Human Information Processing –Task Modeling and Human Problem Solving model

Models of Information Processing

- Atkinson & Shiffrin’s multistore model
  - Three parts of system
    - Sensory register
    - Short term memory (STM) (book = short term store)
    - Long term memory (LTM) (book = long term store)
  - Inborn and universal
  - Analogy = computer
    - Stores = hardware
    - Control processes/mental strategies = software

Atkinson & Shiffrin’s Model

- Sensory Register
  - Sights/sounds represented directly
  - Limited capacity
- Short-Term Memory (STM)
  - Conscious part
  - Limited capacity
    - 7 +/- 2 units of information
  - Limited time

Atkinson & Shiffrin’s Model

Long-Term Memory (LTM)

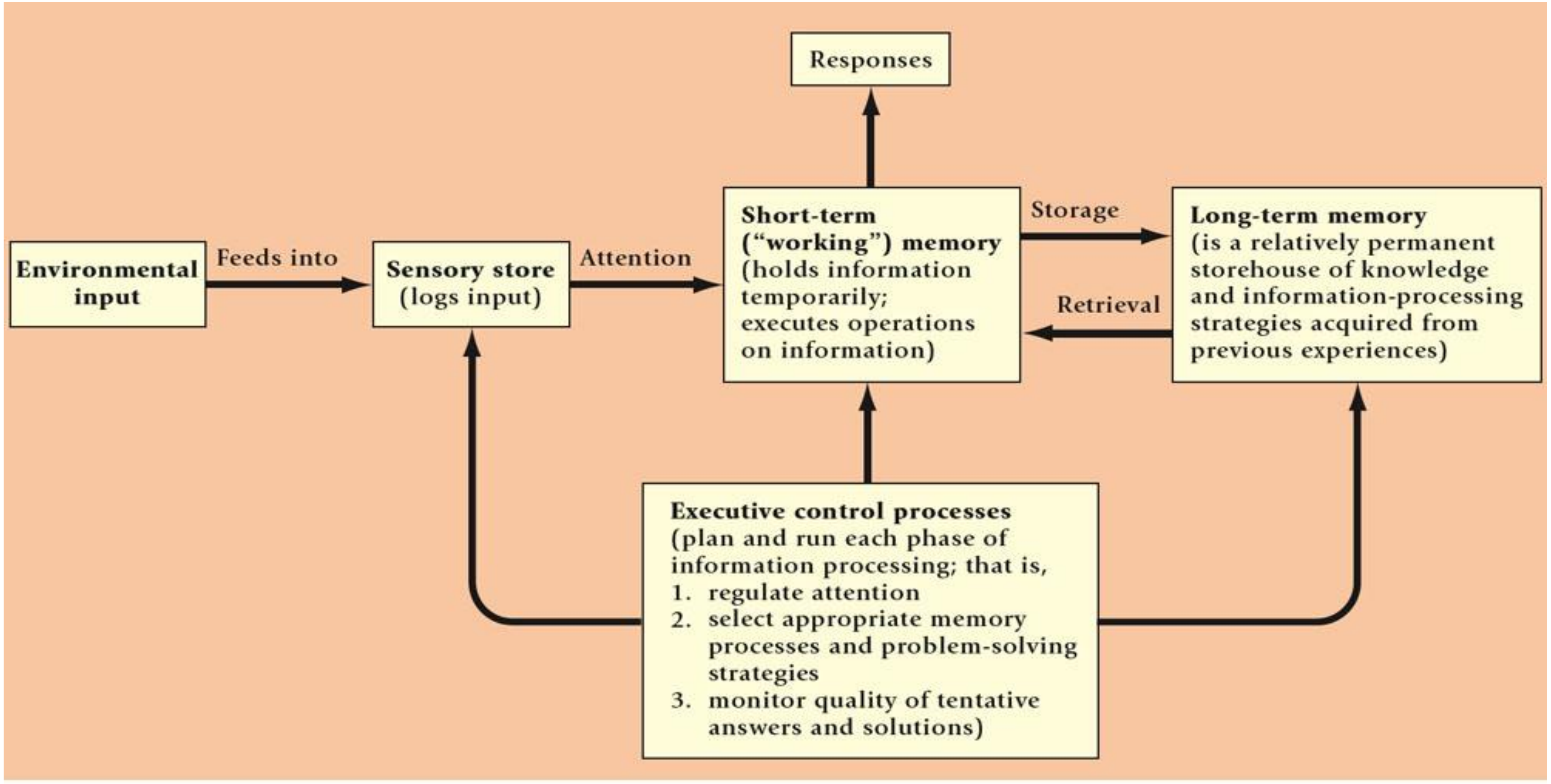
- Unlimited capacity
- Unlimited time frame
- Organization and memory strategies

Information-processing Model of Human Problem Solving

- Types of Knowledge that might be in a chunk of information:
  - General knowledge
    - Information that most people know and apply without regard to a specific domain
      - "red is a color." "4 is bigger than 3."
      - Gained through everyday experiences and basic schooling
      - Domain Specific Knowledge:
      - Information on the form or function of an individual object or a class of objects
      - Bolts are used to carry shear or axial stress
      - The proof stress of a grade 5 bolt is 85 kpsi.
    - Gained from study and experience in the specific domain
      - It may take about 10 years to gain enough specific knowledge to be considered an expert in a domain
  - Procedural Knowledge:
    - The knowledge of what to do next
      - If there is no answer to problem X, then decompose X into two independent subproblems of x1 and x2 that are easier to solve.
    - Gained mostly from experience
    - Required for solving mechanical design problems

Information-processing Model of Human Problem Solving

- Short-Term Memory (STM):
  - – Corresponding to RAM in a computer
  - – Main information processor in the human brain
  - – Information chunks can be processed in about 0.1 sec.
    - • Processing implies such actions as comparing one chunk of information to another, modifying a chunk (decomposing or assembling), etc.
  - – The more memory is used to solve harder problems.
    - – "Magical Number Seven, Plus or Minus Two" Capacity of the short-term memory
    - • Only two or three chunks can be compared at one time due to limits in short-term memory capacity.
- Long-Term Memory (LTM):
  - – Permanent retention of information (Disk storage in a computer)
  - – Unlimited capacity
    - • No documented case of anybody' s brain becoming full regardless of the head size
    - – Fairly slow in recording information (2-5 minutes to memorize a single chunk of info)
  - – Speedy recovery of information, although retrieval time depends on the complexity of the information and the recentness of its use.
  - – Information can be retrieved at different levels of abstraction, in different languages, and with different features.
    - • Human memory is powerful in matching the form of the data retrieved to that which is needed for processing in the short-term memory.
- Control of the Information Processing System:
  - – enables us to encode outside information obtained through our senses or retrieve information from LTM for processing in STM.
  - – When completed manipulating the info, the controller can store the results in the LTM or in the external environment by describing it in text, verbally, or in graphic im ages.



Responses
Environmental input
Feeds into
Sensory store (logs input)
Attention
Short-term ("working") memory (holds information temporarily; executes operations on information)
Storage
Retrieval
Long-term memory (is a relatively permanent storehouse of knowledge and information-processing strategies acquired from previous experiences)
Executive control processes (plan and run each phase of information processing; that is,
1. regulate attention
2. select appropriate memory processes and problem-solving strategies
3. monitor quality of tentative answers and solutions)

- A schematic model of the human information processing system. ADAPTED FROM ATKINSON & SHIFFRIN, 1968.

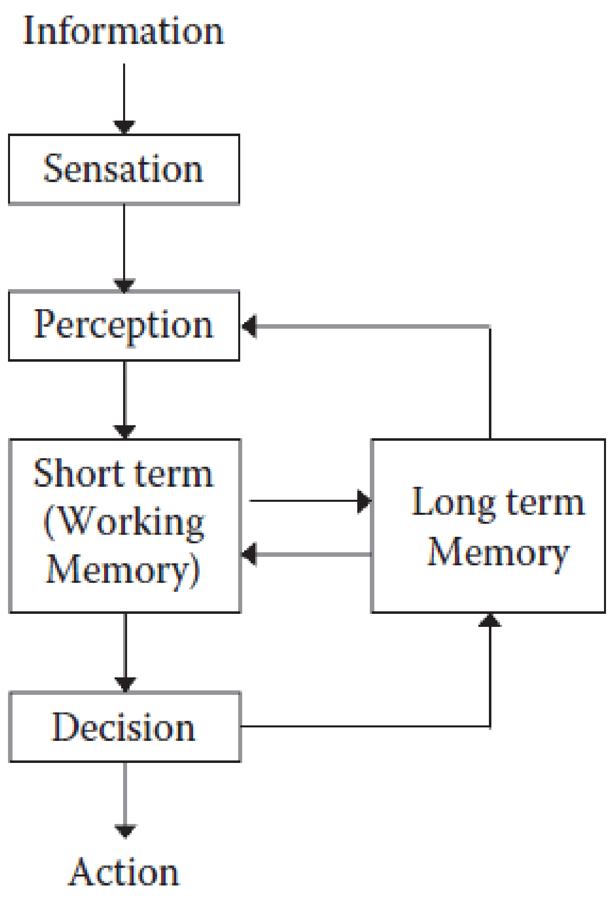
human problem-solving or information-processing efforts consist of these important parts:

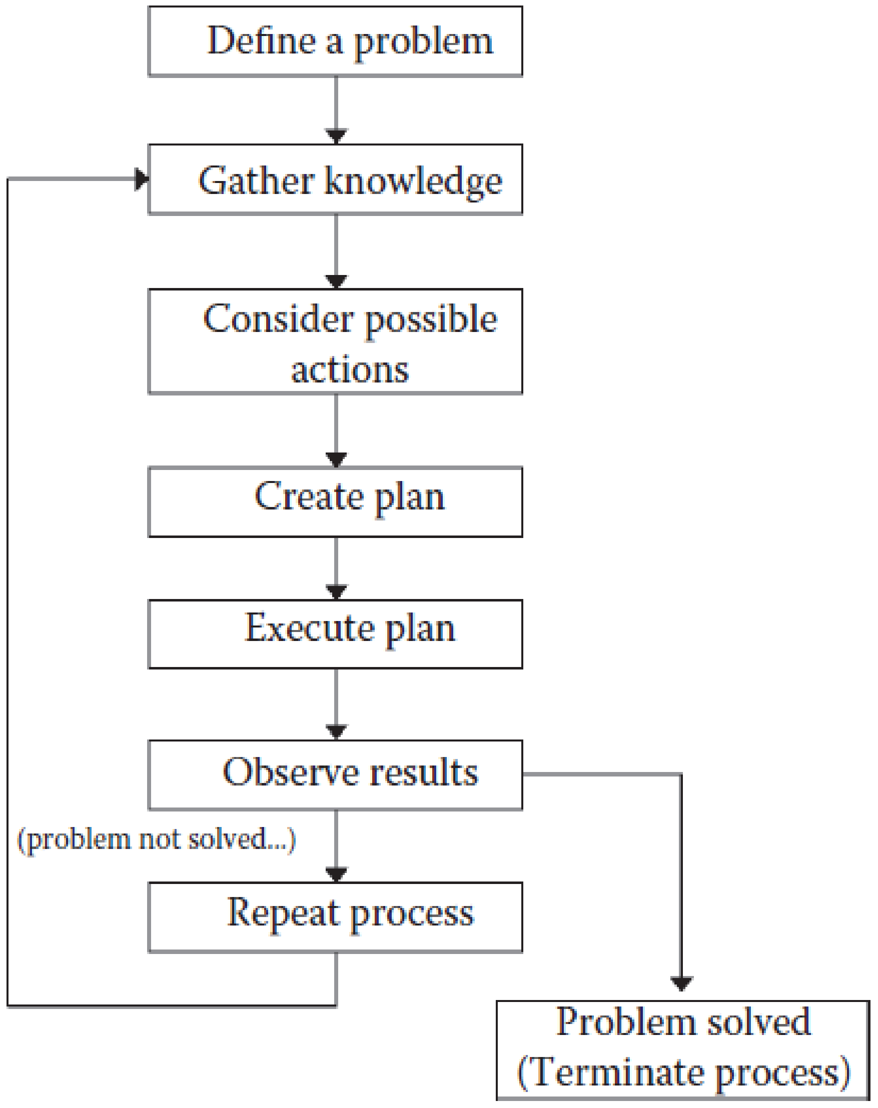
•*Sensation*, which senses external information (e.g., visual, aural, haptic), and *Perception*, which interprets and extracts basic meanings of the external information.

•*Memory*, which stores momentary and short-term information or long-term knowledge. This knowledge includes information about the external world, procedures, rules, relations, schemas, candidates of actions to apply, the current objective (e.g., accomplishing the interactive task success- fully), the plan of action, etc.

•*Decision maker/executor*, which formulates and revises a “plan,” then decides what to do based on the various knowledge in the memory, and finally acts it out by commanding the motor system (e.g., to click the mouse left button).

(a) The overall human problem-solving model and process and (b) a more detailed view of the "decision maker/executor."





Why Model-based approaches?

- Highlight important information
- Help to manage complexity
- Useful to support methods

Models

- What are the properties of a model?
- A model should
    - Focus on one particular aspect of the real world (here, the UI, the Interactive Application) to be represented and emphasized
    - Raise the abstraction level by promoting appropriate abstractions of the real world (multiple and ample possibilities)
  - Be declarative, rather than procedural

Model-Based Interface Design and Development

- Goals:
  - To provide comprehensive development environments (i.e., design and implementation phases)
  - To improve usability and portability of interfaces
  - To integrate usability analysis with interface development
  - To promote declarative UI knowledge (rather than imperative, procedural)

How can we reach them?

- By using a new paradigm: model-based interface development involving 3 facets:
  - Models: explicitly capture knowledge about UI and Interactive Applications with appropriate abstractions
  - Methods: structure the definition and use of underlying models and related transformations
    - Supporting tools: support the use of the method by providing tools for models and their related transformations.

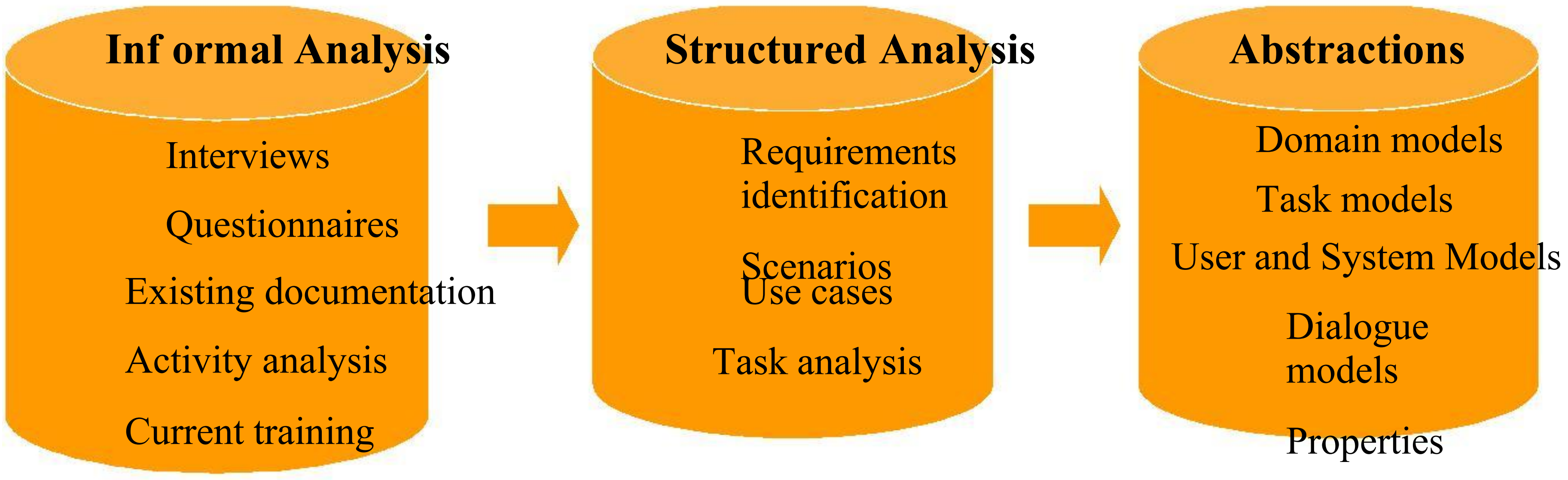
Significant Models in HCI

- Task models
- Cognitive architectures
- User models
- Domain Models
- Context Models
- Presentation Models
- Dialogue models

Definitions

- Task – activity that has to be performed to reach a goal
- Goal
- desired modification of state
- Attempt to receive state information
- Each task is associated with one goal
- Each goal is associated with one or multiple tasks
- Multiple abstraction levels - Basic task
- Task Analysis
- Task Models

Moving from informal to structured representations



Scenarios

- Informal,compact description of:
- one (or multiple) specific user
- Who interacts with a specific interface
- To reach a specific goal
- In a specific environment

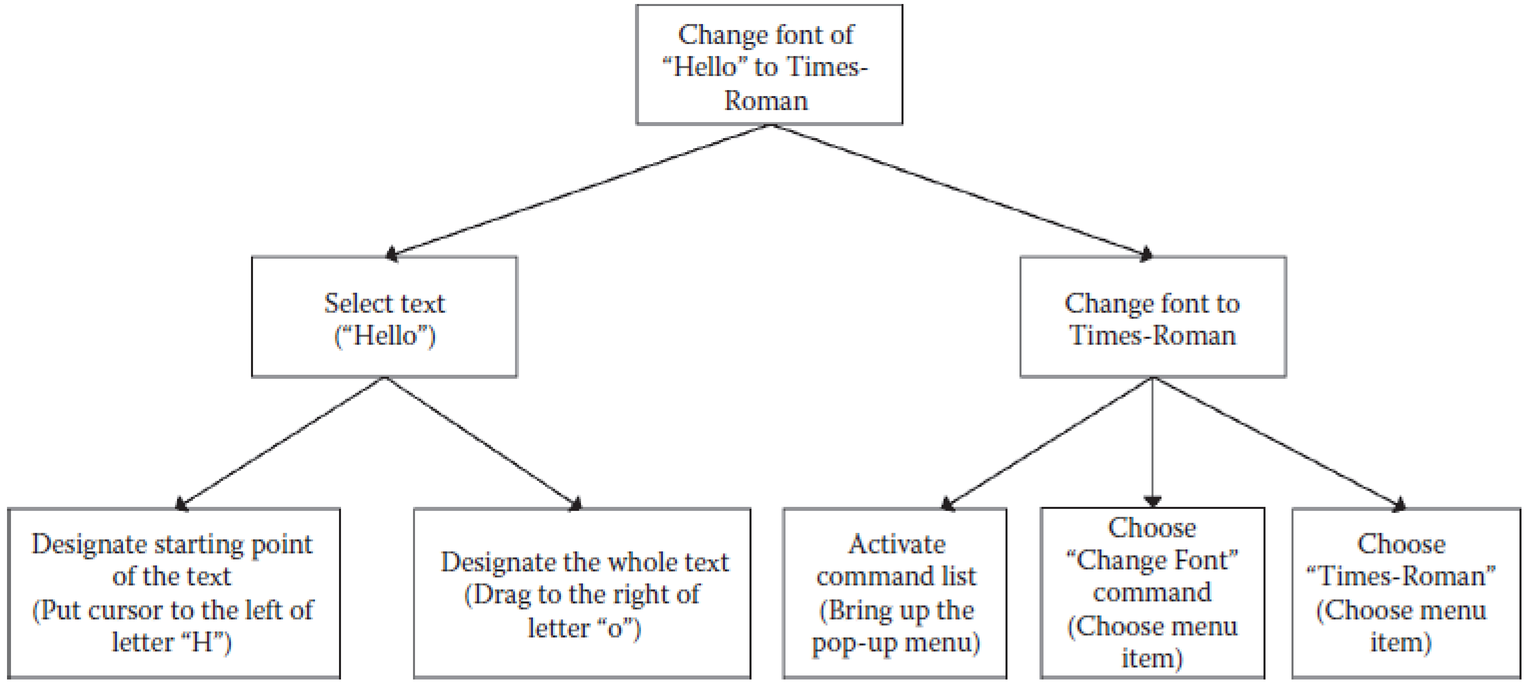
Example of scenario

Silvia is looking for interesting papers on patterns. She makes a request to the on-line library by giving the name of the topic as one of the parameters of her request, and indicating that she is interested in papers written in English. The order of providing these two parameters is not important. She receives a long list of references. As she is interested in recent contributions she adds a further constraint in the request so that she receives information only on papers published in the last five years. The new list of publications is more manageable. She understands that the works by Gamma are very relevant. She would like to have them grouped so that they are presented together. Thus she makes a new request adding the constraint that the author has to be Gamma. The result is the information that she was looking for. Now she can move to another request for another topic.

Use of Scenarios

- Capture the context where the application is used
- Elicit requirements
- Identify important episodes from the user behaviour
- To provide a context for performing evaluation
  - Ability to highlight issues and stimulate discussion while requiring limited effort to develop

example of a hierarchical task plan


Change font of "Hello" to Times-Roman
Select text ("Hello")
Change font to Times-Roman
Designate starting point of the text (Put cursor to the left of letter "H")
Designate the whole text (Drag to the right of letter "o")
Activate command list (Bring up the pop-up menu)
Choose "Change Font" command (Choose menu item)
Choose "Times-Roman" (Choose menu item)

Task analysis

Example: Task analysis of tourists visiting a virtual museum application

- Tourists are characterised by a low average knowledge of the topics considered. Usually they prefer to have guided tours through the rooms of the museum and the town with pictures and information about the works of art. However linear pre-defined tours alone would be too restrictive so some degree of navigational freedom is important. Access to the information is provided with the support of spatial representations: the museum and town maps. This allows users to have immediate information about the locations of the works.
- Tourists want general information on the artistic works, and this information has to be presented clearly and in a limited amount because it has to be interpreted easily. Thus a work will be presented by an image, the title, a short description, the name of the author, the material and technique used for its creation, and when it was made. Additional information about the museum and the town can be provided on request, such as the path to get to the museum from the closest railway station or airport, information (title, data, location) on further exhibitions, and historical information on the town and the museum.

Task analysis

general hierarchical task model, certain subtasks need to be applied in series

- task model can be hierar- chically refined and can serve as a basis for the interface structure.
- Note that, based on this model, we could “select” interfaces to realize each subtask in the bottom of the hierarchy, which illustrates the crux of the HCI design process.
- The interaction model must represent as much as possible what the user has in mind, especially what the user expects must be done (the mental model) in order to accomplish the overall task. This way, the user will be “in tune” with the resulting interactive application.
- The interface selection should be done based on ergonomics, user preference, and other requirements or constraints.
- Finally, the subtask structure can lend itself to the menu structure, and the actions and objects to which the actions apply can serve as the basis for an object-class diagram (for an object-oriented interactive software implementation).

Task analysis

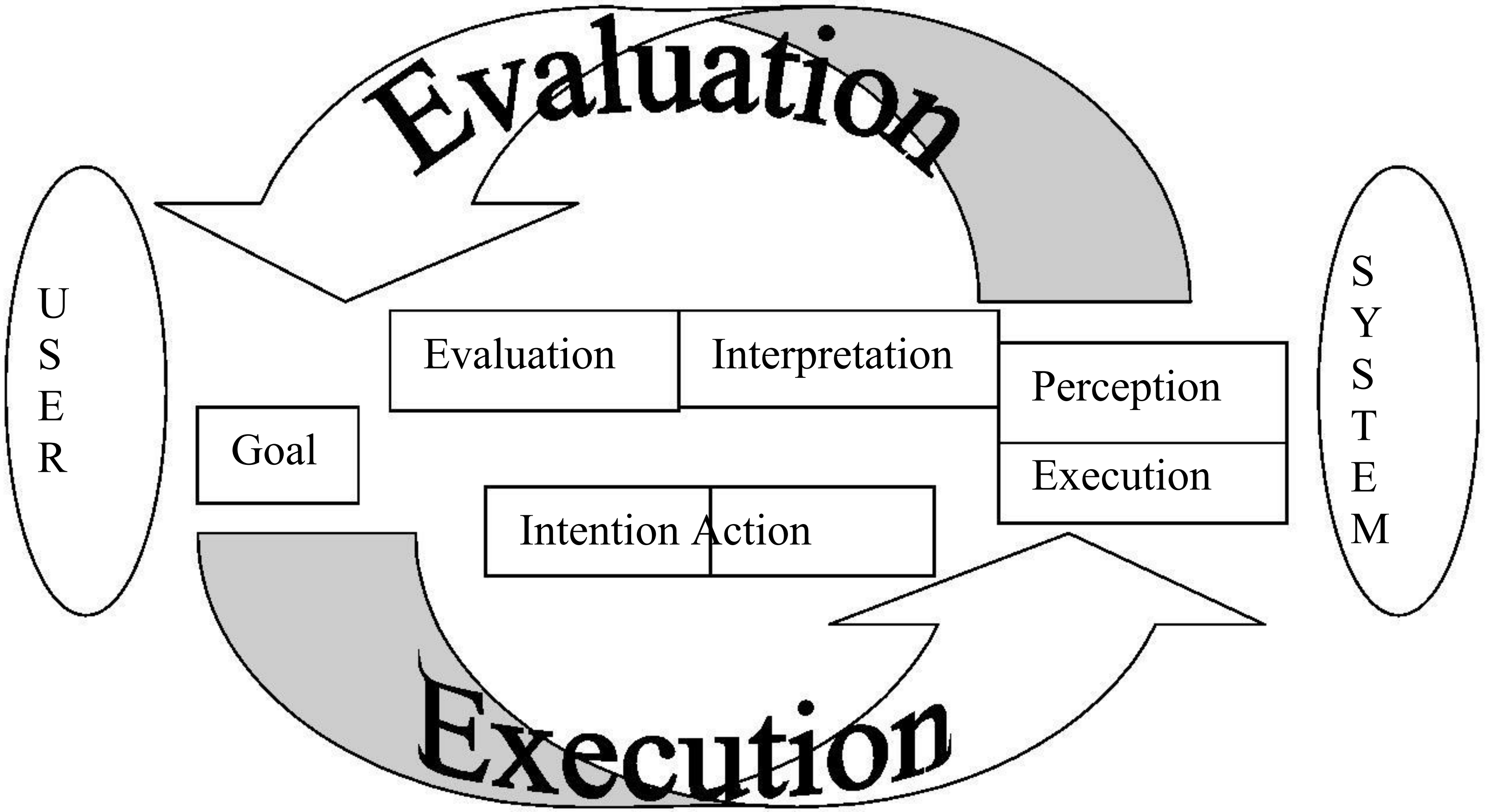
Example: Task analysis of tourists visiting a virtual museum application

- Tourists are characterised by a low average knowledge of the topics considered. Usually they prefer to have guided tours through the rooms of the museum and the town with pictures and information about the works of art. However linear pre-defined tours alone would be too restrictive so some degree of navigational freedom is important. Access to the information is provided with the support of spatial representations: the museum and town maps. This allows users to have immediate information about the locations of the works.
- Tourists want general information on the artistic works, and this information has to be presented clearly and in a limited amount because it has to be interpreted easily. Thus a work will be presented by an image, the title, a short description, the name of the author, the material and technique used for its creation, and when it was made. Additional information about the museum and the town can be provided on request, such as the path to get to the museum from the closest railway station or airport, information (title, data, location) on further exhibitions, and historical information on the town and the museum.

Task Analysis (task list)

- Access to guided tours through the museum and the town
- System enable some degree of navigational freedom
- Access to the information through spatial representations.
- Access to general information on the artistic works
- System presents information clearly and in a limited amount
- System presents a work by an image, the title, a short description, the name of the author, the material and technique used for its creation, and when it was made.
- Additional information about the museum and the town can be provided on request.

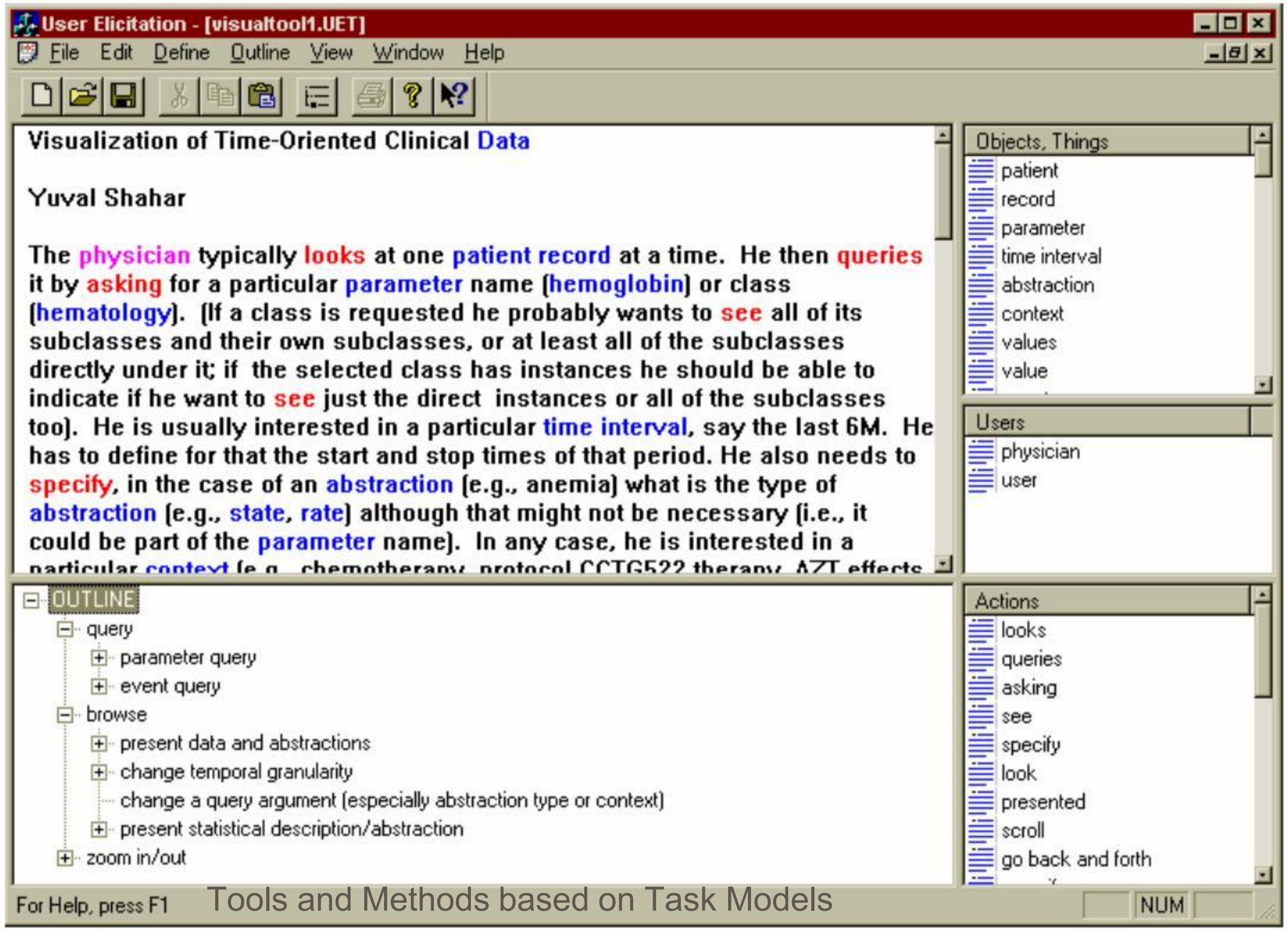
Norman’s cycle of interaction



Detectable problems

- Lack of correspondence between user intentions and actions supported by the interface
- Lack of correspondence between representations provided by the system and those expected by the user
- The best interface is that invisible that does not provide obstacles when users perform their tasks

User-Elicitation Example for a Medical-Domain Interface


User Elicitation - [visualtool1.UET]
File Edit Define Outline View Window Help

Visualization of Time-Oriented Clinical Data

Yuval Shahar

The physician typically looks at one patient record at a time. He then queries it by asking for a particular parameter name (hemoglobin) or class (hematology). (If a class is requested he probably wants to see all of its subclasses and their own subclasses, or at least all of the subclasses directly under it; if the selected class has instances he should be able to indicate if he want to see just the direct instances or all of the subclasses too). He is usually interested in a particular time interval, say the last 6M. He has to define for that the start and stop times of that period. He also needs to specify, in the case of an abstraction (e.g., anemia) what is the type of abstraction (e.g., state, rate) although that might not be necessary (i.e., it could be part of the parameter name). In any case, he is interested in a particular context (e.g., chemotherapy, protocol CCTG522 therapy, AZT effects

Objects, Things
- patient
- record
- parameter
- time interval
- abstraction
- context
- values
- value

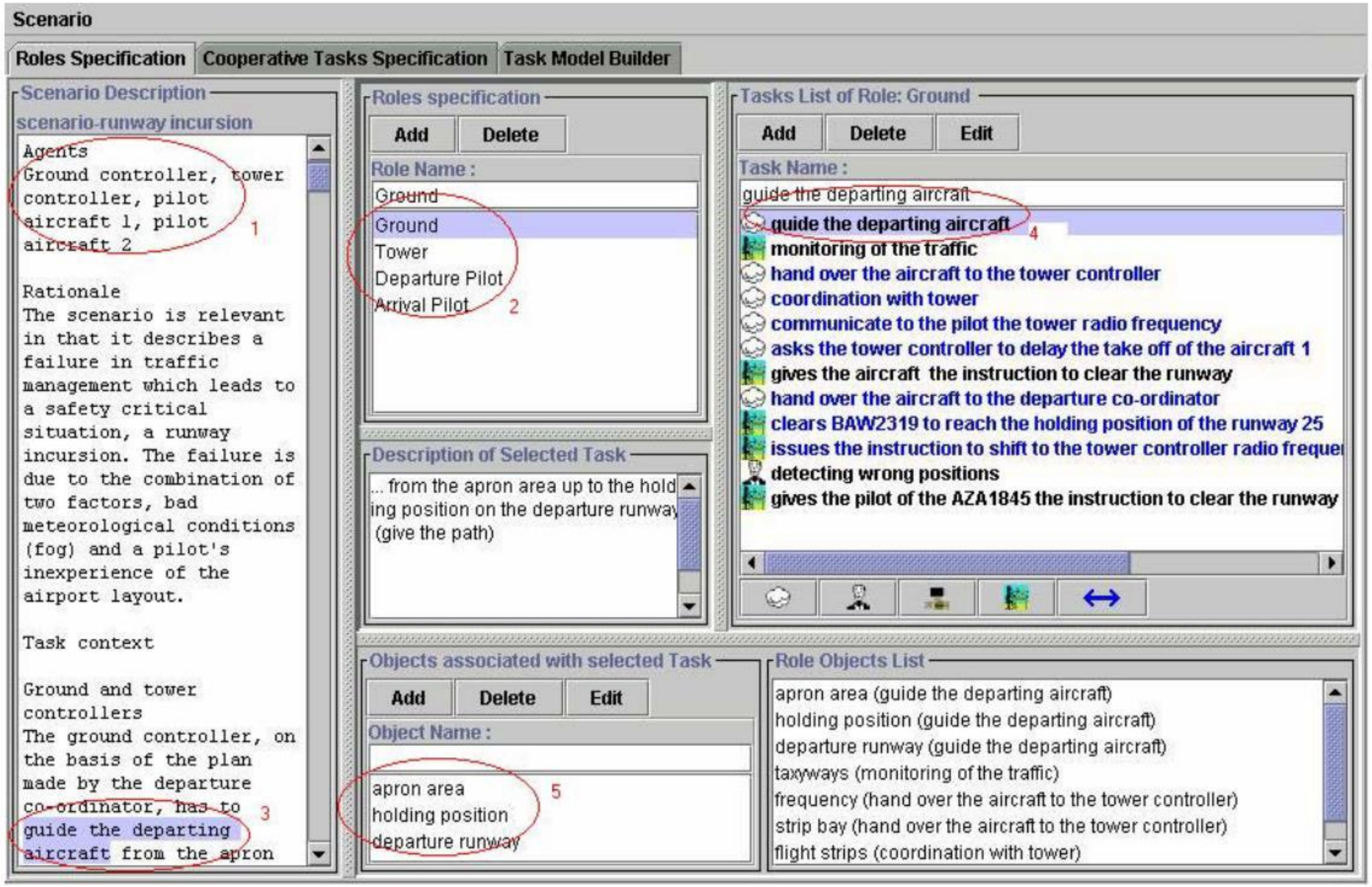
Users
- physician
- user

OUTLINE
- query
  - parameter query
  - event query
- browse
  - present data and abstractions
  - change temporal granularity
  - change a query argument (especially abstraction type or context)
  - present statistical description/abstraction
- zoom in/out

Actions
- looks
- queries
- asking
- see
- specify
- look
- presented
- scroll
- go back and forth

For Help, press F1 NUM

The environment supporting task identification



Scenario

Roles Specification | Cooperative Tasks Specification | Task Model Builder

Scenario Description

scenario-runway incursion

```
Agents
Ground controller, tower
controller, pilot
aircraft 1, pilot
aircraft 2

Rationale
The scenario is relevant
in that it describes a
failure in traffic
management which leads to
a safety critical
situation, a runway
incursion. The failure is
due to the combination of
two factors, bad
meteorological conditions
(fog) and a pilot's
inexperience of the
airport layout.

Task context

Ground and tower
controllers
The ground controller, on
the basis of the plan
made by the departure
co-ordinator, has to
guide the departing
aircraft from the apron
```

1

Roles specification

Add | Delete

Role Name :

Ground

- Ground
- Tower
- Departure Pilot
- Arrival Pilot

2

Description of Selected Task

... from the apron area up to the holding position on the departure runway (give the path)

Tasks List of Role: Ground

Add | Delete | Edit

Task Name :

guide the departing aircraft

- guide the departing aircraft 4
- monitoring of the traffic
- hand over the aircraft to the tower controller
- coordination with tower
- communicate to the pilot the tower radio frequency
- asks the tower controller to delay the take off of the aircraft 1
- gives the aircraft the instruction to clear the runway
- hand over the aircraft to the departure co-ordinator
- clears BAW2319 to reach the holding position of the runway 25
- issues the instruction to shift to the tower controller radio freque
- detecting wrong positions
- gives the pilot of the AZA1845 the instruction to clear the runway

3

Objects associated with selected Task

Add | Delete | Edit

Object Name :

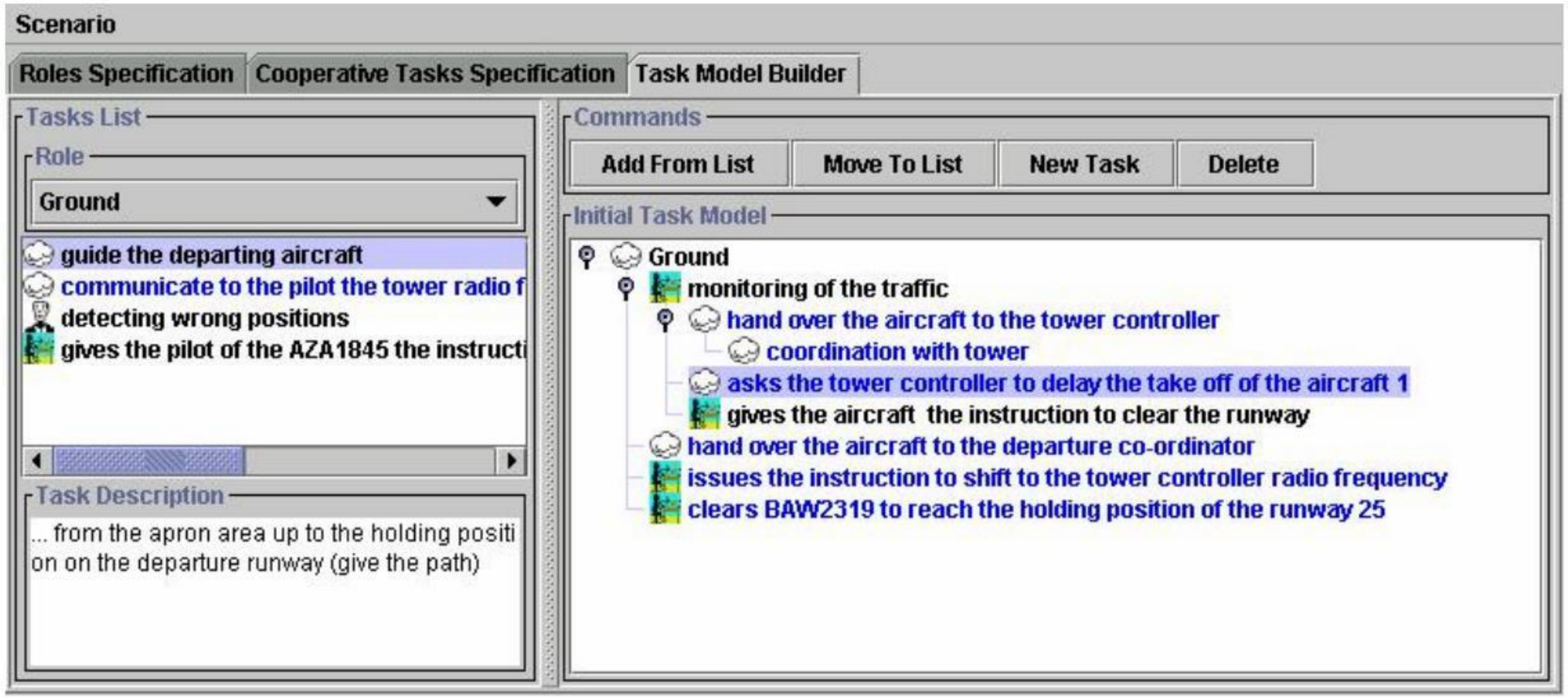
- apron area
- holding position
- departure runway

5

Role Objects List

- apron area (guide the departing aircraft)
- holding position (guide the departing aircraft)
- departure runway (guide the departing aircraft)
- taxyways (monitoring of the traffic)
- frequency (hand over the aircraft to the tower controller)
- strip bay (hand over the aircraft to the tower controller)
- flight strips (coordination with tower)

Tool support to structure the task model


Scenario

Roles Specification | Cooperative Tasks Specification | Task Model Builder

Tasks List

Role: Ground

guide the departing aircraft
communicate to the pilot the tower radio f
detecting wrong positions
gives the pilot of the AZA1845 the instructi

Task Description

... from the apron area up to the holding positi
on on the departure runway (give the path)

Commands: Add From List | Move To List | New Task | Delete

Initial Task Model

```
Ground
  monitoring of the traffic
    hand over the aircraft to the tower controller
      coordination with tower
    asks the tower controller to delay the take off of the aircraft 1
    gives the aircraft  the instruction to clear the runway
  hand over the aircraft to the departure co-ordinator
  issues the instruction to shift to the tower controller radio frequency
  clears BAW2319 to reach the holding position of the runway 25
```

Engineering task models

- Flexible and expressive notations with precise semantics
- Systematic methods able to indicate how to use information in the task models
- Availability of automatic tools to use such information efficiently

Advantages of Task-based approaches

- For the designer: high-level, structured approaches which allow integration of both functional and interactional aspects
- For the end user: support the generation of more understandable systems

The many possible task models

- (Describe) Existing System
- (Define) Envisioned System
- User

Use of Task Models

- Improve understanding of the application domain
- Record the result of interdisciplinary discussion
- Support effective design
- Support usability evaluation
- Support the user during a session
- Documentation

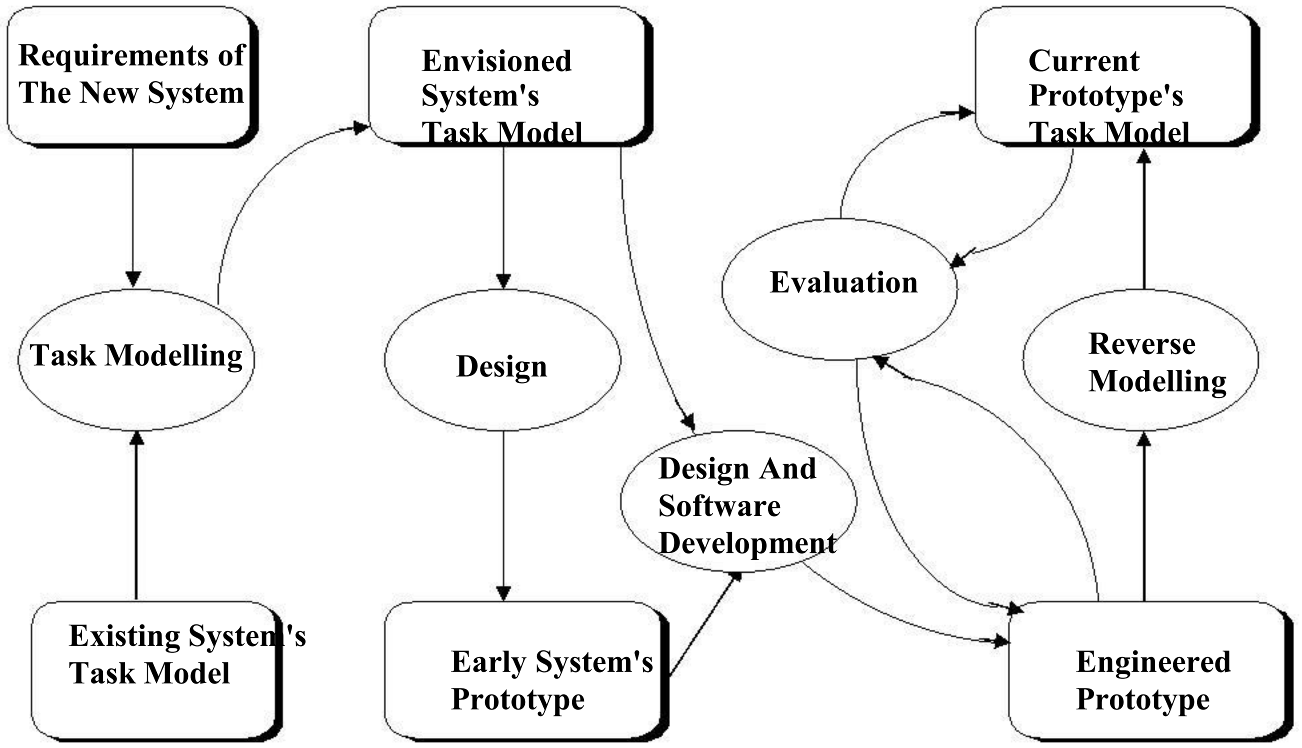
Task Models vs Scenarios

- Scenarios are informal descriptions of a specific use in a specific context
- Task models describe the main possible activities and their relationships
- Scenarios can support task model development
- Task models can support scenarios identification

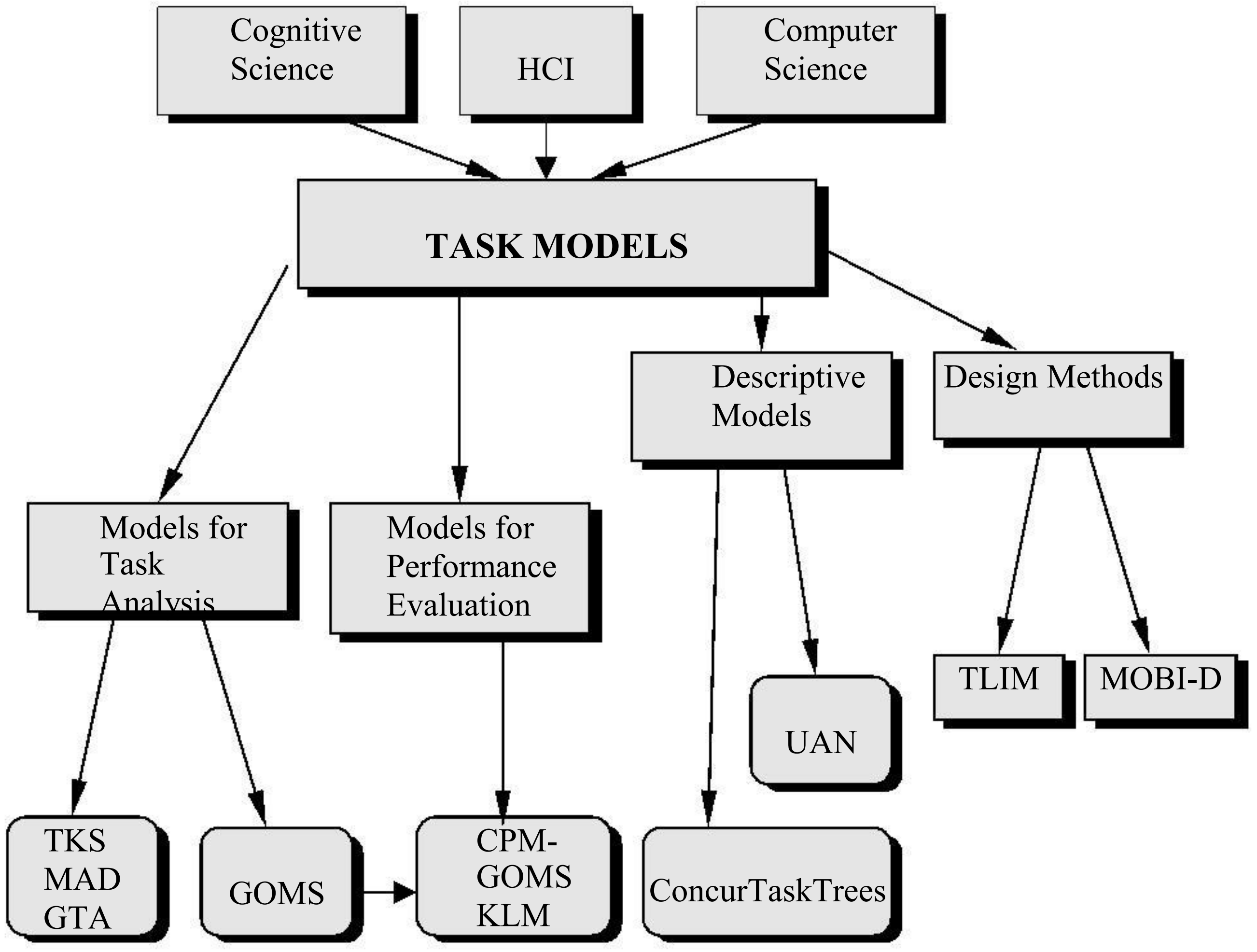
Representations of Task Models

- Hierarchical task analysis
- GOMS family
- UAN
- Different syntax (textual vs graphical)
- Different level of formality
- Different set of operators for task composition

Use of Models in the Life Cycle


Requirements of The New System
Envisioned System's Task Model
Current Prototype's Task Model
Evaluation
Task Modelling
Design
Reverse Modelling
Design And Software Development
Existing System's Task Model
Early System's Prototype
Engineered Prototype

Approaches to task models



Tools and Methods based on Task Models

Representations of Task Models

- Hierarchical task analysis
- GOMS family
- UAN
- Different syntax (textual vs graphical)
- Different level of formality
- Different set of operators for task composition

GOMS Example

```
GOAL: EDIT-MANUSCRIPT
          GOAL: EDIT-UNIT-Task repeat until no more unit tasks
          GOAL: ACQUIRE-UNIT-TASK
          GET-NEXT-PAGE if at end of manuscript
          GET-NEXT-TASK
               GOAL: EXECUTE-UNIT-
               TASK
               GOAL: LOCATE-LINE
                    [select: USE-QS-METHOD
                    USE-LF-METHOD]
                    GOAL: MODIFY-TEXT
                         [select: USE-S-
                         METHOD
                         USE-M-METHOD]
                         VERIFY-EDIT
```

Limitations of GOMS

- It does not consider user errors
- It does not consider the possibility of interruptions
- It considers only sequential tasks
- It can be inadequate for distributed applications (such as web-based applications)

UAN - User Action Notation

- The user interface is represented by a hierarchy of asynchronous tasks
- user action and system feedback are specified at a low level
- textual notation

Example of UAN specification

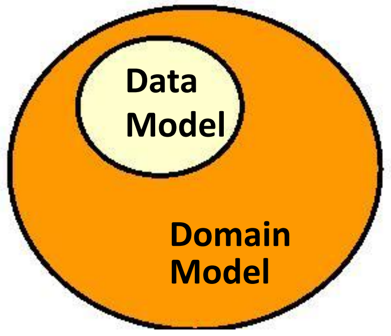
Task: BuildRequest:
*((SelR | ClearR | IconifyR)**
--> SpecField+)

Task: SelApplication

User Action	Interface Feedback	Interface State
~[x,y in AppICON] □□ (t<tdoubleClick) □□□	w'!: w'-! UnMap(PrevAppliMenu) Map(AppMenu) UnMap(AppICON)	CurAppli=App CurMenu=AppMenu

Domain model

- Definition
  - A domain model defines the objects that a user can view, access, and manipulate through a user interface
- A domain model represents objects of the domain with their relationships
- Historically, data models have been considered for a while, but they are only a subset of domain models



Domain Model

- Domain models extend data models
- Relationships among objects are made explicit and declarative
- Data models are useful only for automatic layout generation
- Domain models are can help to identify effective layout and user interface behavior

ConcurTaskTrees

- Focus on Activities
- Hierarchical Structure
- Graphical Syntax
- Rich set of temporal operators
- Task allocation
- Objects and task attributes

Categories of tasks

interaction

application

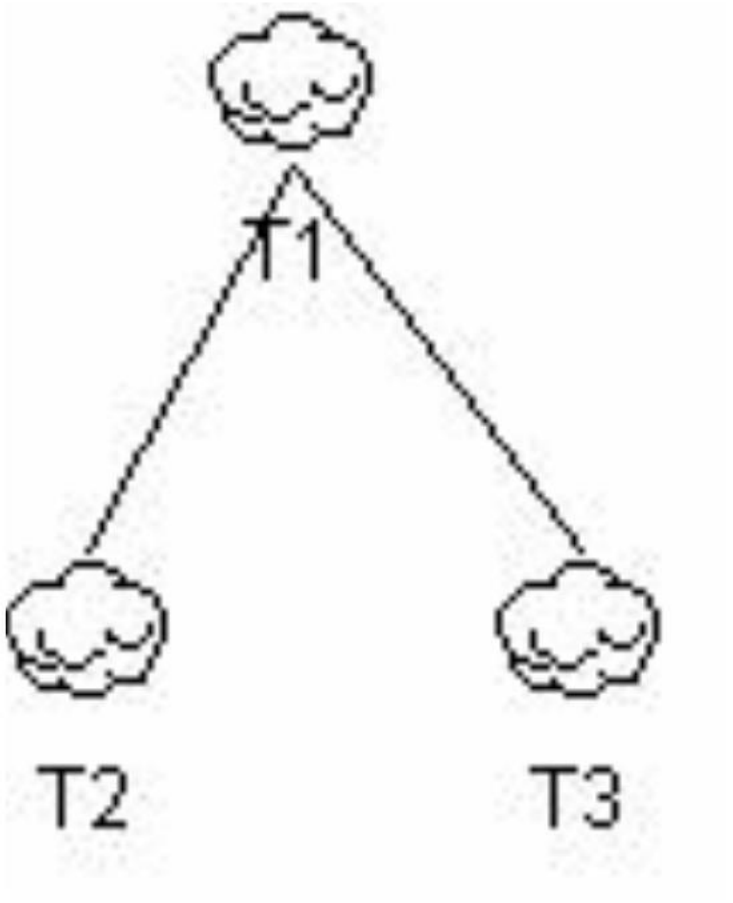
user

abstract

Temporal operators

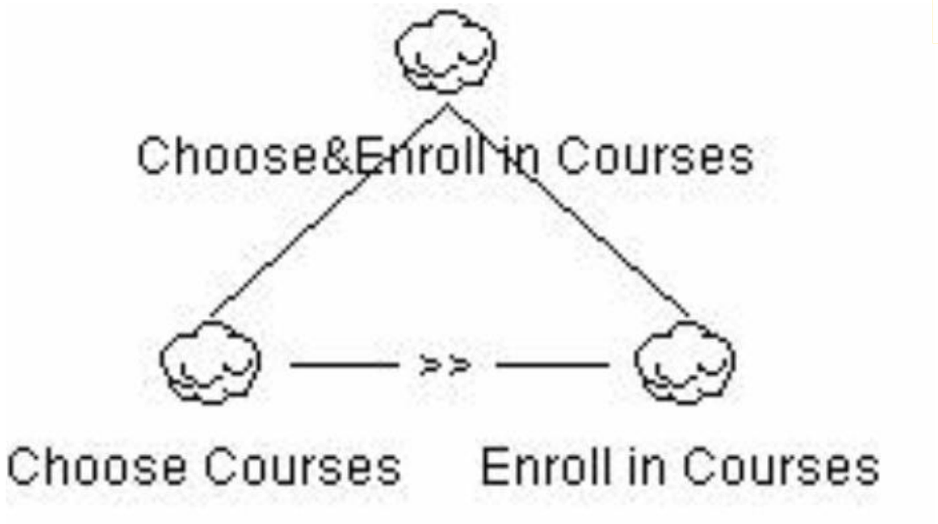
Enabling	T1 >> T2 or T1 [ ]>> T2
Disabling	T1 [> T2
Interruption	*T1* |> *T2*
Choice	T1 [ ] T2
Iteration	T1* or T1$_{\{n\}}$
Concurrency	T1 ||| T2 T1 |[]| T2
Optionality	[T]
Order Independency	T1 |=| T2

Hierarchy



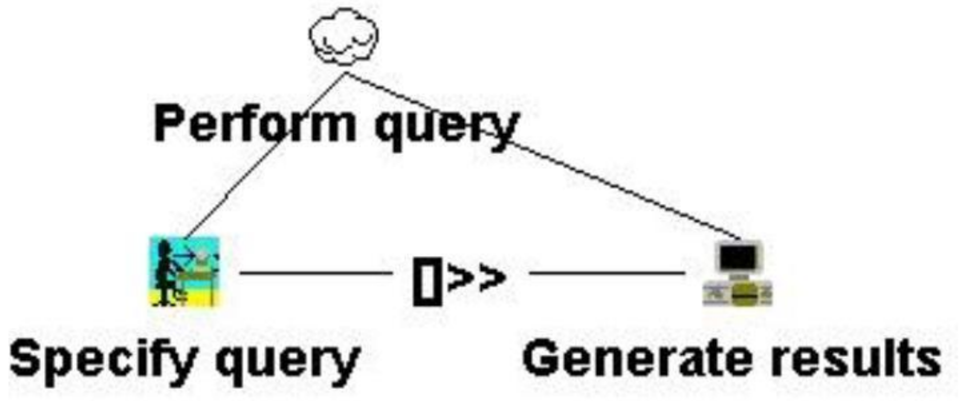
- Tasks at same level represent different options or different tasks that have to be performed
  - Read levels as “In order to do T1, I need to do T2 and T3”, or “In order to do T1, I need to do T2 or T3”

Enabling



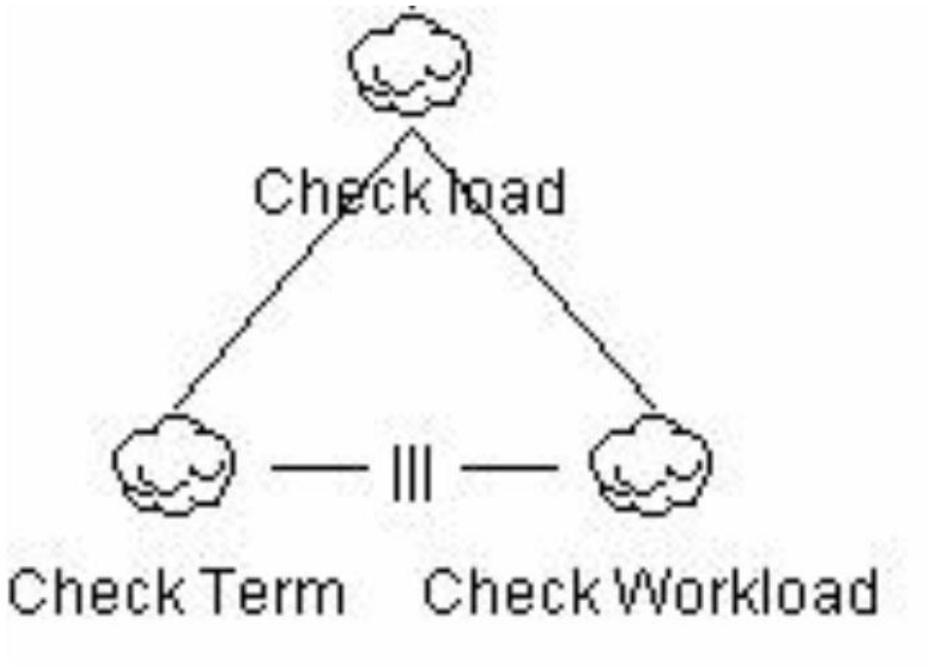
- Specifies second task cannot begin until first task performed
  - I.e., I cannot enroll at university before I've chosen which courses to take

Enabling with Information Flow



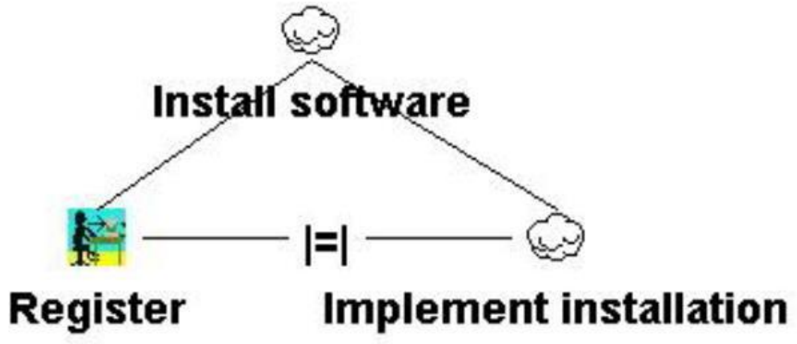
Specifies second task cannot be performed until first task is performed, and that information produced in first task is used in second

Interleaving



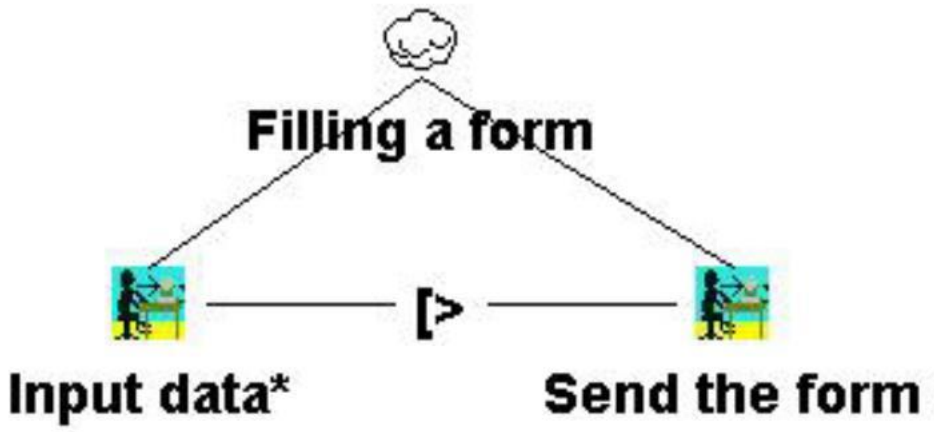
- Tasks can be performed in any order, or at same time
  - In order to check the load of a set of courses, I need to consider what terms they fall in and to consider how much work each course represents
- I can do this in any order

Order Independence



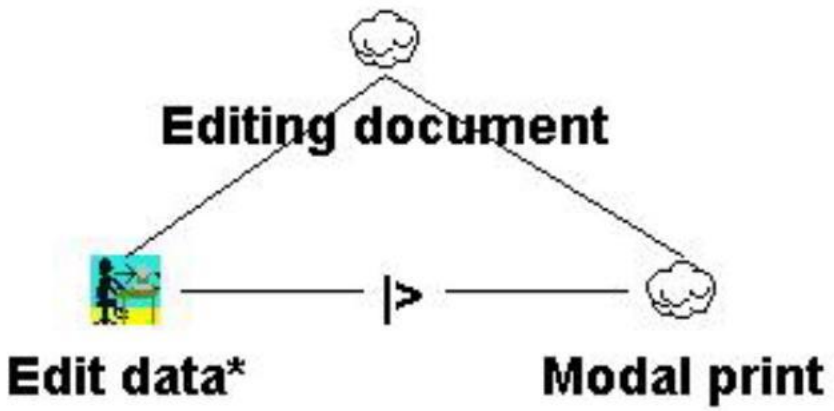
- Tasks can be performed in any order, but when one starts then it has to finish before the other one can start
  - When I install new software I can start by either registering or implementing the installation but if I start one task I have to finish it before moving to the other one

Disabling



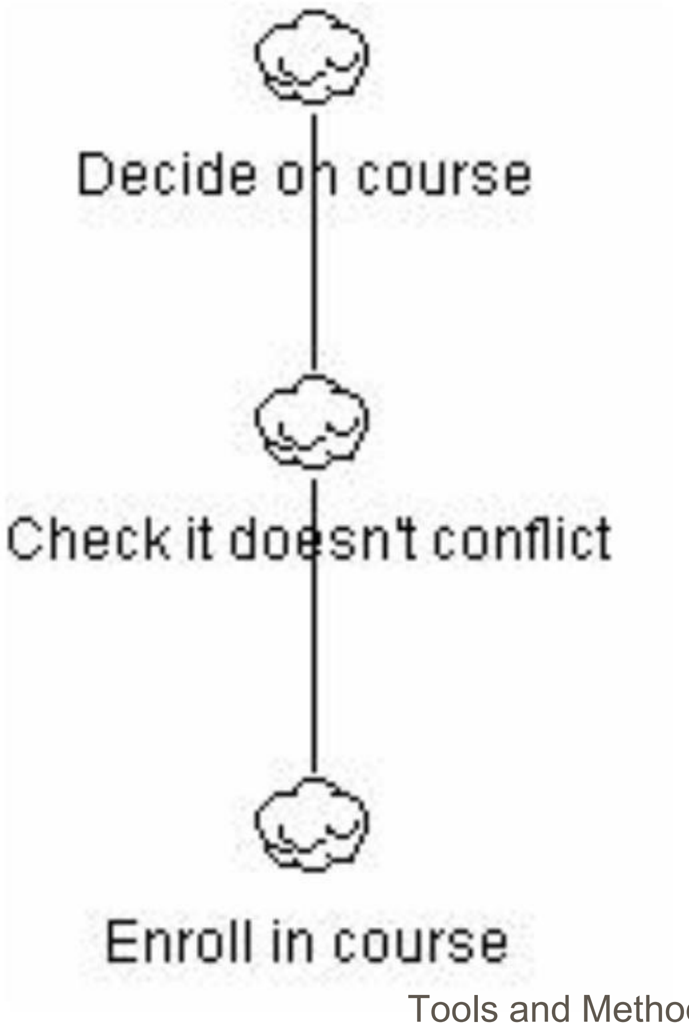
- The first task (usually an iterative task) is completely interrupted by the second task

Suspend-resume



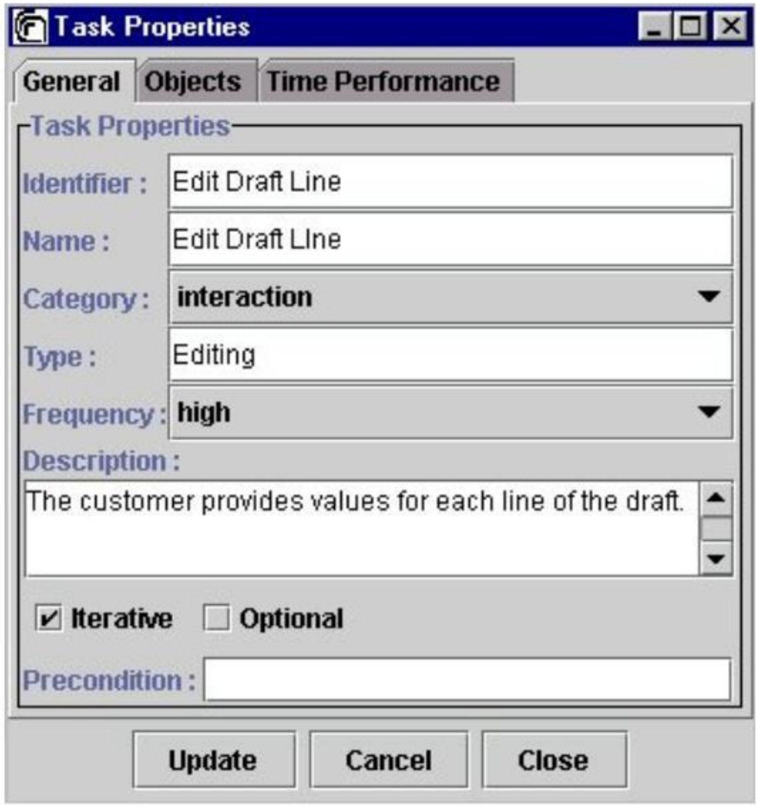
- First task can be interrupted by the second one
  - When the second terminates then the first one can be reactivated from the state reached before

Common Errors



- Hierarchy does NOT represent sequence
  - “In order to check a course doesn’t conflict, I have to enroll in the course”

Task and attributes



Task Properties

Identifier :	Edit Draft Line
Name :	Edit Draft LIne
Category :	interaction
Type :	Editing
Frequency :	high

Description : The customer provides values for each line of the draft.

[✓] Iterative [ ] Optional

Precondition :

Interaction tasks

Selection

Edit

Control

…

Application task

Overview

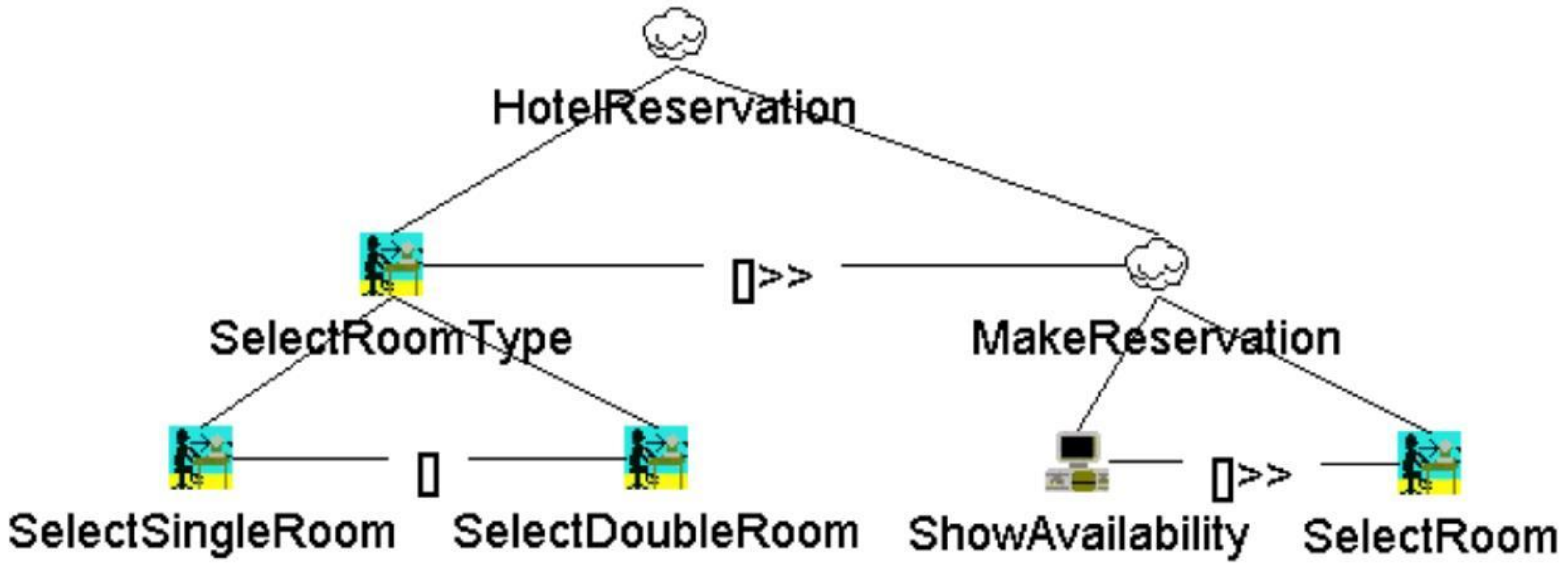
Feedback

Generating alerts

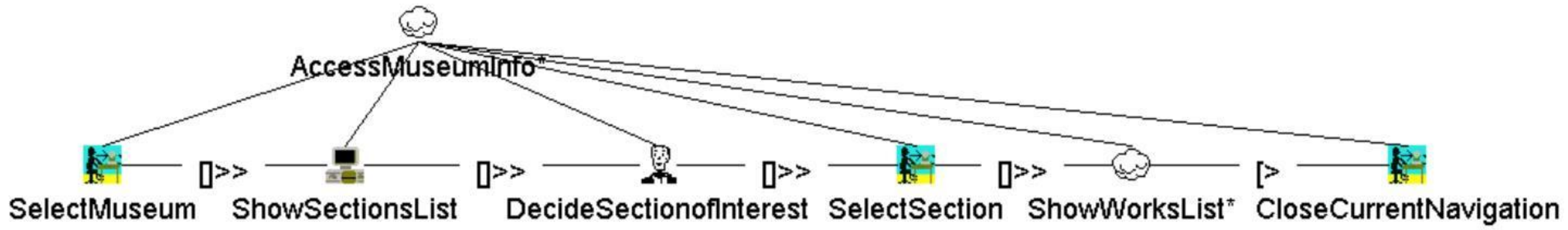
Grouping

…

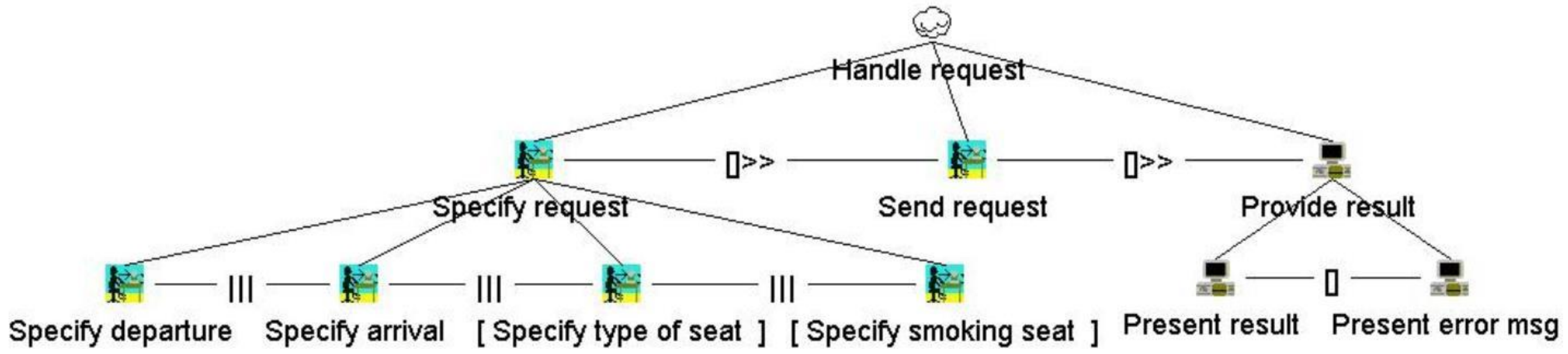
Inheritance of relationships



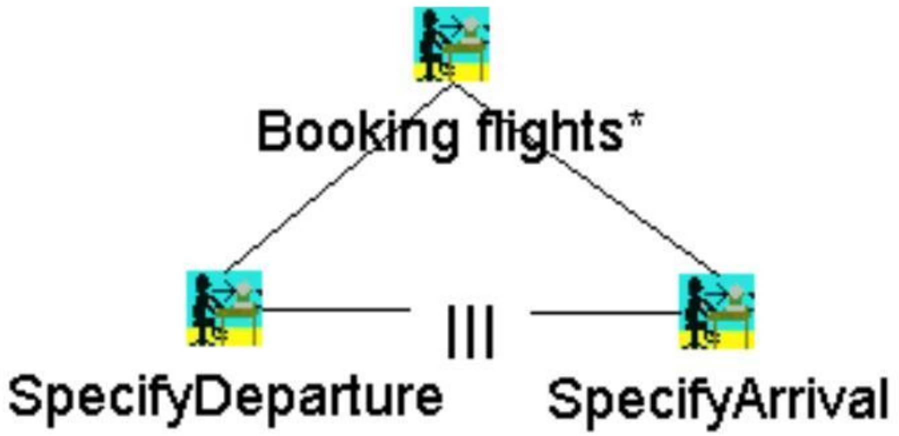
Relationships task/subtasks

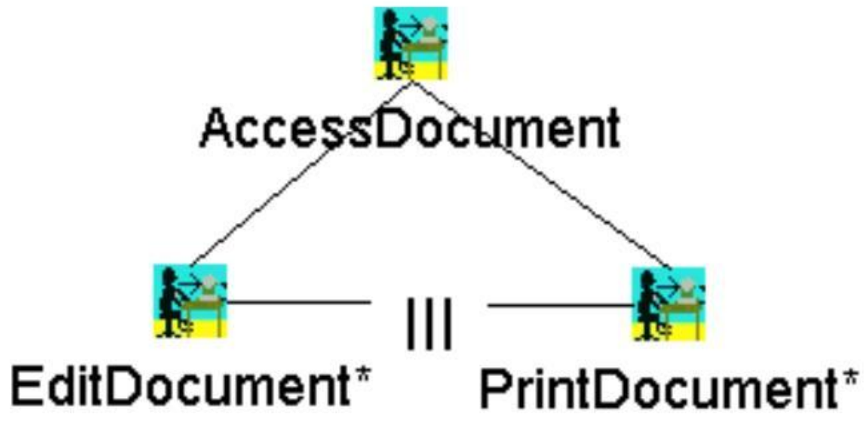


Optional tasks

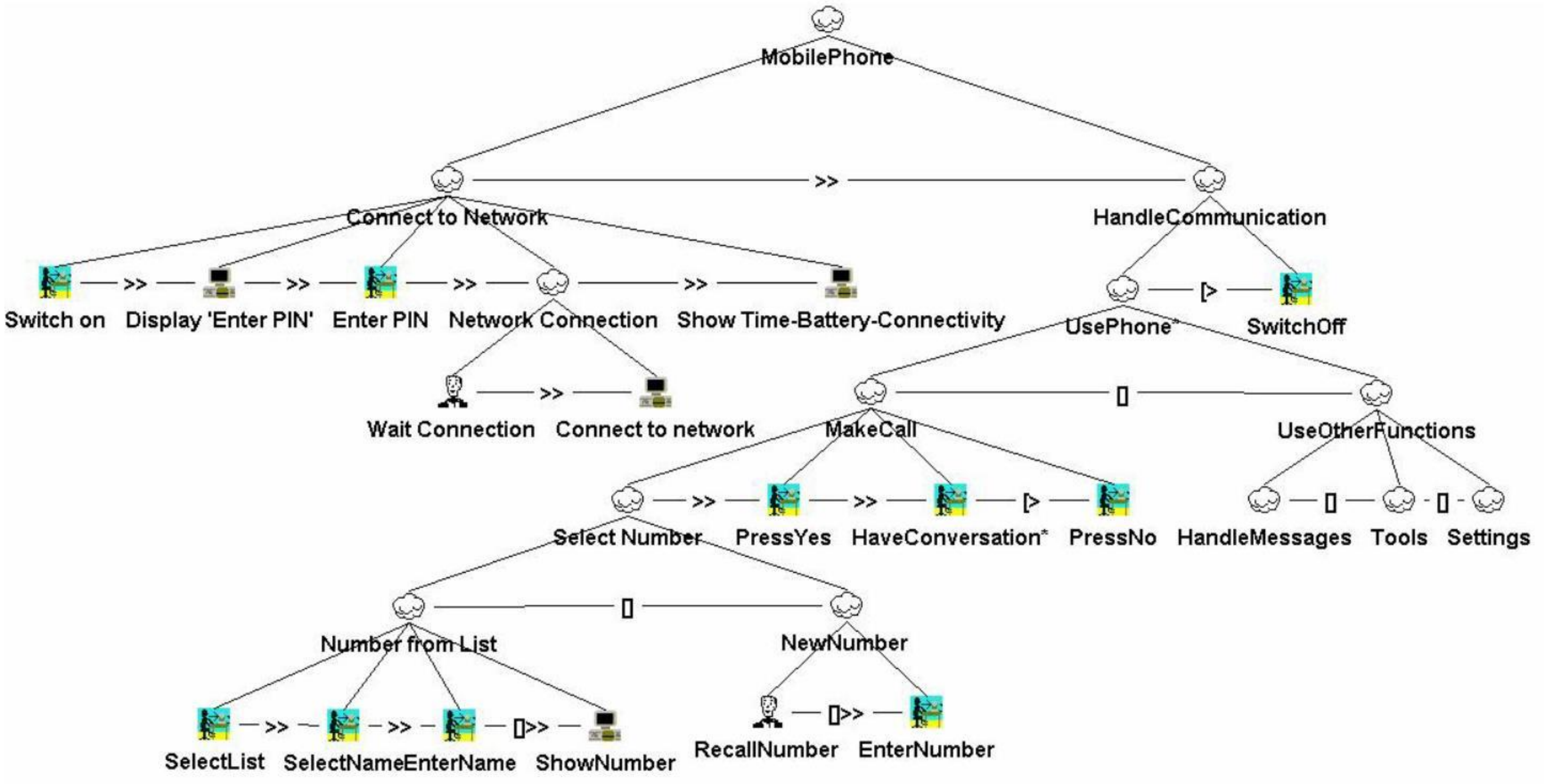


Multiple performance / continuous interleaving





An Example



MobilePhone
>>
Connect to Network
HandleCommunication
Switch on
>>
Display 'Enter PIN'
>>
Enter PIN
>>
Network Connection
>>
Show Time-Battery-Connectivity
UsePhone*
[>
SwitchOff
Wait Connection
>>
Connect to network
MakeCall
[]
UseOtherFunctions
Select Number
>>
PressYes
>>
HaveConversation*
[>
PressNo
HandleMessages
[]
Tools
[]
Settings
Number from List
[]
NewNumber
SelectList
>>
SelectNameEnterName
>>
[]>>
ShowNumber
RecallNumber
[]>>
EnterNumber

Representations of Task Models

	GOMS	UAN	CTT	MAD	GTA
Sequence	X	X	X	X	X
Order independence		X	X		X
Interruption		X	X	X	
Concurrency	**Only CPM-GOMS**	X	X	X	X
Optionality			X	X	
Iteration		X	X	X	X
Allocation			X		X
Objects			X		X
Performance	X		X		X
Pre-post conditions	X	X	X	X	X

Tool support

- GTA – Euterpe
- VTMB -TAMOSA
- ICO – PetShop
- Teallach
- QGOMS