TASK ANALYSIS IN ID

LECTURE 4

Agenda

- Task Analysis – goals, method
- Hierarchical Task Analysis (HTA)
- Groupware Task Analysis (GTA)
- The Bridge Method
- Concur Task Trees (CTT) notation
- Task Analysis Tools: Euterpe, CTTE
- A case study

TASK ANALYSIS

Task Analysis

- fundamental methodology in the assessment and reduction of human error

- Nearly all task analysis techniques provide, as a minimum, a description of the observable aspects of operator behavior at various levels of detail, together with some indications of the structure of the task - **action oriented approaches.**

- Other techniques focus on the mental processes which underlie observable behavior, e.g. decision making and problem solving - **cognitive approaches**

Task Analysis

- What: Analysis of a task in terms of its cognitive, motor, and perceptual aspects.

- Why:
  - To understand how people work.
  - To systematically examine the tasks that a user will perform on a new or existing system.
  - Task analysis for a new system forms the basis of the design for user interaction.

- How: Using a variety of data collection methods and task description techniques.

What is TA?

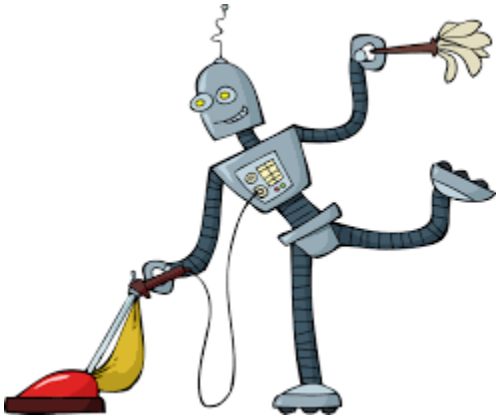
Methods to analyze people's jobs:

- what people **do**
- what **things they work with**
- what they must **know**

Goals of Task Analysis

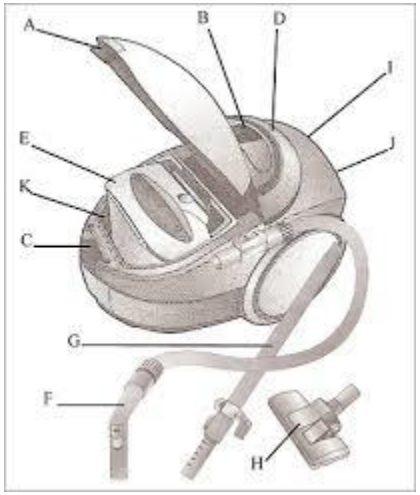
- Elicit descriptions of **what people do**
- Represent those descriptions
- Predict **difficulties**, **performance**
- Measure **learnability**, transfer of knowledge between systems
- **Evaluate** systems against usability and/or functional requirements

An Example



- in order to clean the house
  - get the vacuum cleaner out
  - fix the appropriate attachments
  - clean the rooms
  - when the dust bag gets full, empty it
  - put the vacuum cleaner and tools away

- must know about:
  - vacuum cleaners, their attachments, dust bags, cupboards, rooms etc.



Approaches to Task Analysis

- Task decomposition
  - splitting task into (ordered) subtasks
- Knowledge based techniques
  - what the user knows about the task and how it is organised
- Entity/object based analysis
  - relationships between objects, actions and the people who perform them
- lots of different notations/techniques

General Method

- observe
- collect unstructured lists of words and actions
- organize using notation or diagrams

Differences from other techniques

Systems analysis			**vs.**	**Task analysis**
system design	-	focus	-	the user

Cognitive models			**vs.**	**Task analysis**
internal mental state	-	focus	-	external actions
practiced `unit' task	-	focus	-	whole job

Task Decomposition

Aims:

- describe the actions people do
- structure them within task subtask hierarchy
- describe order of subtasks

Variants:

- Hierarchical Task Analysis (HTA)
  - most common
- **GTA (Vrije University, Amsterdam)**
- CTT (CNUCE, Pisa)
  - uses LOTOS temporal operators

Hierarchical Task Analysis (HTA)

HTA

- Hierarchical Task Analysis is a systematic method of describing how work is organized in order to meet the overall objective of the job.

- It involves identifying in a top down fashion the overall goal of the task, then the various sub-tasks and the conditions under which they should be carried out to achieve that goal.

- complex planning tasks can be represented as a hierarchy of **operations** - different things that people must do within a system and **plans -** the conditions which are necessary to undertake these operations.

HTA

- One of the most common task analysis techniques
- Recursively break task down into subtasks
- Describes task in terms of:
  - goals
  - operations
  - plans

- Goals (what person is seeking to achieve)
- Operations (activities to meet goals)
- Plans (conditions under which operations are carried out)

Textual HTA description

Hierarchy description ...

0. in order to clean the house
    1. get the vacuum cleaner out
    2. get the appropriate attachment
    3. clean the rooms
        3.1. clean the hall
        3.2. clean the living rooms
        3.3. clean the bedrooms
    4. empty the dust bag
    5. put vacuum cleaner and attachments away

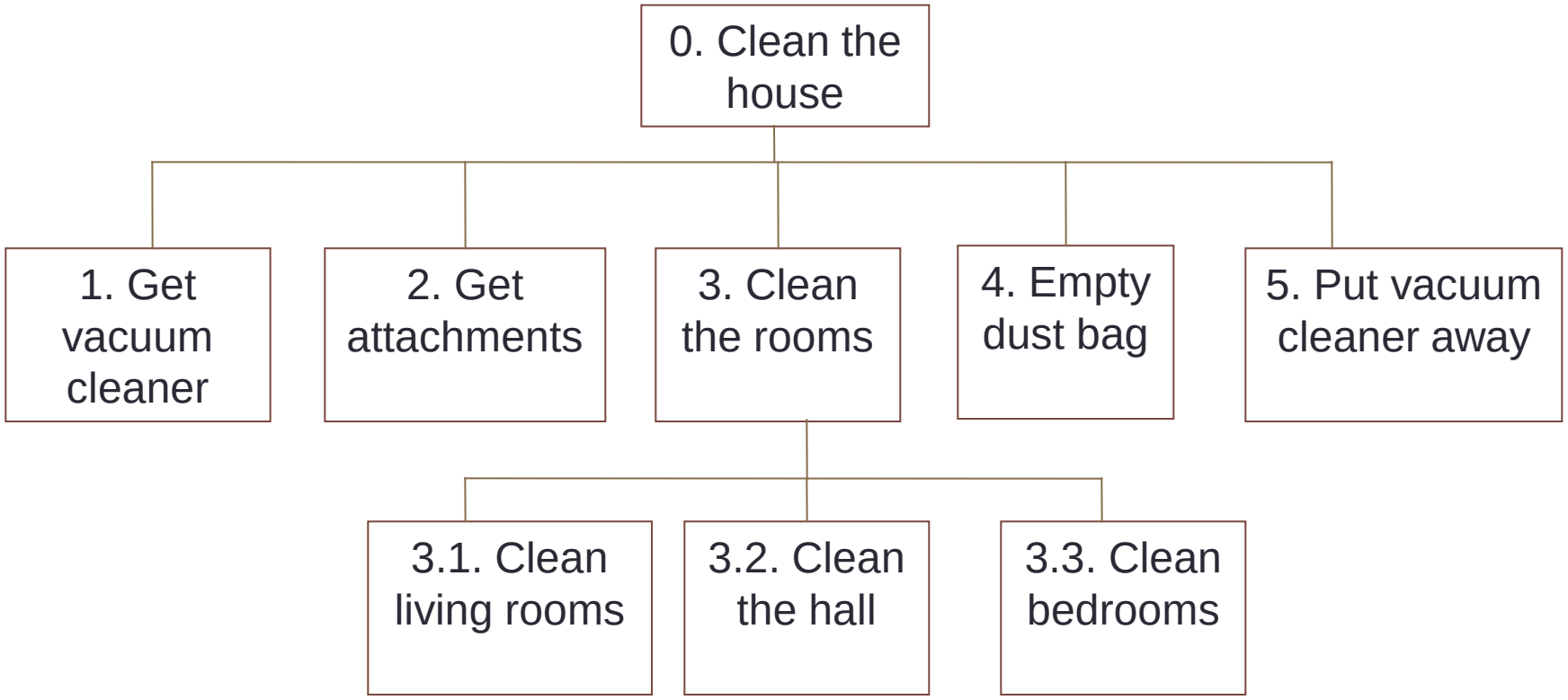
... **and plans**

Plan 0: do 1 - 2 - 3 - 5 in that order; when the dust bag gets full do 4

Plan 3: do any of 3.1, 3.2 or 3.3 in any order depending on which rooms need cleaning

only the plans denote order

HTA Diagrammatic Description



Plan 0: do 1 - 2 - 3 - 5 in that order. When the dust bag gets full do 4

Plan 3: do any of 3.1, 3.2 or 3.3 in any order depending on which rooms need cleaning

Generating the Hierarchy

1. get list of tasks
2. group tasks into higher level tasks
3. decompose lowest level tasks further

Stopping rules

How do we know when to stop?

Is “empty the dust bag” simple enough?

Purpose: expand only relevant tasks

Motor actions: lowest sensible level

HTA

- **Advantages**
  - HTA is a simple and flexible method that does not depend on a methodological context.
  - HTA enables the representation of a task hierarchy that could be further detailed.
  - Although HTA is task oriented and to some extent user oriented it still maintains a strong relationship with traditional software engineering.
  - HTA provides information like inefficiencies in tasks, that can be used for developing product requirements.

HTA

- **Disadvantages**
  - There are no strict rules for creating an HTA diagram so different analysts will generate inconsistent hierarchies at varying levels of detail.
  - HTA requires both training and experience. It is not a tool that can be applied immediately.
  - HTA is not a predictive tool. It focuses on existing tasks.
  - HTA diagrams can become quite complex
  - Concurrent tasks and tasks that overlap cannot be described
  - Interruption hard to express in diagrams

Groupware Task Analysis

Designing for Groupware

- GTA – Groupware Task Analysis – modeling framework for task knowledge
- Task – an activity performed to reach a certain goal
- Goal – a desired state in the system or task world
- Sometimes it is hard to make the distinction between task and goal
- A task changes something

Gerrit van der Veer

Martijn van Welie

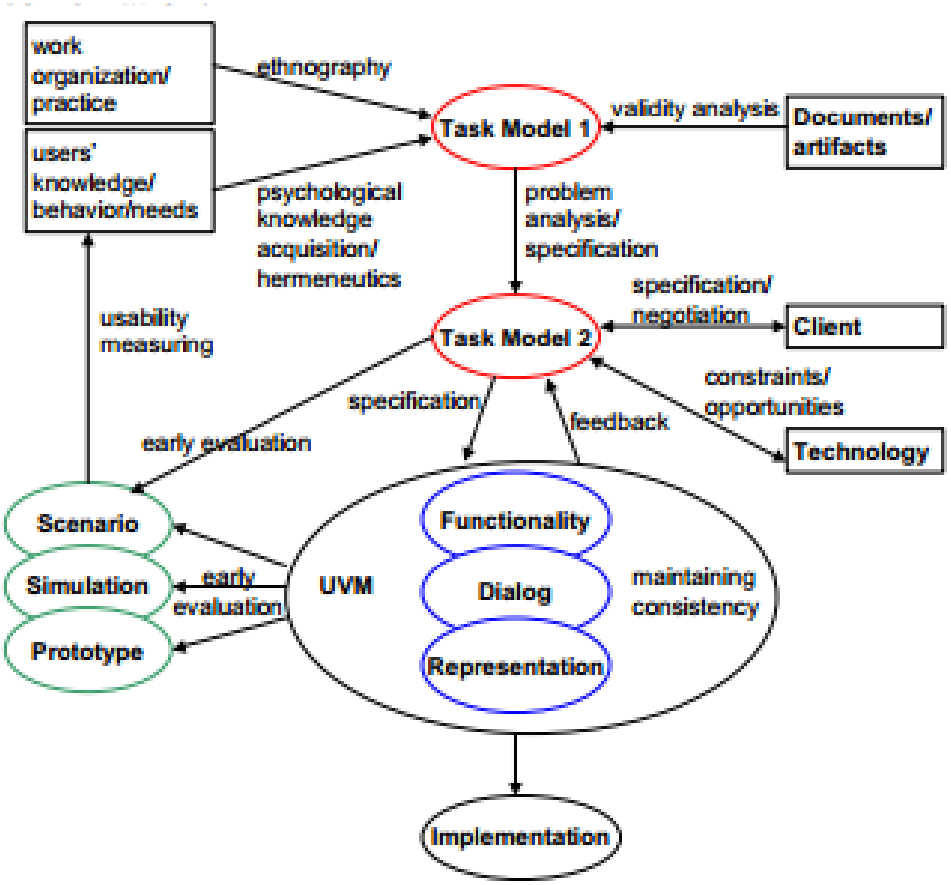
GTA

- Emphasis on modeling a group or organization and their tasks
- Is a conceptual framework describing essential things when designing for groupware
- Analyzing a complex system:
  - **users**
  - **tasks**
  - Devices (hard, **soft**)
  - **Social environment**
  - **Physical environment**

GTA

- 3 steps:
  - Analyzing current task situation - **Task Model 1**
  - envisioning a task situation for which information technology is to be designed - **Task Model 2**
  - specifying the semantics of the information technology to be designed - **The user's virtual machine**

The design process (van Welie)



GTA concepts

- **Object -** used to transfer information between agents
- Objects identification may be performed using interviews (identify nouns in relation to task description)
- **Agent -** an entity that is considered active. Usually agents are humans, but groups of humans or software components may also be considered agents. Agents perform tasks and always play certain *roles* within the task world.
- **Role -** a meaningful collection of tasks performed by one or more agents. The role is meaningful when it has a clear goal or when it distinguishes between groups of agents.
- A role is consequently *responsible* for the tasks that it encompasses.

GTA concepts

- **Task -** an activity performed by agents to reach a certain goal.
- typically changes something in the task world and requires some period of time to complete.

- Complex tasks can be decomposed into smaller subtasks

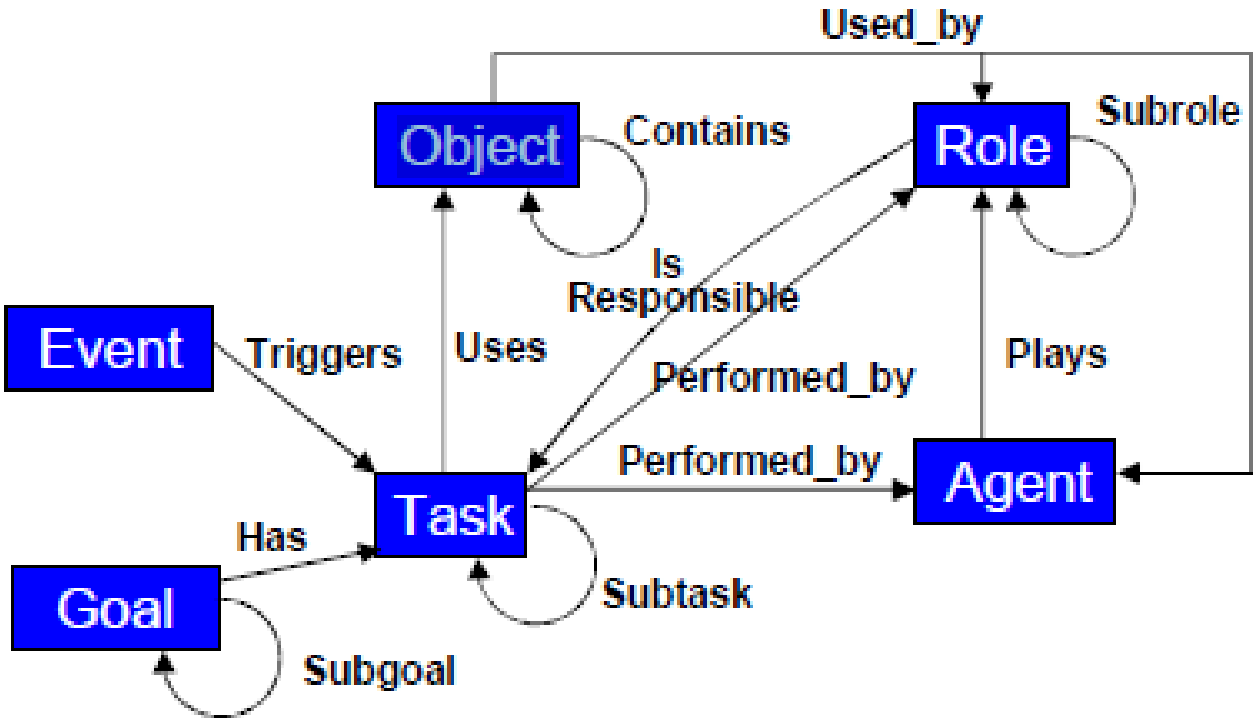
- Tasks are executed in a certain order and the completion of one task can *trigger* the execution of one or more other tasks.

- A task could also be started because an event has occurred in the task world.

GTA concepts

- **Event -** a change in the state of the task world at a point in time.
- The change may reflect changes of attribute values of internal concepts such as Object, Task, Agent or Role or could reflect changes of external concepts such as the weather or electricity supply.
- Events influence the task execution sequence by *triggering* tasks.
- This model does not specify how the event is created or by whom.

GTA Ontology



Analyzing the current task situation (Task model 1)

- The design of a new product triggered by a current task situation (not optimal or improvements are expected by introducing new technology)

- Task analysis helps formulate design requirements and later on evaluation

- Task model 1 describes the current real situation by **observing** or **asking** the people who are involved

Tasks as explanation

- imagine asking the user the question:
  what are you doing now?

for the same action the answer may be:

typing ctrl-B

making a word bold

emphasising a word

editing a document

writing a letter

Envisioning the future task situation (Task Model 2)

- redesign of the task structure in order to include technological solutions for problems and technological answers to requirement

- **Problems** identified in TM1:
  - **Task structure not optimal** (high frequency, redundant, too many subtasks)
  - **Differences** between the formal and actual task performance
  - **Inefficient interaction** in organization
  - **Inconsistencies** in tasks
  - People are doing things they are **not allowed to do**

Specifying technology (The user's virtual machine)

- detailed description of the system as far as it is of direct relevance to the end-user

- "**virtual machine**" - indicate "the functionality of the system ... where implementation details and details of the underlying hardware are suppressed“

- **user's virtual machine** (UVM) indicates the total of user relevant knowledge on the technology, both semantics (what the system offers the user for task delegation) and syntax (how task delegation to the system has to be expressed by the user)

Detailed design

- Results of task analysis and modeling – used to create the UI
- Aspects to be considered:
  - Functionality
  - Dialog structure
  - Presentation
- Gap between analysis and design
  - Analysis results: detailed description of the domain and aspects to be improved (design goals)
  - Design : create a solution that meet the requirements (design goals)
  - Engineering+ *Creativity* needed

Detailed design

- The GAP refers to:
  - What are the main displays
  - Which data must be represented and which are merely attributes
  - Which is the appropriate interaction style
  - How should the user navigate through the interface structure
  - How is the functionality accessible

    +
  - Technological constraints and client wishes

- In practice: initial design – evaluation - iteration

The Bridge

- A comprehensive methodology for
  - understanding user needs
  - identifying users' conceptual building blocks for their tasks
  - building GUI prototypes from the building blocks
  - testing the results with actual users

- Originally developed at Bellcore

Bellcore
Bell Communications Research

Bridging the Gap

- Bridging the gap between user requirements and GUI design
- The problem:
  - How do we turn our understanding of users into successful systems?

Detailed design

- Guidelines for bridging the gap
  - Express user requirements in task flows
  - Mapping task flows to task objects
    - Identify which task objects need to be included in the system
    - Identify the relevant attributes of the task objects
    - Identify relevant actions on task objects (actions can be ordered in menus)
    - Identify groups of attributes st only the relevant task attributes are shown while performing a task – views
    - Identify object containment relationships – screens
  - Mapping task objects to GUI objects using a specific platform style

The Bridge

- Participatory design method
- An object-oriented design method
- Derives *task objects* – objects that support user tasks and make sense to users – from user tasks and uses them as building blocks
- Task objects have attributes, actions, and containment relationships, as in object-oriented programming

Case Study: Shopping

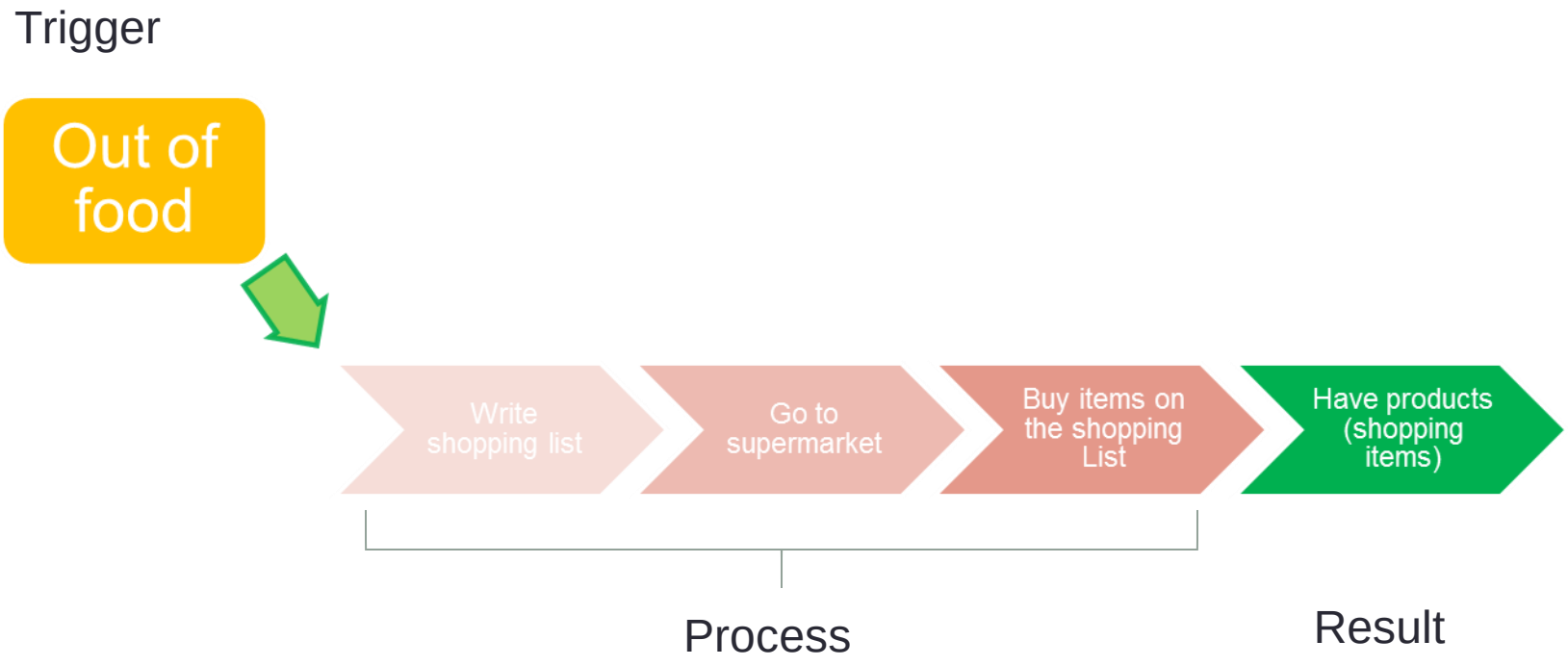
- Tasks:
  - Write shopping list
  - Bring list to shop
  - Buy things on the list

- Task objects:
  - Shopping list
  - Shopping items (products)
  - Supermarket

The Bridge – Part 1

- Task analysis – understand user needs
- Describe current tasks as task flows
- Begin with high-level Current Big Picture task flow
- Identify trigger and result of task flow

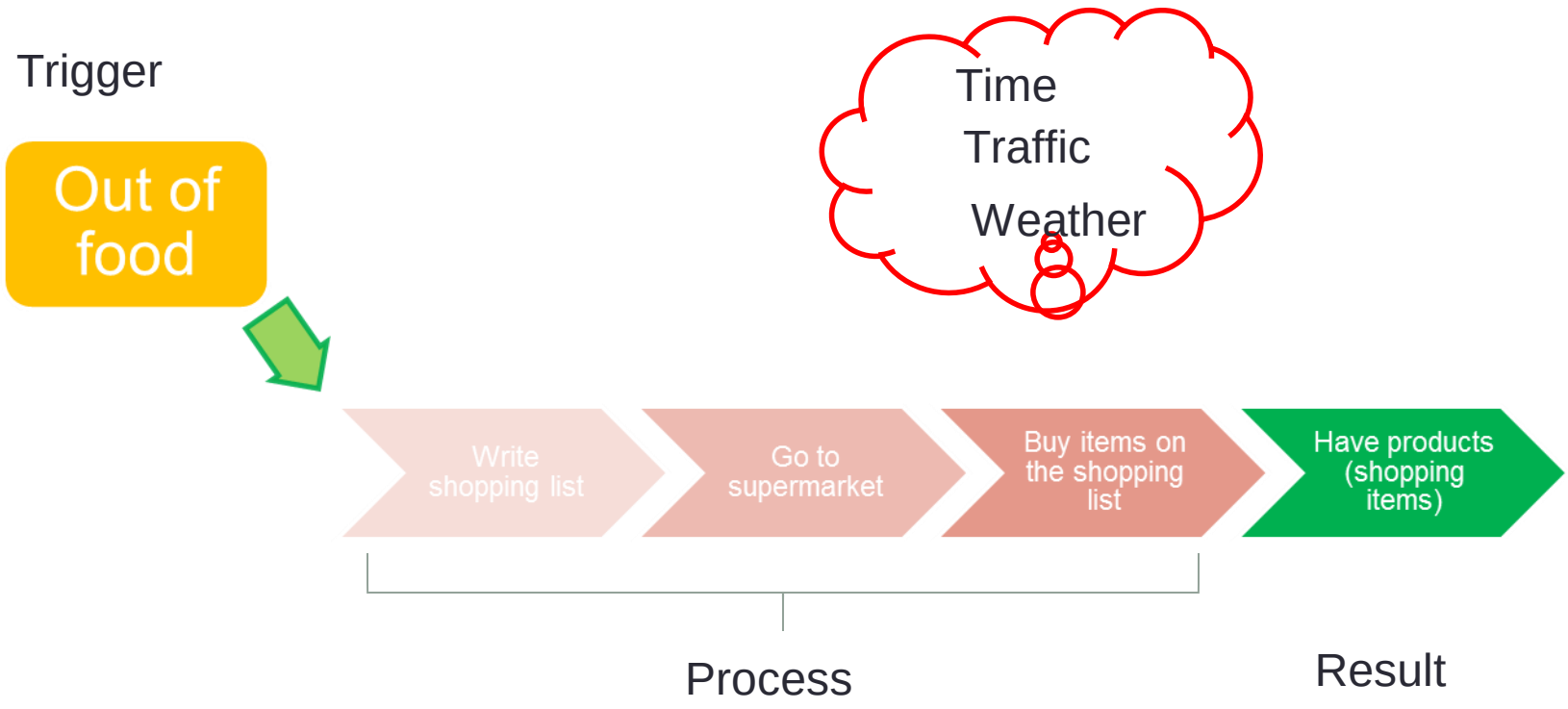
The Bridge – Case Study



The Bridge – Part 1

- Task analysis (continued)
- Identify problems associated with tasks
- Scoping: Agree on what parts to address in this design session

The Bridge – Case Study



The Bridge – Part 1

- For each problem, agree on priority (high, medium, or low) for solving it
- Brainstorm “blue sky” ideal task flow that addresses the problems of current tasks
- No criticism during brainstorming
- Consider radical solutions without regard for feasibility
- After brainstorming, agree on the desirability and feasibility (high,medium, or low) of each part

The Bridge – Case Study

- Eliminates driving
- Saves time
- Comfort
- Highly desirable
- Highly feasible

Trigger

Out of food

Process

Result

The Bridge – Part 1

- Construct realistic task flows for the new system with as many desirable features of the ideal tasks as possible

- After creating each task flow, agree on what is in scope for this design session

The Bridge – Part 2

- Part 2: Task object design –identify users' conceptual building blocks
- Write down all nouns that appear in the realistic task flows
- For each noun, write down its attributes: properties, such as its name, and any objects it contains
- Some nouns will emerge as task objects users need to work with, others as properties of objects

The Bridge – Part 2

Shopping list	Product	Supermarket	**Identity**
Items (products)	Name Description Quantity Price	Name Location Products	**Attributes**

The Bridge – Part 2

- For each object, identify actions that users (not the computer) perform on the object

- For each object, identify any other objects it's contained in (parent objects) and any objects it contains (child objects)

The Bridge – Part 2

Shopping list	Product	Supermarket	**Identity**
Items (products)	Name Description Quantity	Name Location Products	**Attributes**
Create Edit Destroy	Add to shopping cart Remove from shopping cart	Pay	**Actions**

I'm in	In me
	products

I'm in	In me
supermarket	

I'm in	In me
	products

Containment

The Bridge – Part 3

- Part 3: Mapping task objects to GUI objects – build a prototype

- Use style rules for the mapping, such as

  - ✦ **A task object** is a conceptual unit, so put it in its **own window**
  - ✦ Put the **object's actions** in the window's **menu bar** and **tool bars**
  - ✦ Put the **object's attributes** in the client area of the window

Mockup

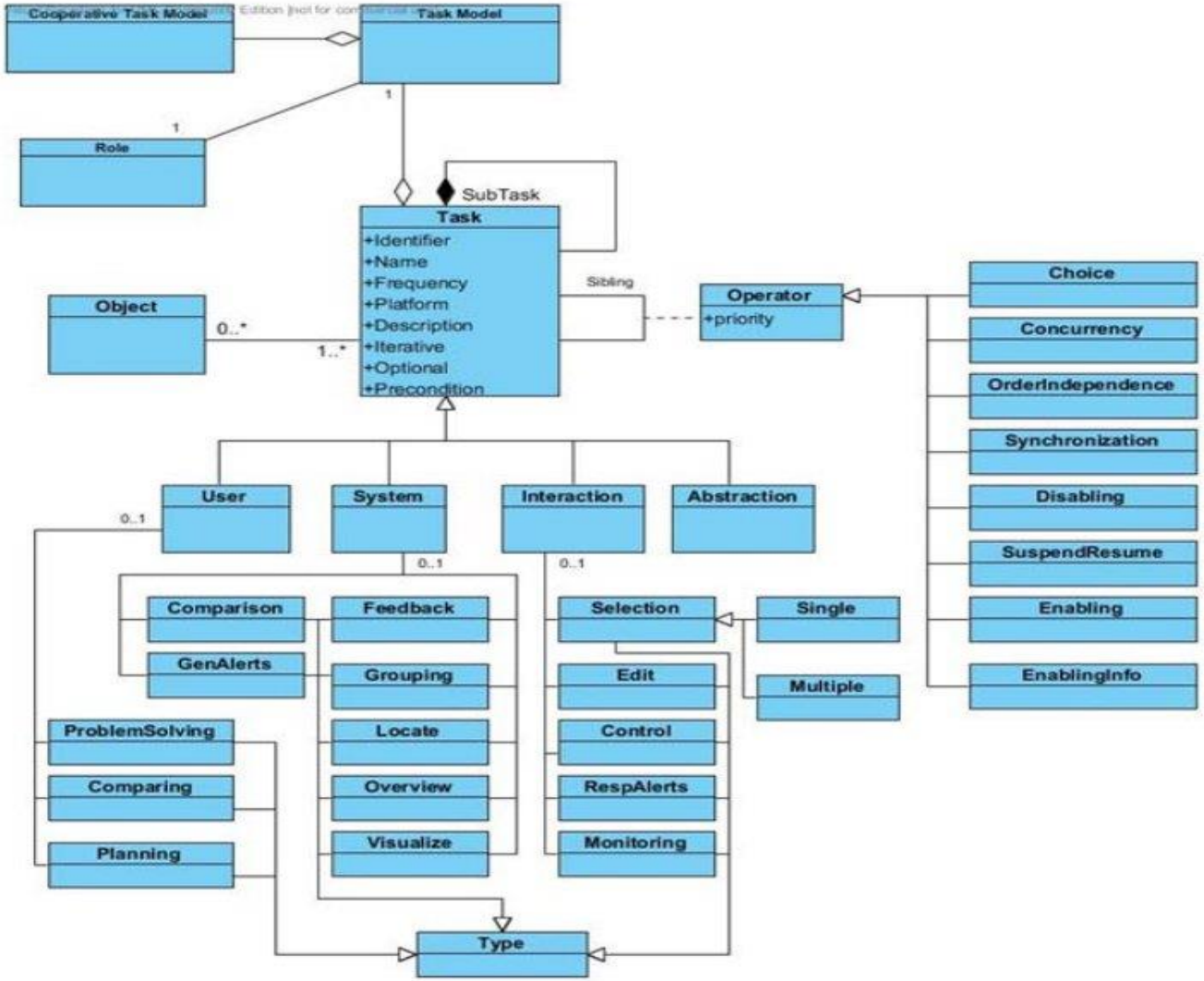

ShoppingList
File
Edit
Product
Description
Quantity
Purchased
Apple
Red
1 kg
Chicken
Amaretto
1
Add
Delete
Close

CONCUR TASK TREES

CTT notation (Fabio Paterno)

- notation for task model specifications
- developed to overcome limitations of notations previously used to design interactive applications.
- main purpose - an easy-to-use notation that can support the design of real industrial applications (medium-large dimensions)
- Main features:
  * Hierarchical structure
  * Concurrent notation – LOTOS operators
    * Focus on activities

ConcurTaskTrees metamodel – CTT



ConcurTaskTrees – CTT

- **Abstract task**

- **Application task**

- **User task**

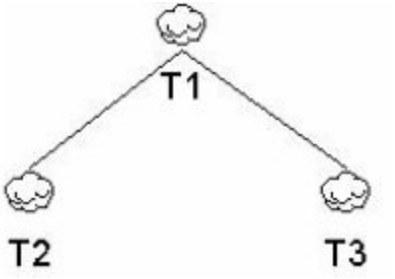
- **Cooperative task**

- **Interaction task**

CTT notations

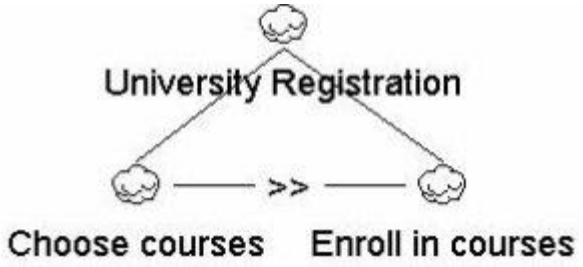
- Tasks at same level represent different options or different tasks at the same abstraction level that have to be performed.

- “In order to do T1, I need to do T2 and T3”, or “In order to do T1, I need to do T2 or T3



CTT temporal operators

- Enabling

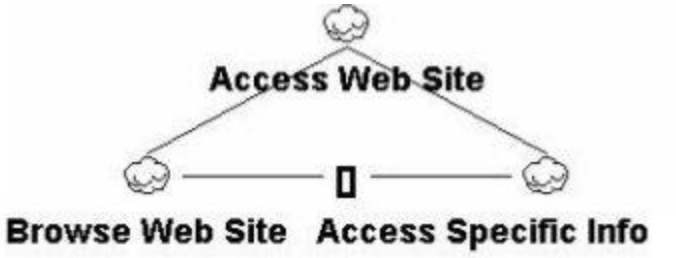


- Specifies second task cannot begin until first task performed.

- Example: I cannot enroll at university before I have chosen which courses to take.

CTT temporal operators

- Choice

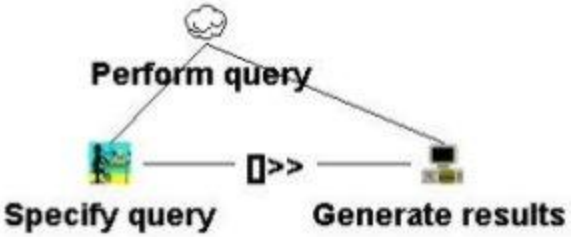


- Specifies two tasks enabled, then once one has started the other one is no longer enabled.

- Example: When accessing a web site it is possible either to browse it or to access some detailed information.

CTT temporal operators

- Enabling with information passing

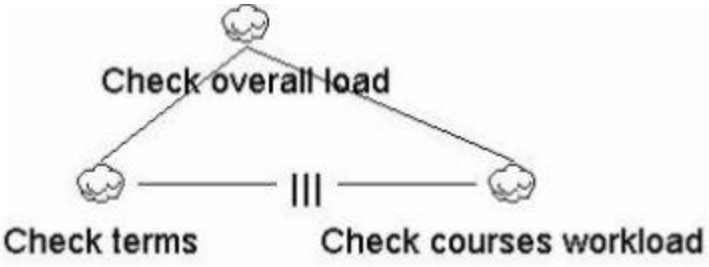


- Specifies second task cannot be performed until first task is performed, and that information produced in first task is used as input for the second one.

- Example: The system generates results only after that the user specifies a query and the results will depend on the query specified.

CTT temporal operators

- Concurrent tasks

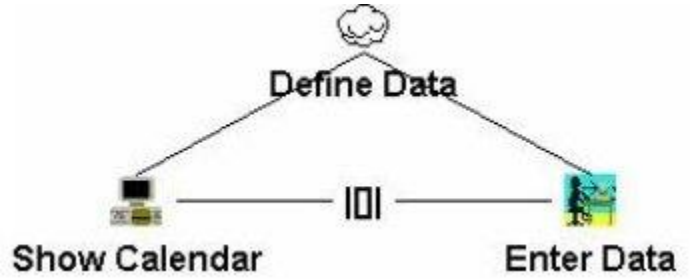


- Tasks can be performed in any order, or at same time, including the possibility of starting a task before the other one has been completed.

- Example: In order to check the load of a set of courses, I need to consider what terms they fall in and to consider how much work each course represents

CTT temporal operators

- Concurrent communicating tasks

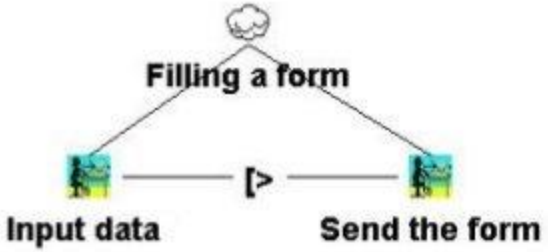


- Tasks that can exchange information while performed concurrently.

- Example: An application where the system displays a calendar where it is highlighted the data that is entered in the meantime by the user.

CTT temporal operators

- Disabling



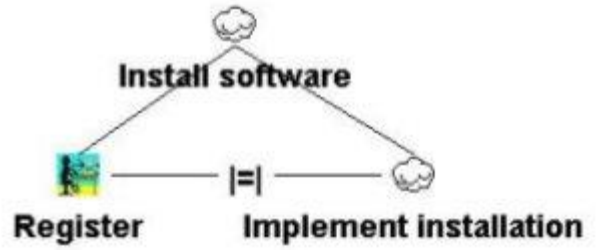
- The first task (usually an iterative task) is completely interrupted by the second task.

- Example: A user can iteratively input data in a form until the form is sent.

-

CTT temporal operators

- Task independence

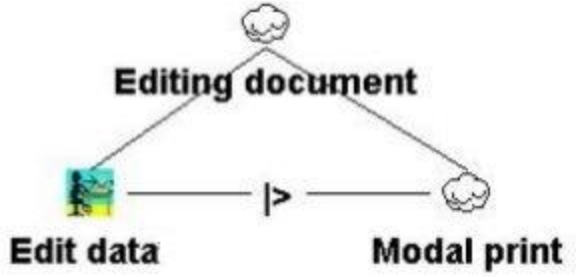


- Tasks can be performed in any order, but when one starts then it has to finish before the other one can start.

- Example: When people install new software they can start by either registering or implementing the installation but if they start one task they have to finish it before moving to the other one.

CTT temporal operators

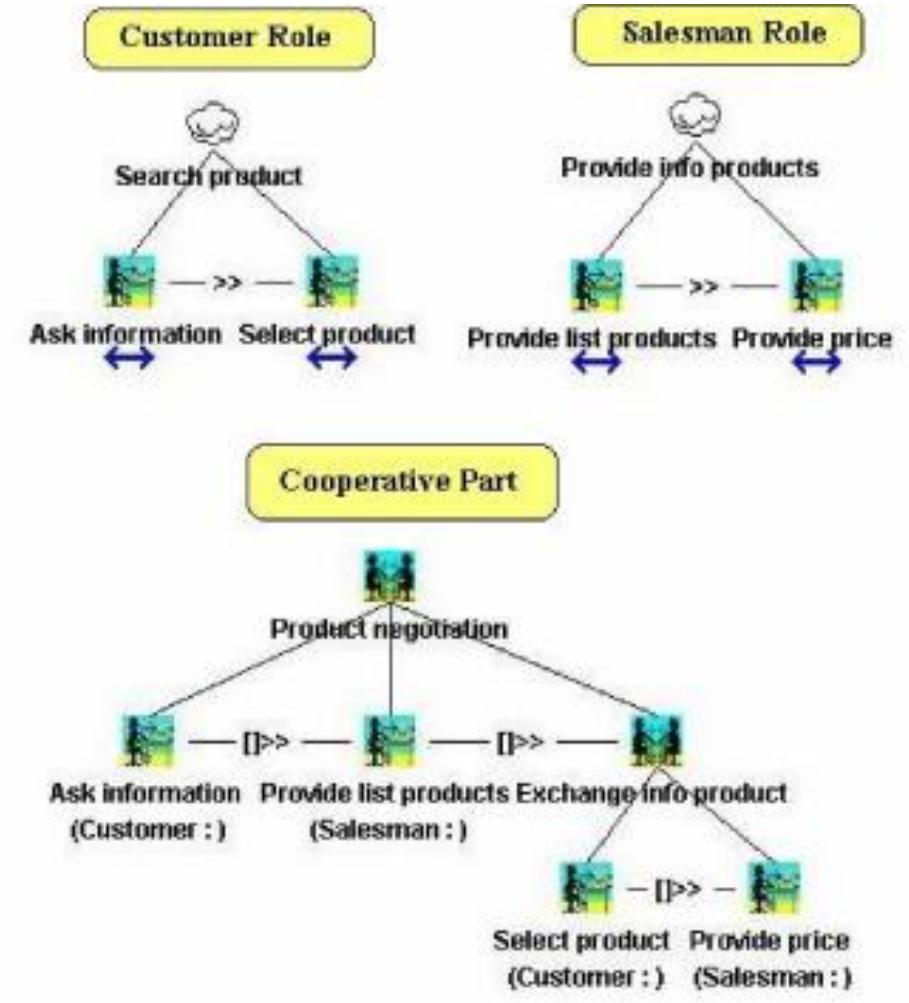
- Suspend-Resume



- First task can be interrupted by the second one. When the second terminates then the first one can be reactivated from the state reached before

- Example: Editing some data and then enabling the possibility of printing them in an environment where when printing is performed then it is no possible to edit.

CTT for cooperative tasks



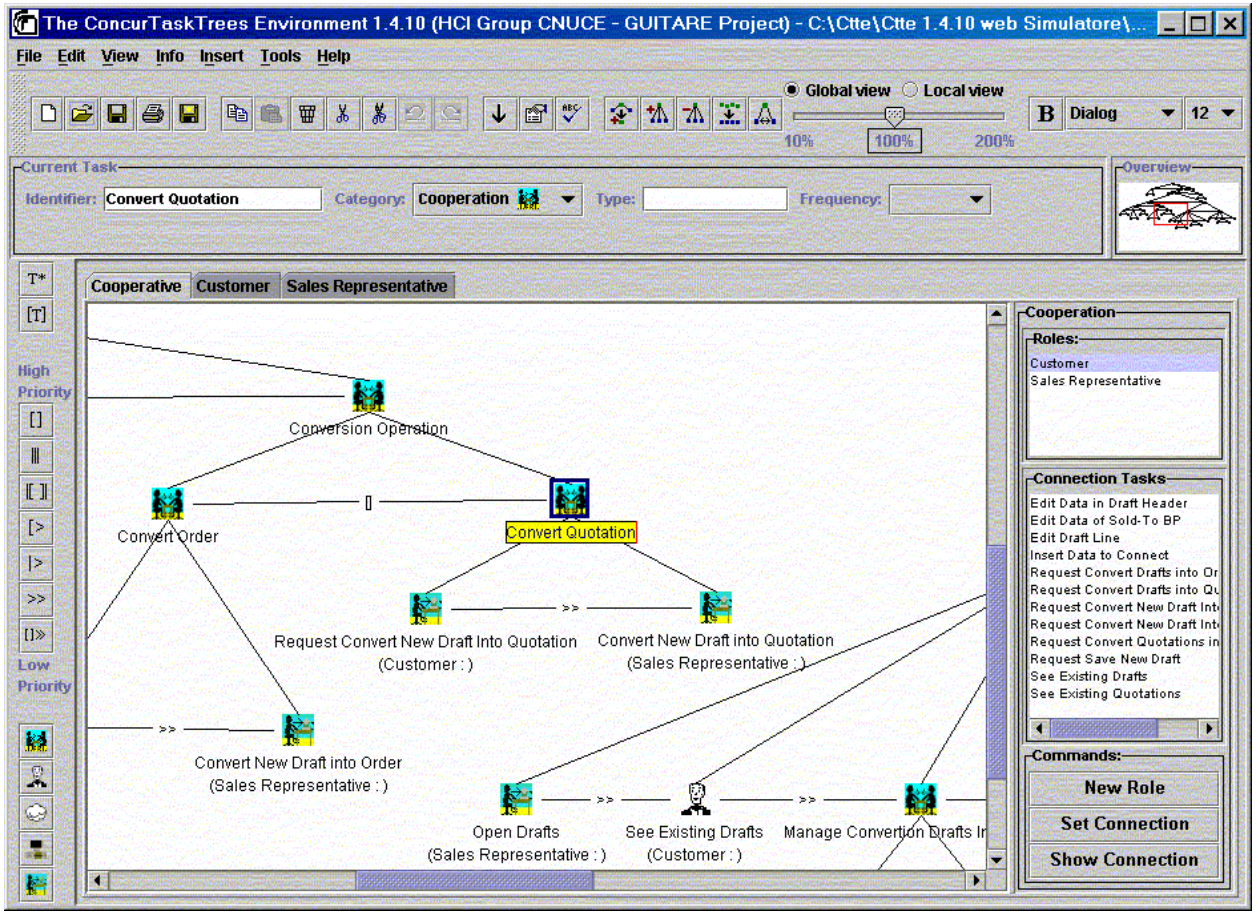
TASK ANALYSIS TOOLS

EUTERPE

CTTE

CTTE – CTT Environment

- http://giove.cnuce.cnr.it/ctte.html



Case Study

Interactive system for job evaluation using the points method

Job evaluation

- Goal
  - Create a hierarchy of jobs in an organization

Points Evaluation Method

- A set of *compensable* factors are identified as determining the worth of jobs.

- Typically the compensable factors include the major categories of:

  - Skill
  - Responsibilities
  - Effort
  - Working Conditions

Points Evaluation Method - Factors

- Skill
  - Experience
  - Education
  - Ability
- Responsibilities
  - Fiscal
  - Supervisory
- Effort
  - Mental
  - Physical
- Working Conditions
  - Location
  - Hazards
  - Extremes in Environment

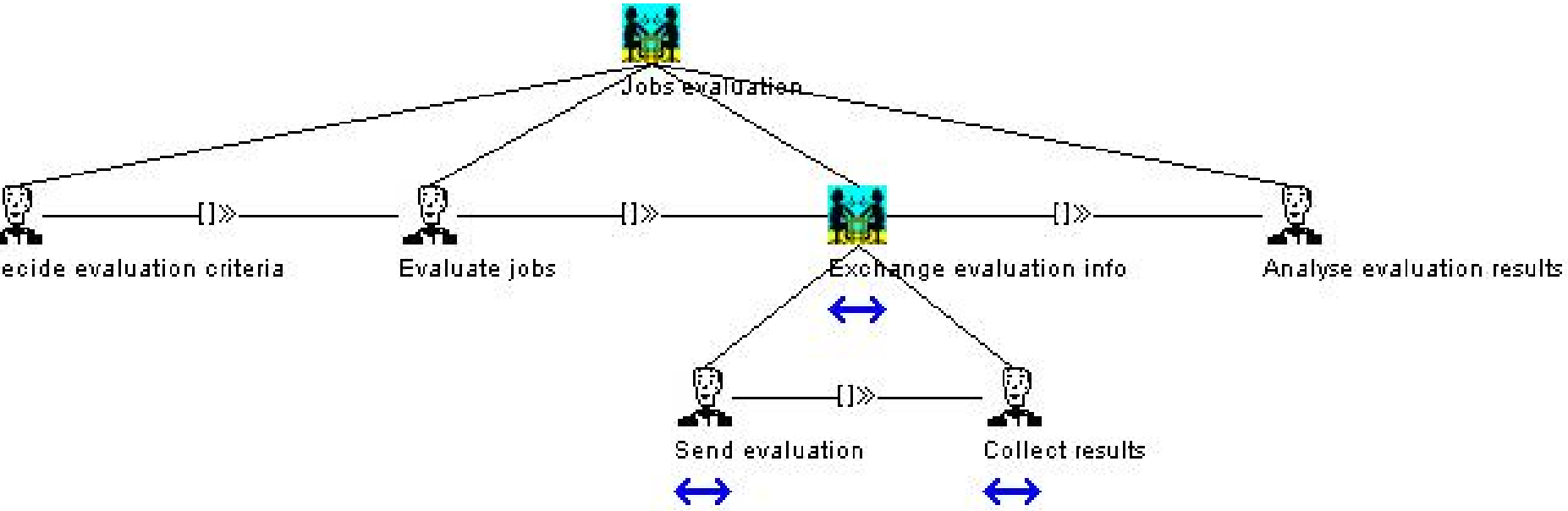
Points Evaluation Method

- Each factor is then divided into levels or degrees which are then assigned points by experts (evaluators).
- Each job is rated using the job evaluation instrument.
- The points for each factor are summed to form a total point score for the job.
- Jobs are then grouped by total point score and assigned to wage/salary grades so that similarly rated jobs would be placed in the same wage/salary grade.

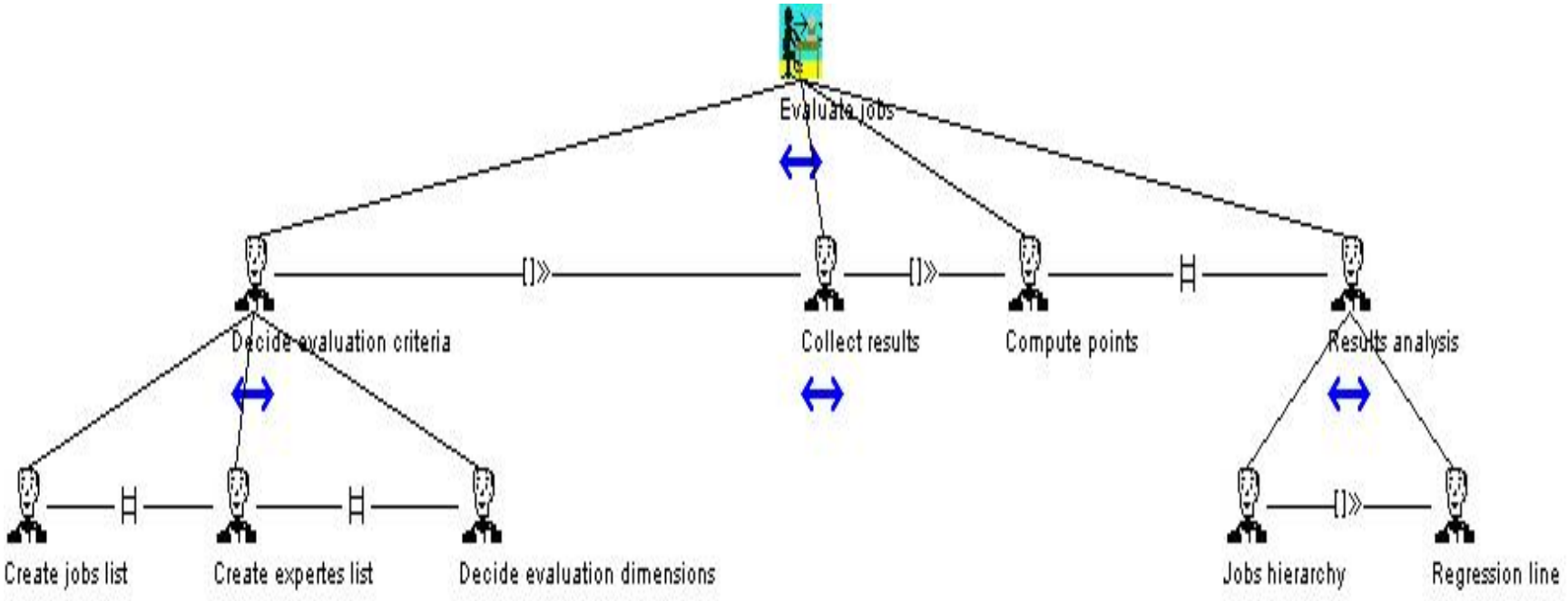
GTA concepts

- Agents: psychologist, experts (evaluators)
- Objects: organogram, list factors, job list, evaluation form, market analysis, evaluation instructions, job description
- Tasks: job inventory, job evaluation (follow instructions, decide points for each factor in job evaluation), compute sums, averages, create job hierarchy, regression line

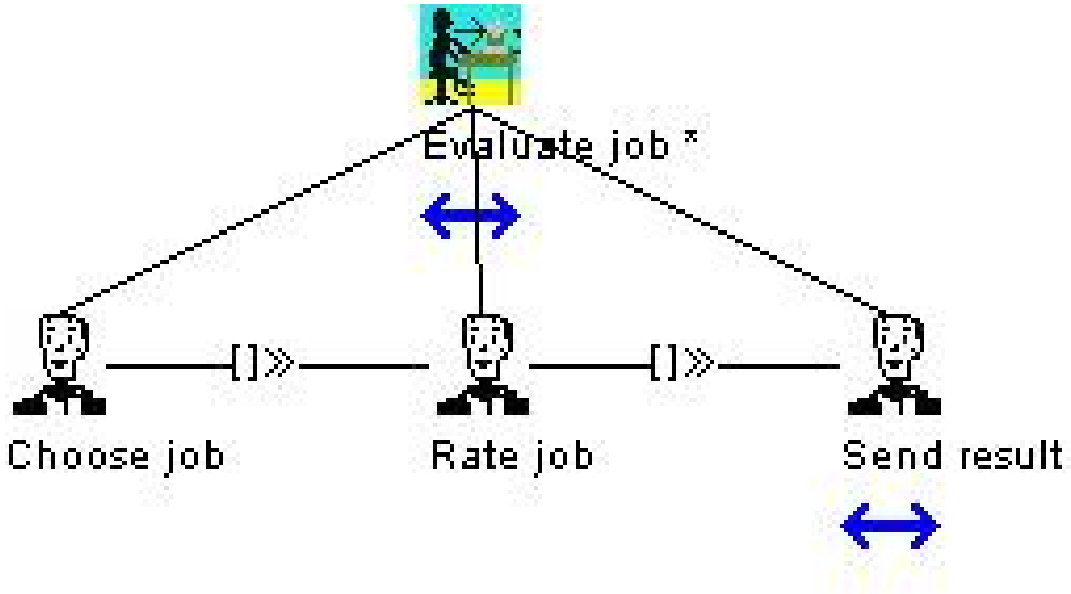
Cooperative TM 1



Psychologist TM1



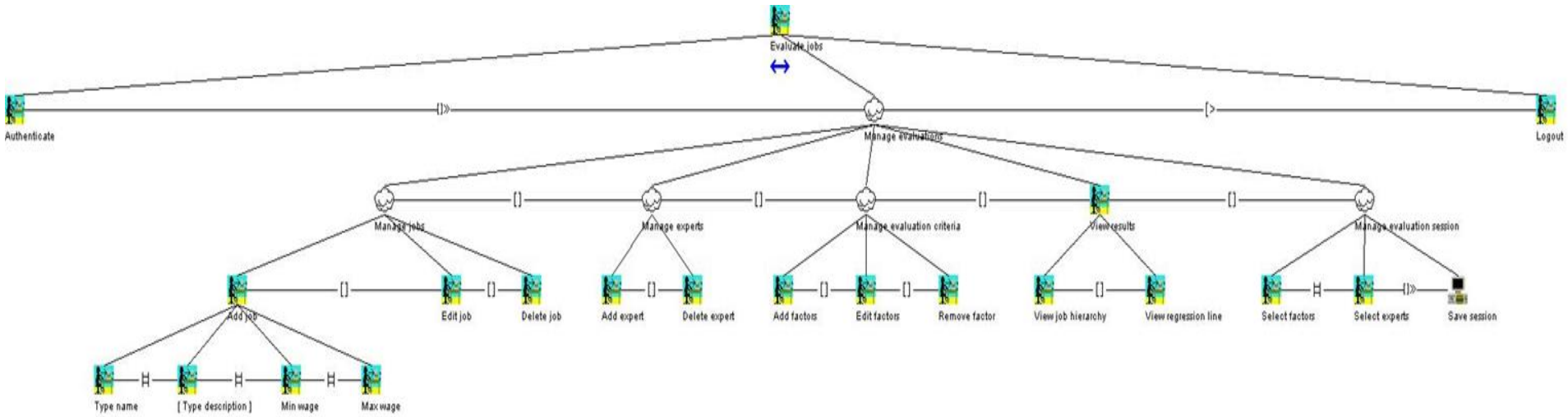
Expert TM1



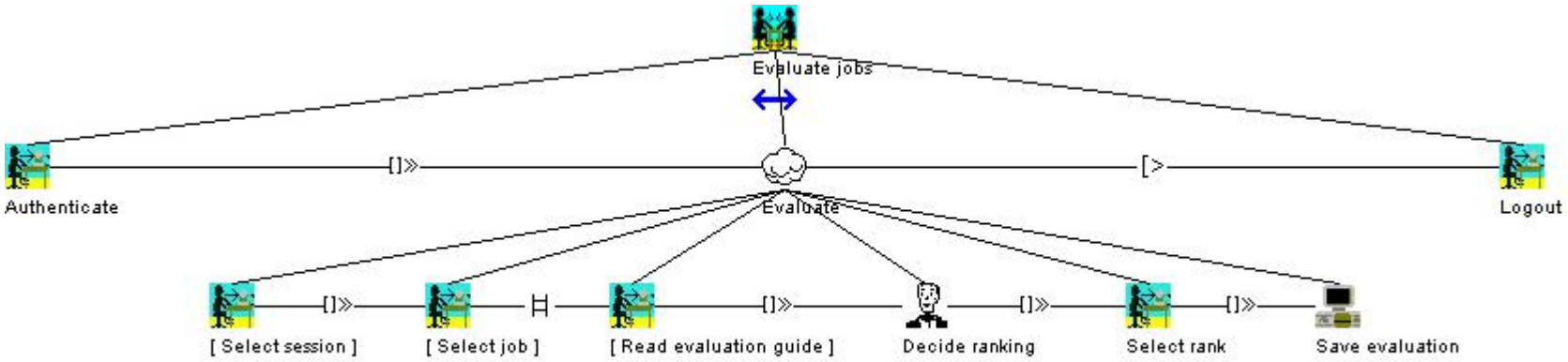
Identified problems in TM1

- Psychologist
  - workload – collect and process evaluations – compute averages, create job hierarchy, draw regression line
- Expert
  - Read evaluation guidelines from various sources, manage evaluations (how many jobs, what was evaluated…)
- Documents transfer between participants

TM 2 psychologist



TM 2 expert


Evaluate jobs
Authenticate
[]»
Evaluate
[>
Logout
[Select session]
[]»
[Select job]
[Read evaluation guide]
[]»
Decide ranking
[]»
Select rank
[]»
Save evaluation

Psychologist functionality

Jobs

Evaluators

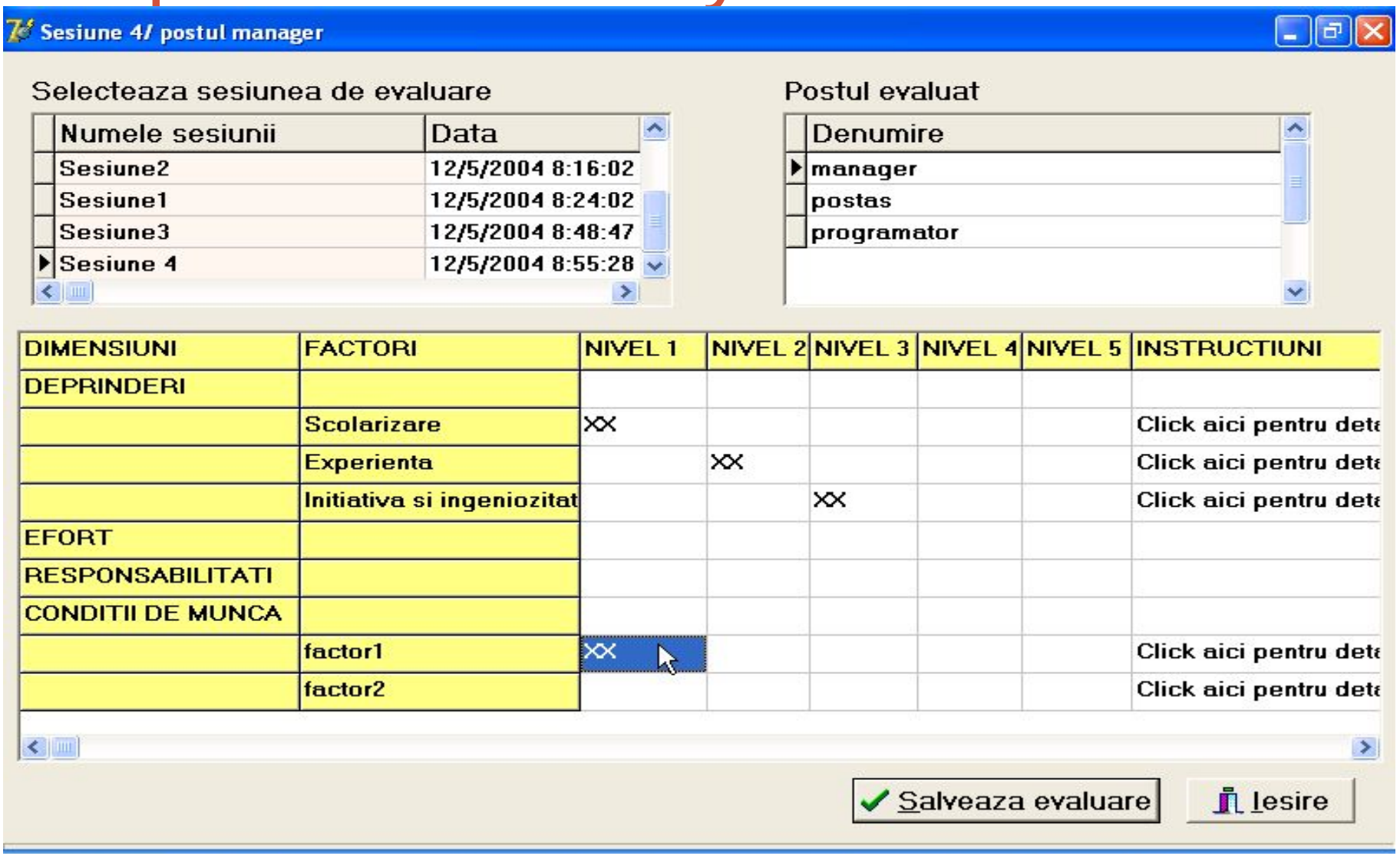
Factors

Evaluation session

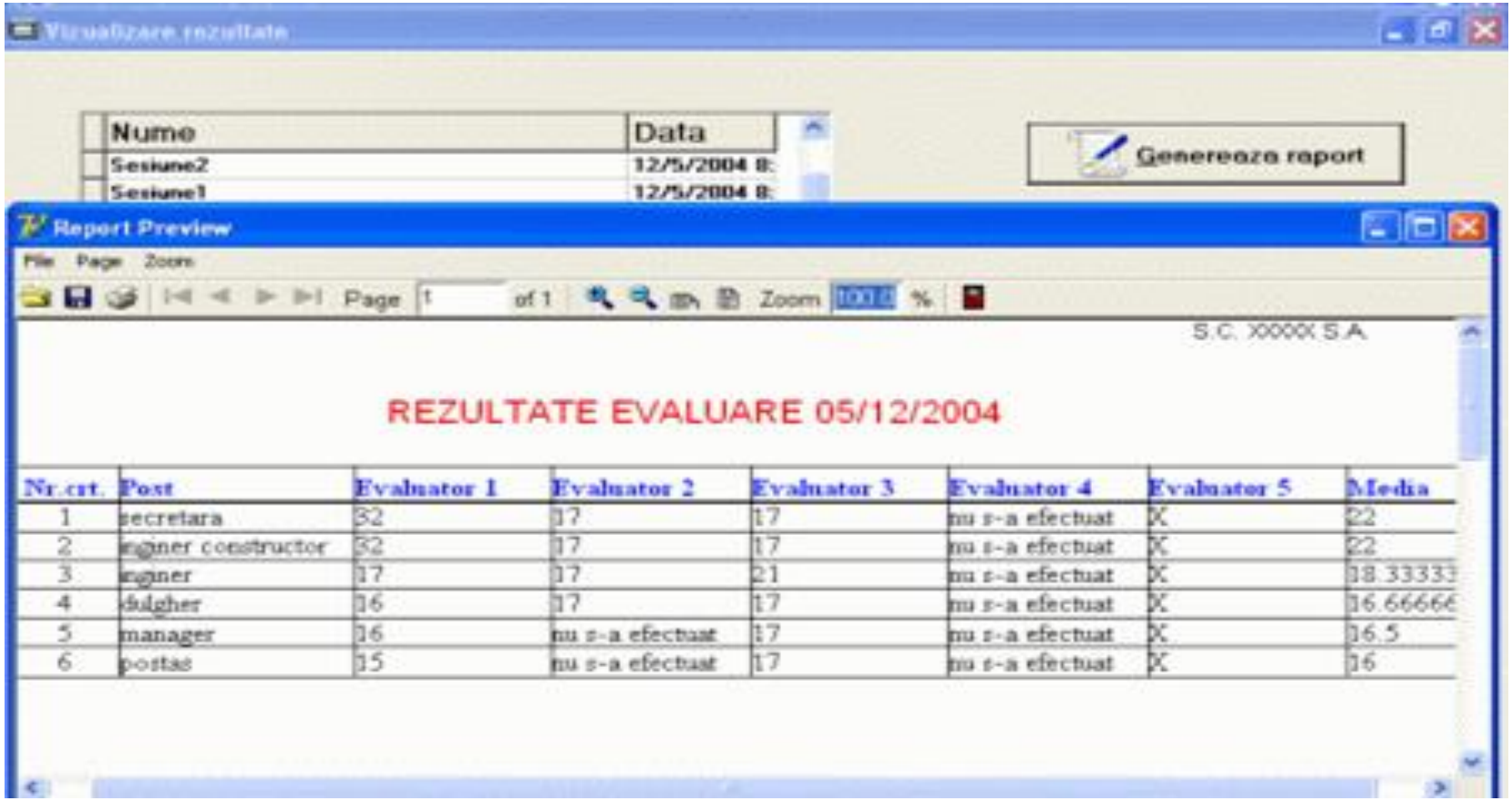
Evaluation results

Close application

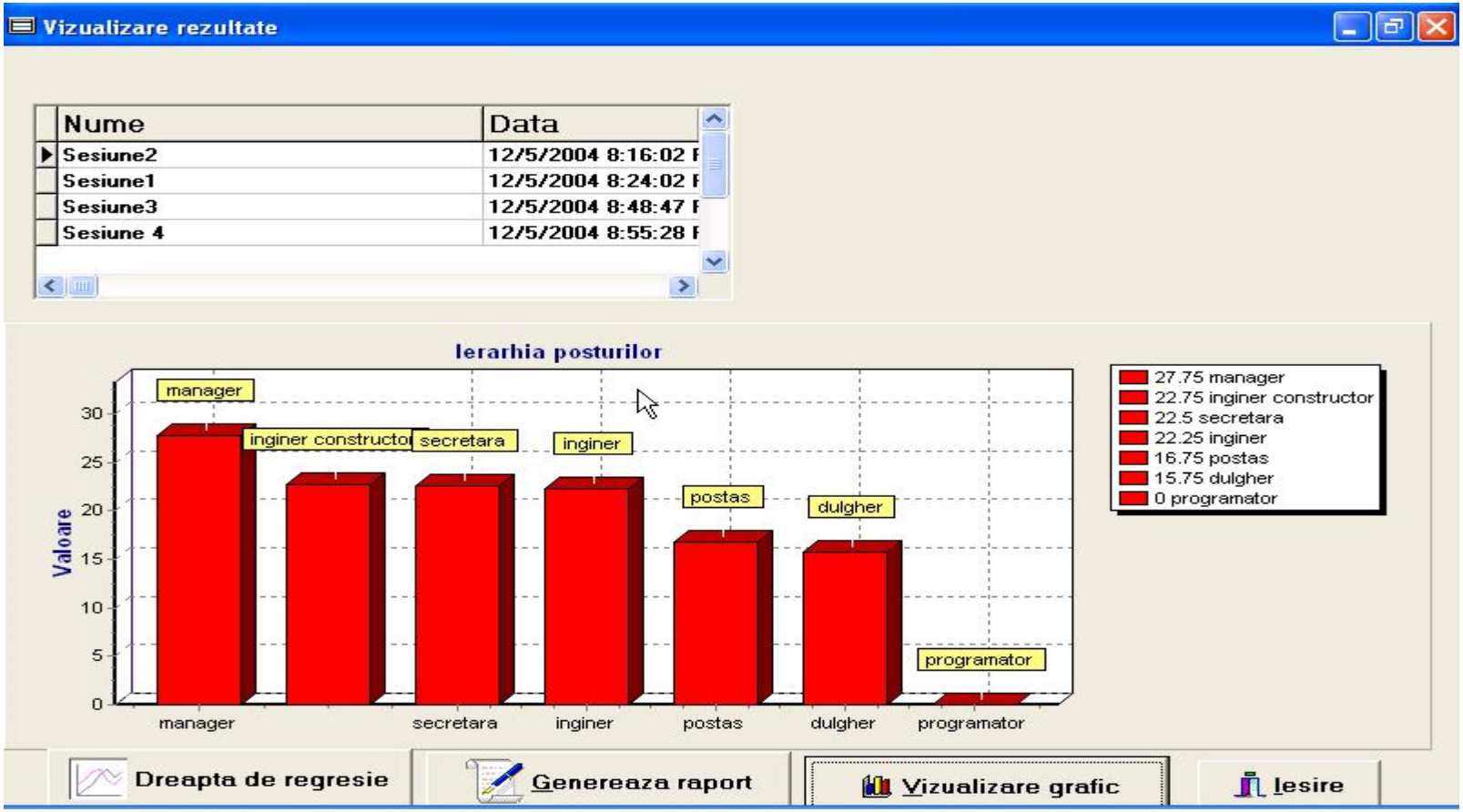
Expert functionality



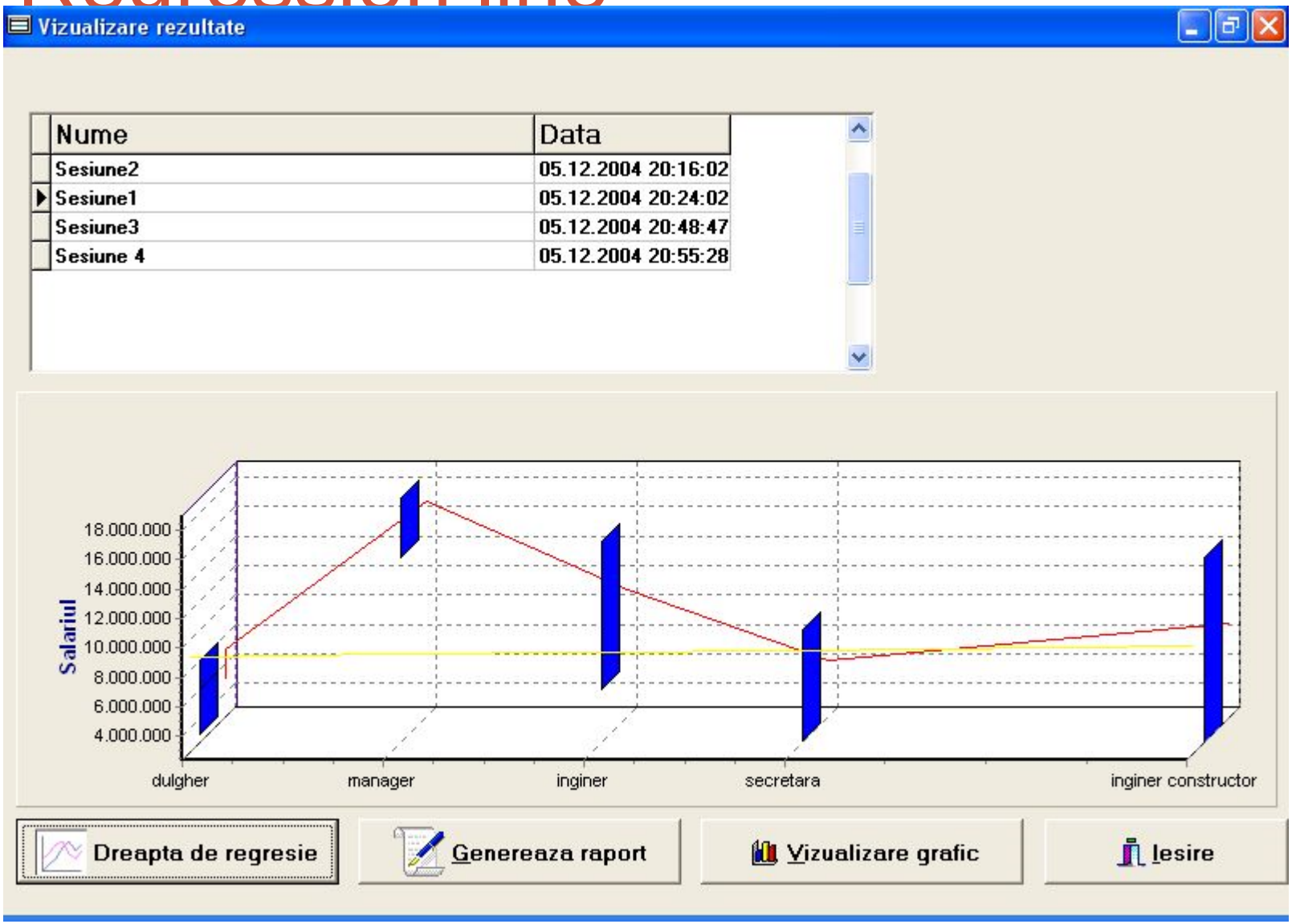
Evaluation results view



Evaluation results view



Regression line



Usability Test

- **System Usability Scale** (SUS)
- Dimensions:
  - Complexity
  - Usage difficulties
  - Need for prior knowledge
  - Consistency
  - Functionality
- 23 participants
- **76% acceptance score after first iteration**

Resources

- CTTE

  http://giove.isti.cnr.it/tools/ctte/

- Euterpe

  http://www.few.vu.nl/~gerrit/gta/euterpe.html