SYSTEMS ANALYSIS LECTURE 5 STRUCTURAL TASKS 2

Zuzana Bělinová

Further system structural tasks

- Identification of specific elements and relations
- Flow Network Task
- System decomposition task
- System integration task
- Systems goals task

Identification of specific elements and relations

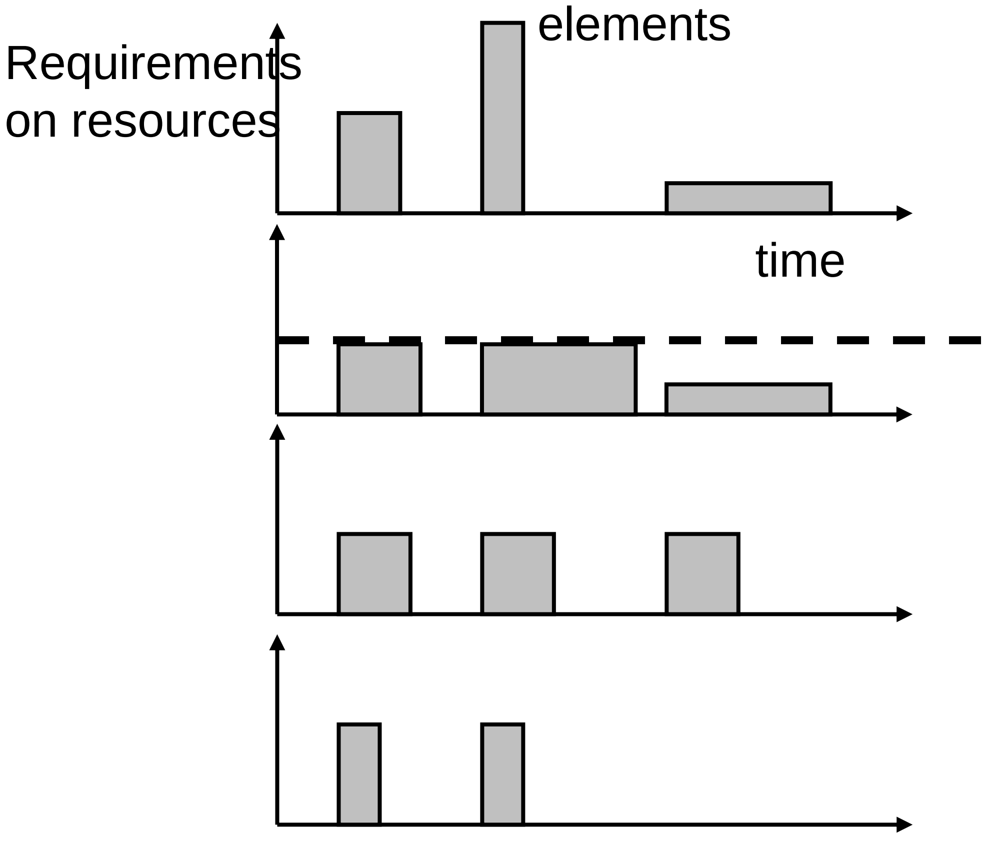
- Quantitative viewpoint
  - Border elements
  - With specific function
  - According values

- Qualitative viewpoint
  - Non-aggressive – IN>>OUT
  - Important, complex – many IN, OUT
  - Simple – few IN, OUT
  - Control, aggressive – IN<<OUT

Flow network task

- Directed network – directed graph with non-negative evaluation of relations, with one initial and one final node
- Requirement - homogenization
- Powerful algorithm within this theory is the Ford-Fulkerson's algorithm

Capacity balancing



System decomposition task

- Reasons for decomposition
  - Change of level of distinguish
  - Excessive complexity
  - Top-down design
  - Need for changes, system overview

After decomposition often follows integration!!!

Whether any kind of decomposition is applied, 3 postulates should be met

- Integrity - i.e.: The System before and after decomposition remains the same. („nothing is lost“)
- Consistency – i.e.: No part of the System is isolated. („we are able to connect it back together“)
- Balance (just recommendation)– i.e.: Decomposed parts are of comparable (not too different) complexity.

Types of decomposition

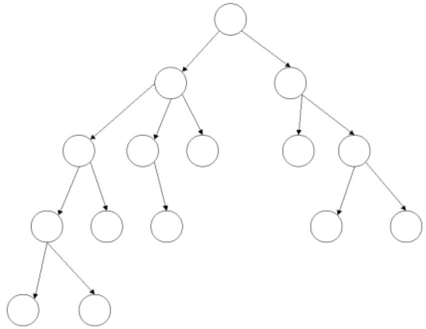
- **Topological decomposition** – the minimum number of joints of decomposed parts, or minimum sum of weights of these joints. To fulfil this criterion various modifications of Ford - Fulkerson's algorithm are successfully applied
- **Functional decomposition** – decomposed parts – subsystems have to be carriers of Macro-functions. Identification of macro-functions can be done via the concept of levels of distinguish. This decomposition is significantly dependent on the Subject's point of view; nevertheless it is frequently used as it reflects the aspect of minimal inference to Systems processes related to the level of decomposition.
- **Semantic decomposition** – the elements of decomposed part – Subsystem are carriers of pre – defined feature. This type of decomposition is de facto set up on the basis of the categorization of Systems elements. As a rule, sorting, clustering and fuzzy clustering algorithms are frequently utilized.
- **Hierarchic decomposition** – decomposition of hierarchical structure.
  - on (hierarchic) levels
  - on branches

Hierarchical structure

- The graph of hierarchic structure is tree

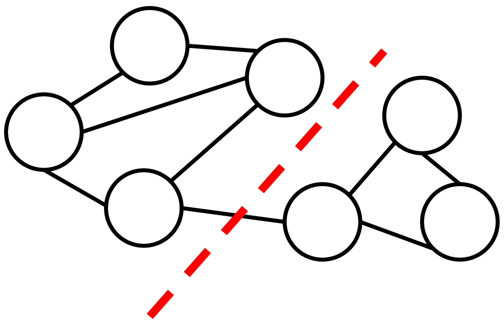
Hst := **(X, R)** $\Leftrightarrow \exists$ **r** (a, b)$\in$**R**;

- where **r** (a, b)$\in$**R** is relation representing subsidiarity/ superiority of couples of elements a, b$\in$ **X)**
- **$\exists$A$\subset$X,** where **(a $\in$A)$\cap$(b$\notin$A)** is subset of so-called i level of hierarchic structure
- **$\exists$B$\subset$X ,** kde **(b$\in$B)$\cap$(a$\notin$B)** is subset of so-called j level of hierarchic structure
- **(j $\neq$ i)**



Types of decomposition

- topological decomposition
  typically



- functional decomposition
  e.g. control function, economic, production, ...
- semantic decomposition
  e.g. based on colour, price, age, education, ... – anything described in the parameters of elements
- Hierarchic decomposition
  - on (hierarchic) levels – e.g. Top management, heads of departments, workers
  - on branches – e.g. To every employee all the superiors

Changes in the structure tasks

- Structure reduction
  - Element deletion (removal)
  - Integration of several elements into one
- Structure expansion
  - Adding an element
  - Disintetration an element into several elements

Element deletion (removal)

- Reason – it is useless
  - Nobody uses its outputs
  - There are no inputs available
  - Its function has changed in an unwanted matter
- Why not leave it in the system
  - Uses system sources
  - Increases complexity
  - Increases probability of errors
  - Increases probability of activation of undesirable processes

Integration

- Reasons
  - Homogenization
  - Introduction of universal elements
  - Too complex system
- Often done after decomposition
- Creation of new functions

Structure expansion

- Adding an element
  - Often as solution of irregular interface – adding a conversion element
- Disintegration
  - Reasons – need for detailed control
  - New functions originating
  - Feedback may appear
  - …

System goals task

- The goal of the System generally can be:
  - Outer goal – allocated to the neighborhood. It means either
    - setting the neighbourhood to a pre-definedstate
    - the initiation of specific process within in the neighborhood

  Expressed simply, this means a 'supply' of some entity from the system into the surroundings.
  - Internal goals – allocated to the interior of the system. They represent dynamic balance of the set of the goals of its components and subsystems.
    - Specific state of the System
    - Certain processes in the System defined in qualitative, quantitative and dynamic parameters.

Types of goals

- According to the length of the Systems path
  - near goals
  - distant goals
- According to time scale
  - Tactical
  - Operational
  - Strategic goals.
- According to ability to achieve
  - Achievable – if there is a path to the goal state and if there are sufficient system resources
  - unachievable

Specific goals

- The ordering goal – Its achievement is mediated by "ordering" / "self-ordering" processes, which are generally of “minimax” character and are therefore within the class of “strong processes”. (example – homeostatic process). The goals of this category imply that the control processes and system resources are managed in such a way that resources are optimally spent.
- Species survival goal (conservation of genetic code) - established under the influence of biology, its characteristics are nevertheless generally systemic
- Enforcement of identity goal

Goals sources

- inside the system
- in the near neighbourhood
- in the far neighbourhood

Methods of goals generation

- exact, "hard": characterized by rigorous rules. They use logical and mathematical approaches, mathematical modelling, and operations research,
- exact "semi-hard": the rules-making goals continue to be exact, but they are based on the methods of coping with uncertainty, such as fuzzy approaches, genetic algorithms and neural networks,
- soft: methods based on subjective experience (heuristic approach, "brainstorming").

Thank you for your attention