SYSTEMS ANALYSIS LECTURE 8 – SYSTEM BEHAVIOUR

Zuzana Bělinová

Behaviour

- Way of achieving goals
- Set of processes active within the System in specified time interval and in given state of the neighborhood

Process – sequence (chain) of events

State of the System – Set of actual states of Systems elements

Event – Change of the state of Systems element, or change of Systems structure or step of external time

Prerequisities

- Events are discreate, parameter changes are discreate
- Set of inputs is finite and distinguishable

→System state is finite

Behaviour

- Partial behavior (F_i) – Set of processes activated for pre – defined (fixed) vector of Systems inputs (I_i)
- System behaviour

  $$\mathbf{F = U_{\forall j}\ F_i}$$

- Finding behaviour
  - Behaviour is usually analysed using model (System model, specialized – e.g. DT, PN, etc.)
  - By experiments

How to describe process

- Path in the graph of the system
- Time of activation of elements
- Logical rules for choosing alternatives

Basic behaviour model

- Just the processes

- Ways of recording the behaviour
  - Graph of vehaviour
  - Matrix of behavour
  - Set of processes

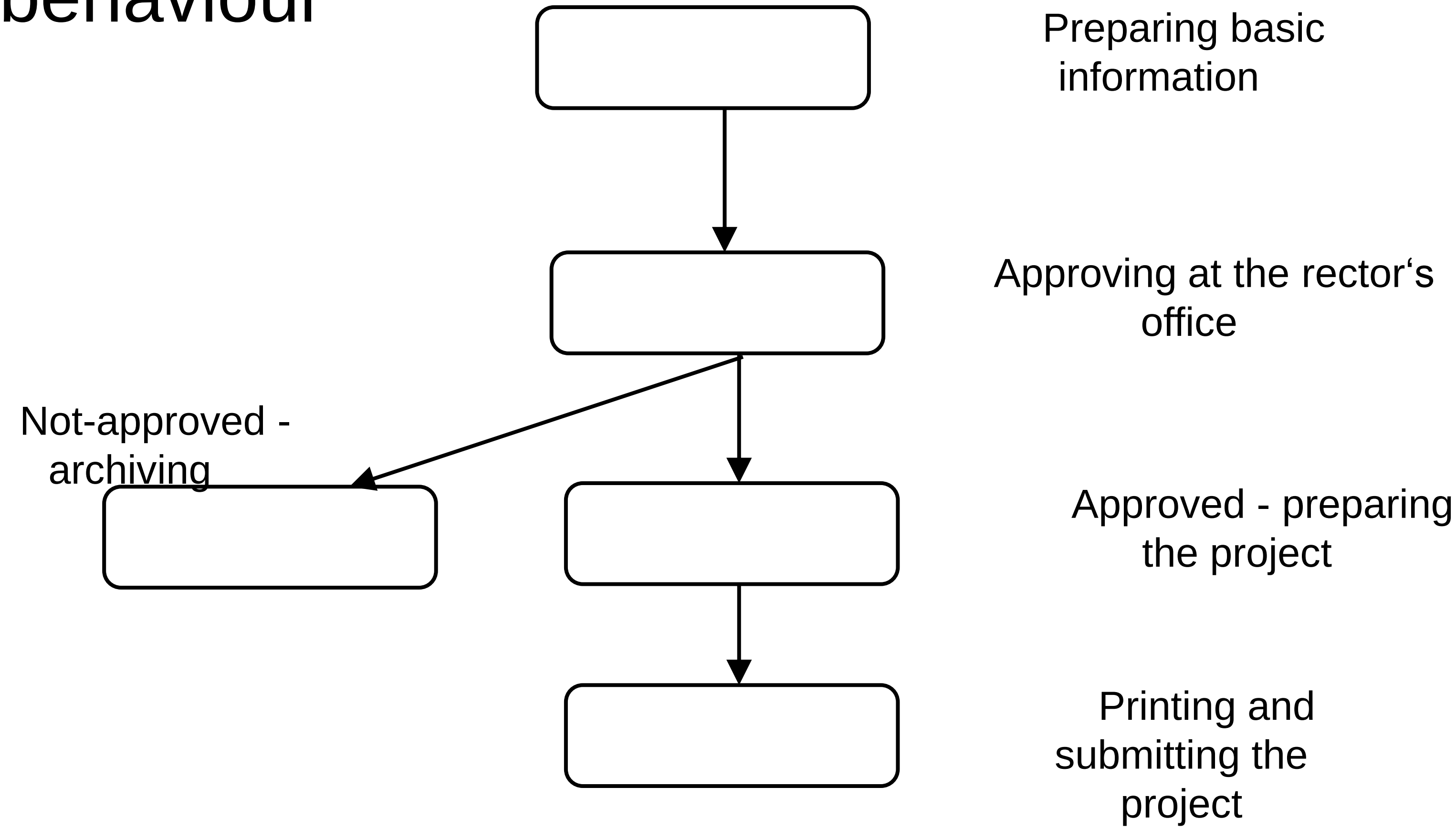
Graph of behaviour

System	Graph
state	node
event	edge
process	path

- Using PN – transition diagram is the graph of behaviour

Example:

Preparing a project proposal - Graph of partial behaviour



Matrix of behaviour

- Adjacency matrix of the graph of partial behaviour

Di	s^i_1	s^i_2	...	s^i_n
s^i_1	d^i_{11}	d^i_{12}	...	d^i_{1n}
s^i_2	d^i_{21}	d^i_{22}	...	d^i_{2n}
...	...	...	...	...
s^i_n	d^i_{n1}	d^i_{n2}	...	d^i_{nn}

- $d^i_{j,k}$ **= 0** .... No transition $s^i_j \rightarrow s^i_k$
- = **1** .... Transition $s^i_j \rightarrow s^i_k$
- s^i – system states
- d^i - events

Matrix of (partial) behaviour

	Preparing basic information	Approving at the rector‘s office	Archiving	Preparing the project	Printing and submitting the project
Preparing basic information	0	1	0	0	0
Approving at the rector‘s office	0	0	1	1	0
Archiving	0	0	0	0	0
Preparing the project	0	0	0	0	1
Printing and submitting the project	0	0	0	0	0

Standard behaviour matrix - **SDi**

- The same structure as behaviour matrix **Di**, but it is based on the whole system state
- There are not only the states active for particular input I, but also all other system states

Set of processes

- All possible processes (paths)

Example:

F1:

- f11: Preparing basic information → Approving at the rector‘s office →Preparing the project → Printing and submitting the project

- f12: Preparing basic information →Approving at the rector‘s office → Not-approved - archiving

System behaviour

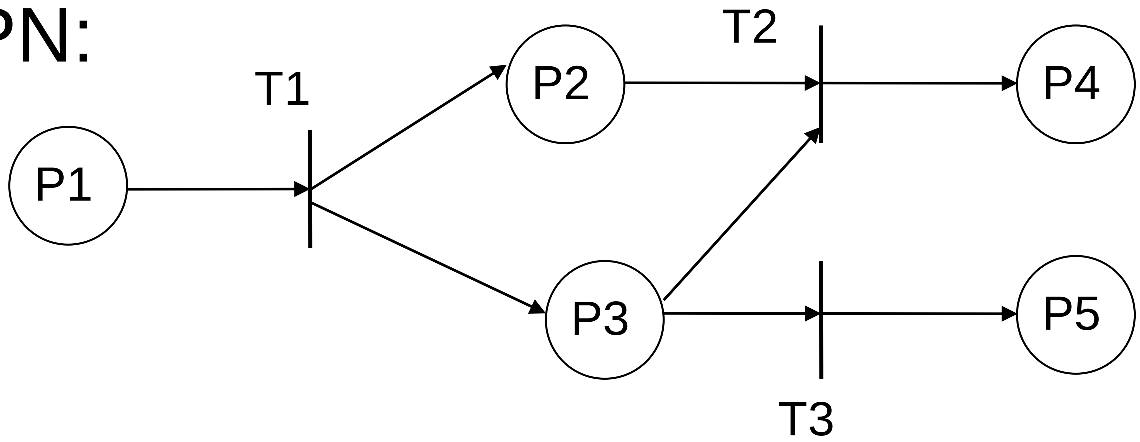
- *Set of all behaviour graphs, or matrices D_i resp. SD_i of all sets of processes represents behavior F as a whole.*

$$F = \cup_{\forall i} F_i$$

Example:

Basic behaviour model using a PN

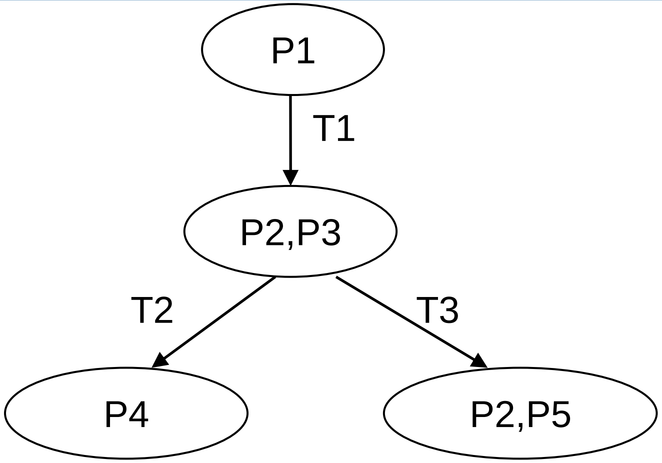
PN:



M0=(1, 0, 0, 0, 0)

Example

Graph of behaviour:



Matrix of partial behaviour:

	P1	P2, P3	P4	P2, P5
P1	0	1	0	0
P2, P3	0	0	1	1
P4	0	0	0	0
P2, P5	0	0	0	0

Set of processes:

F1:

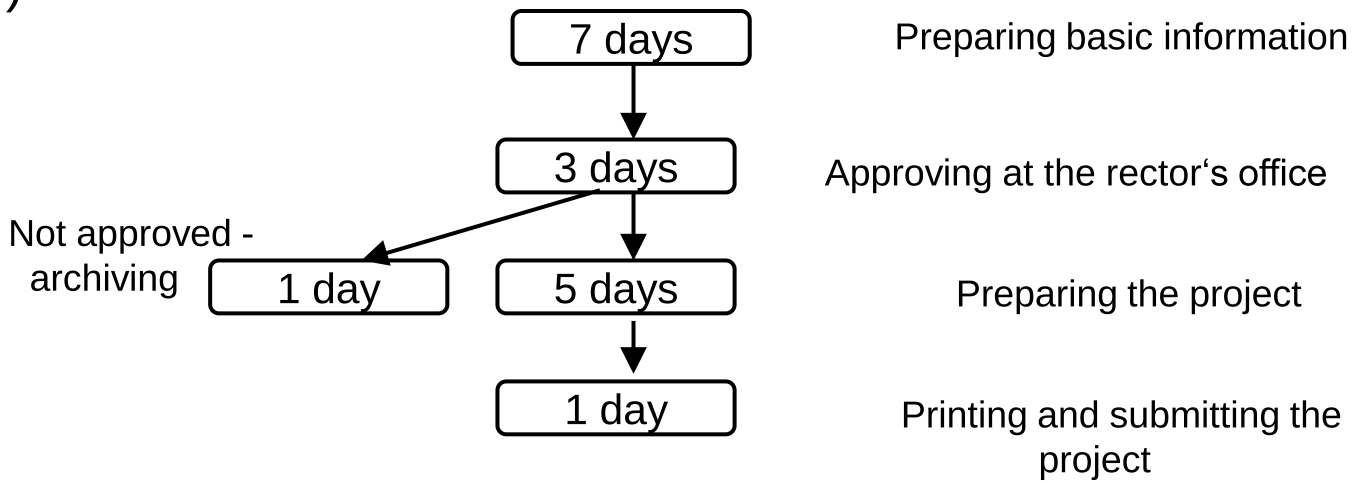
- *f11: P1→P2,P3→*P4
- *f12: P1→P2,P3→*P2,P5

Extended behaviour model

- It is dealing also with
  - Parameters of the relations and their values
  - Functions of elements
- Creation of extended behaviour model
  - Create the basic model
  - Add there the description of parameters and functions
  - Introduce the input values
- Usually it is solved for some interesting process – e.g. the longest, shortest, etc.
- Usage
  - Analysis of time (duration of processes)
  - Analysis of costs
  - Analysis of reliability

Analysis of time

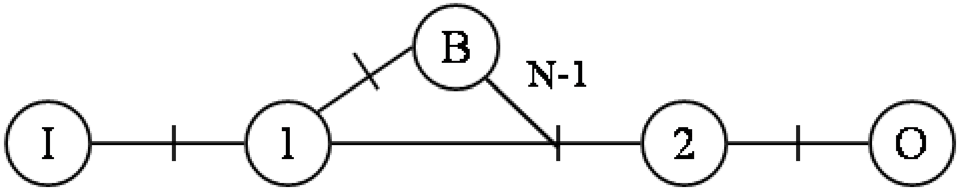
- There is time assigned to every element activation
- We assign the total duration time to the processes
- Search of process according requirements (typically the shortest one)



- f11: Preparing basic information → Approving at the rector's office →Preparing the project → Printing and submitting the project (7+3+5+1=16)
- f12: Preparing basic information →Not approving at the rector's office (7+3=10)

Analysis of costs

- Cumulative
- Using source of assets
  - Real
  - Fictional



Analysis of reliability

(Probability of process realization)

- It is necessary to know the reliability of particular elements

- **Serial arrangement**

  $P = \prod_{\forall i} p_i$ P – process reliability, p – reliability of elements

- **Parallel arrangement**

  $P+Q=1$,

  $q_i+p_i=1 \ \big|_{\forall i}$

  $Q=\prod_{\forall i} q_i$

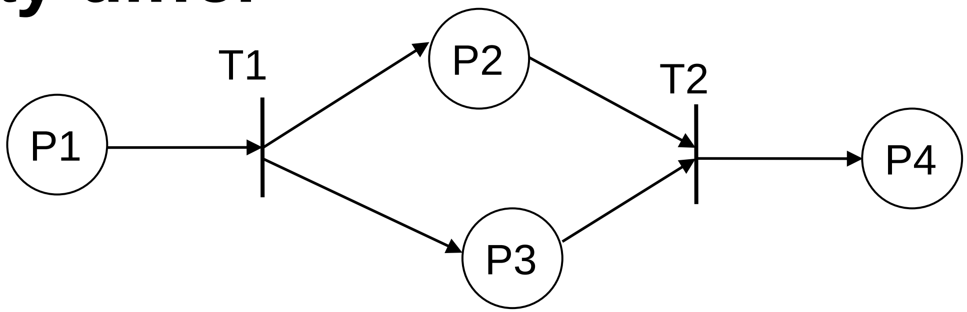
  P – process reliability, p – reliability of elements

  Q – Probability of incorrect process run, q – probability of elements malfunction

Note:

BE CAREFUL

The functional and topological arrangement may differ



Functionally serial!

Serial behaviour task

- Linear sequence of states
- $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots\ldots \rightarrow s_n$

Finding e.g.

- Regularity of interfaces
- Path lengths
- Agreeement of goals and the output parameters
- …

Parallel behaviour task

- More than one process are running at the same time
- more functions are activated

- Necessary to judge, if these processes are mutually dependent or not (e.g. Synchronous activation of the same element, using the same sources, etc.)

- If the processes are dependent, first analysis if there is a problem
- Solution
  - Shifting in time,
  - Changing condition for activation
  - Changing structure, etc.
- In the end, regularity needs to be checked

Example – using time diagrams for process dependance assessment

Example:

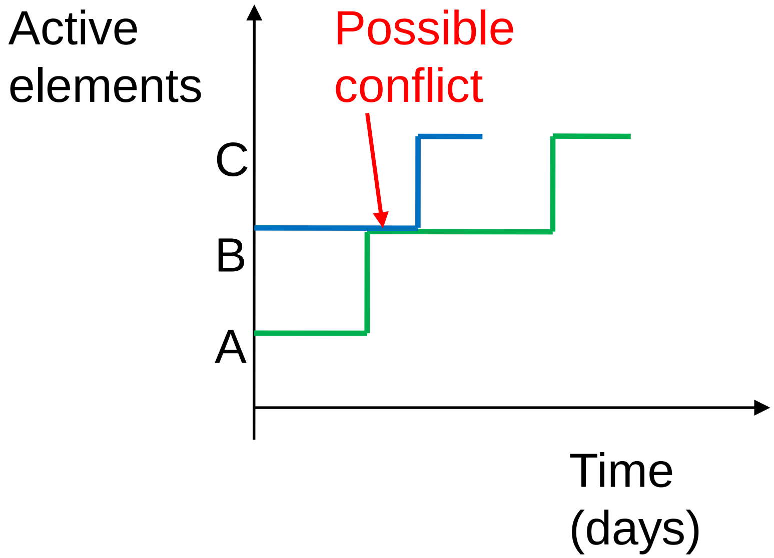
Processes in the system:

- I: A→B→C
- II: B→C

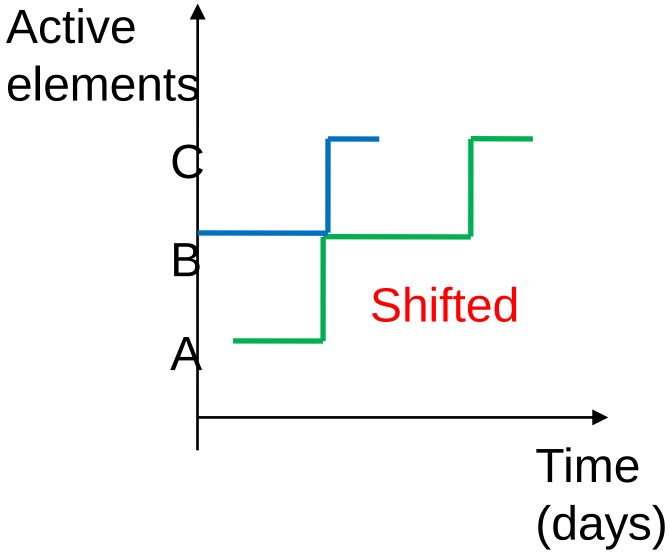
Duration of particular states:

State	Duration (days)
A	2
B	3
C	1

Time diagram:



Possible solution in case of conflict:



Alternate behaviour task

- After one state alternatively more there one state can occure
- Decision – using e.g. logical condition

Modelling uses

- Logical sentences / functions
- Sets of logical equations
- Decision tables (**DT)**
- Combination of Petri nets (**PN**) and tools mentioned above

- Thank you for your attention