domains

- Safe range and frequency specifiers
 - Necessary by the principle of domain relevance.
 - If it is left out there will be no part of the application domain the related requirements could refer.
- Database
 - Is it part of the problem or the solution?
 - The description is not enough to resolve the problem.

13

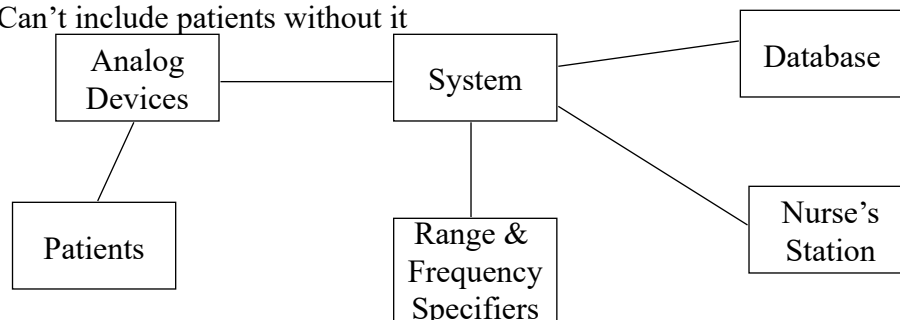
13

Patient Monitoring Context Diagram

Included:

The machine must check their status

Can't include patients without it



Not directly connected:

But they are of central interest

14

14

Describing Domain Characteristics

- Static vs. dynamic
 - Nothing happens, nothing changes, no events ... to plan a journey take the road network as fixed.
 - Time dimension, things happen things change ... civil engineer constructing an intersection takes the road network as dynamic.
- One-dimensional vs. multi-dimensional
 - Program text of a C function is one-dimensional
 - Graphic image is multi-dimensional
 - One dimensional structures are simpler.
 - We usually project multidimensional problems to one dimensional structures.

15

15

Describing Domain Characteristics

- Tangible vs. intangible
 - Chemical plant in process-control system is tangible
 - Texts in word processing systems is intangible
 - Any dynamic physical domain in the problem context introduces real time considerations ...
- Inert, reactive, active dynamic domains
 - It can change but only in response externally controlled events ... text in an editor
 - It does perform some actions of its own in response to external stimulus ... vending machine
 - It performs actions without external stimulus. ... User of a word processing program.

16

16

Processes of the Problem Domain

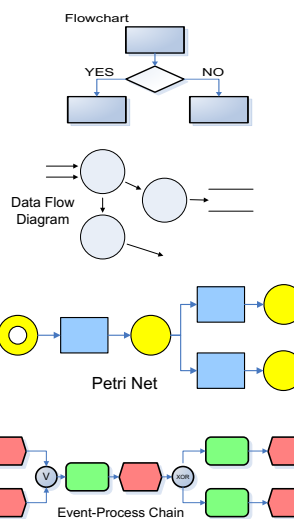
- To understand the problem domain better, understand the processes
- Process Model: An abstract description of an actual (AS-IS) or proposed (TO-BE) process that represents selected process elements considered important for the purpose of the model

17

17

Formal / Semi-formal methods ...

- Flowcharts
 - Strong on control/sequencing but weak on data flow
 - Little support for parallel processing
 - (Behavioral)
- Dataflow diagrams
 - Strong on data flow but weaker on control
 - Good support for parallel processing
 - Weak on iteration
 - (Functional)
- Petri-nets
 - Very precise control and data flow
 - Parallel processing
 - Non-intuitive for human interpretation
 - Too precise for many human processes
 - (Functional-Behavioral)
- Event Process Chains
 - Strong on control/sequencing
 - Support for data flow, relations to org. units, roles (eEPCs)
 - (Functional-Behavioral - EPCs)



18

18

18

Event Based Analysis

- Event is something that happens for a moment.
 - Events are related with states but are not the same.
 - Events represents a significant change in state.

Example:

- a throttle control system is turned on

- Easier for the domain expert to identify events of the problem domain
- Modern software architectures require us to decompose the system boundaries with respect to events

19

19

EPC – Event-Driven Process Chain

- Semi-formal, behavioral notation
- The process is described with a sequence of functions (activities) with events
- Events trigger functions and are results of the functions.
- EPC diagrams start and end with events
- Alternative or parallel paths are modeled with logical operators (And, Or, Xor)

20

20

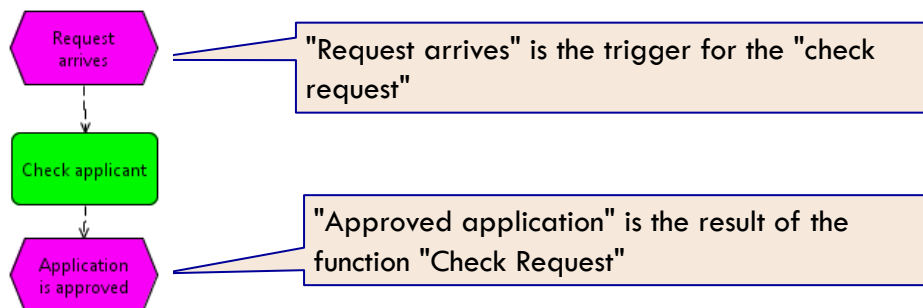
EPC – Major Elements

	Description	Example
	Function is a technical task or an activity performed on an object to support one or more business objectives.	Review a Document, Send a Message, Create a Record, Update a Report, Archive a Document, etc.
	An event represents a state of an object that is relevant in terms of business context, which controls or influences the further procedure of the business process. Events trigger activities and are the results of activities. An activity is a time consuming occurrence while an event is related to one point in time	Document Reviewed, Message Sent, Customer Arrived, Message Received, etc.
	Logical Operators represent alternations that link events and activities in process chains	

21

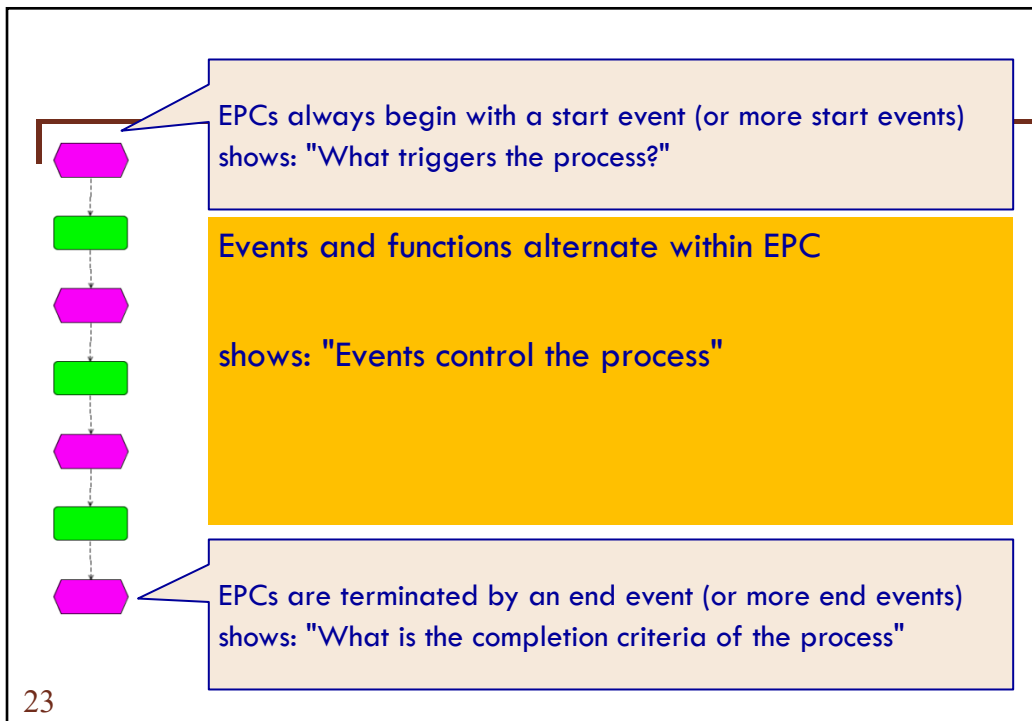
Succession of events and functions

- Arrows indicate the time-logical order between events and functions.



22

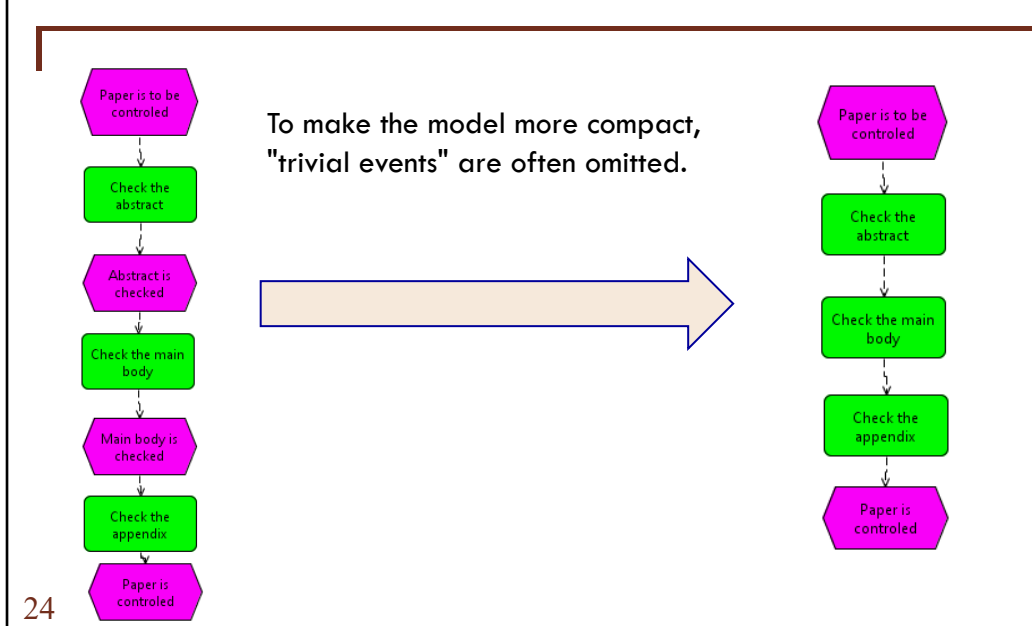
22



23

23

Trivial Events



24

24

Naming Conventions

- Always name the functions as «imperative»
 - Review the document
 - Approve the output
- In Turkish, two alternatives:
 - Dokümanı gözden geçir
 - Dokümanın gözden geçirilmesi
- Events are named to indicate the state
 - The document is reviewed
 - The review of the document is completed
- Always name the objects singular
 - Review the document (not review the documents, conduct the activities...)

25

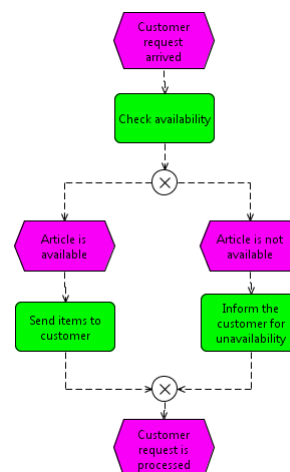
25

Modeling of Alternative Paths

- XOR Connector



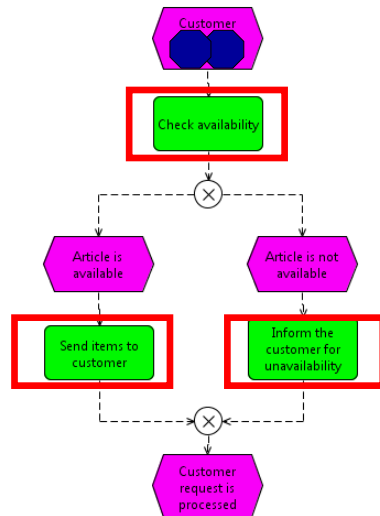
Modeling one alternative flow:
Exactly one of several possible paths is taken



26

26

XOR-Connector



in the example:

either the item is in stock

► left path is traversed

or it is not available

► right path is traversed

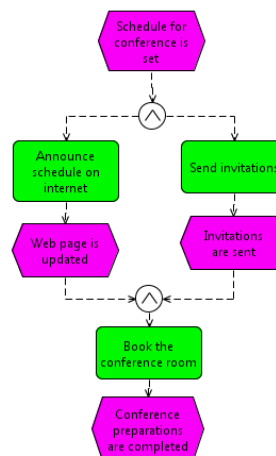
27

Modeling of Parallel Executions

■ AND Connector



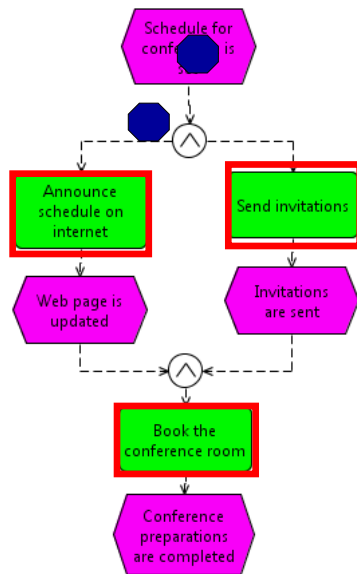
More than one function can be performed simultaneously



28

28

AND-Connector



The two paths are traversed in parallel:

"Announce schedule on internet" can be executed at the same time with "send invitations"

29

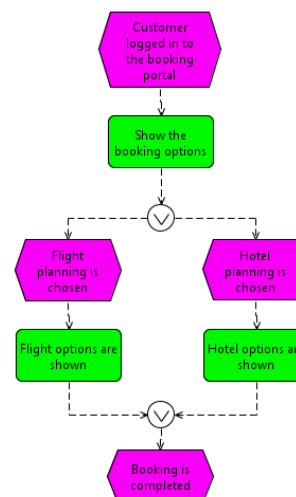
Modeling of «One or More»

■ OR Connector



Modeling one or more alternative flow:

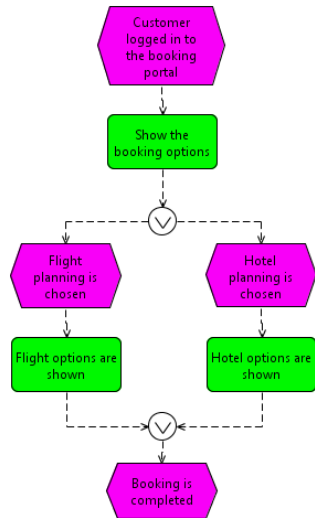
More than one of the options can be processed in parallel - but at least one must be taken



30

30

OR-Connector



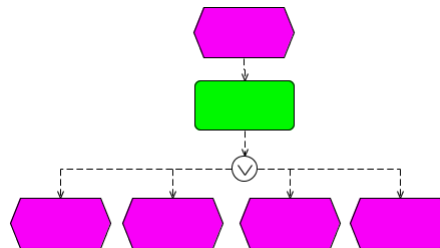
Three options:

Case 1: Only Left

Case 2: Only Right

Case 3: Both

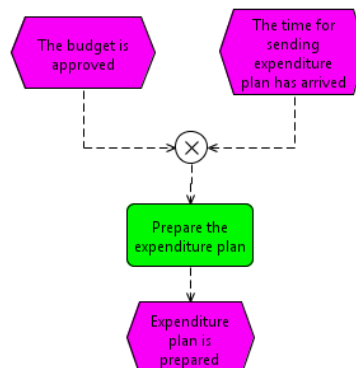
How many options if branches are multiple?



31

Multiple Start Events

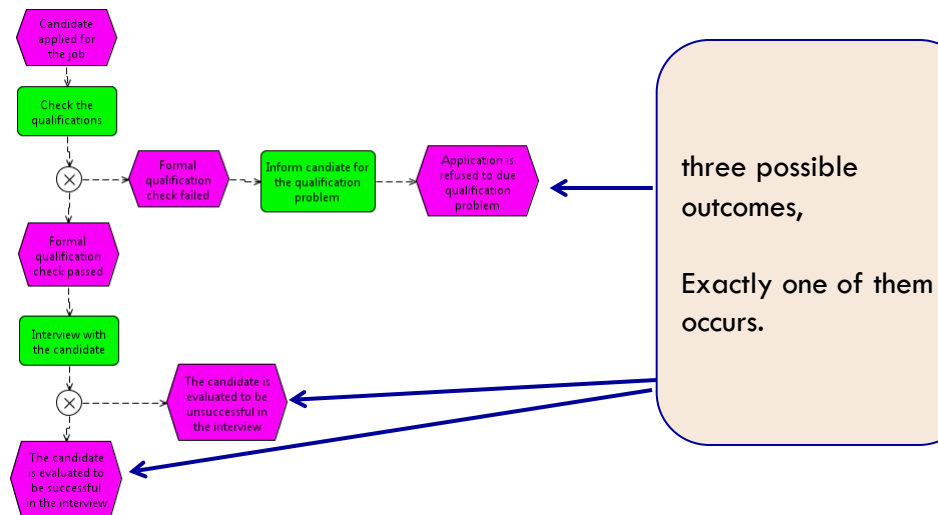
- Connectors can also be used to model more complex conditions for the start of a process



32

32

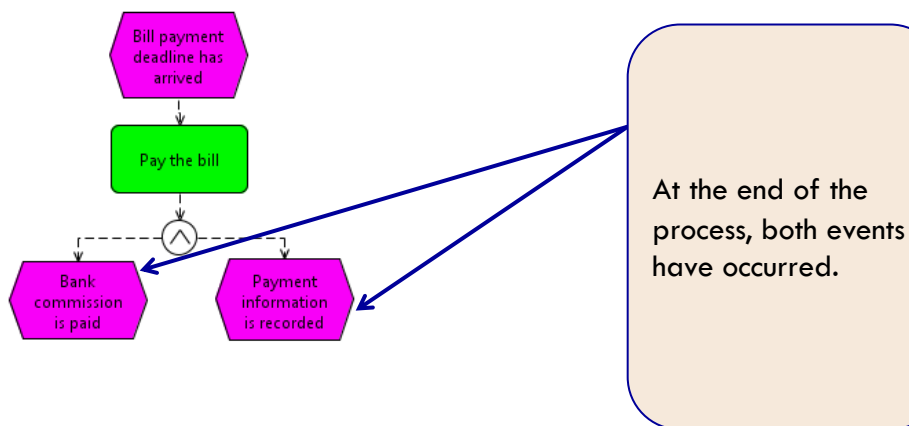
Multiple End Events – Case 1



33

33

Multiple End Events – Case 2

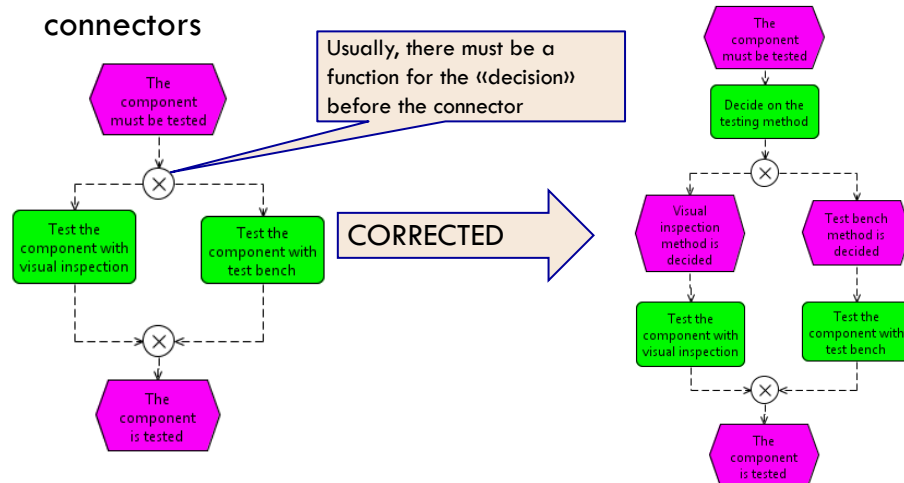


34

34

Rule:

- Be careful about usage of events before or and xor connectors



35

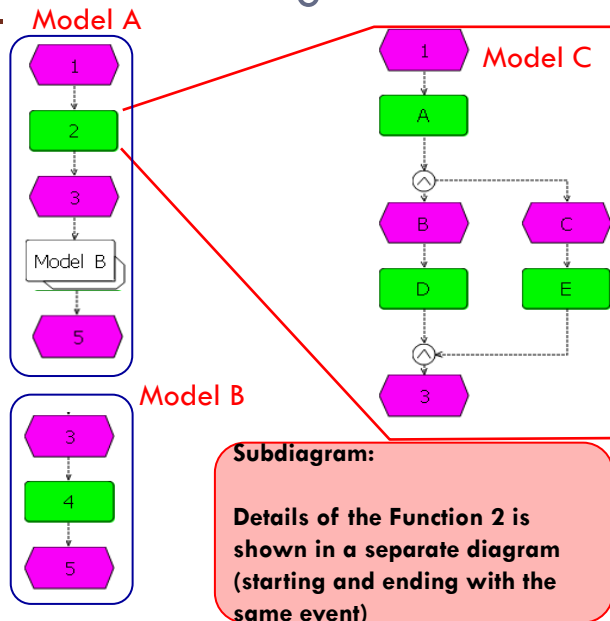
35

Process Interface and Subdiagrams

Process Interface:

Model A ends with Event 3
then it continues with **Model B**, which starts with Event 3

Modularization Allows Reuse and Avoids Duplicate Modeling!

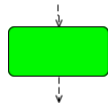


36

36

EPC Syntax Rules Summarized:

- EPCs start and end with events
- Events have:
 - Exactly one incoming arc and one outgoing arc
 - Or exactly no incoming arc and one outgoing arc (start)
 - Or exactly one incoming arc and no outgoing arc (end)
- Functions have exactly one incoming and one outgoing arc



WRONG!

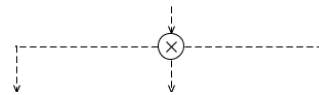
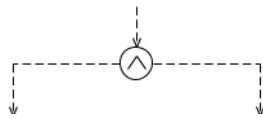


37

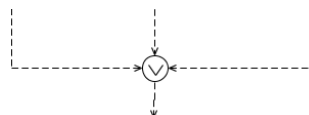
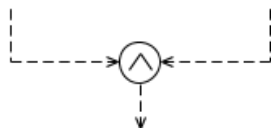
37

EPC Syntax Rules Summarized:

- Connectors have either exactly one incoming arc and more than one outgoing arc (Split)



- Or more than one incoming arc and exactly one outgoing arc (Join)

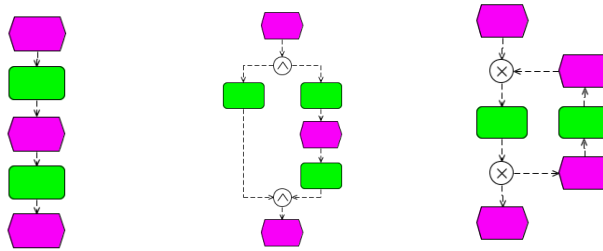


38

38

EPC Syntax Rules Summarized

- Events and functions alternate. If necessary, intermediate connectors do not change this rule.
- If several functions executed in succession, "trivial events" between them can be omitted

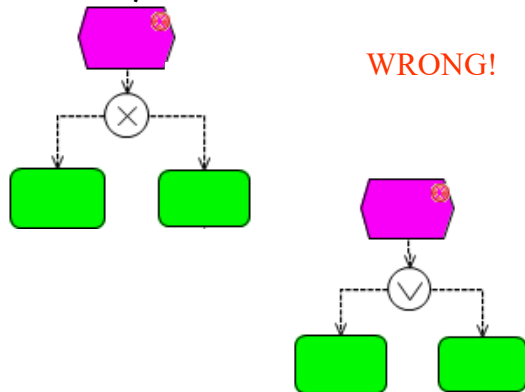


39

39

EPC Syntax Rules Summarized

- Before XOR and OR-split, usually there must be a function, not an event

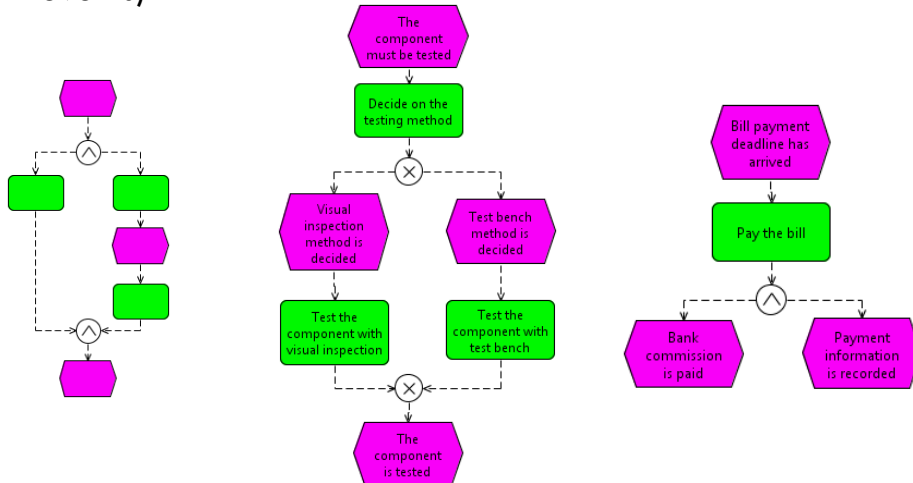


40

40

EPC Syntax Rules Summarized

- Use splits and joins as pairs (other than start and end events)



41

41

More Details on Connectors



Example (events)	Explanation	Example (functions)	Explanation
	Both of the events must occur in order to execute the function		The event triggers both of the functions
	The execution of function results in the occurrence of the two events		When the two of the functions are completed, the event occurs

42

42

More Details on Connectors



Example (events)	Explanation	Example (functions)	Explanation
	For the function to be executed, one or more of the events must occur		The event triggers one or more of the functions
	When the function is executed, it results in one or more of the events occurring		When one or more of the functions are executed, the event occurs

43

43

More Details on Connectors



Example (events)	Explanation	Example (functions)	Explanation
	For the function to be performed, one of the events must occur		The event triggers only one of the functions
	The execution of the function results in only one of the events		When only one of the functions are executed, the event occurs

44

44

eEPC – extended Event Driven Process Chain

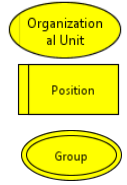
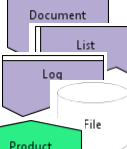
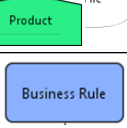

- EPCs are extended to include:
 - Roles, organizational units, positions that performs the activities
 - Information (entity, document, email, CD, etc.) that is being input/output to activities
 - Application systems, tools, etc. that supports the activities
 - Objectives of performing activities

45

45

45

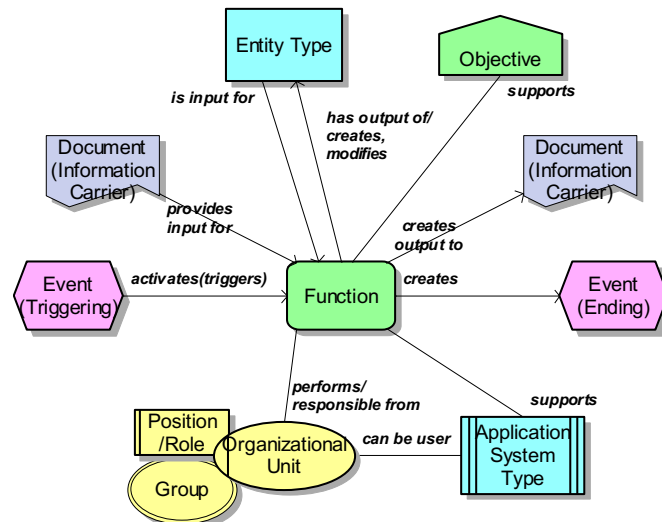
eEPC – Additional Elements

	Description	Example
	<p>Organizational units are the performers of the tasks required to attain the business objectives.</p> <p>The smallest organizational unit in a company is a position. It is assigned to employees (persons).</p> <p>A group may represent a group of employees (persons) which are working together for a specific period of time.</p>	Design Dept., Finance Dept., Project Team, Review Team, Project Manager, Designer, Customer, etc.
	<p>Document is a type of information carrier, which represents a means to store and transmit information. It can be in the form of a document, an email, a fax, a CD, or a verbal message that is produced out of an activity or input to be processed by an activity.</p>	Software Requirements Spec. (SRS), Review Form, List of..., Notification Email, Checklist, etc.
	<p>The rules which constraint how the function is executed</p>	The assignment must be submitted in two weeks
	<p>An application type represents a system or a tool that is used to support agents (actors, roles) in performing their activities.</p>	Req. Mang. Tool, Project Mang. Tool, or specifically MS. Project, Rational Rose, etc.

46

46

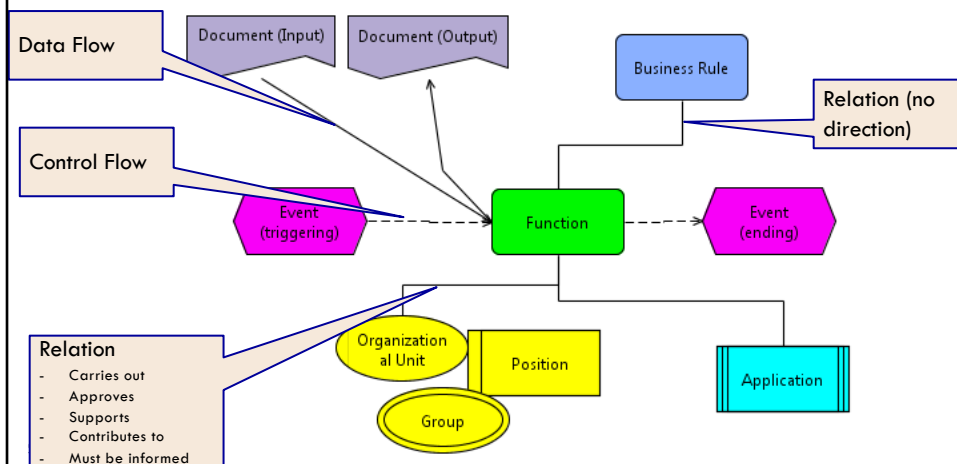
eEPC – Major Element Connections



47

47

eEPC – Major Element Connections



48

48

eEPC – Modeling Considerations

- A Function must always have an Organizational Element connected! (unless it is a subdiagram)
- For each function, analyze if there are any inputs, outputs or both.
 - There may be functions without any inputs or outputs
 - Decide if you will show template documents as inputs. Alternatively, you can assume that all outputs has templates, or you can attach the information of the template to each output.
- For each function, analyze if there are any business rules applicable, any applications used etc.

49

49

Process Modeling Tools

- Many available for free (especially for academic use)
 - Aris Express
 - eEPC examples: <https://www.ariscommunity.com/event-driven-process-chain>
 - Bizagi (for BPMN)
 - Intalio (for BPMN)
 - ...

50

50