LECTURE

2

Systems analysis

Behaviour, architecture

Ing. Zuzana Bělinová, Ph.D.

LECTURE OVERVIEW

System behaviour

Genetic code

System architecture

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 discrete, parameter changes are discrete

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 behaviour
- Matrix of behaviour
- Set of processes

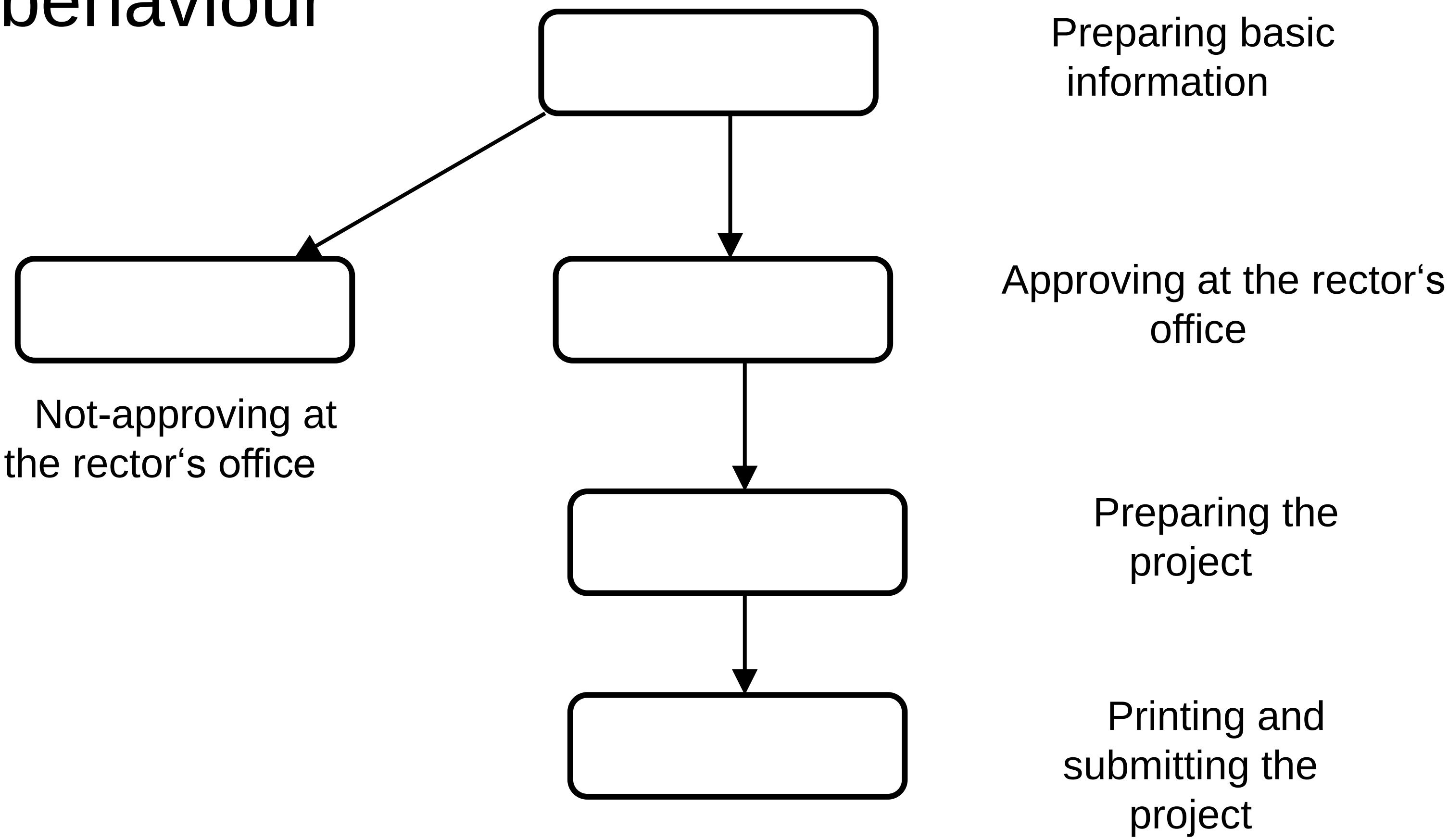
GRAPH OF BEHAVIOUR

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	Not-approving at the rector's office	Preparing the project	Printing and submitting the project
Preparing basic information	0	1	1	0	0
Approving at the rector's office	0	0	0	1	0
Not-approving at the rector's office	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 →Not approving at the rector‘s office

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.

$$F = \cup \forall i \; F_i$$

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

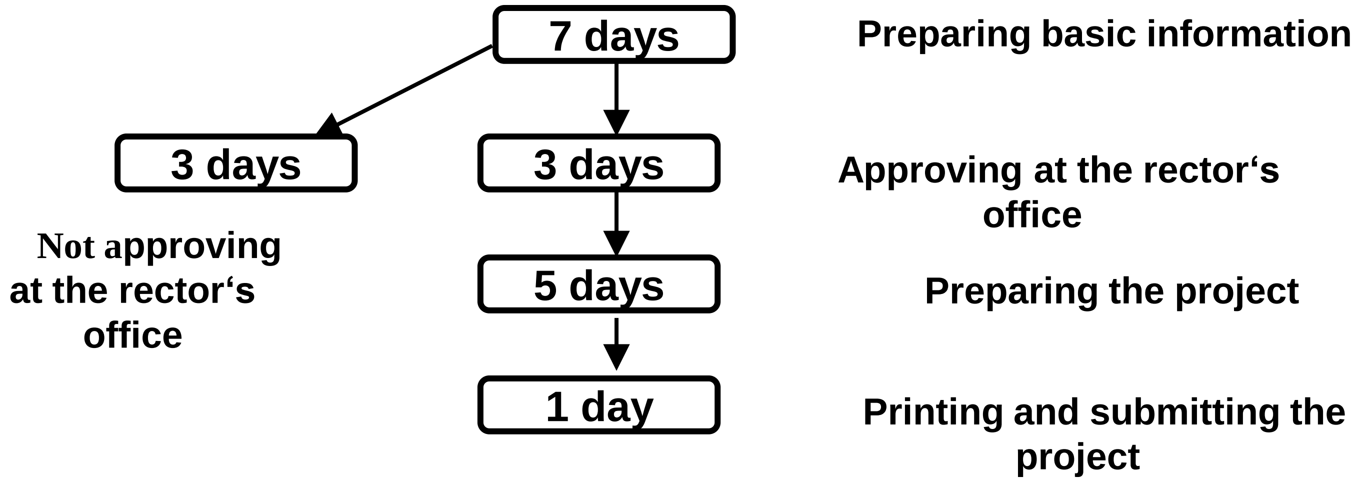
- Analysis of time (duration of processes)
- Analysis of costs
- Analysis of reliability

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)

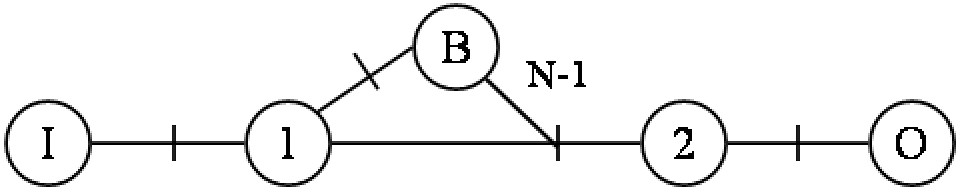
f12: Preparing basic information →Not approving at the rector's office (7+3=10)

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

SERIAL BEHAVIOUR TASK

Linear sequence of states

$S_1 \rightarrow \ S_2 \rightarrow \ S_3 \ \rightarrow ...... \rightarrow \ S_n$

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:

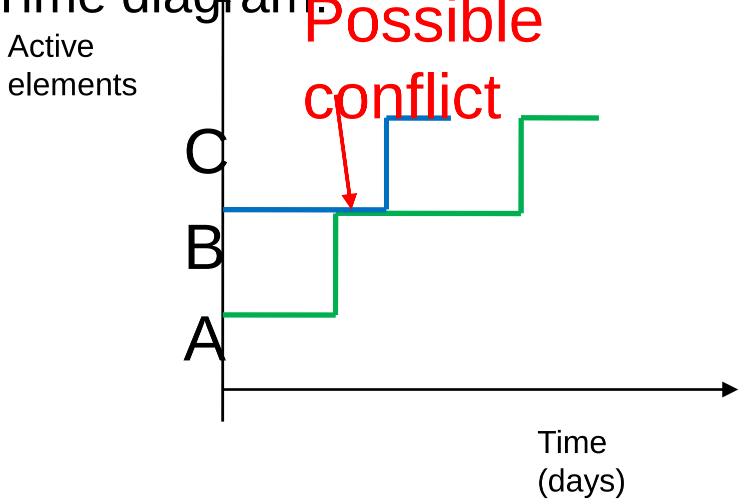
I: A→B→C

II: B→C

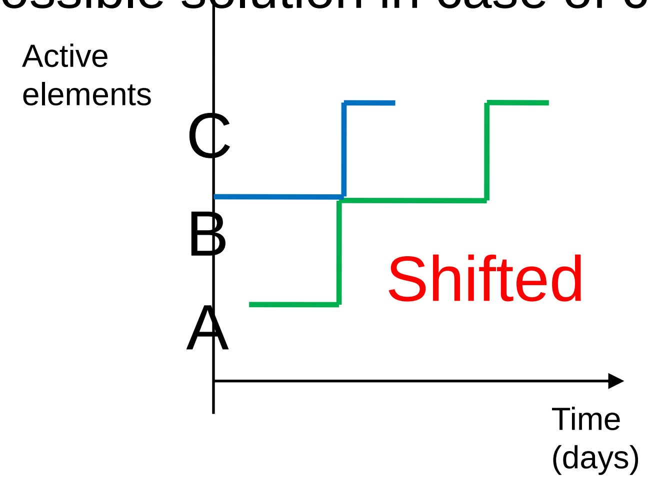
Duration of particular states

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 occur

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

TYPES OF BEHAVIOUR

Equilibrium

- With Attractor

Stable

Divergent – regenerative / degenerative…

Goal oriented – with attractor / homeostasis…

Response to the inputs / activated from interior…

Species oriented (With genetic code) – ideal / standard / adaptive / mutation - failure / catastrophe / self-correction / self-learning

GENETIC CODE

In the system there are some processes functioning more efficiently than the others (system has disposition („genetic“ dispositions) for these processes

When activating these kind of processes (according genetic code), strong functions are activated

Typical processes – can be found in most of partial behaviour

Typical behaviour (containing typical processes) – genetic code

GENETIC CODE IDENTIFICATION

F_T can be found as intersection of partial behaviours using e.g. standard behaviour matrixes SDi

$$\mathbf{F_T = \cap_{\forall i} F_i}$$

Often there is no intersection of all partial behaviours, solution:

- Decrease of the condition – not looking for the same process in all partial behaviours, but only e.g. in 90% of behaviours, 80%, etc.
- Include process frequencies, weight of processes
- Or to declare the GC process with the highest number of strong functions

TYPES OF BEHAVIOR IN RELATION TO THE GENETIC CODE I

1) Ideal behaviour. It has consistently zero deviation from the genetic code and tracks the goal. In practice this behaviour is rare.

2) Standard behaviour. Small deviations arise from the GC, with the possibility of returning to the path of GC. It follows the goal, but requires control of the goal behaviour.

3) Adaptive behaviour. Pose considerable deviation from the trajectory of GC. There is the possibility of returning to a trajectory of GC at a price: either changes the parameters on the same structure or structural changes (introduction of new elements and joints). It allows achieving the goal of reduced efficiency (due to additional costs such as energy, time ...... the adaptation).

TYPES OF BEHAVIOUR IN RELATION TO THE GENETIC CODE II

4) Mutation behaviour. Variations arise from the GC trajectory of no return. There is a possibility to ensure overall consistency and integrity of System with an acceptable change of GC and with a change in goals. System can maintain the basic attributes of the goal behaviour. In this case a change of the structure is quite frequent. Based on the analogy with the biological sciences we can specify resulting System as a different kind.

5) Degenerative (faulty) behaviour. System is unable to maintain the goal, in some cases even losing a good sense of goal behaviour. It disrupts GC irreversibly and leads to the disintegration of the System. For this behaviour, we can distinguish two versions:

- a) The uncontrolled (sudden or gradual) degradation
- b) Controlled System termination (apoptosis).

ANALYSIS OF VARIATIONS FROM THE GENETIC CODE (IN GENERAL DISTANCE OF PROCESSES)

Structural analysis – how the processes differ

Dynamic analysis – number of necessary steps to change the process to the genetic code

Goal analysis

SYSTEM ARCHITECTURE

SYSTEM ARCHITECTURE

Purpose-built system model of particular object or system, that has to fit into the given space and at the same time to effectively execute given of recognized system functions

Definition specification directly or indirectly show real existence conditions of particular object (“system constrains“)

Dimensions

- Time
- Costs
- Resources
- Etc.

SYSTEM ARCHITECTURE

Must be based on the system requirements

Creating system architecture

- Finding out customer expectations
- Transforming the imto formal requirements
- Finding solution – creating the architecture from different perspective
    - Functional
    - Physical
    - Organizational
    - Etc.

ARCHITECTURE

System architecture can be viewed also as unification construction of three system models

- object (what)
- infrastructure (where, when)
- purpose (how, why).

If we emphasize the

- object, it is the developing architecture
- infrastructure, it is the real architecture
- purpose, it is the theoretical architecture, preferring the system theoretical viewpoints

ARCHITECTURE TYPES

Pragmatic sorting – e.g. in transport telematics

- Functional
- Physical
- Communication
- Organization

E.g.the different viewpoints in the telematic ITS European architecture FRAME

ARCHITECTURE TYPES

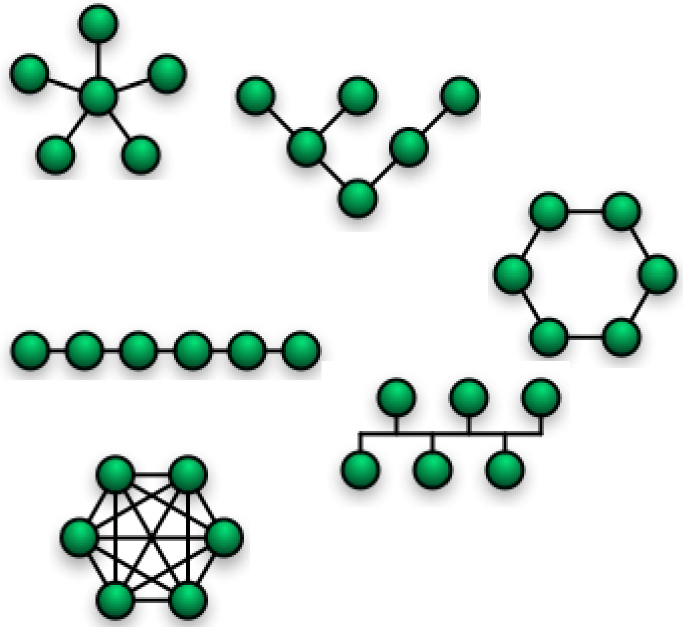
Based on the level – e.g. in transport telematics

- Global
- European
- National
- Local

ARCHITECTURE TYPES

Topology types – used in computer technology

- Star
- Tree
- Ring
- Line
- Bus
- Mesh
- Fully conected



Source: https://en.wikibooks.org/wiki/Communication_Networks/Network_Topologies

Thank you for your attention