CEIHM

Modèle de Tâches

Université Nice Sophia Antipolis (Polytech)

Marco Winckler

Université Nice Sophia (Polytech) | I3S | SPARKS team | bureau 446

winckler@i3s.unice.fr

http://www.i3s.unice.fr/~winckler/

UNIVERSITÉ CÔTE D'AZUR

POLYTECH NICE SOPHIA

References

Palanque P., Martinie C., Winckler M. (2017) Designing and Assessing Interactive Systems Using Task Models. In: Bernhaupt R., Dalvi G., Joshi A., K. Balkrishan D., O'Neill J., Winckler M. (eds) Human-Computer Interaction – INTERACT 2017. INTERACT 2017. Lecture Notes in Computer Science, vol 10516. Springer

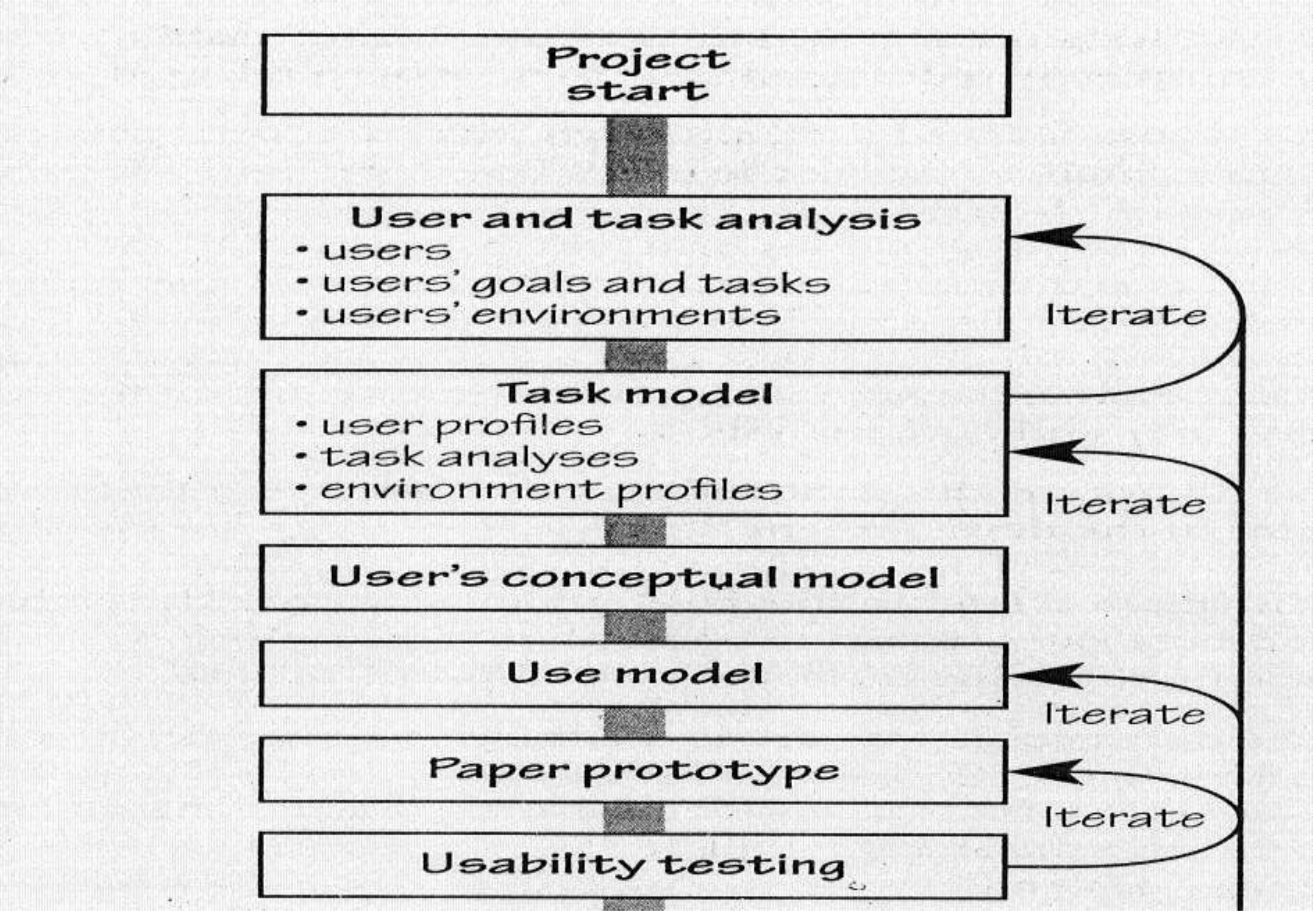
Structure of the lecture

- Introduction to Model-based Approaches
- State of art in Task Models
- ConcurTaskTrees
- Task Models for Cooperative Applications
- HAMSTERS
- Task Models in Usability Evaluation

Why Model-based approaches?

- Highlight important information
- Help to manage complexity
- Useful to support design methods
- One important aspect of most development methods
- Reason about models
- Documentation and user help
- …

How does an interative UCD Process look like



Project start

User and task analysis
- users
- users' goals and tasks
- users' environments

Task model
- user profiles
- task analyses
- environment profiles

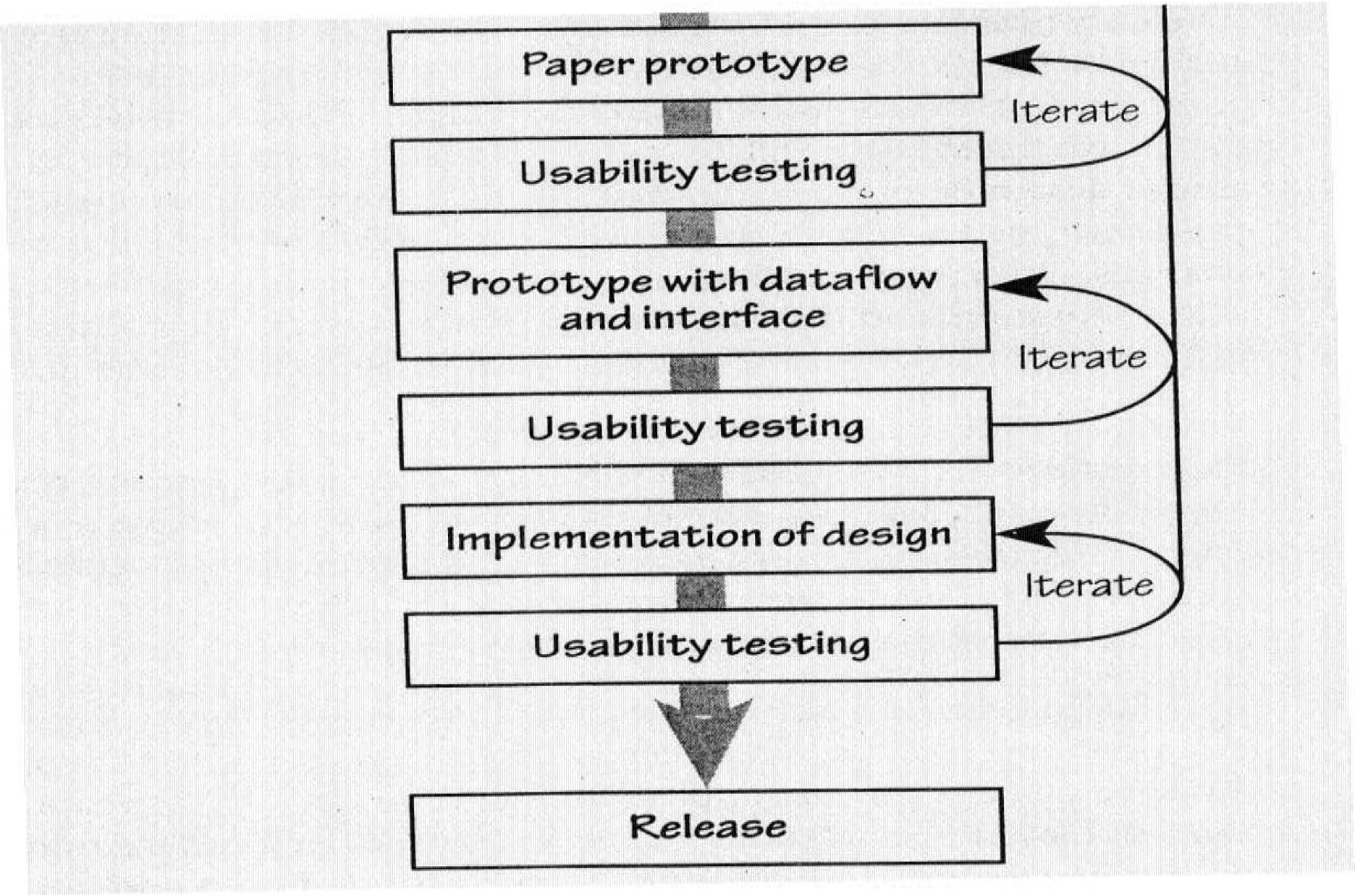
User's conceptual model

Use model

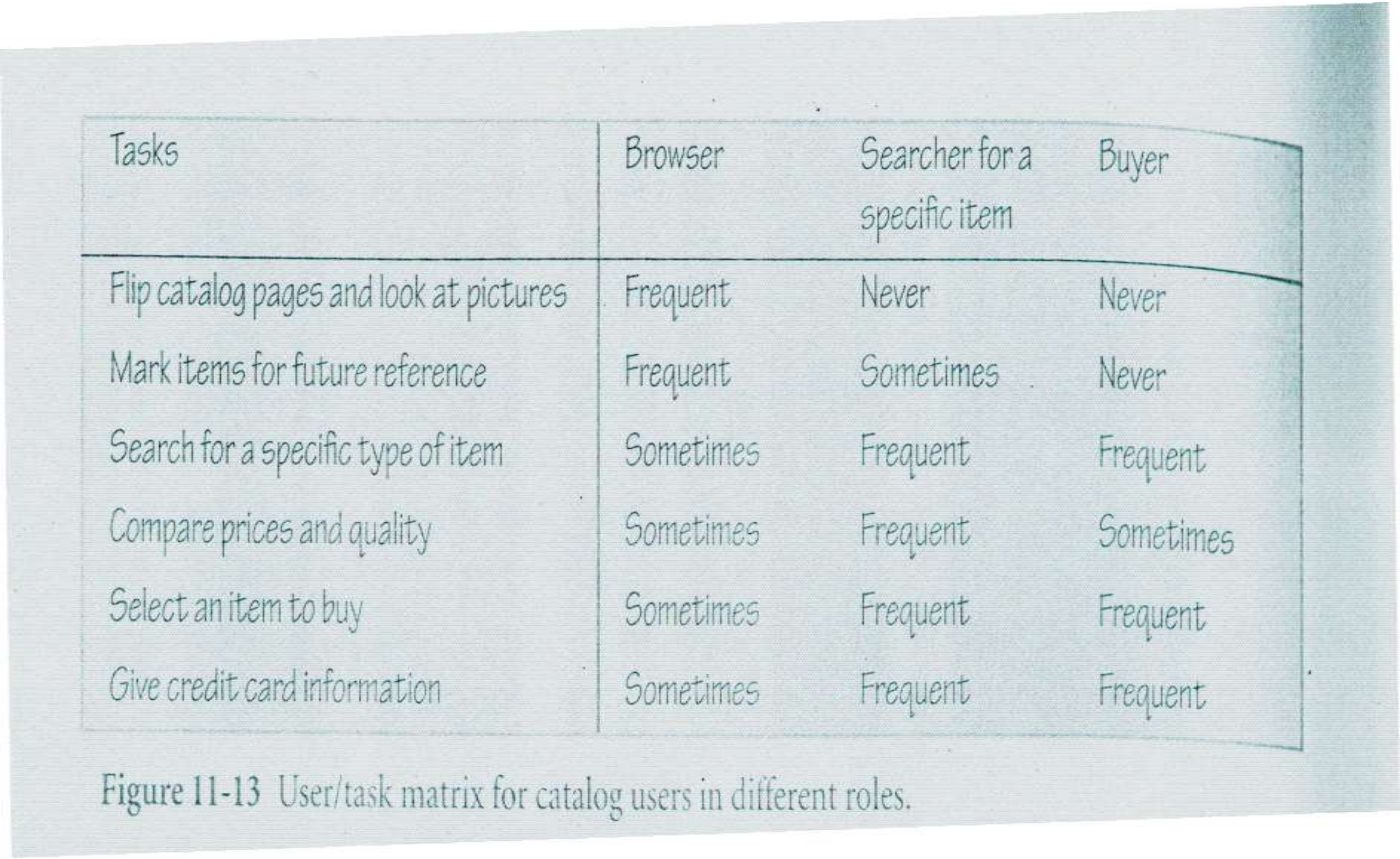
Paper prototype

Usability testing

How does an interative UCD Process look like



user/task matrix



Tasks	Browser	Searcher for a specific item	Buyer
Flip catalog pages and look at pictures	Frequent	Never	Never
Mark items for future reference	Frequent	Sometimes	Never
Search for a specific type of item	Sometimes	Frequent	Frequent
Compare prices and quality	Sometimes	Frequent	Sometimes
Select an item to buy	Sometimes	Frequent	Frequent
Give credit card information	Sometimes	Frequent	Frequent

Figure 11-13 User/task matrix for catalog users in different roles.

multi-level flowchart

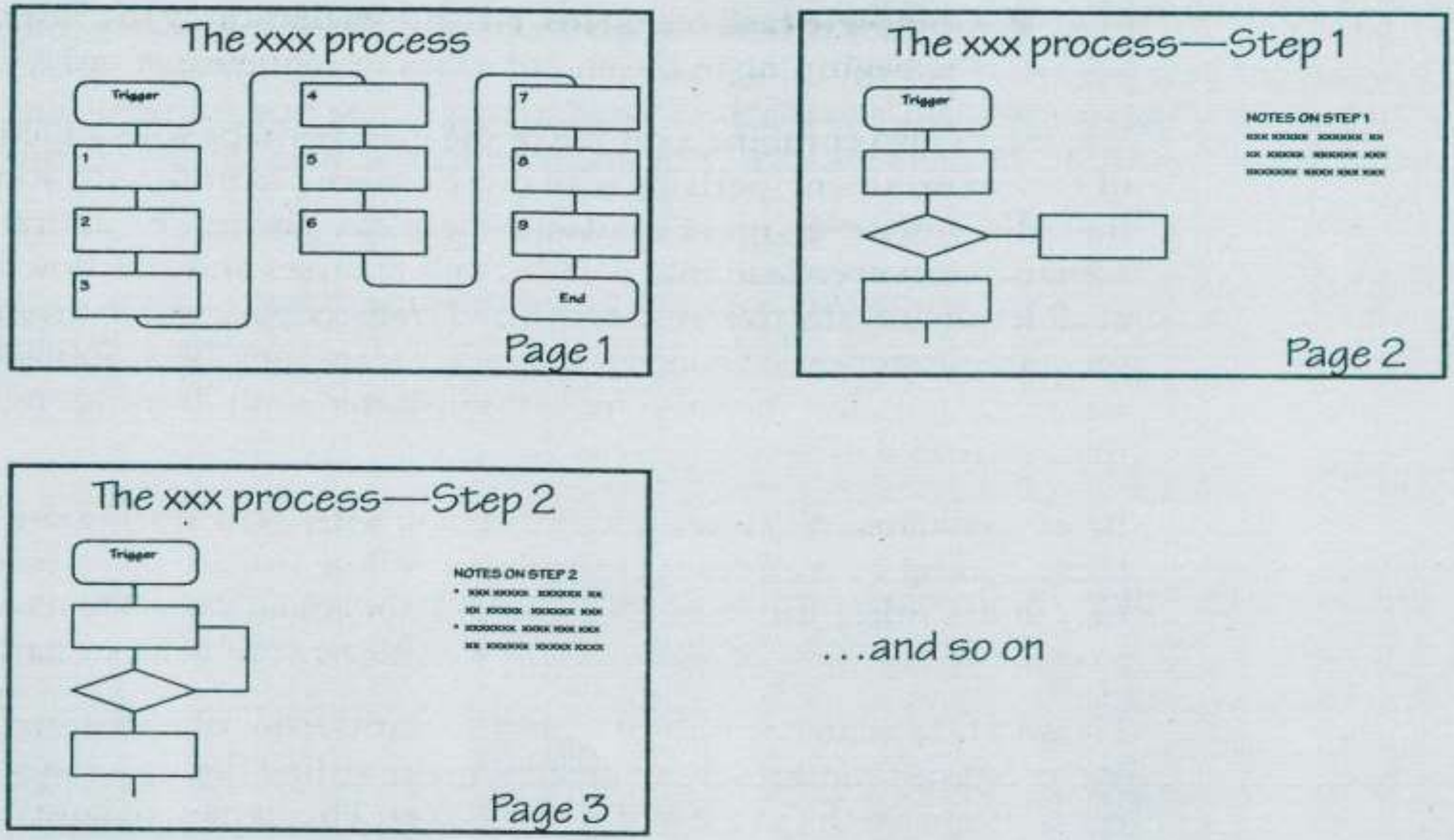


Figure 11-15 If you have a complex process and want to show it in detail, consider a multi-level flowchart like this one.

Task analysis: Definitions

- Task
- Goal
- Basic Task
- Task Analysis <> Activity analysis
- Task Modeling

What is the point of task analysis?

- Determine
  - What the users do
  - The tools they use to do their work
  - The information they know or the information they must know for performing their work
- Cover all/most cases
- Cover all/most users

One small example

- To clean the house
  - Take the vacuum cleaner
  - Plug it in
  - Use it in all the rooms
  - When the bag is full, empty it
  - Store the vacuum cleaner
- You must know
  - Where is the vacuum cleaner
  - Where are the plugs
  - How to remove the bag
  - Where are the new bags
  - Evaluate if a room is clean or not

Information gathering

- Planned procedure
  - Ask the boss
  - Look at the manual
- Actual procedure
  - Ask the operator
  - Record operator behaviour
- Optimal procedure
  - The target for a good system

Task Analysis (Web application for museum)

- Tourist
  - general and clear information in limited amount
  - access by spatial representations
  - guided navigation
- Art student
  - some basic knowledge
  - choice among different types of information
- Expert
  - precise requests
  - detailed description of the information requested

Beware about the abstraction level

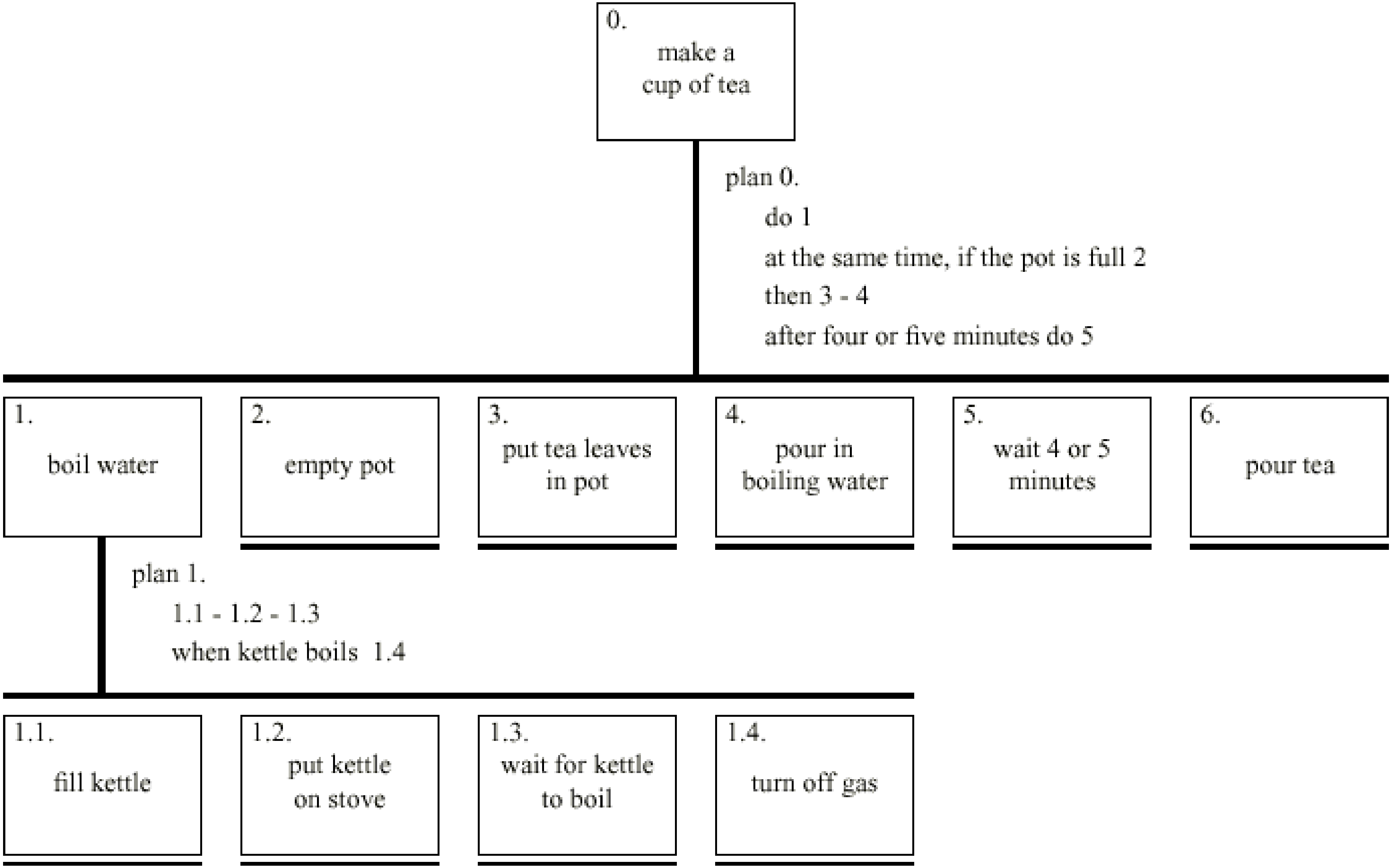
- If you ask the following question:
  What are you doing?
- You can get:
  - I am striking ctrl-B keys
  - I make "bonjour" in bold
  - I make on word bold
  - I highlight a word
  - I am modifying a document
  - I am writing a letter
  - I keep contact with my familly
  - I am waiting for a phone call ☹

Building the task tree

- 1 Build a list of tasks
- 2 Gather high-level tasks
- 3 Breakdown low level tasks

- Where to stop ?
  - Is the action « empty bag » decomposed enough?
  - Objective: only decompose pertinent tasks (objective optimization/automation of work)
  - User actions: cognitive/motor usually not that interesting to decompose (except for performance evaluation)
  - System activity MUST not be represented (will be in other models)
- Highest level = reproduction of the specie
- Lowest level = biological life (breathe, drink a coffee , …)

Example: make a cup of tea



0.
make a cup of tea

plan 0.
do 1
at the same time, if the pot is full 2
then 3 - 4
after four or five minutes do 5

1.
boil water

2.
empty pot

3.
put tea leaves in pot

4.
pour in boiling water

5.
wait 4 or 5 minutes

6.
pour tea

plan 1.
1.1 - 1.2 - 1.3
when kettle boils 1.4

1.1.
fill kettle

1.2.
put kettle on stove

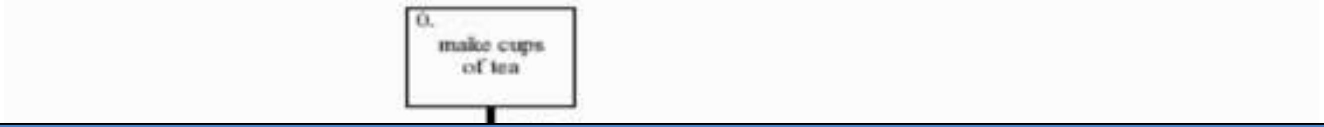
1.3.
wait for kettle to boil

1.4.
turn off gas

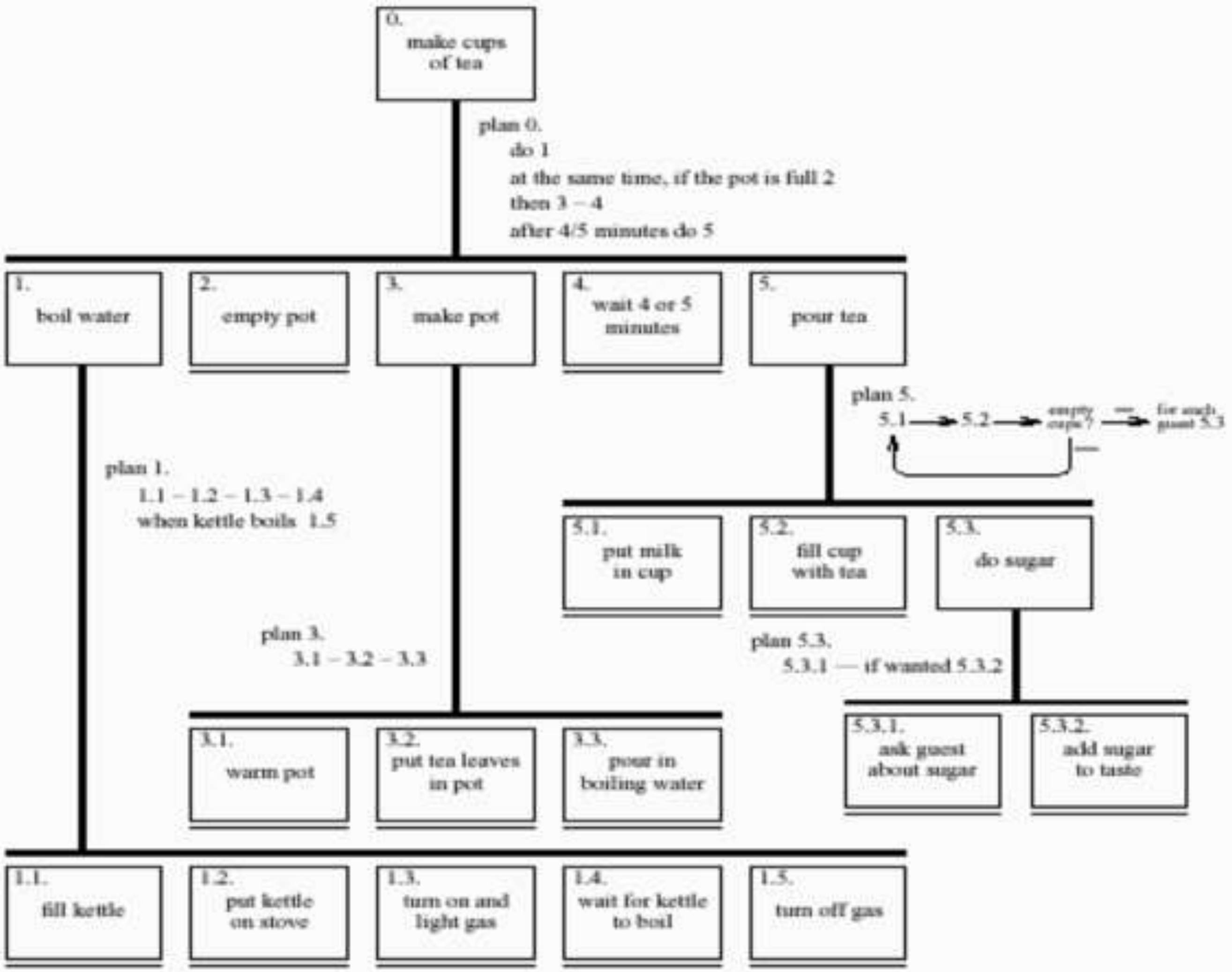
Refining Models

- Once the model is built (whatever notation you use)
  - How to be sure that it is correct?
  - How to improve it?
- Some heuristics
  - Coupled actions: e.g., where is "turn gas on"?
  - Restructure : e.g. make a task generic “make pot”
  - Balance : e.g. is “pour tea” simpler than “make pot”?
  - Generalise: e.g. make one cup … make many

Example: make a cup of tea



Example: make a cup of tea



0. make cups of tea

plan 0.
do 1
at the same time, if the pot is full 2
then 3 – 4
after 4/5 minutes do 5

1. boil water

2. empty pot

3. make pot

4. wait 4 or 5 minutes

5. pour tea

plan 5.
5.1 → 5.2 → empty cups? → for each guest 5.3

5.1. put milk in cup

5.2. fill cup with tea

5.3. do sugar

plan 5.3.
5.3.1 — if wanted 5.3.2

5.3.1. ask guest about sugar

5.3.2. add sugar to taste

plan 1.
1.1 – 1.2 – 1.3 – 1.4
when kettle boils 1.5

plan 3.
3.1 – 3.2 – 3.3

3.1. warm pot

3.2. put tea leaves in pot

3.3. pour in boiling water

1.1. fill kettle

1.2. put kettle on stove

1.3. turn on and light gas

1.4. wait for kettle to boil

1.5. turn off gas

Examples of task analysis

- Tourist
  - Book a train ticket
  - Find a monument in the city
  - Game of 15
- A client using a cash machine
  - Withdraw money
  - Check the balance on the account
  - …

Party : The game of 15

- The game is played by 2
- You have the following tokens 1, 2, 3, 4, 5, 6, 7, 8, 9
- People play in turn, each player take one token at a time, one token can only be taken once
- The first with 3 token of which the sum adds up to 15 is the winner

□ Play this game without tool (paper, pencil or other!)

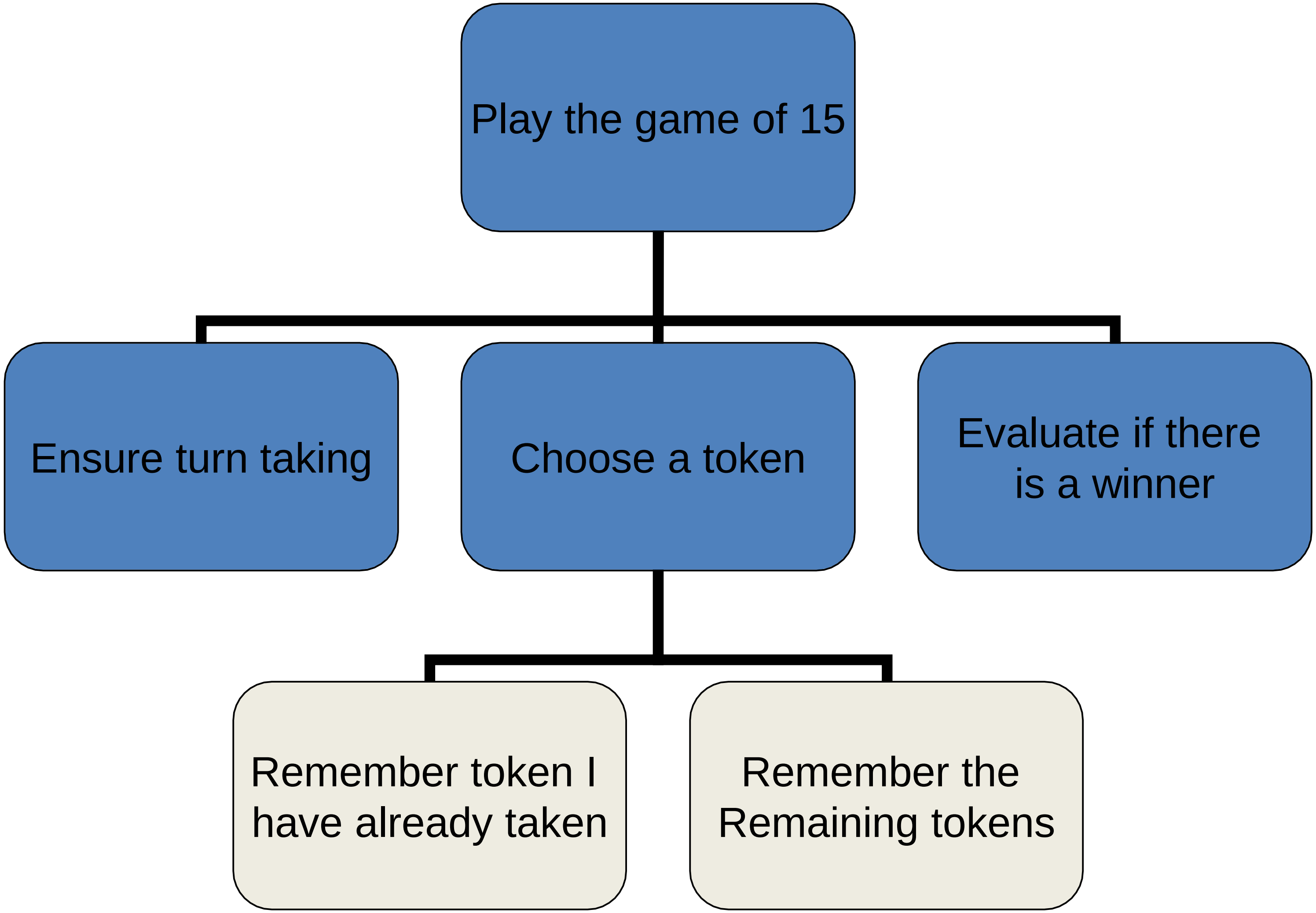
Party : The game of 15

- Analyze one player task
  - Propose a decomposition in sub tasks
- Propose an interface (paper +pencil or computer tool) for this game
  - The user interface must (of course) help the player to achieve his task

Beware !

- Not a cognitivist approach
- Don’t include the system
- Make sure models are correct
- No algorithmic description (calculation of who is the winner)
- No task migration (the system performs tasks assigned to the player)
- Who starts (difference between you play together and you play with ME - elbow communication)

Task Analysis



Play the game of 15

Ensure turn taking

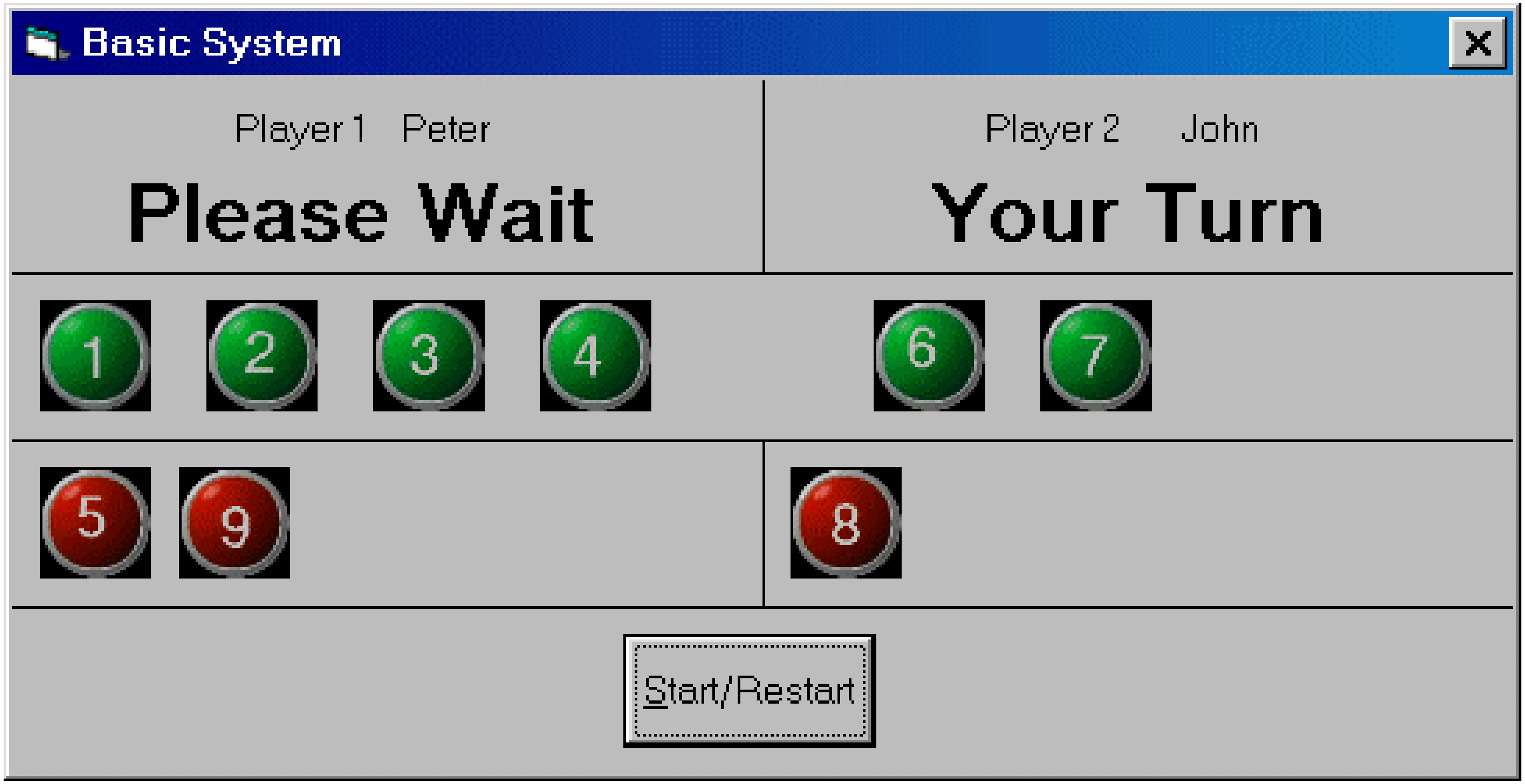
Choose a token

Evaluate if there is a winner

Remember token I have already taken

Remember the Remaining tokens

Your solution



Basic System

Player 1 Peter

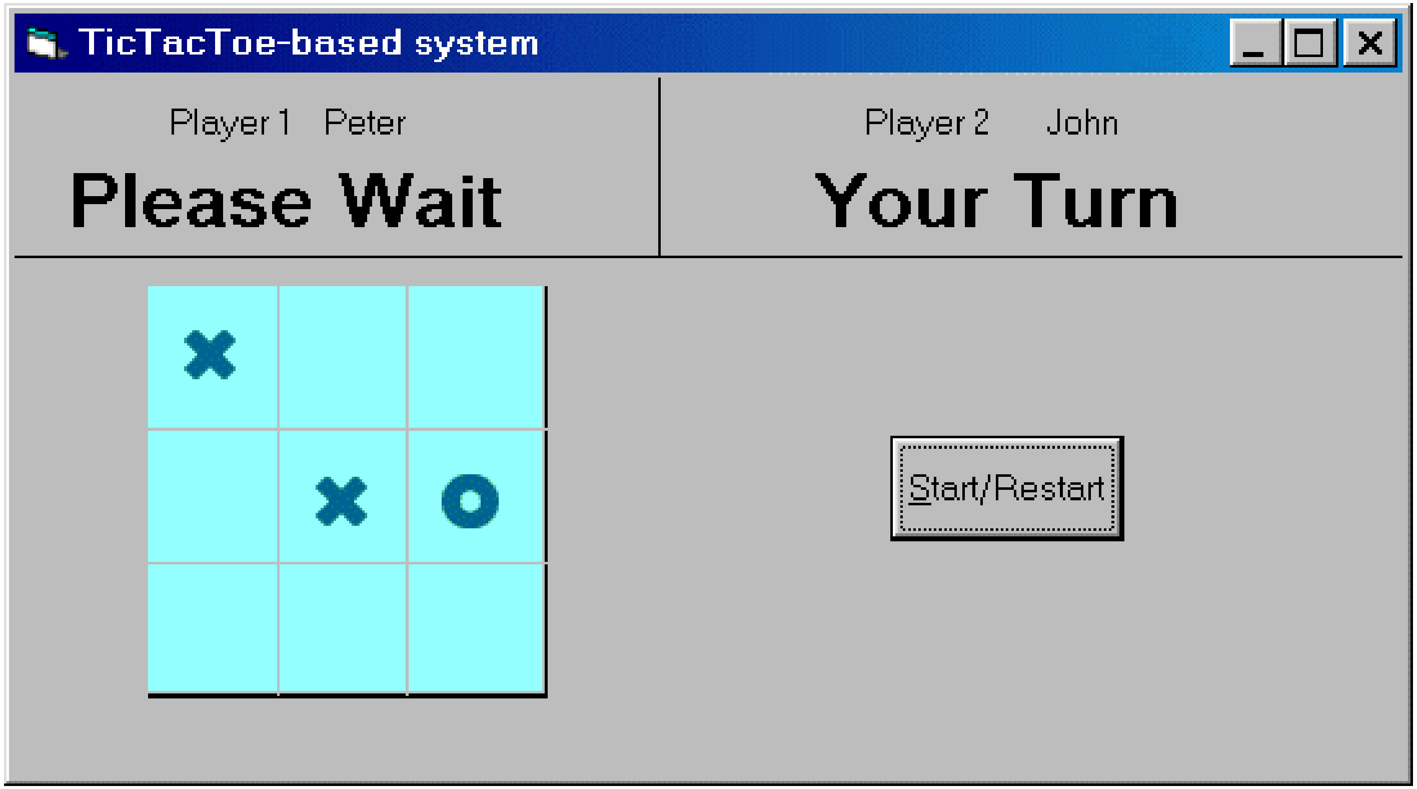
Please Wait

Player 2 John

Your Turn

Another solution

8	3	4
1	5	9
6	7	2



TicTacToe-based system

Player 1 Peter

Please Wait

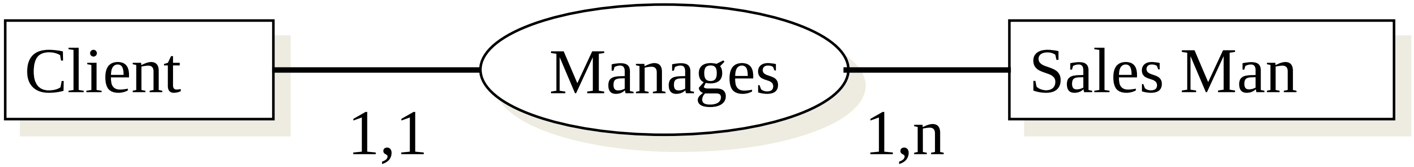
Player 2 John

Your Turn

X		
	X	O

Functioning versus Usage Structuring

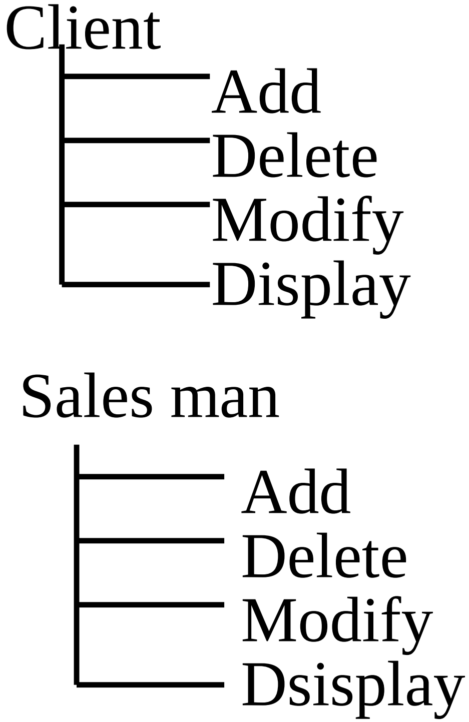
- Extraterrestrial
- Example of a computer system
  - We have a database with the following schema:



  - Propose a menu tree allowing to manipulate that database

Functioning structure

- View of the computer Scientist
  - The tree structure reflects the database structure
  - Grouping is made by function code similarity



Client

Add

Delete

Modify

Display

Sales man

Add

Delete

Modify

Dsisplay

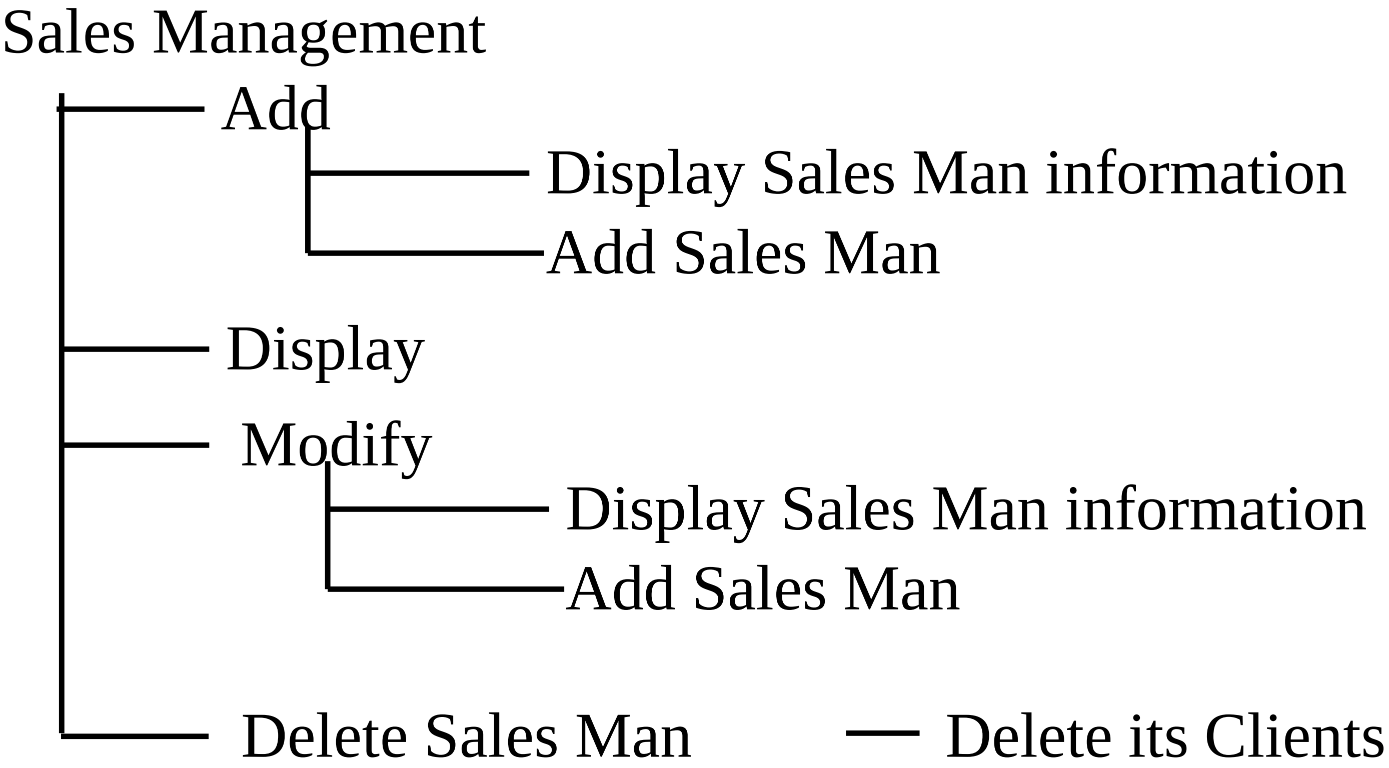
Example of task described as a scenario

- « I spent my day at work entering information for new clients and I must associate them with the more suitable sales man (according to various criteria) … sometimes I receive an invoice from a sales man recently recruited. In such a case, I must create both the new sales man and the client. Sometimes a sales man quits the company. Usually they leave the company for another one and the clients follow them. I then have to delete the sales man and all its related … »

- Propose a tree structure adapted to the tasks

Usage structure (1/2)

- User interface is structured according to the task



Sales Management

Add

Display Sales Man information

Add Sales Man

Display

Modify

Display Sales Man information

Add Sales Man

Delete Sales Man

Delete its Clients

Things are still problematic

- There is no
  - Menu for deleting clients (without sales man)
  - Client modification
  - …

U=1/F

- There is
  - Access to clients through their sales man
  - Display of client information without getting to their sales man first
  - …

Engineering task models

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

The many possible task models

- Existing System
  - One or many systems
  - One or many operators
- Envisioned System
  - One or many systems
  - One or many operators
- Variation of models (co-evolution)

Use of Task Models

- Better understanding of the application (and in particular its use)
- Record discussions (multidisciplinary)
- Help design
- Help usability evaluation
- Help performance evaluation
- Help user in performing the tasks (contextual help)
- Documentation (content + structure)

Representations of Task Models

- Hierarchical task analysis
- GOMS family
- UAN
- K-made (web)
- AMBOSS (web)
- **HAMSTERS**
- 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
```

Exemple GOMS-KLM

Description	**Operation**	**Time (sec)**
Reach for mouse	H[mouse]	0.40
Move pointer to "Replace" button	P[menu item]	1.10
Click on "Replace" command	K[mouse]	0.20
Home on keyboard	H[keyboard]	0.40
Specify word to be replaced	M4K[word]	2.15
Reach for mouse	H[mouse]	0.40
Point to correct field	P[field]	1.10
Click on field	K[mouse]	0.20
Home on keyboard	H[keyboard]	0.40
Type new word	M4K[word]	2.15
Reach for mouse	H[mouse]	0.40
Move pointer on Replace-all	P[replace-all]	1.10
Click on field	K[mouse]	0.20
Total		**10.2**

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

- 2 complementary set of information
  - A hierarchy of tasks LOTOS (proche de CTT)
  - A table for describing states and feedback
- Textual notation
- Introduced in 1992 (Hix & Hartson huge success Developing user interfaces Ensuring, Usability Through Product & Process)

Example of UAN specification

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

Task: SelApplication

User Action	Interface Feedback	Interface State

~[x,y in AppICON]
∨∧$(t<t_{doubleClick})$ ∨∧
	w'!: w'-!
	UnMap(PrevAppliMenu)
	Map(AppMenu)
	UnMap(AppICON)
		CurAppli=App
		CurMenu=AppMenu

ConcurTaskTrees

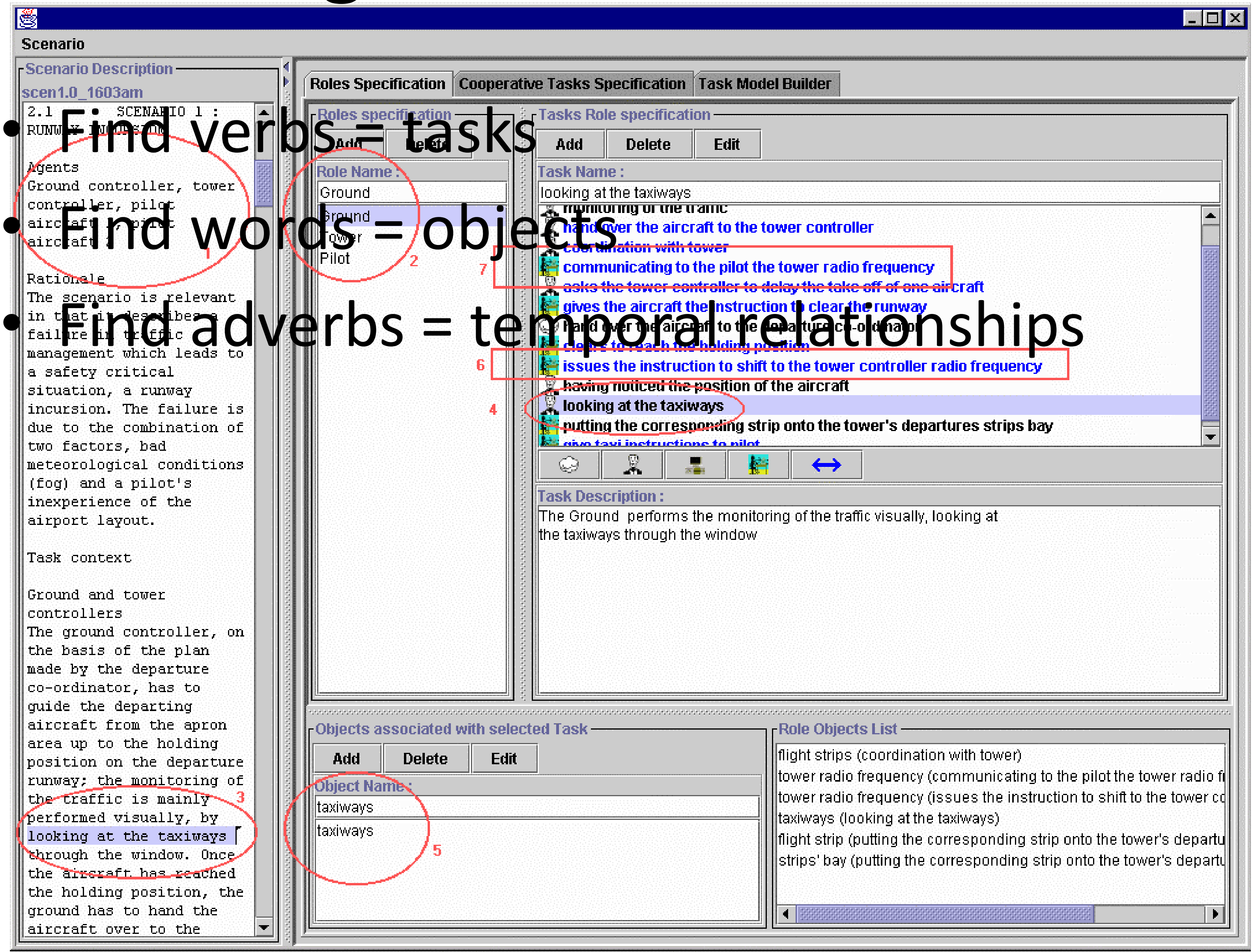
- Focus on Activities
- Hierarchical Structure
- Graphical Syntax
- Rich set of temporal operators
- Task allocation
- Objects and task attributes

Task Models vs Scenarios

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

Moving from scenarios to tasks

- Find verbs = tasks
- Find words = objects
- Find adverbs = temporal relationships



2.1 SCENARIO 1 : 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 area up to the holding position on the departure runway; the monitoring of the traffic is mainly performed visually, by looking at the taxiways through the window. Once the aircraft has reached the holding position, the ground has to hand the aircraft over to the

Role Name: Ground

- Ground
- Tower
- Pilot

Task Name: looking at the taxiways

- monitoring of the traffic
- hand over the aircraft to the tower controller
- coordination with tower
- communicating to the pilot the tower radio frequency
- asks the tower controller to delay the take off of one aircraft
- gives the aircraft the instruction to clear the runway
- hand over the aircraft to the departure co-ordinator
- clears to reach the holding position
- issues the instruction to shift to the tower controller radio frequency
- having noticed the position of the aircraft
- looking at the taxiways
- putting the corresponding strip onto the tower's departures strips bay
- give taxi instructions to pilot

Task Description: The Ground performs the monitoring of the traffic visually, looking at the taxiways through the window

Object Name: taxiways

- taxiways

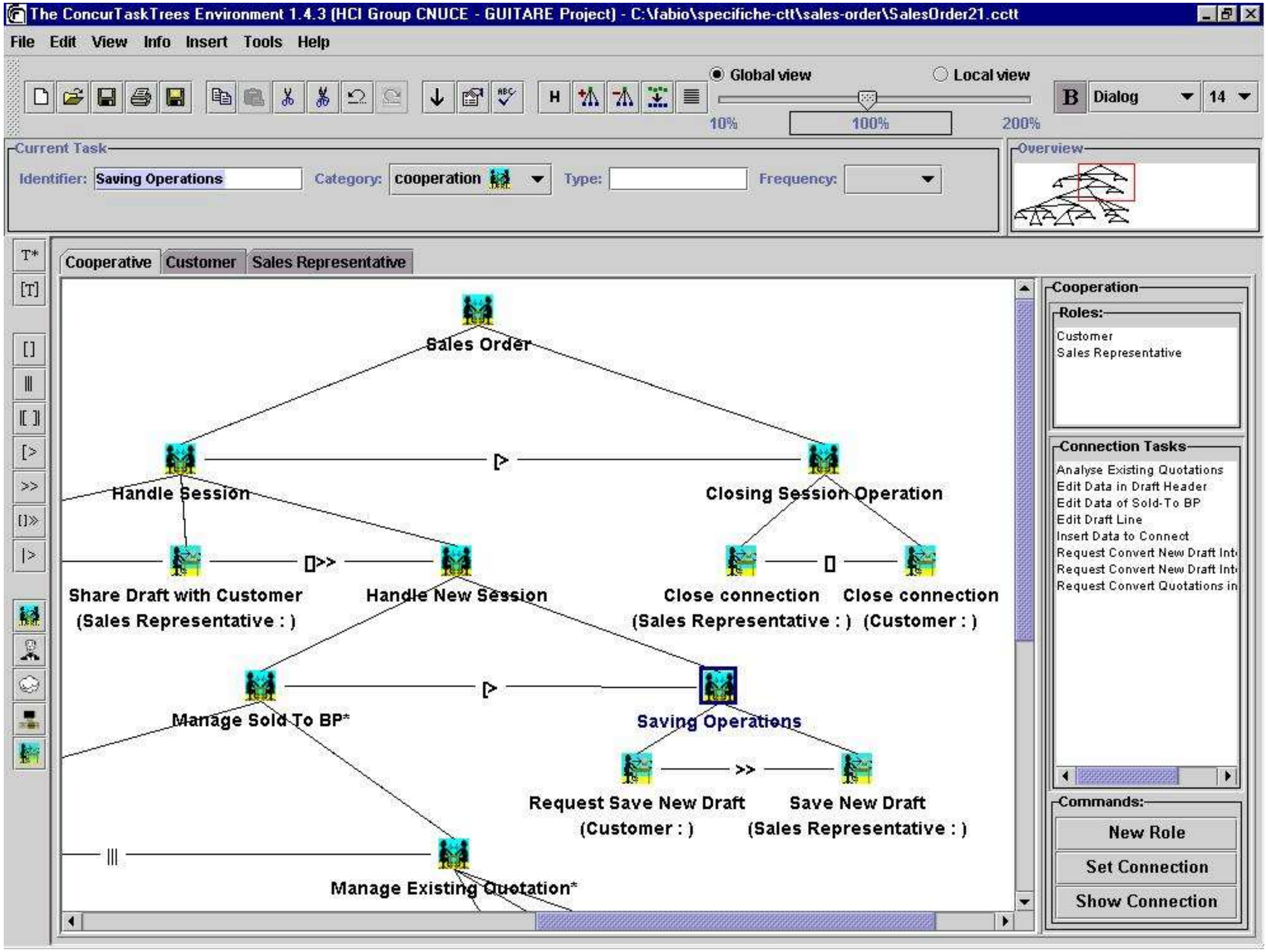
Role Objects List:

- flight strips (coordination with tower)
- tower radio frequency (communicating to the pilot the tower radio fr
- tower radio frequency (issues the instruction to shift to the tower co
- taxiways (looking at the taxiways)
- flight strip (putting the corresponding strip onto the tower's departu
- strips' bay (putting the corresponding strip onto the tower's departu

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
- Concurrency (must finish first) T1 |=| T2
- Optionality [T]

CTT Editor



The ConcurTaskTrees Environment 1.4.3 (HCI Group CNUCE - GUITARE Project) - C:\fabio\specifiche-ctt\sales-order\SalesOrder21.cctt

● Global view ○ Local view

10% 100% 200%

Dialog 14

Current Task

Identifier: Saving Operations

Category: cooperation

Type:

Frequency:

Cooperation

Roles:

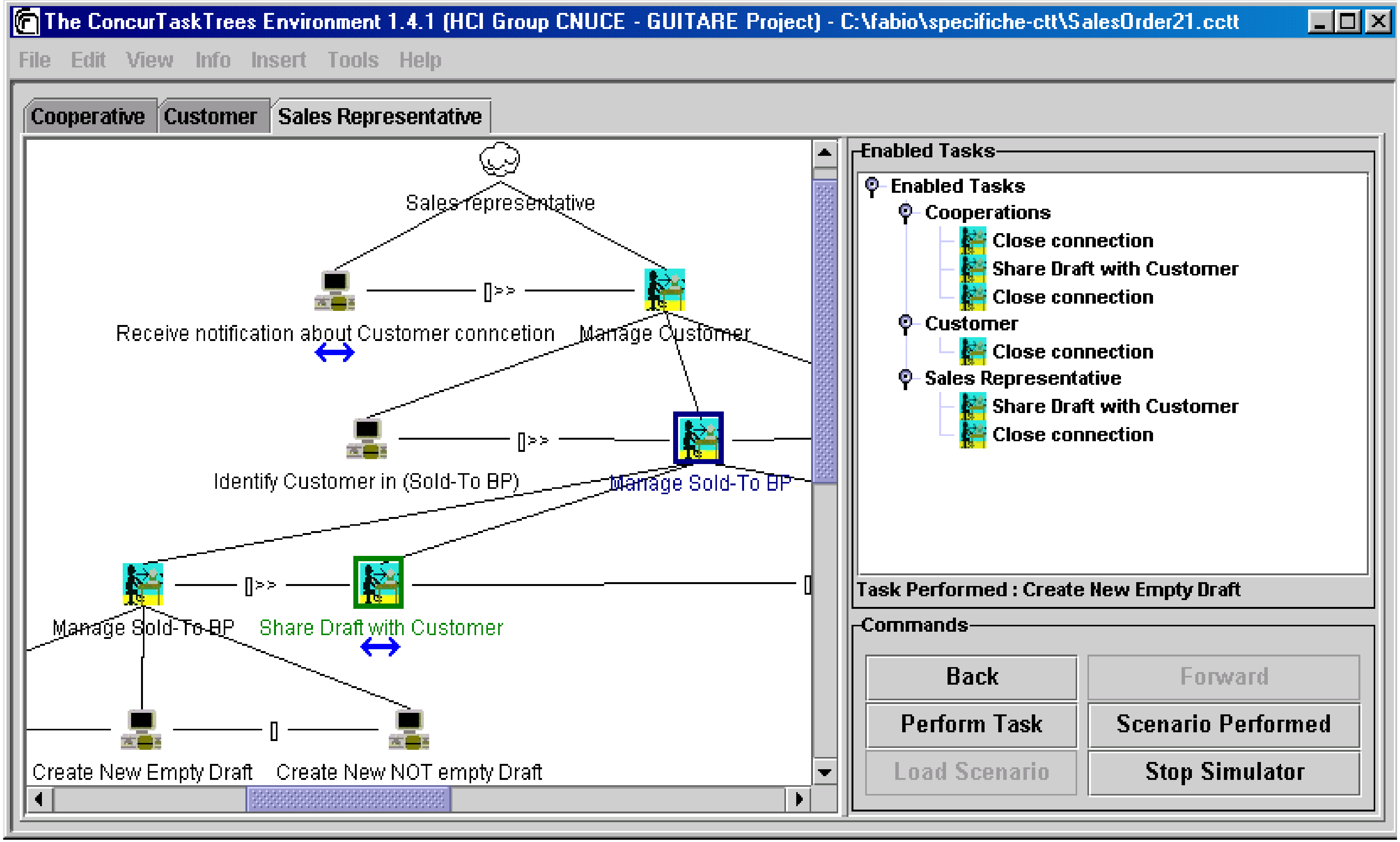
Customer
Sales Representative

Connection Tasks

Analyse Existing Quotations
Edit Data in Draft Header
Edit Data of Sold-To BP
Edit Draft Line
Insert Data to Connect
Request Convert New Draft Int
Request Convert New Draft Int
Request Convert Quotations in

File
Edit
View
Info
Insert
Tools
Help
Overview
Cooperative
Customer
Sales Representative
Sales Order
Handle Session
[>
Closing Session Operation
Share Draft with Customer (Sales Representative :)
[]>>
Handle New Session
Close connection (Sales Representative :)
[]
Close connection (Customer :)
Manage Sold To BP*
[>
Saving Operations
Request Save New Draft (Customer :)
>>
Save New Draft (Sales Representative :)
|||
Manage Existing Quotation*
Commands:
New Role
Set Connection
Show Connection

Task Simulator



The ConcurTaskTrees Environment 1.4.1 (HCI Group CNUCE - GUITARE Project) - C:\fabio\specifiche-ctt\SalesOrder21.cctt

Enabled Tasks

- Enabled Tasks
  - Cooperations
    - Close connection
    - Share Draft with Customer
    - Close connection
  - Customer
    - Close connection
  - Sales Representative
    - Share Draft with Customer
    - Close connection

Task Performed : Create New Empty Draft

Commands

HAMSTERS Software installation / Slides

- Slides are available
- Need java to be installed on your computer http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
- Need to get Hamsters tool and examples
- Need to get the Hamsters video (file HAMSTERS - Create a project - Create a task model.WMV)

Past-Current Research Projects

- Air Trafic Management (enroute ATC workstations) 1995-2001 & 2010-2014 HALA! Network of excellence & SPAD
    - Dynamic instantiation of widgets, Post WIMP interfaces
    - Time constraint about 3mn (speed vector)
    - Automation and automation degradation / tasks migrations
- Space domain : R&T IMAGES (2004-2006) R&T TORTUGA (2008-2011) R&T ALDABRA (2011-2012) R&T MARACCAS (2012-2013)
    - Specification of satellite ground segments with multimodal interfaces
    - Target application: AGENDA & spacecraft-debris collision avoidance system
    - Collaborative distributed operations / time constraint 24h
- Civil aviation Thales (2001-2004) 2009-2016 (Airbus – dependable interactive cockpits)
    - Interactive Cockpits (ARINC 661 standards)
    - Specification of system architectures for dependable interactive systems – Time constraints 30 sec

Work on task modeling -> HAMSTERS

- Philippe A. Palanque, Rémi Bastide, V. Sengès: Validating interactive system design through the verification of formal task and system models. IFIP Engineering HCI (1995): 189-212
- Philippe A. Palanque, Rémi Bastide: Synergistic Modelling of Tasks, Users and Systems using Formal Specification Techniques. Interacting with Computers 9(2): 129-153 (1997)
- David Navarre, Philippe A. Palanque, Fabio Paternò, Carmen Santoro, Rémi Bastide: A Tool Suite for Integrating Task and System Models through Scenarios. DSV-IS (2001): 88-113
- Marco Winckler, Philippe A. Palanque, Christelle Farenc, Marcelo Soares Pimenta: Task-Based Assessment of Web Navigation Design. TAMODIA Task Models and Diagrams for User Interface Design (2002): 161-169
- Philippe A. Palanque, Sandra Basnyat: Task Patterns for Taking Into Account in an Efficient and Systematic Way Both Standard and Erroneous User Behaviours. Human Error, Safety and Systems Development (2004): 109-130

- Célia Martinie, Philippe A. Palanque, Marco Winckler: Structuring and Composition Mechanisms to Address Scalability Issues in Task Models. INTERACT (2011) 589-609

Course objectives

- Know the benefits of using task description techniques to design interactive systems (tasks-based design)
- Be able to describe users’ activities in a systematic and structured way (examples and practical exercises)
- Know how to use the HAMSTERS notation and tool for editing, analyzing and simulating task models (according to the current state of interaction what are the available tasks) help understand if it is a correct description
- Have experience in analyzing an interactive systems focusing on the tasks the users have to perform (manual and tool supported by HAMSTERS)

- Interact altogether on experiences on task-based design

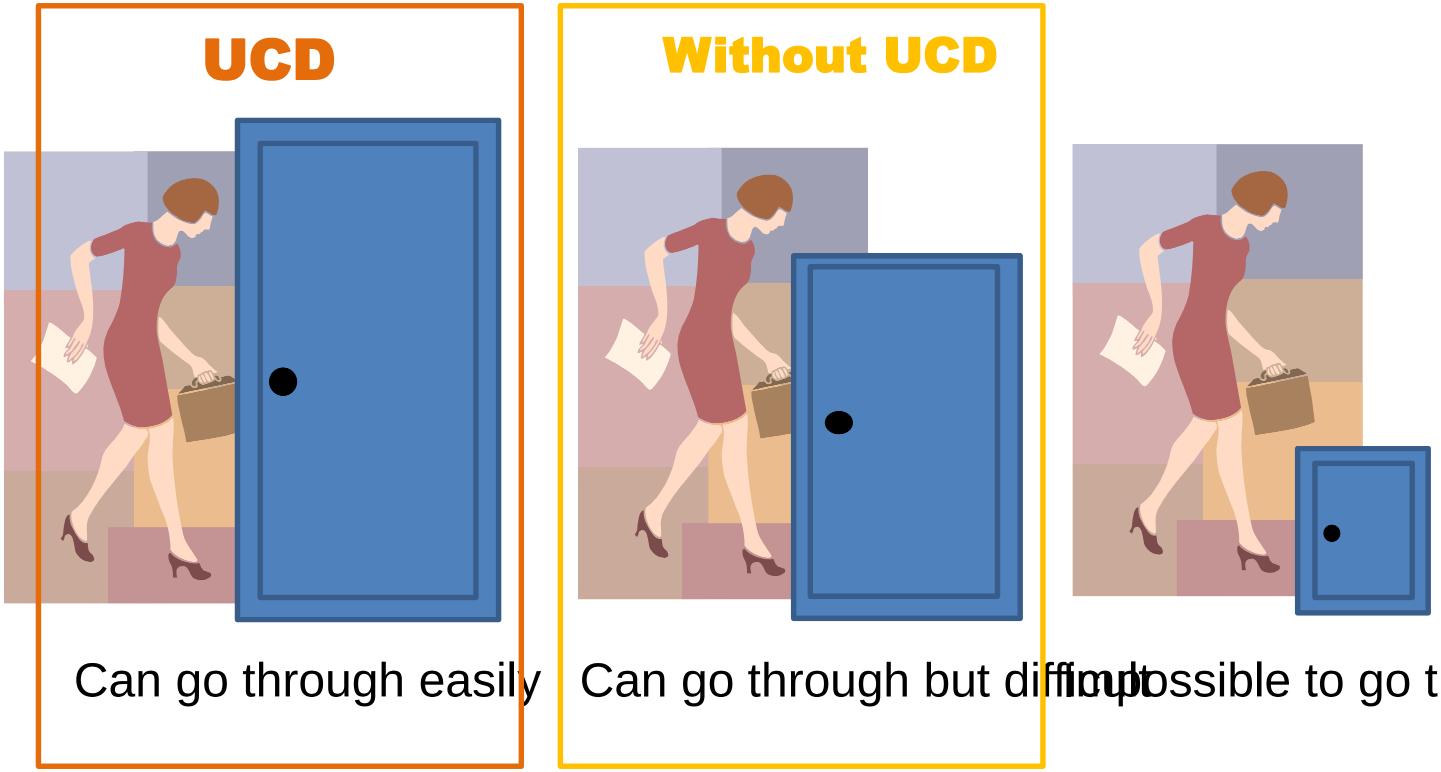
Descriptions of Tasks

- Record information gathered during tasks analysis
- One important aspect of most development methods
- Useful for designing an interactive system that fits user work
  - Identify complex activities in the current work
  - Allocation of tasks to different actors
  - Identify good candidate for automation

Why Model-based approaches?

- Highlight important/relevant information
- Focus on abstract information (stay away from non relevant details)
- Help to manage complexity
- One important aspect of most development methods
- Possibility to reason about models
- Support collaboration amongst stakeholders

Affordance/Usability of models



UCD

Without UCD

Can go through easily

Can go through but difficult

Impossible to go through

Significant Models in HCI

- Task models
- Cognitive models e.g. SRK
- User models e.g. edutainment
- Domain Models e.g. UML class diagrams
- Context Models e.g. location aware systems
- Presentation Models e.g. widgets libraries
- Dialogue models e.g. state machines
- Platform models e.g. cellphones descriptions (display, CPU, …)

Task models (versus task analysis)

- Manage information gathered from task analysis
  - Structure
  - Record
  - Refine
  - Analyze
  - Reuse
- 3 Fundamental concepts in tasks modelling
  - Hierarchical structure of tasks: Hierarchical Task Analysis (Annett & Duncan, 1967) (Annett, 2004)
  - Knowledge representation: Task Knowledge Structure (Johnson et al. 1989)
  - Multi-user/Collaborative activities: Groupware Task Analysis (van der Veer 1996)

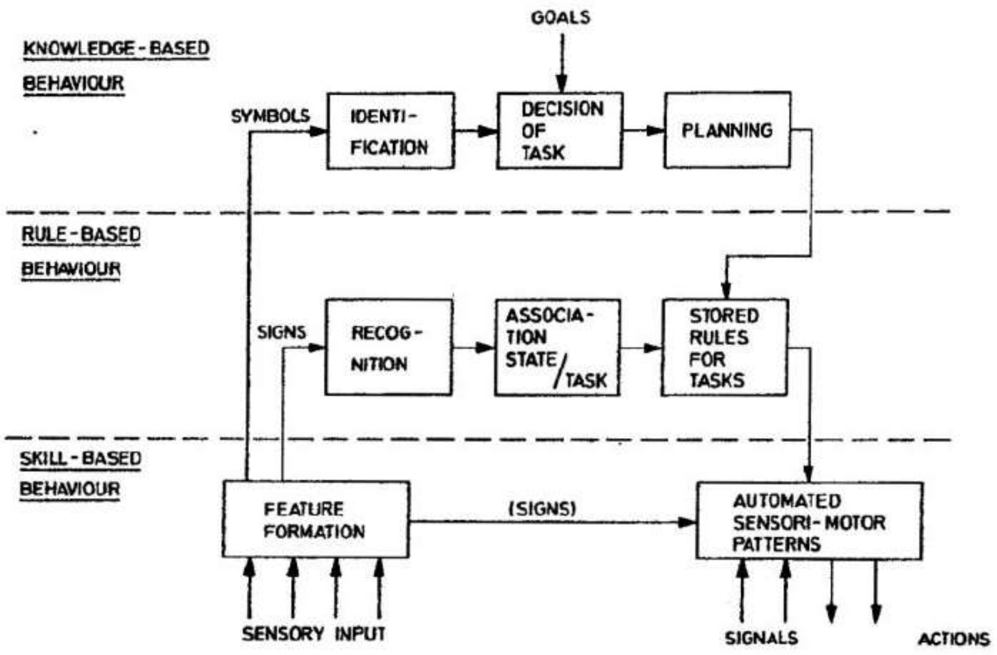
What is the point of task analysis?

- Determine
  - What the users do
  - The tools they use to do their work
  - The information they know or the information they must know for performing their work
  - The knowledge they have (or must have)
  - The objects they must or can use to perform their tasks
  - The strategies they apply (or must know) for performing the tasks
- Cover all/most cases
- Cover all/most users

One small example

Jens Rasmussen. Skills, rules, and knowledge; signals, signs, and symbols, And other distinctions in human performance models. IEEE transactions on systems, man, and cybernetics, vol. Smc-13, no. 3, may / june 1983, 257-266.

- To clean the house
  - Take the vacuum cleaner
  - Plug it in
  - Use it in all the rooms
  - When the bag is full, empty it
  - Stop when finished
  - Store the vacuum cleaner



- You must know
  - Where is the vacuum cleaner?
  - Where are the plugs?
  - How to remove the bag?
  - Where are the empty bags?
  - Should I start by the entrance or by the end of the room? – strategy (related to context)
  - Should I have vacuum cleaner at maximum speed? – strategy
  - Evaluate if a room is clean or not – situational knowledge

Information gathering

- Operators/users work
  - Planned procedure
    - Ask the boss (regulations)
    - Look at the manual or training material
  - Actual procedure
    - Ask the operator
    - Record operator behavior
  - Optimal/Minimal procedure
    - The target for an optimal system (maybe not the best one – e.g. aircraft cockpits)
- Tools around the workspace
  - Post-it, memory joggers, annotations …

Beware about the abstraction level

Example from A. Dix

- If you ask the following question:

  What are you doing?

- You can get:
  - I am striking ctrl-B keys
  - I make the world "bonjour" in bold
  - I make one word bold
  - I highlight a word
  - I am modifying a document
  - I am writing a letter
  - I keep contact with my familly
  - I am waiting for a phone call ☹

Task models as the corner stone of successful design

Task models as the corner stone of successful design

- Main elements of the course
  - HAMSTERS notation
  - Task model creation, edition and validation
- Practical work on an illustrative real life interactive system
- Not presented but included in the annex:
- « Which notation to choose? »: a review of of notations and tools for describing users' tasks

In this part, you will learn how to

- Create a task model
- Build a task model
- Validate a task model
- Modify a task model

Example: Obtain money from an ATM

Photo credit: Adam De Almeida

Why the ATM ?
(Automated Teller Machine)

- Simple but meaningful example:
- System and activities already known by the audience
- Requires manipulation of several types of data

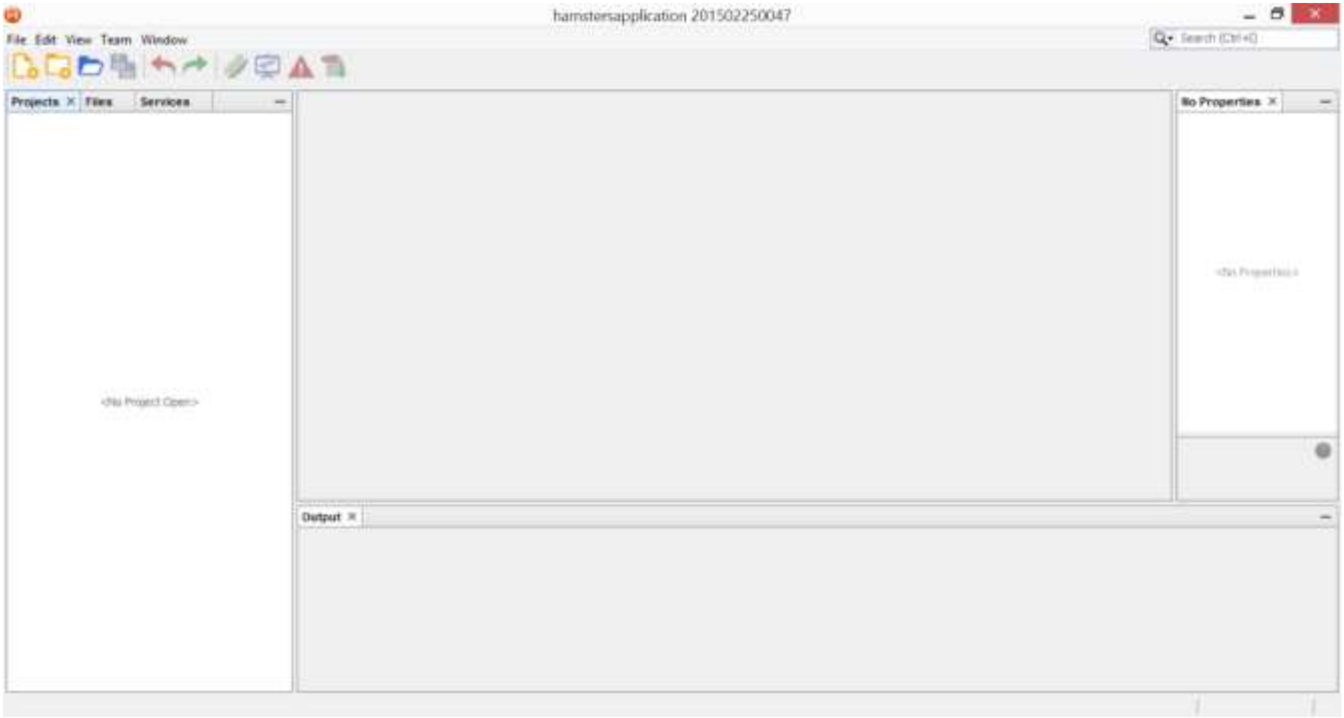
- Several versions of the system for withdrawing money:
- Manual way (at the bank office)
- ATM in public spaces
- Java version of the ATM (for the course)

HAMSTERS TOOL

How to start HAMSTERS

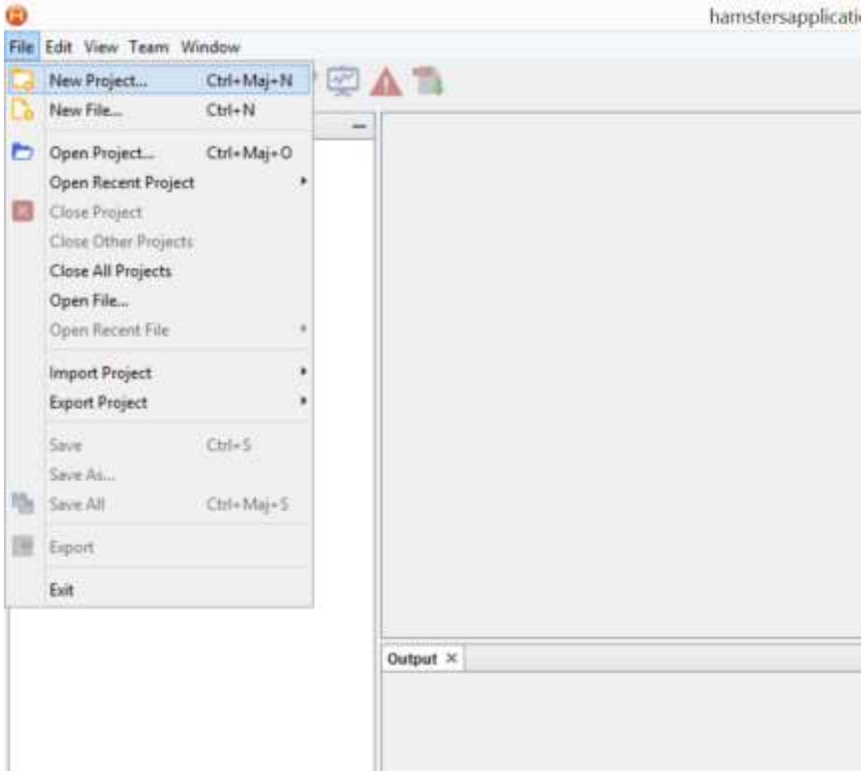
- Make sure Java JRE 8 is installed http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
- Go in directory hamster/bin
- Launch hamsters.exe
- Menu File/new project or click on icon

Launch HAMSTERS

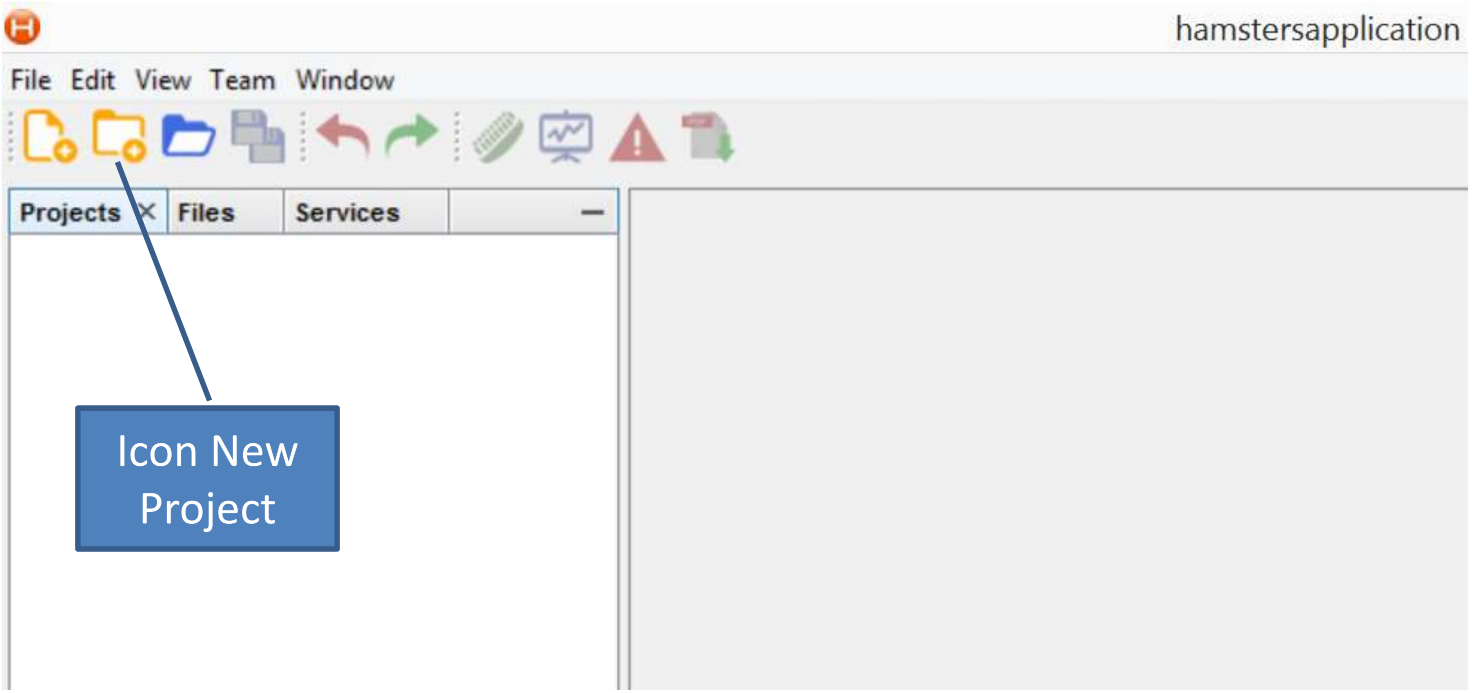


Create a project

Menu File/New Project

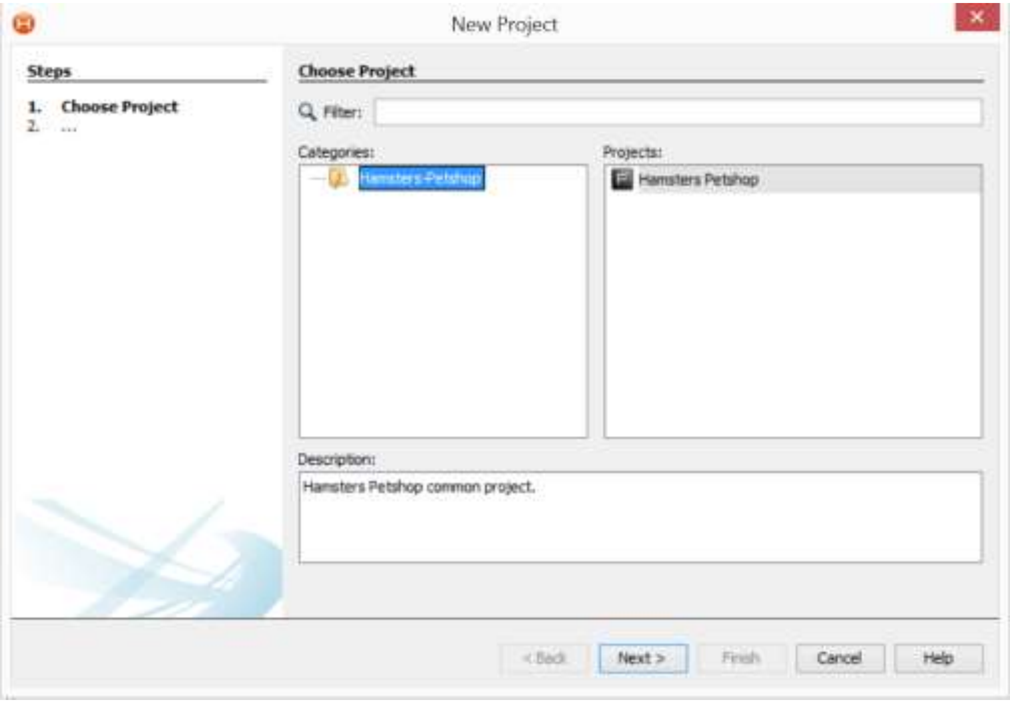


Create a project



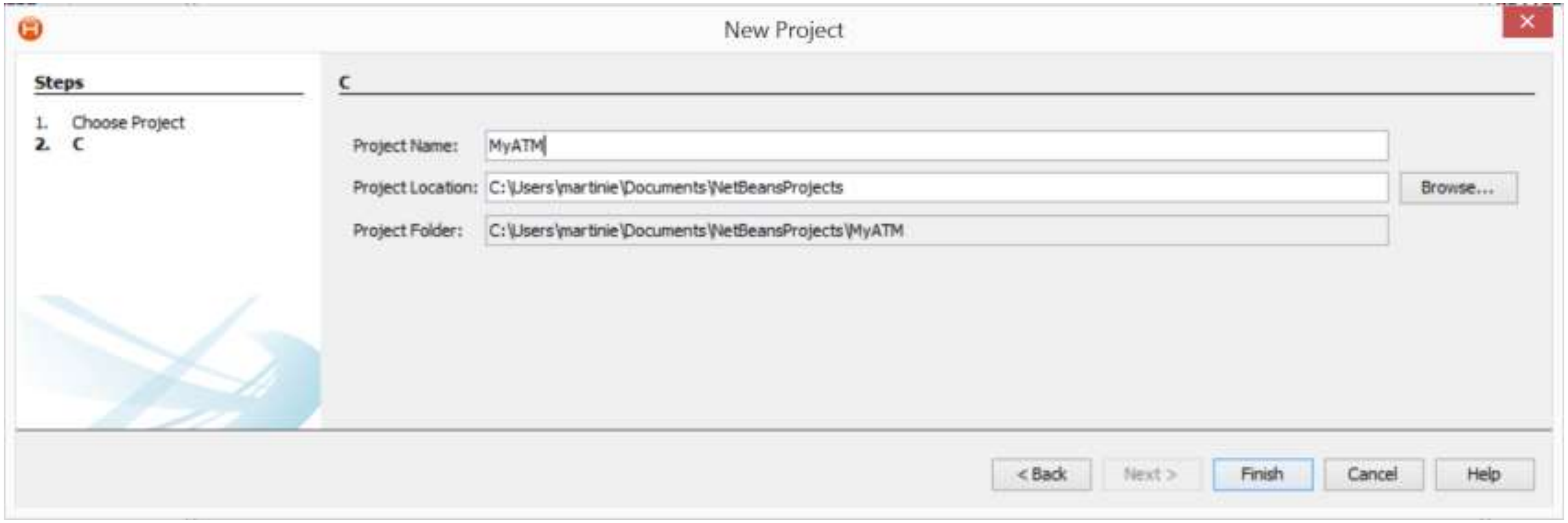
Icon New Project

Click Next



Hamsters Petshop common project.

Enter the name of your project

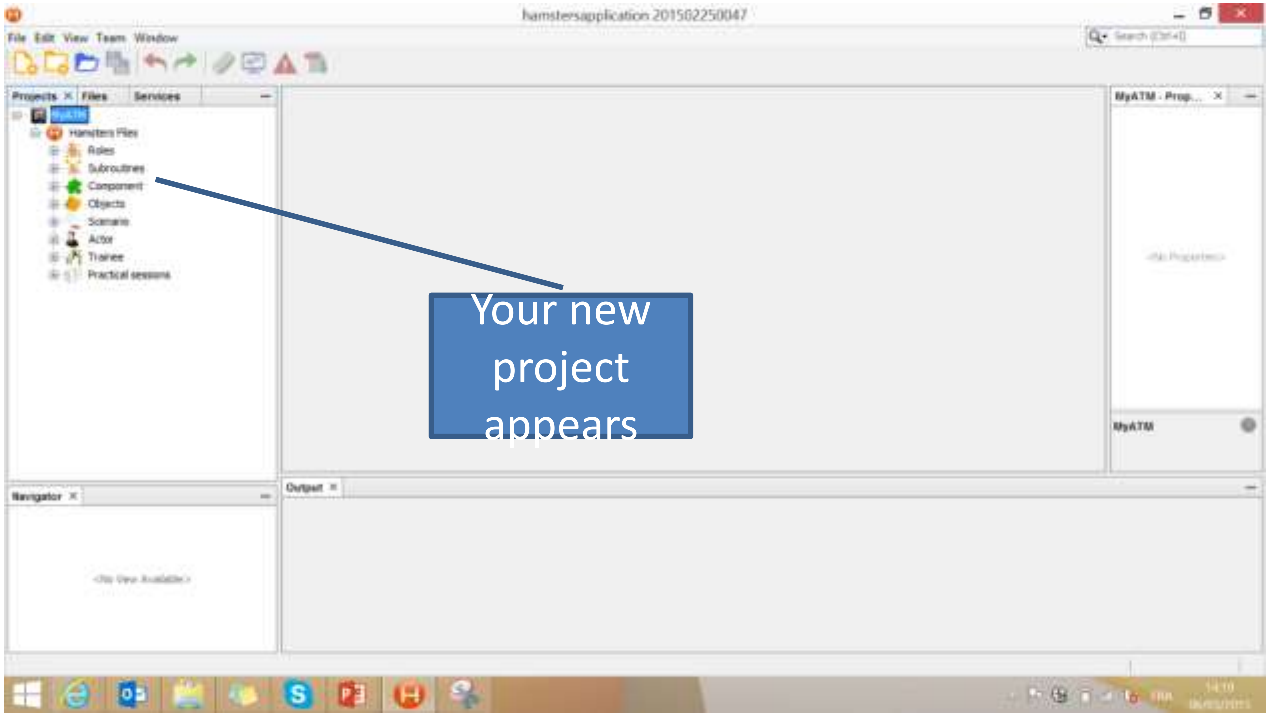


Project Name: MyATM

Project Location: C:\Users\martinie\Documents\NetBeansProjects

Project Folder: C:\Users\martinie\Documents\NetBeansProjects\MyATM

Project created



Your new project appears

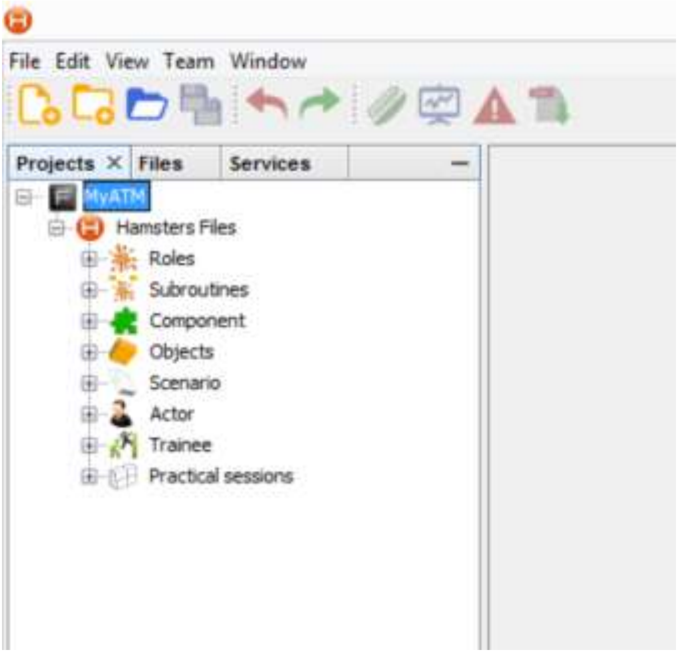
Main directories in the project

Roles
- Contains task models

Subroutines
Components
- Contain reusable task models (presented in part 2)

Objects
- Contain manipulated data: information, physical objects…

Actors
Trainee
Practical sessions
Interactive system runtime and training (not presented today)



HAMSTERS

HUMAN-CENTERED ASSESSMENT AND MODELLING TO SUPPORT ENGINEERING FOR RESILIENT SYSTEMS

Elements of the notation HAMSTERS

- Hierarchy
- Temporal ordering
- Refined task types
- Manipulated objects/information/devices
- Knowledge

Hierarchy

- Decomposition of activities according to goals
- Goals into sub-goals
- Sub-goals into tasks
- Tasks into actions
- …

Exercise 1
Model the main tasks to withdraw money

- Goal: Withdraw money
- Prerequisites:
- Have a withdrawal card
- Know the amount that has to be withdrawn
- Know the PIN number
- Results:
- Take money
- Take back the withdrawal card

Exercise: ATM constraints

- Insert_Card, Enter_Code, Witdraw_Card
- Request_Cash, Select_Amount, Withdraw_Cash

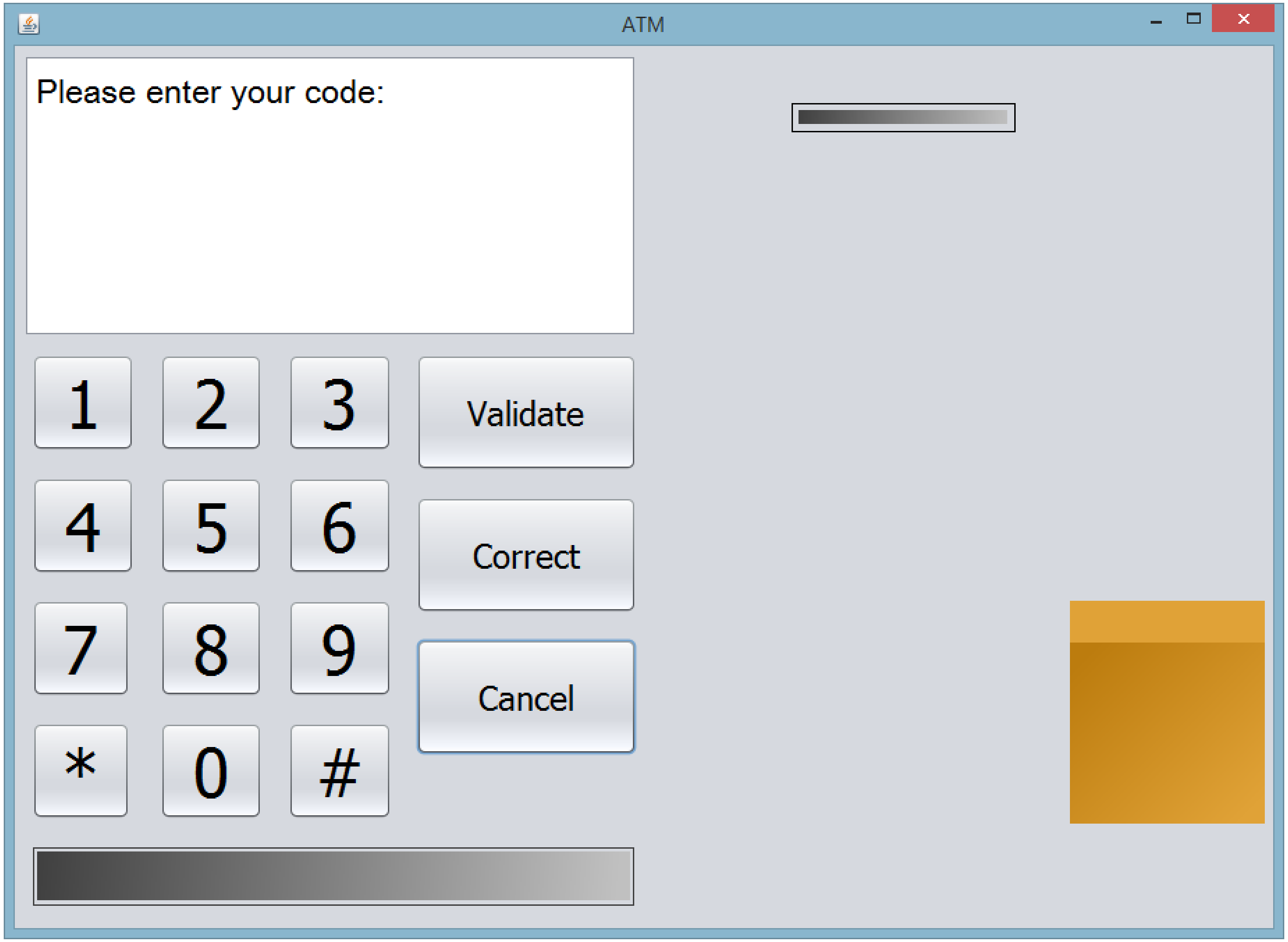
Request_Cash - - - - - before - - - - *Select_Amount*

Select_Amount and *Insert_Card* - - - - - before - - - - - *Withdraw_Cash*

Insert_Card - - - - before - - - - - *Withdraw_Card*

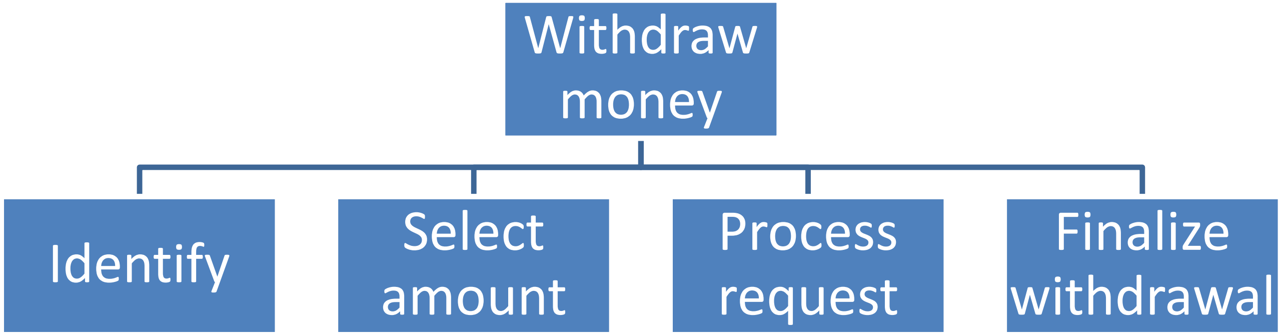
Enter_Code - - - - just after - - - - *Insert_Card*

One example of ATM functioning



Please enter your code:

Exercise 1


Withdraw money
Identify
Select amount
Process request
Finalize withdrawal

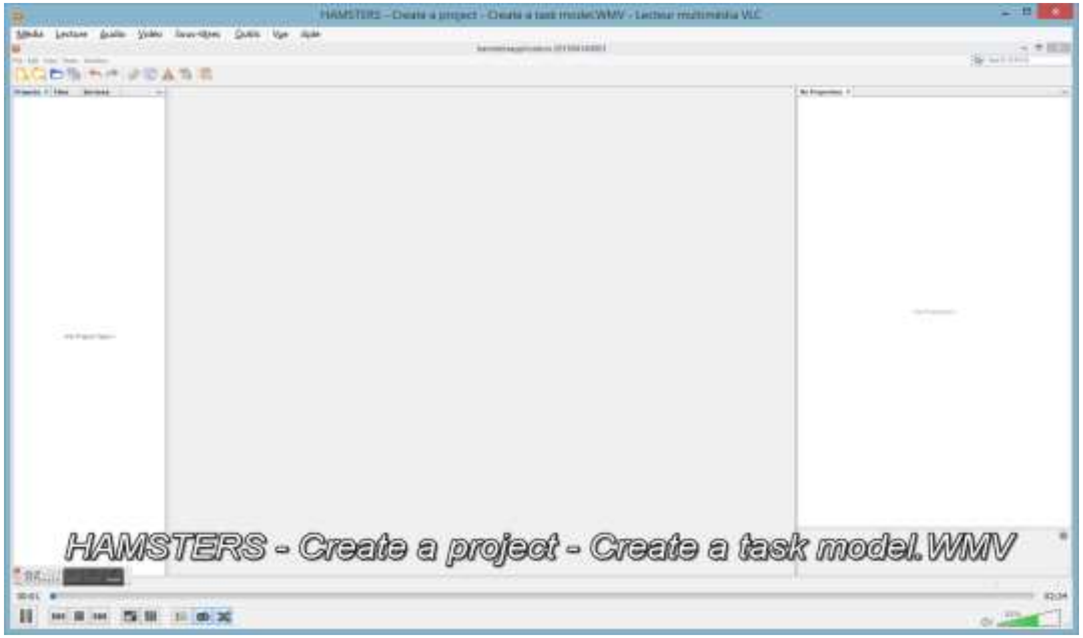
Building the task tree

- 1 - Build a list of tasks
- 2 - Gather high-level tasks
- 3 - Breakdown low-level tasks
- Where to stop ?
  - Is the action “identify” decomposed enough?
  - Objective: only decompose pertinent tasks (objective optimization/automation of work)
  - User actions: perceptive/cognitive/motor usually not that interesting to decompose (except for performance evaluation or for interaction design)
  - System activity MUST not be represented (will be in other models) but connections to it yes
- Highest level = reproduction of the species ;-)
- Lowest level = biological life (breathe, drink, contract muscle …)

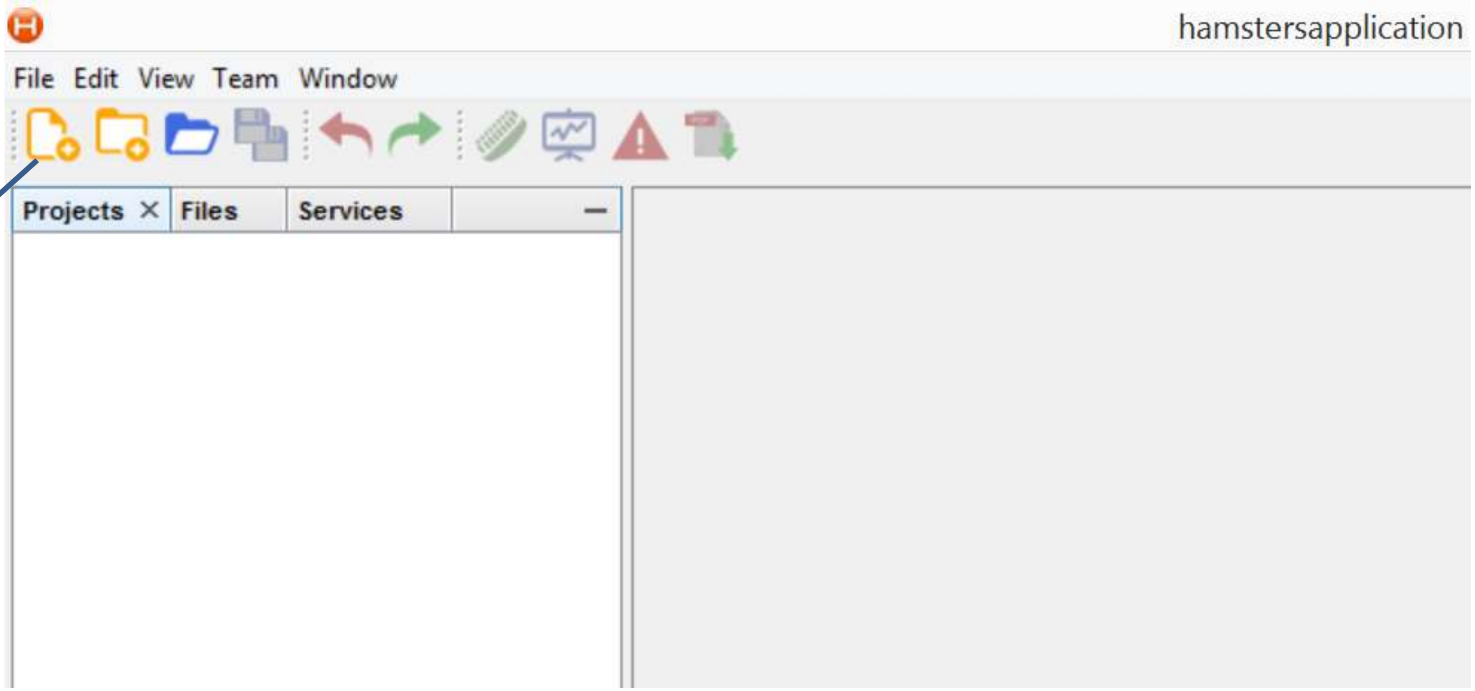
Installation Game (end of the game)

- Identify your hardware/software platform
- Downloading tool / getting a USB Key
- Installing HAMSTERS
- Create your first task model

Video about HAMSTERS installation


HAMSTERS – Create a project - Create a task model.WMV - Lecteur multimédia VLC
Média
Lecture
Audio
Vidéo
Sous-titres
Outils
Vue
Aide
HAMSTERS - Create a project - Create a task model.WMV

Create a task model

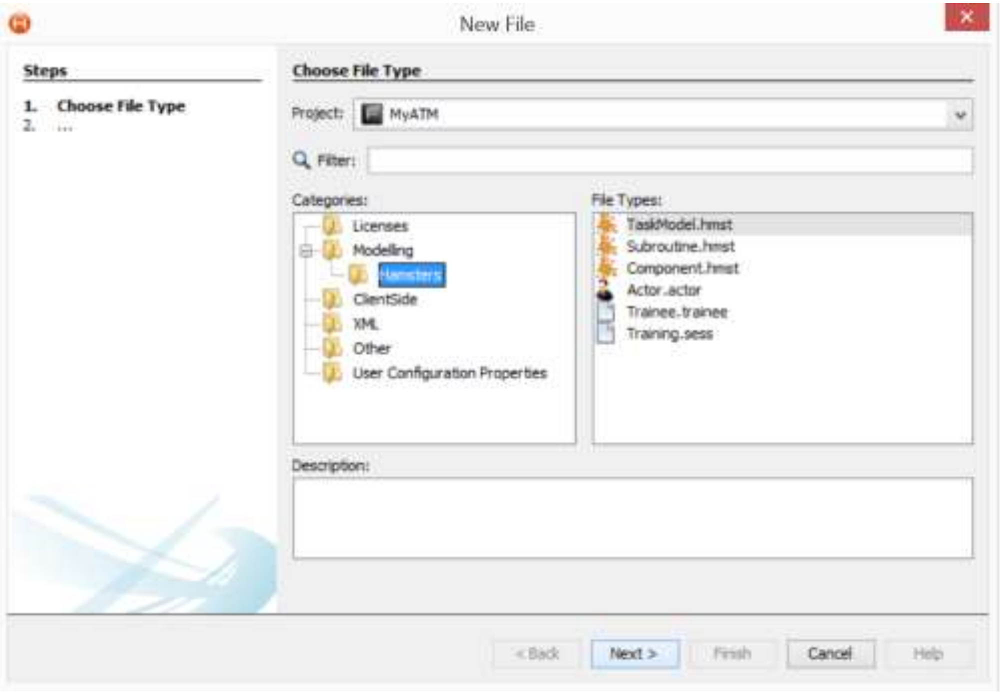


Icon New File

Create a task model

1. In the panel « Categories », select Hamsters

2. In the panel « File Types », select « TaskModel.hmst »



New File

Steps

1. Choose File Type
2. ...

Choose File Type

Project: MyATM

Filter:

Categories:

- Licenses
- Modelling
  - Hamsters
- ClientSide
- XML
- Other
- User Configuration Properties

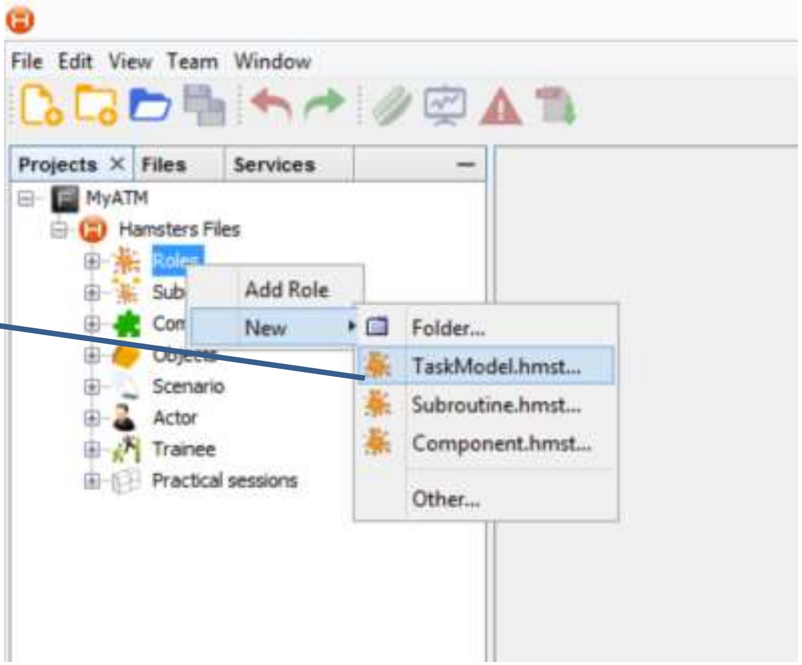
File Types:

- TaskModel.hmst
- Subroutine.hmst
- Component.hmst
- Actor.actor
- Trainee.trainee
- Training.sess

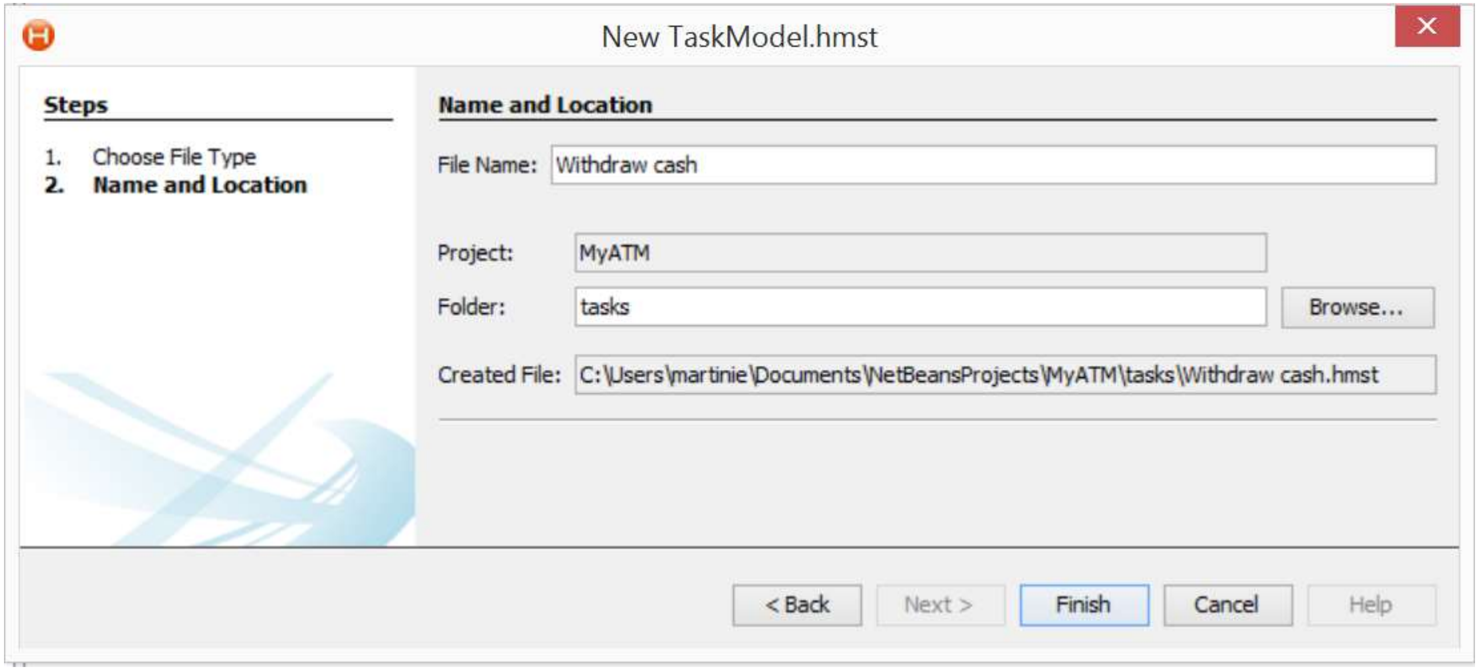
Description:

Create a task model

Select TaskModel.hmst



Enter the name of the task model



Steps

1. Choose File Type
2. **Name and Location**

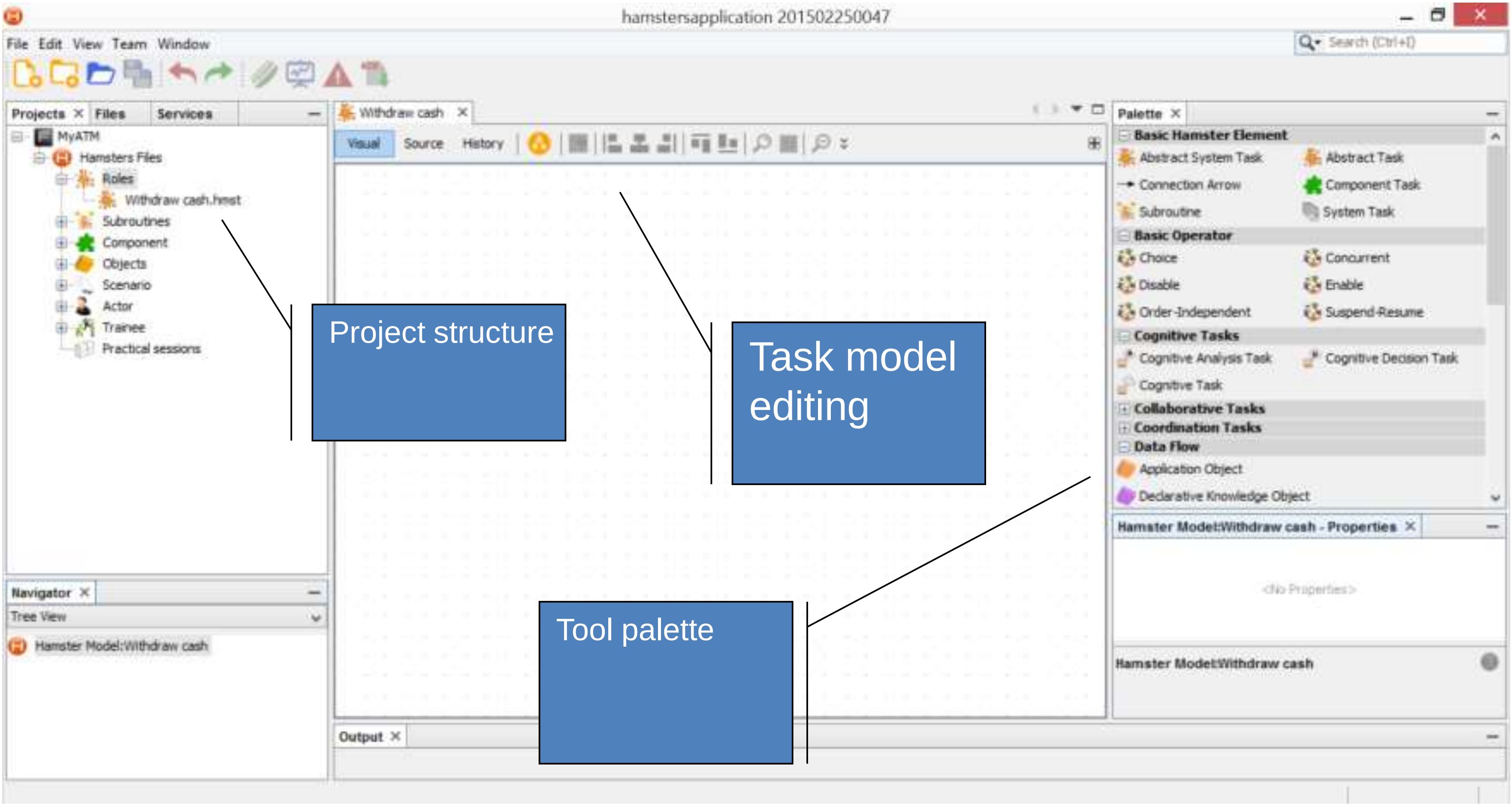
Name and Location

File Name: Withdraw cash

Project: MyATM

Folder: tasks

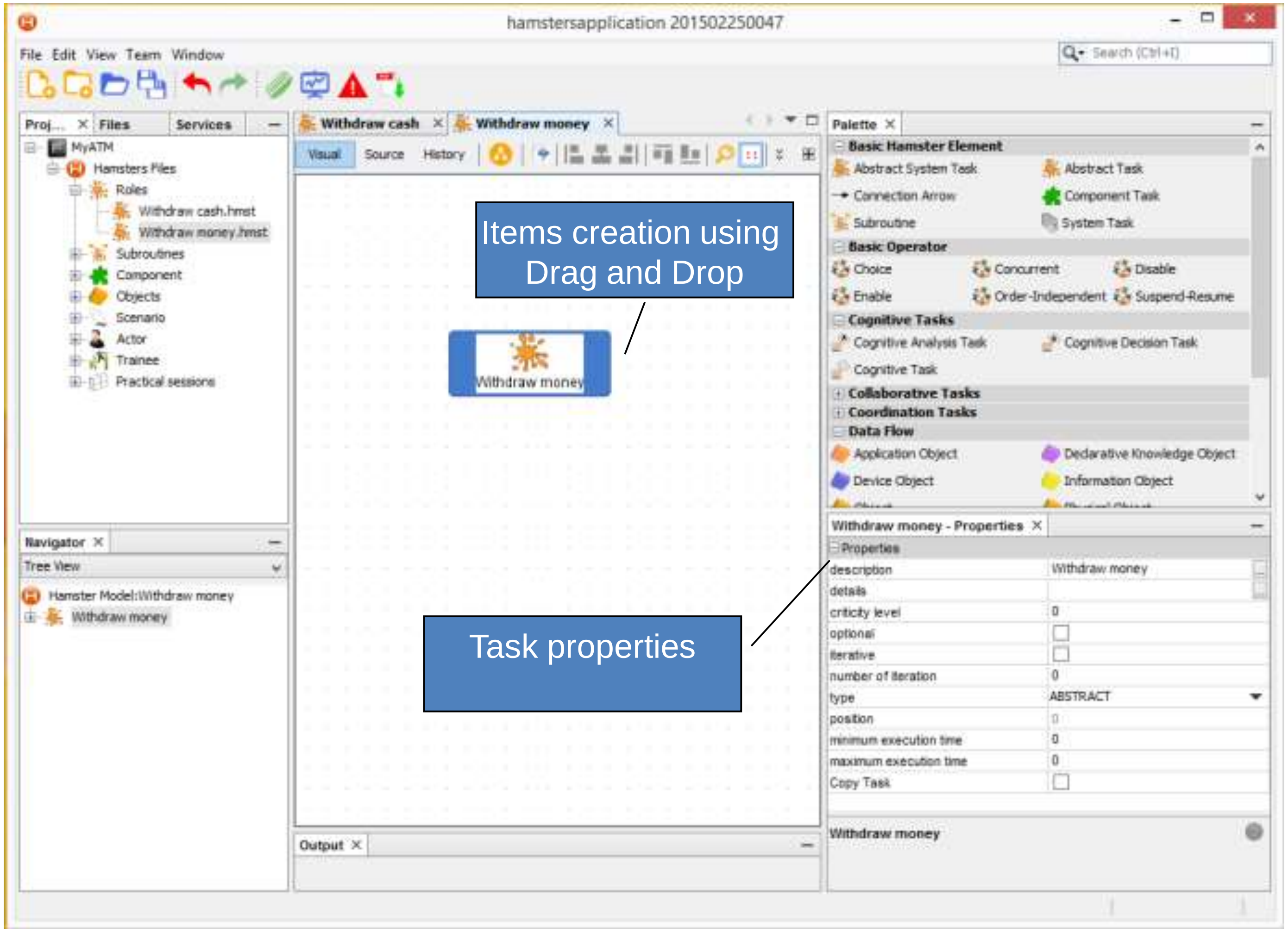
Created File: C:\Users\martinie\Documents\NetBeansProjects\MyATM\tasks\Withdraw cash.hmst



Project structure

Task model editing

Tool palette

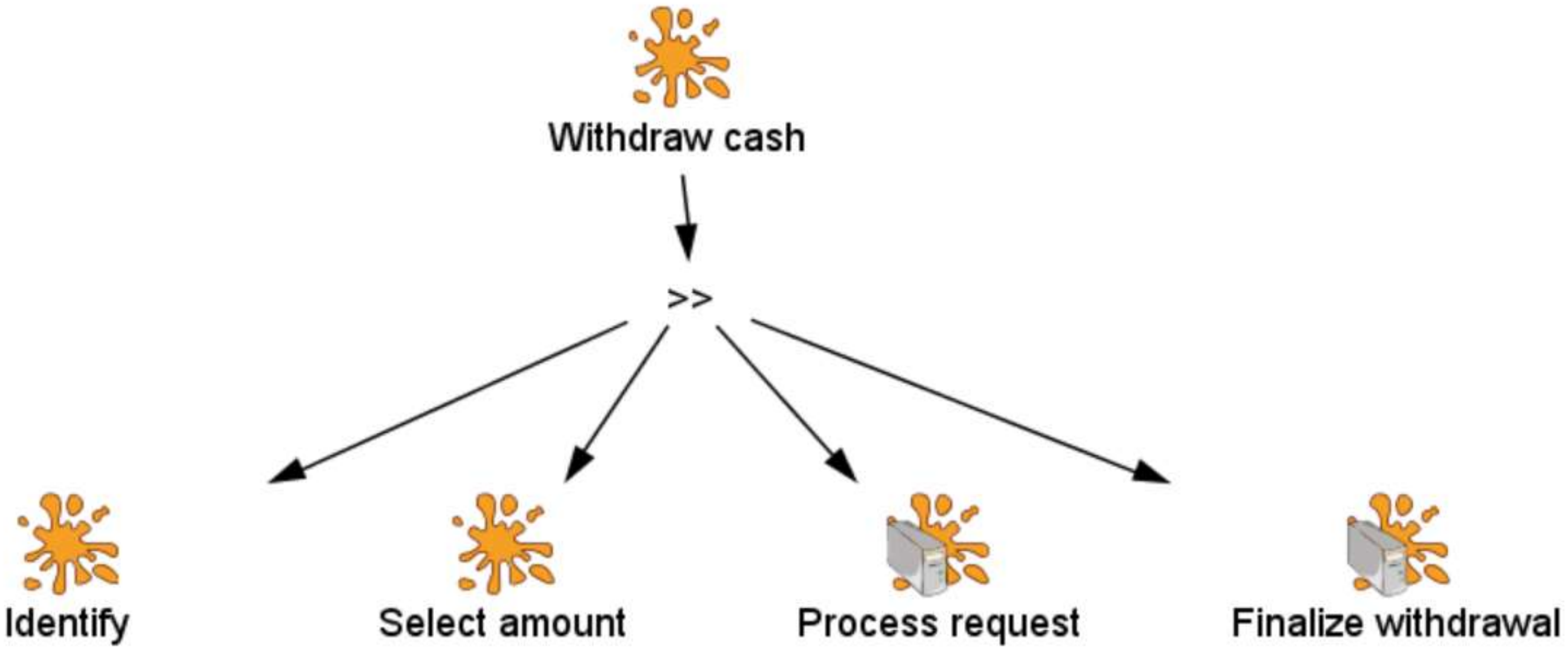


description	Withdraw money
details	
criticity level	0
optional	□
iterative	□
number of iteration	0
type	ABSTRACT
position	0
minimum execution time	0
maximum execution time	0
Copy Task	□

Items creation using Drag and Drop

Task properties

Exercise 1 – you edit this small model



Temporal ordering

- Temporal operators to describe temporal relationships between tasks
- Additional properties for the tasks:
  - Optional (otherwise means mandatory)
  - Iterative (otherwise means executed once)

Wilson S., Johnson P., Kelly C., Cunningham J. and Markopoulos P. **Beyond hacking: A model based approach to user interface design**, Proceedings of HCI'93, 217—23, Cambridge University Press, BCS HCI.

Temporal Operators

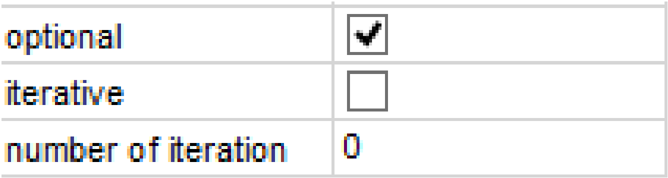
• Enabling	T1 >> T2
• Disabling	T1 [> T2
• Interruption	T1 |> T2
• Choice	T1 [ ] T2
• Concurrency	T1 ||| T2
• Order independant	T1 |=| T2

Wilson S., Johnson P., Kelly C., Cunningham J. and Markopoulos P. **Beyond hacking: A model based approach to user interface design**, Proceedings of HCI'93, 217—23, Cambridge University Press, BCS HCI.

Fabio Paternò, Cristiano Mancini, Silvia Meniconi: **ConcurTaskTrees (CTT): A Diagrammatic Notation for Specifying Task Models**. INTERACT 1997: 362-369

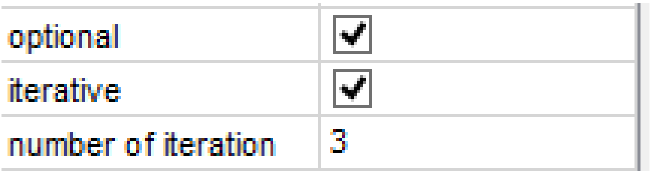
Temporal properties of the task

- Optional



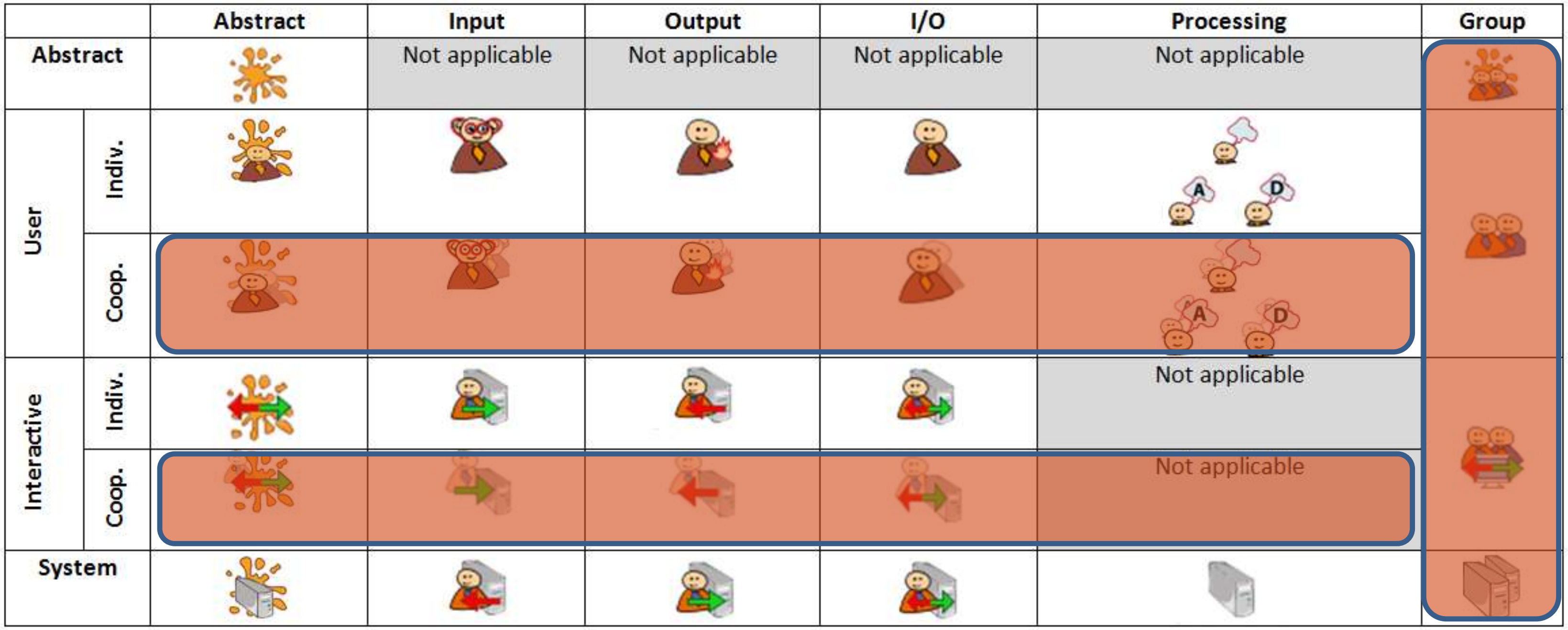
optional	☑
iterative	□
number of iteration	0

- Iterative



optional	☑
iterative	☑
number of iteration	3

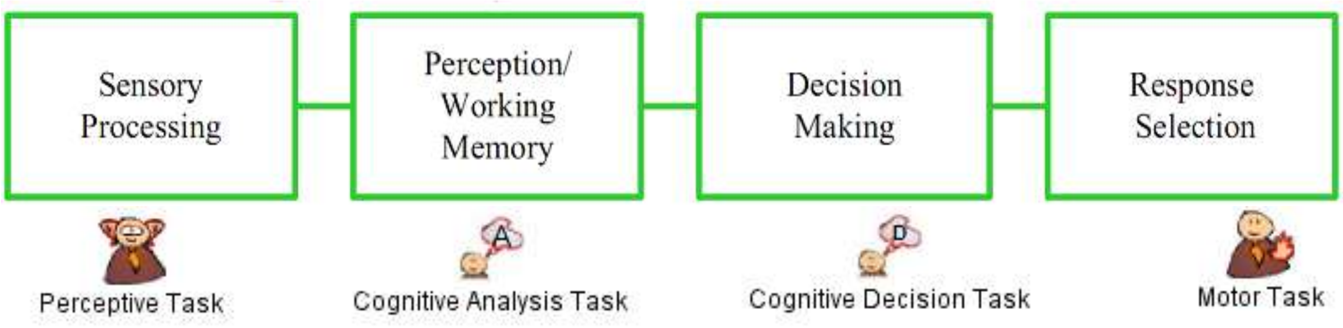
Task types



		Abstract	Input	Output	I/O	Processing	Group
Abstract			Not applicable	Not applicable	Not applicable	Not applicable	
User	Indiv.						
	Coop.						
Interactive	Indiv.					Not applicable	
	Coop.					Not applicable	
System							

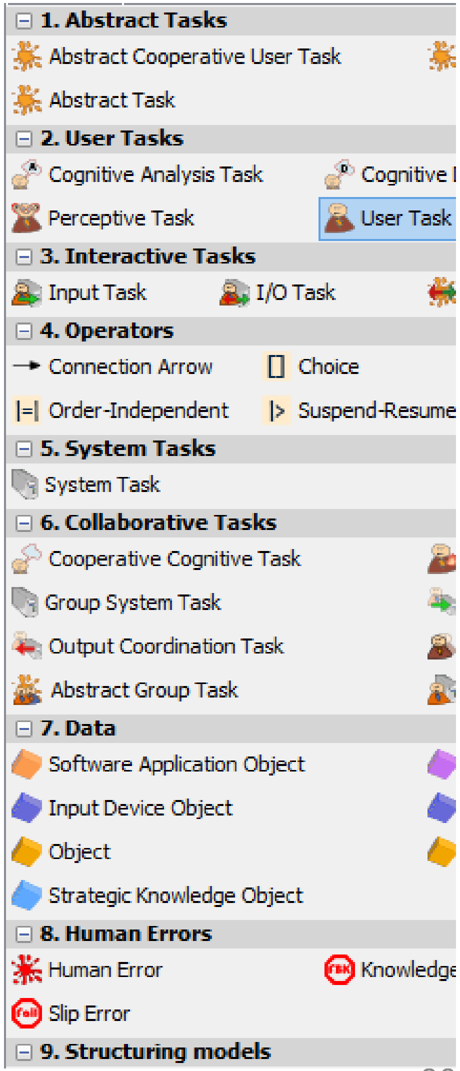
Human/operator behavioral tasks

- Precise and detailed description of human activities
- Perceptive, motor, cognitive (Card, Moran, Newell 1983)
- Cognitive analysis, cognitive decision (Parasuraman 2000)



Tool palette in HAMSTERS (1/2)

- Abstract tasks
- User Tasks
- Interactive Tasks
- Operators
- System Tasks
- Collaborative Tasks
- Data
- Human Error
- Structuring Models



⊟ **1. Abstract Tasks**

Abstract Cooperative User Task

Abstract Task

⊟ **2. User Tasks**

Cognitive Analysis Task   Cognitive I

Perceptive Task   User Task

⊟ **3. Interactive Tasks**

Input Task   I/O Task

⊟ **4. Operators**

Connection Arrow   Choice

Order-Independent   Suspend-Resume

⊟ **5. System Tasks**

System Task

⊟ **6. Collaborative Tasks**

Cooperative Cognitive Task

Group System Task

Output Coordination Task

Abstract Group Task

⊟ **7. Data**

Software Application Object

Input Device Object

Object

Strategic Knowledge Object

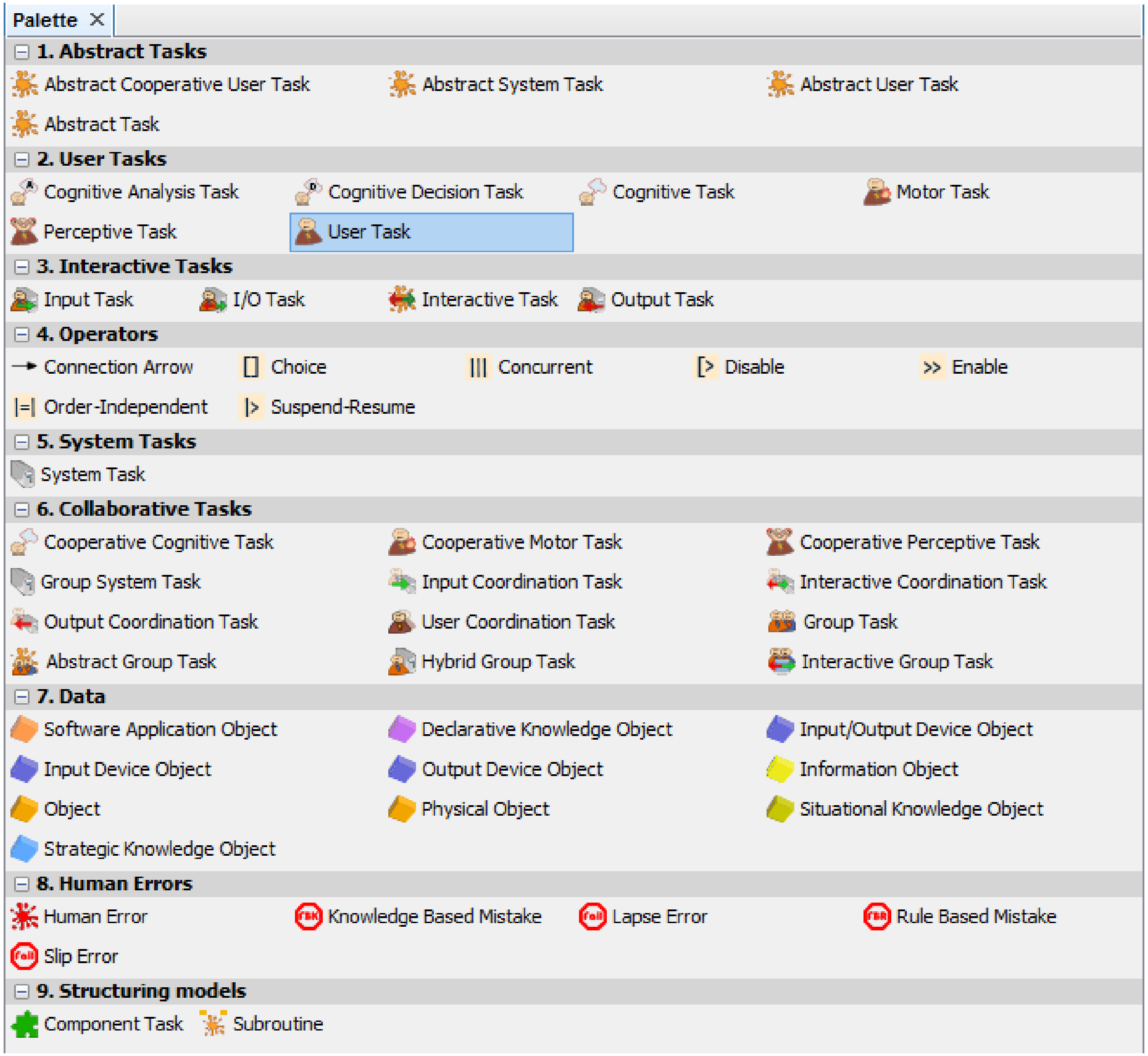
⊟ **8. Human Errors**

Human Error   Knowledge

Slip Error

⊟ **9. Structuring models**

Tool palette in HAMSTERS (1/2)



Palette

1. Abstract Tasks

- Abstract Cooperative User Task
- Abstract System Task
- Abstract User Task
- Abstract Task

2. User Tasks

- Cognitive Analysis Task
- Cognitive Decision Task
- Cognitive Task
- Motor Task
- Perceptive Task
- User Task

3. Interactive Tasks

- Input Task
- I/O Task
- Interactive Task
- Output Task

4. Operators

- → Connection Arrow
- [] Choice
- ||| Concurrent
- [> Disable
- >> Enable
- |=| Order-Independent
- |> Suspend-Resume

5. System Tasks

- System Task

6. Collaborative Tasks

- Cooperative Cognitive Task
- Cooperative Motor Task
- Cooperative Perceptive Task
- Group System Task
- Input Coordination Task
- Interactive Coordination Task
- Output Coordination Task
- User Coordination Task
- Group Task
- Abstract Group Task
- Hybrid Group Task
- Interactive Group Task

7. Data

- Software Application Object
- Declarative Knowledge Object
- Input/Output Device Object
- Input Device Object
- Output Device Object
- Information Object
- Object
- Physical Object
- Situational Knowledge Object
- Strategic Knowledge Object

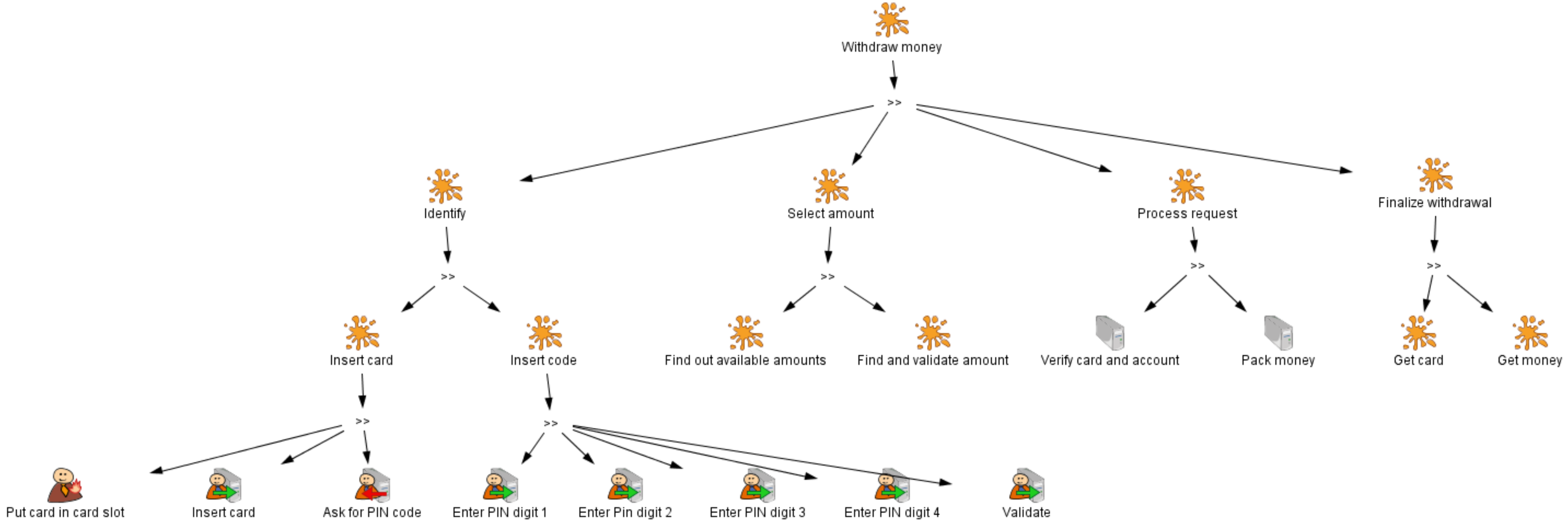
8. Human Errors

- Human Error
- Knowledge Based Mistake
- Lapse Error
- Rule Based Mistake
- Slip Error

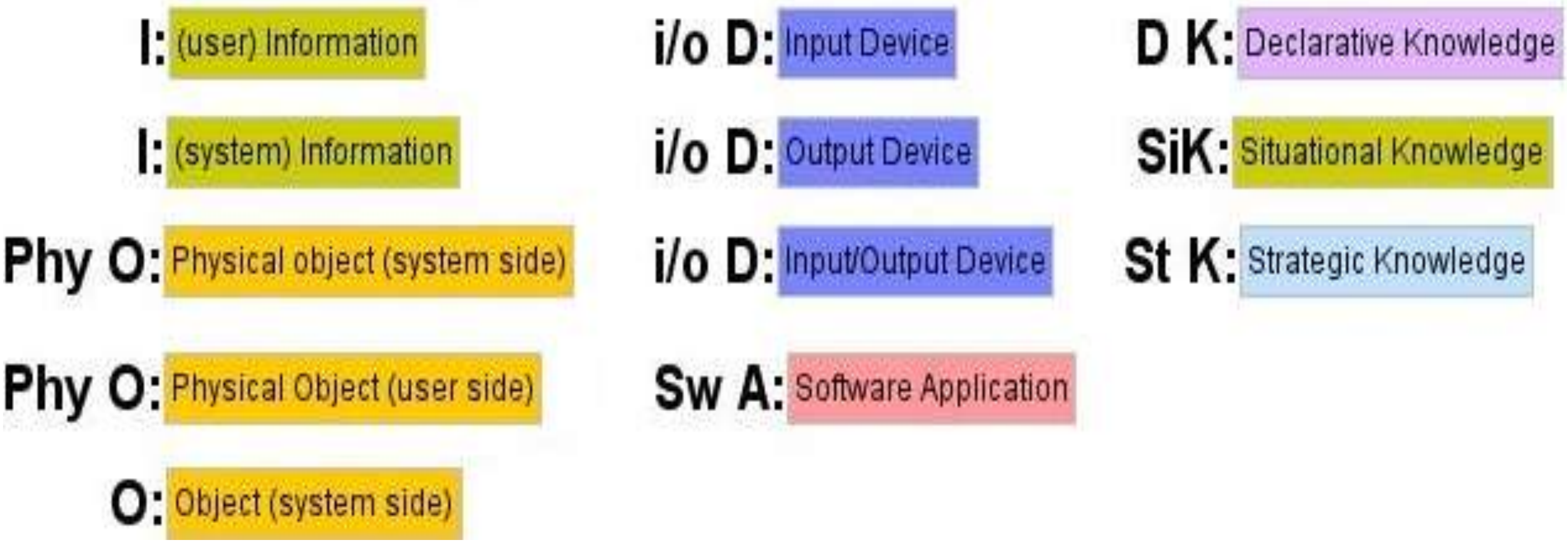
9. Structuring models

- Component Task
- Subroutine

Exercise 2: Refine the task model



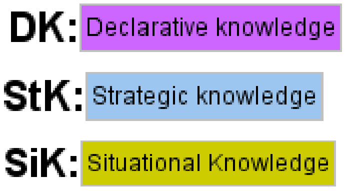
Objects, Information, Knowledge, Devices and Software applications



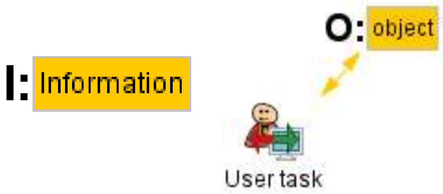
I:	(user) Information	i/o D:	Input Device	D K:	Declarative Knowledge
I:	(system) Information	i/o D:	Output Device	SiK:	Situational Knowledge
Phy O:	Physical object (system side)	i/o D:	Input/Output Device	St K:	Strategic Knowledge
Phy O:	Physical Object (user side)	Sw A:	Software Application		
O:	Object (system side)				

Data flow

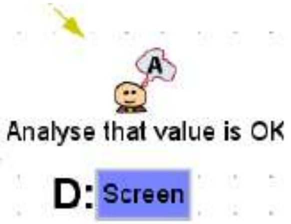
- Precise and detailed description of users’ knowledge, manipulated objects and devices used
- Read (data used for performing the task)
- Write (data produced for performing the task)
- Read/write (data modified while performing the task)



Knowledge connected to the tasks



Object modified while performing a task

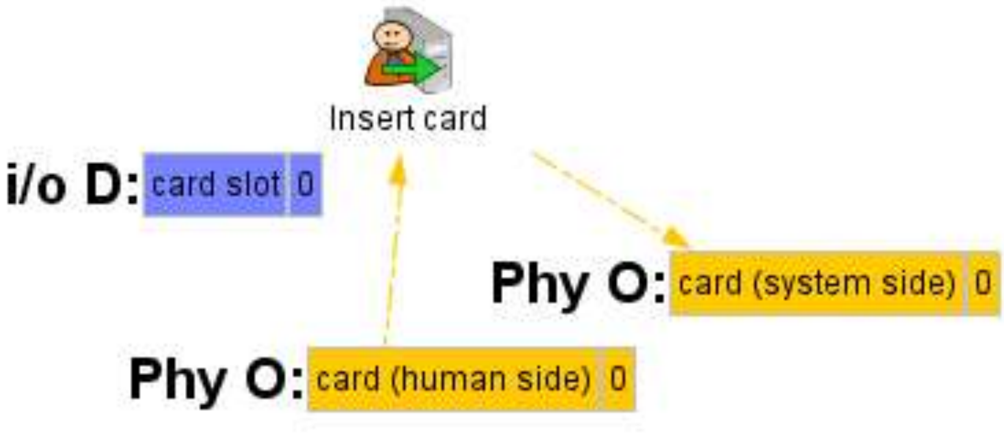


Device used for performing a task

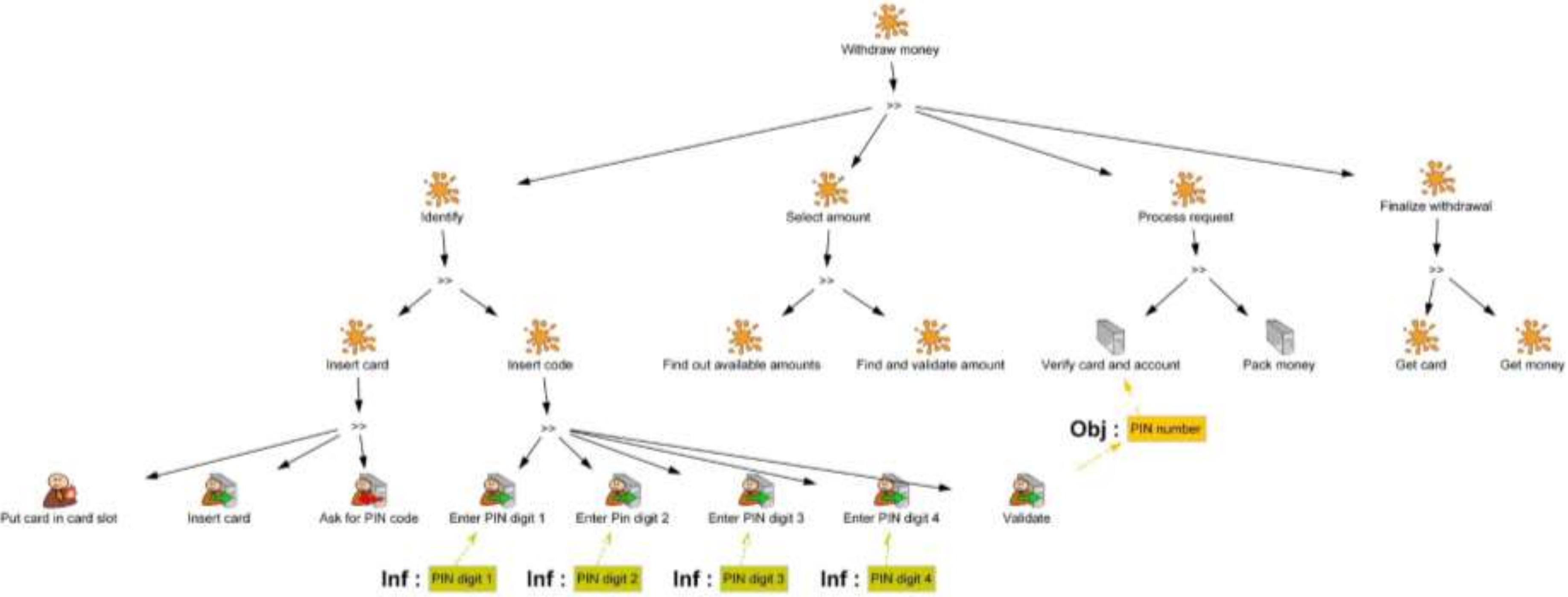
Description of objects manipulated by the user and by the system

- Input and output devices

- Physical objects



Exercise 3: Add data flows in model

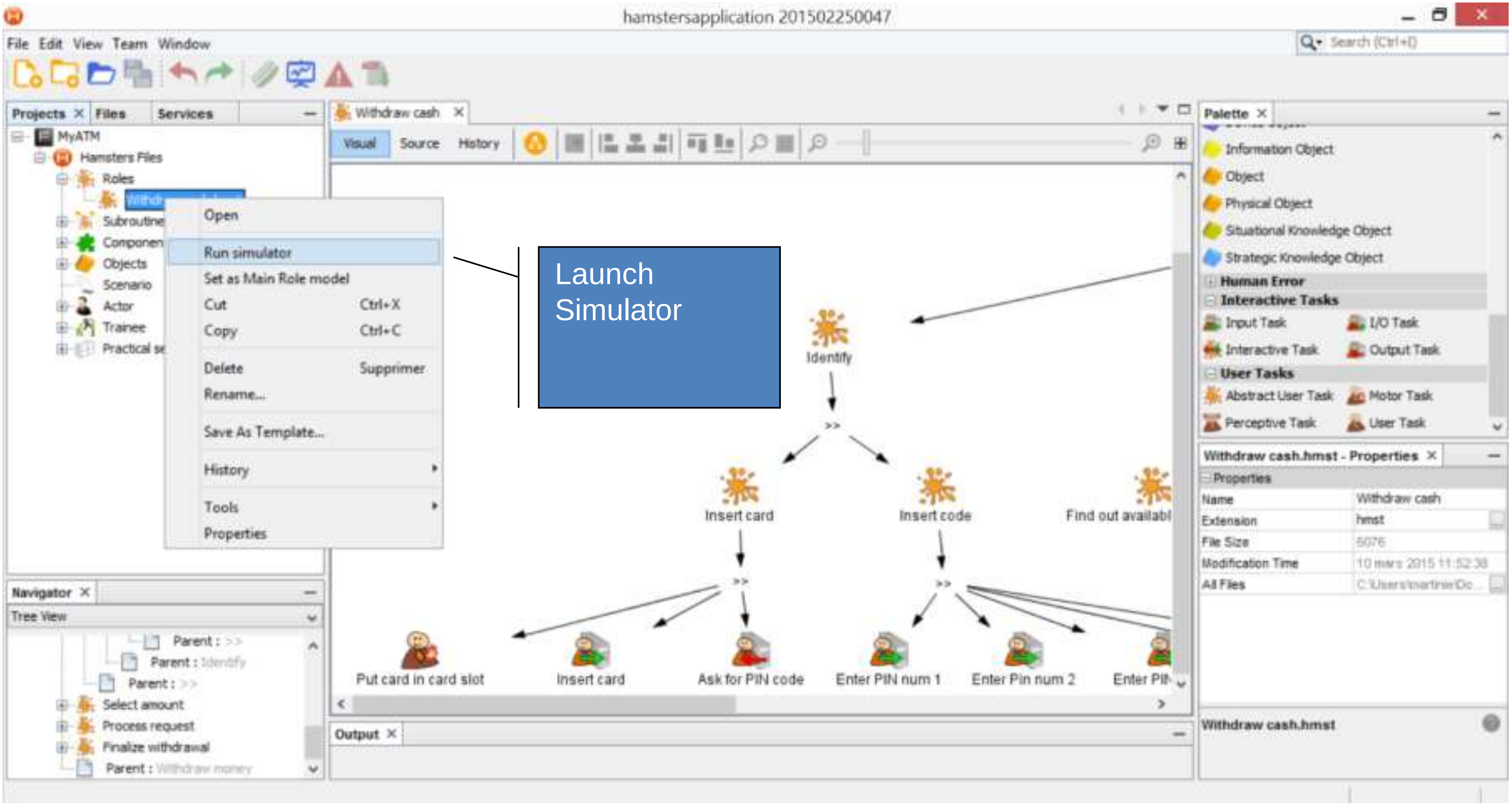


Exercise 3: Add data flows in model

- Add device(s)
- Add credit card
- Add information in the head of the user
- Add objects in the system side

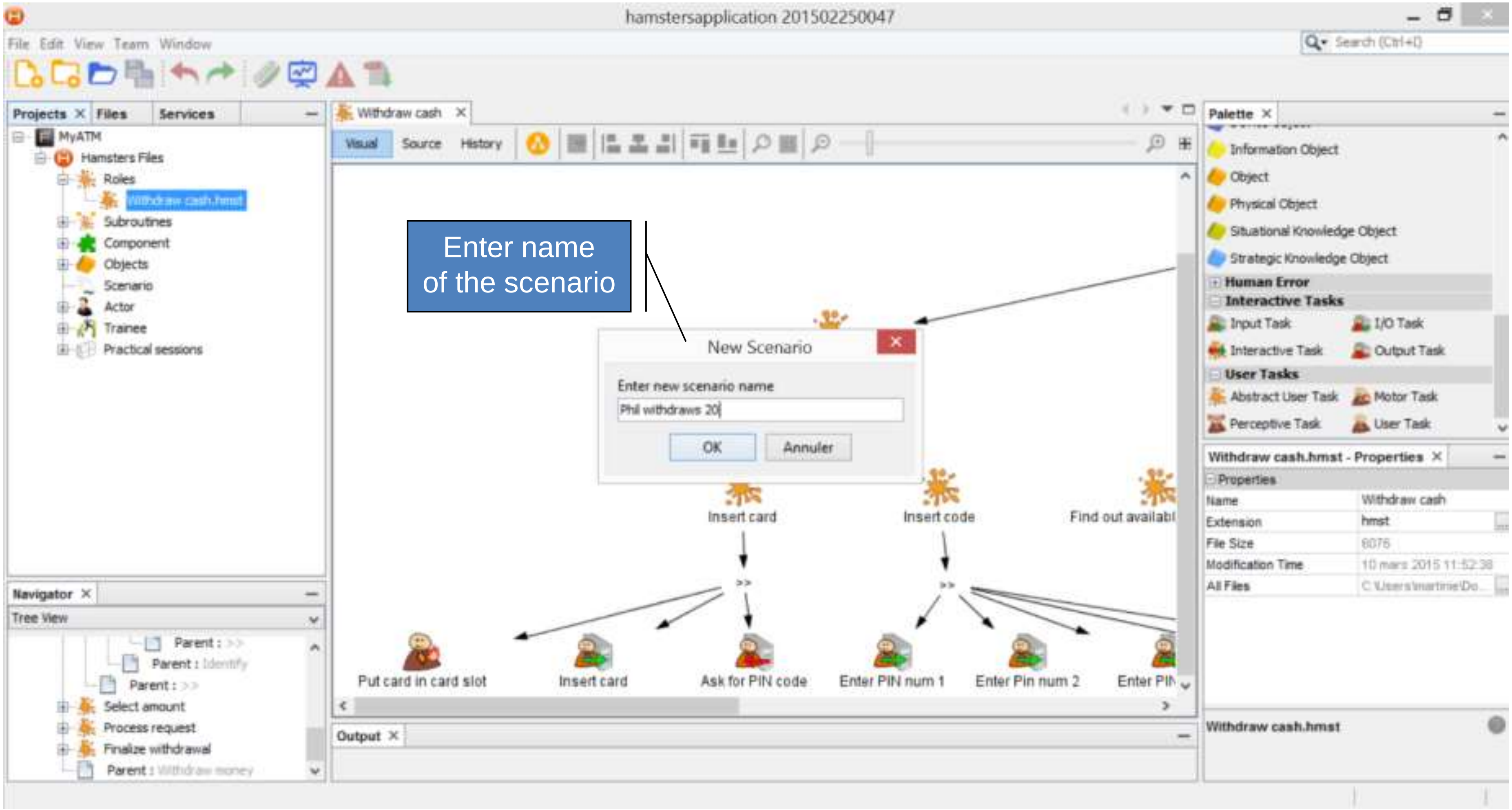
Validation of the model

- Simulation of the model
- Scenarios
  - Identification of scenarios
  - Production of scenarios
  - Storage of scenarios



Launch Simulator

Enter name of the scenario



New Scenario

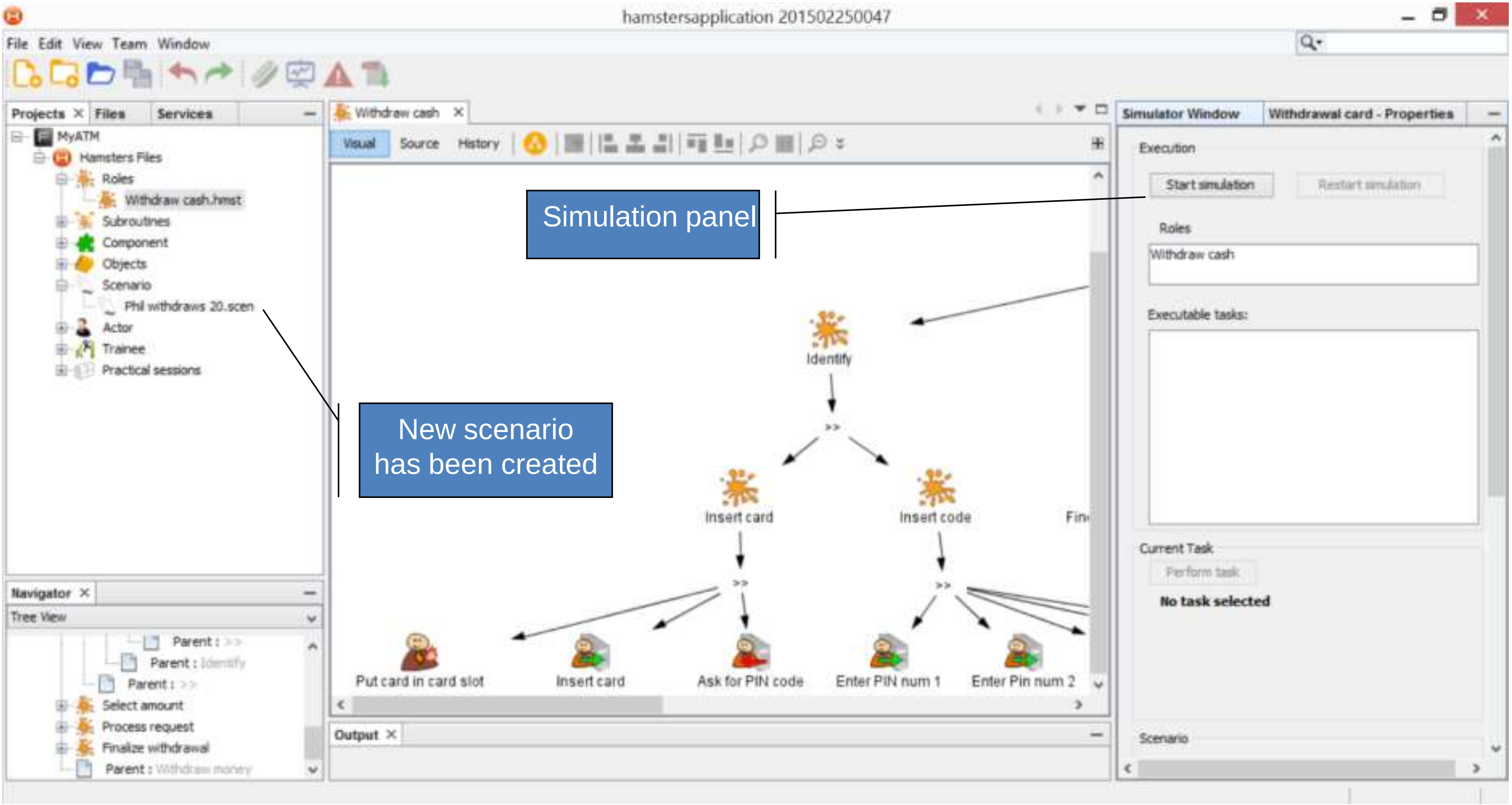
Enter new scenario name

Phil withdraws 20

OK Annuler

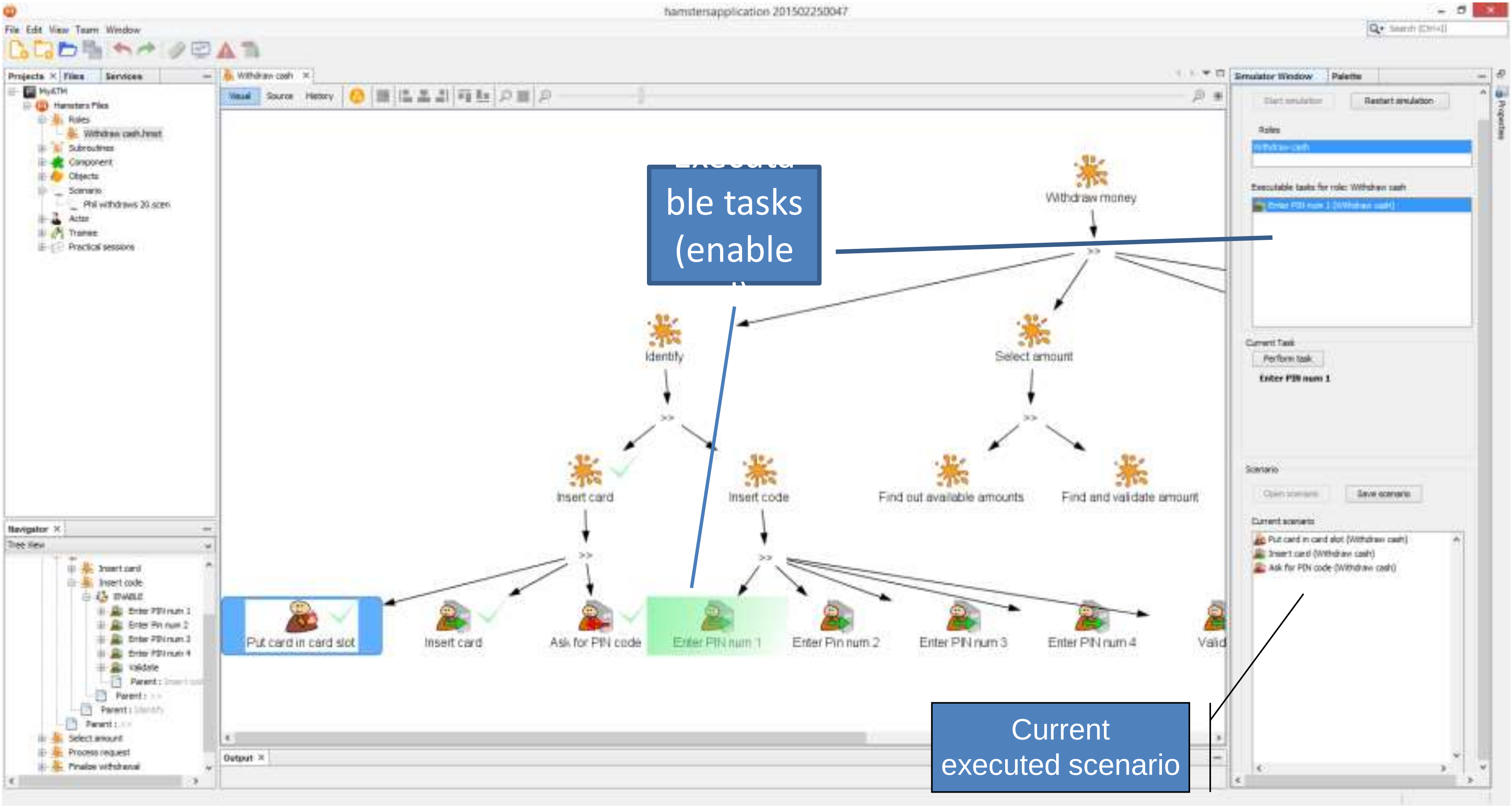
Withdraw cash.hmst - Properties

Properties	
Name	Withdraw cash
Extension	hmst
File Size	6076
Modification Time	10 mars 2015 11:52:38
All Files	C:\Users\martinie\Do...



Simulation panel

New scenario has been created



Roles

Withdraw cash

Executable tasks for role: Withdraw cash

Enter PIN num 1 (Withdraw cash)

Current Task

Perform task

Enter PIN num 1

Scenario

Open scenario

Save scenario

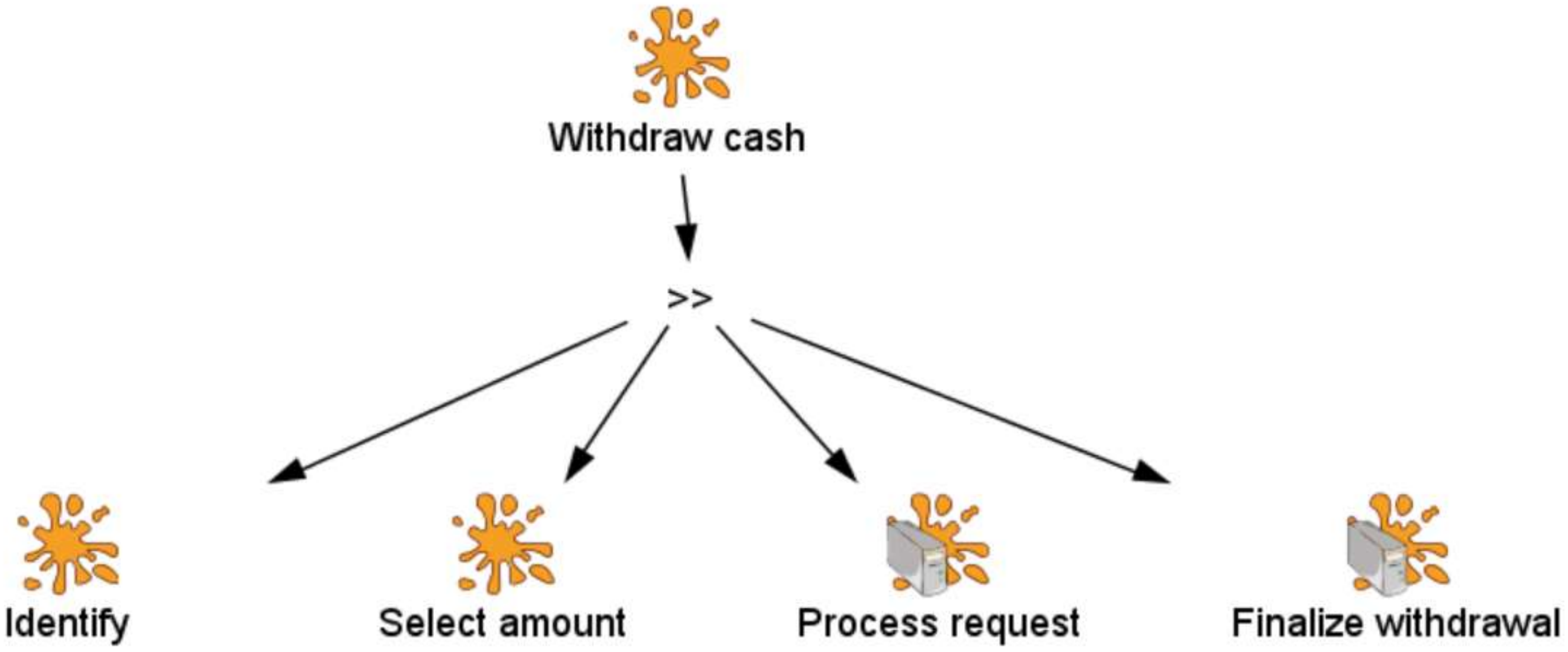
Current scenario

- Put card in card slot (Withdraw cash)
- Insert card (Withdraw cash)
- Ask for PIN code (Withdraw cash)

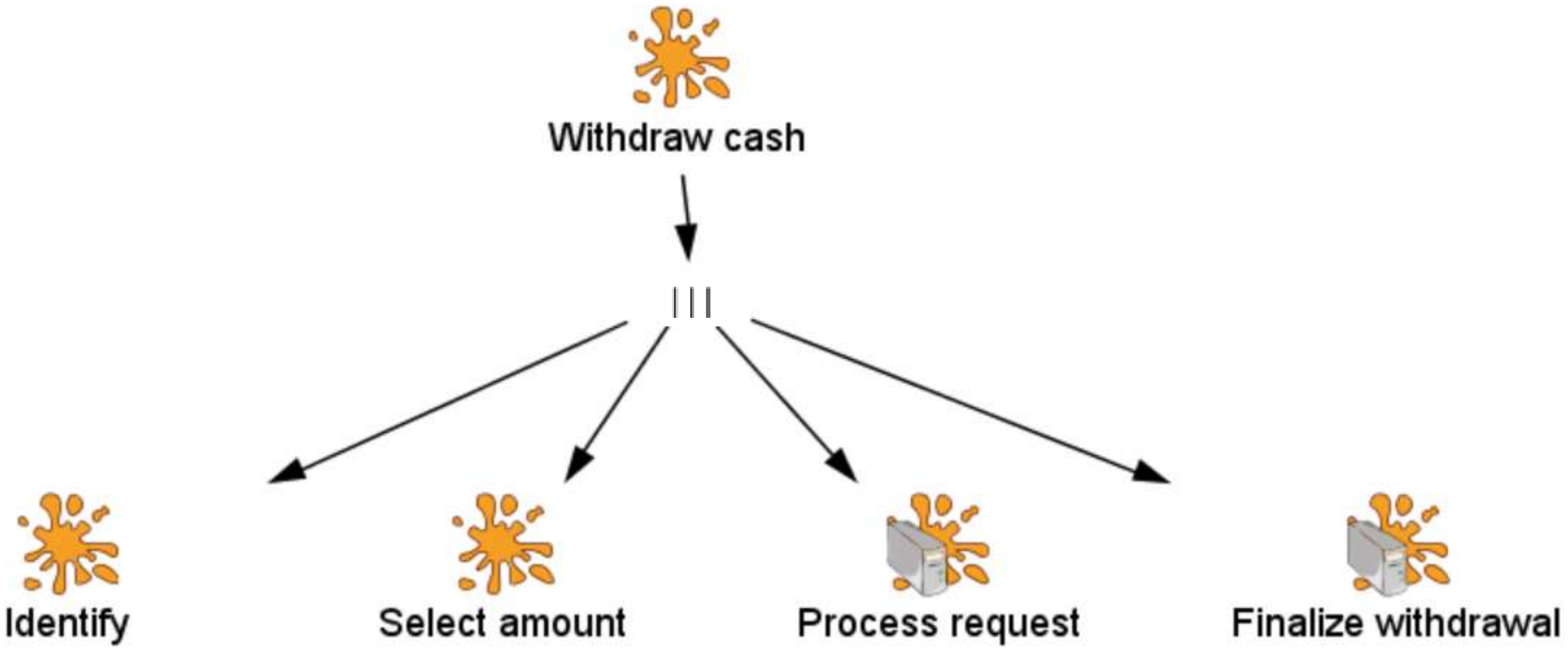
Executable tasks (enabled)

Current executed scenario

Modifying models / sequence



Modifying models / interleaving



Refining Models

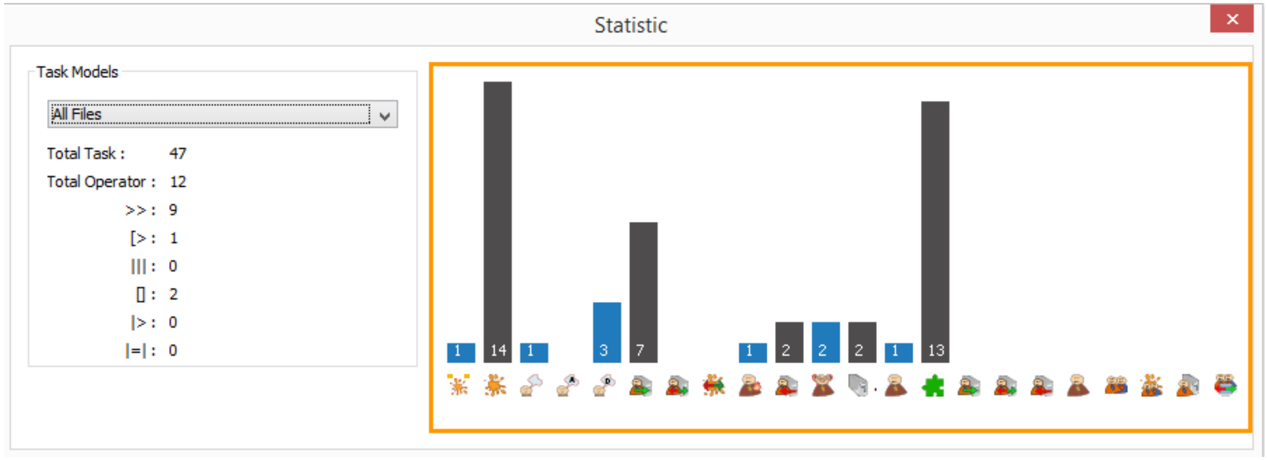
- Once the model is built (whatever notation you use)
  - How to be sure that it is correct?
  - How to improve it?
- Some heuristics
  - Coupled actions: e.g., where is “Put card in card slot"?
  - Restructure : e.g. make a task generic “Enter PIN digit”
  - Balance : e.g. is “Select amount” simpler than “Identify”?
  - Generalize: e.g. always withdraw a particular amount… withdraw different amounts

Analysis of the completed models

- Qualitative
- Consistency of the users actions
- Detection of usability issues
- Quantitative
- Task complexity: How many actions have to be performed to reach the goal?
- Which are the different possible sequence of actions (scenarios) and what is the estimated min/max time for achieving them?
- What is the exhaustive list of required knowledge?
- Which information will have to be remembered and for how long?

Supported by HAMSTERS tool

- Distribution the total number of tasks across task types (Statistics panel)
- Estimated minimum and maximum time to perform a scenario (Statistics panel)
- Structural verification of the task model (Verification panel)



Task Models

All Files

Total Task : 47

Total Operator : 12

>> : 9

[> : 1

||| : 0

[] : 2

|> : 0

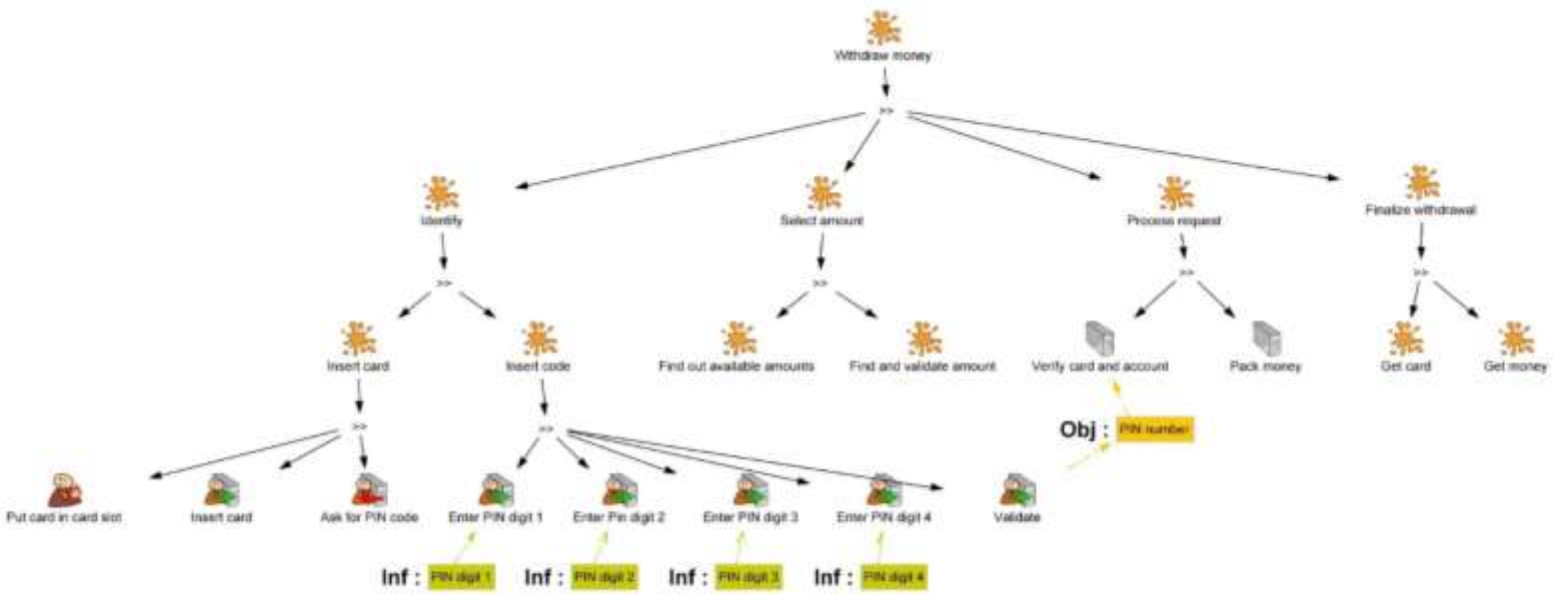
|=| : 0

"Designing" task models

- Dealing with alternative options
- Functioning versus usage
- Interviews/scenarios for informing task model design
- Types of task models
- Relationship with types of procedures

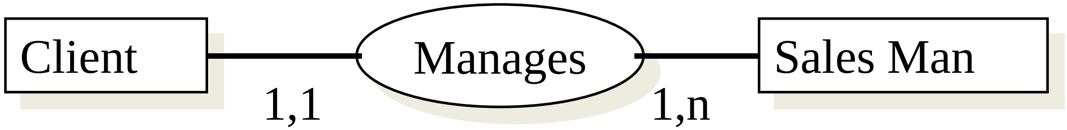
Dealing with alternative options

- Money withdrawal at the cashier



Structuring: Functioning versus Usage

- Extraterrestrial example (J. Grudin, A case against consistency, INTERACT 90)
- Example of a computer system
  - We have a database with the following schema:

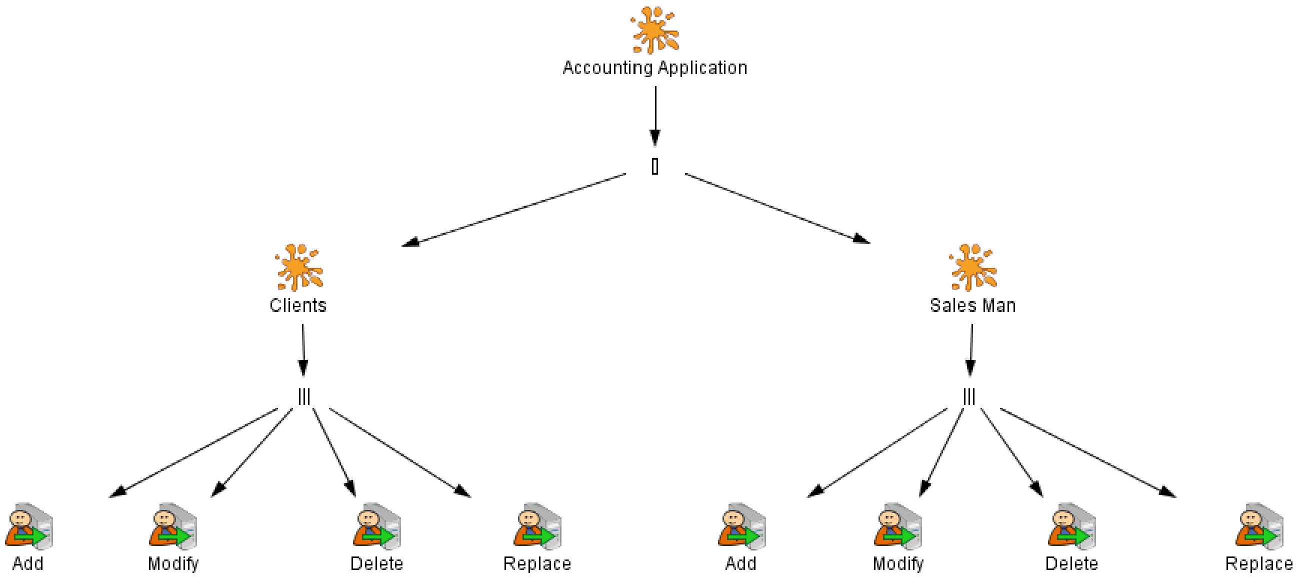


  - Propose a menu tree allowing to manipulate that database

Functioning structure goes to prototype directly

- View of the Computer Scientist
  - No tasks analysis/modeling is needed
  - The tree structure reflects the database structure
  - Grouping is made according to code similarity

Tasks model


Accounting Application
[]
Clients
Sales Man
|||
|||
Add
Modify
Delete
Replace
Add
Modify
Delete
Replace

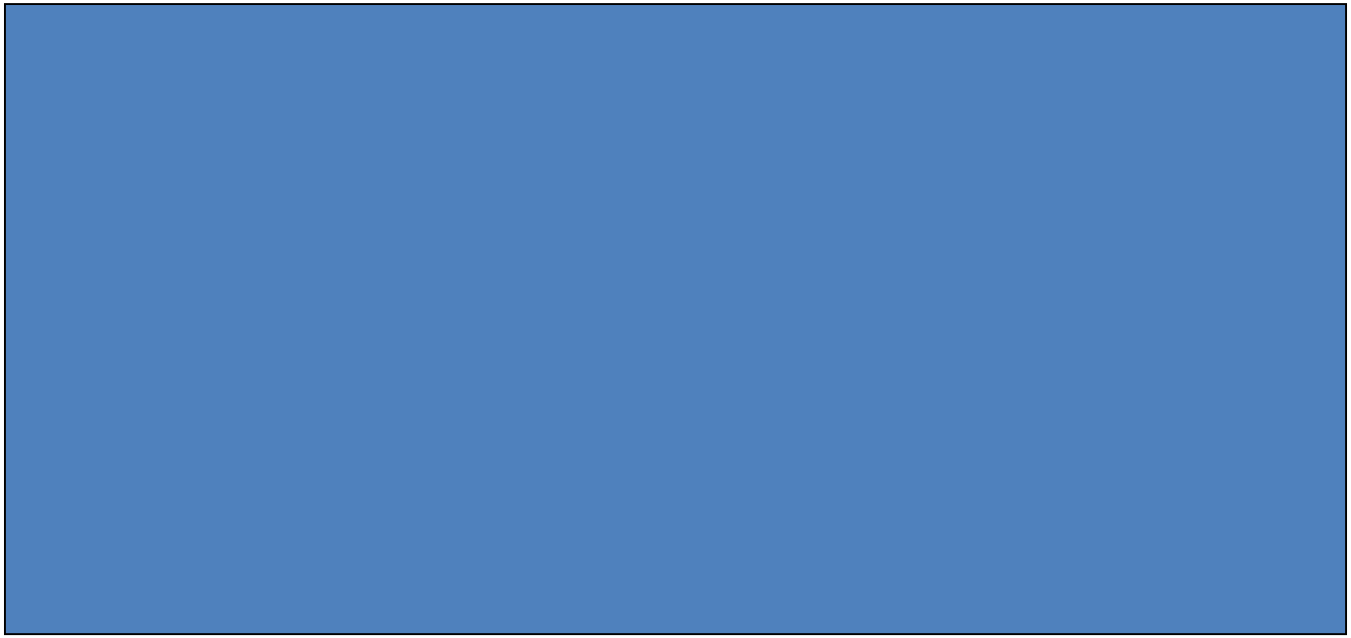
Example of Informal Task Description

- « I spent my day at work entering information for new clients and I must associate them with the more suitable sales man (according to various criteria) … sometimes I receive an invoice from a sales man recently recruited. In such a case, I must create both the new sales man and the client. Sometimes a sales man quits the company. Usually they leave the company for another one and the clients follow them. I then have to delete the sales man and all its related … »

- Exercise: Propose a tree structure adapted to the tasks described in this scenario

Usage structure (1/2)

- User interface is structured according to the task



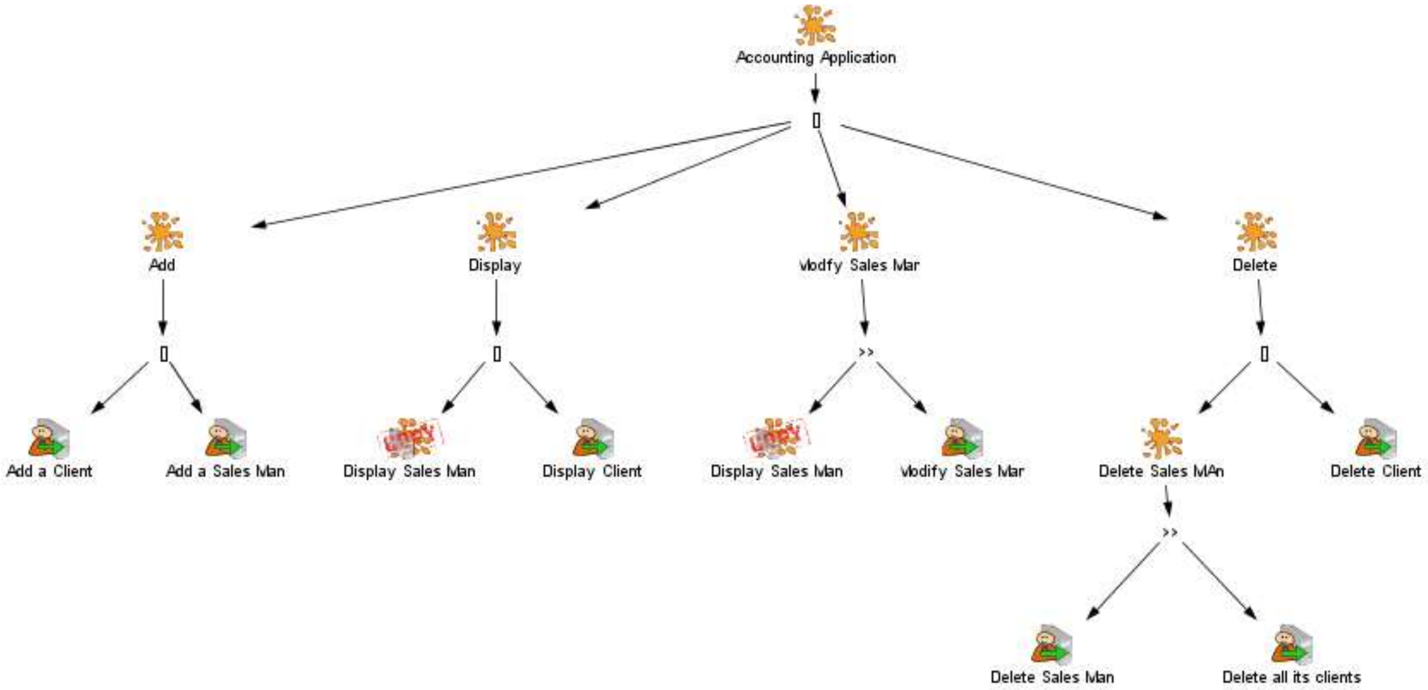
Things are still problematic

- There is no
  - Menu for deleting clients (without sales man)
  - Client modification
  - …
- There is
  - Access to clients through their sales man
  - Display of client information without getting to their sales man first
  - Duplication of menu items in the hierarchy
  - …

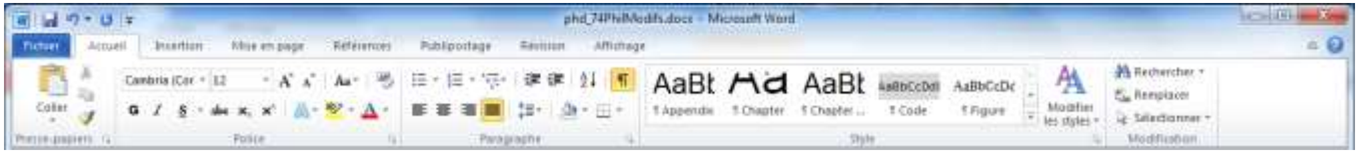
U=1/F

(W. Buxton BCS HCI 96)

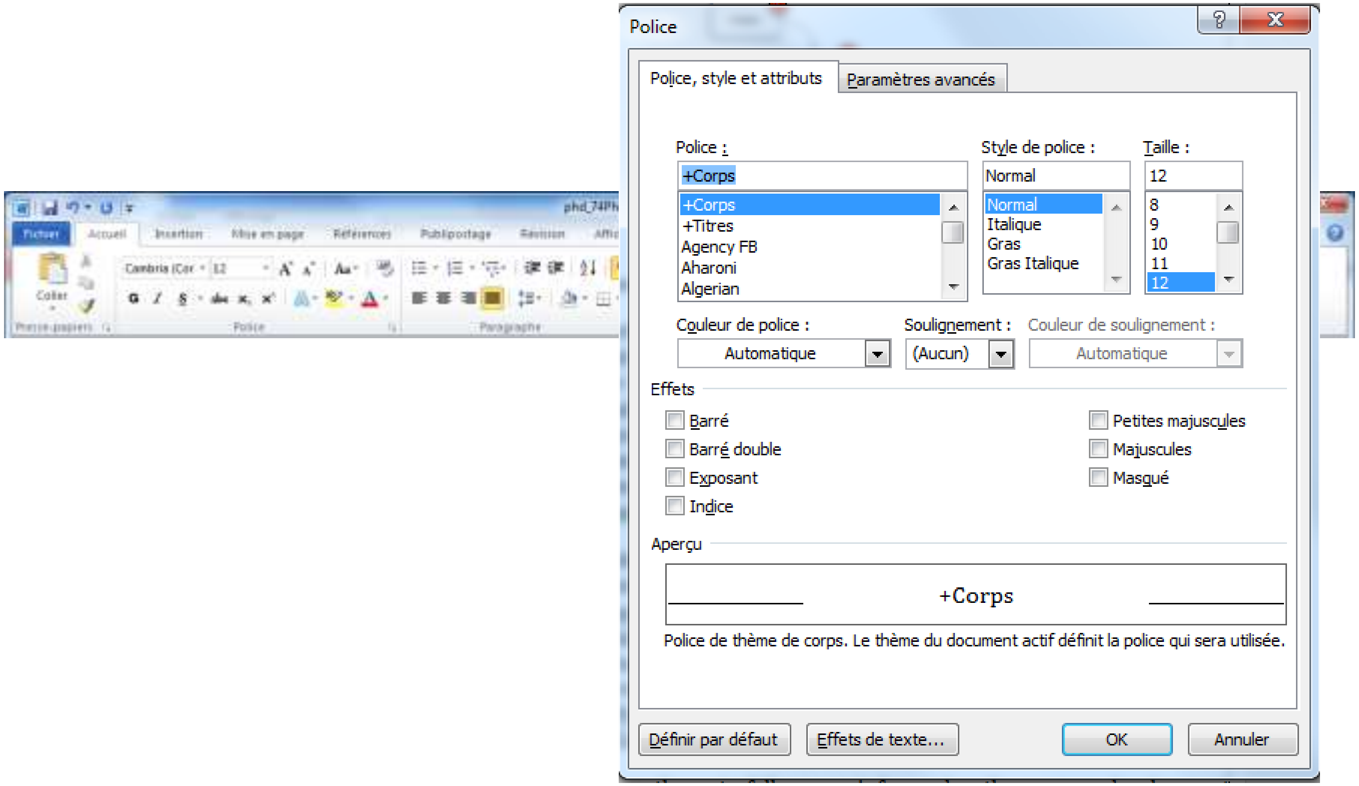
Task model in HAMSTERS



Examples of menu items duplication MS Word



Example MS Word



Police

Police, style et attributs | Paramètres avancés

Police : +Corps

- +Corps
- +Titres
- Agency FB
- Aharoni
- Algerian

Style de police : Normal

- Normal
- Italique
- Gras
- Gras Italique

Taille : 12

- 8
- 9
- 10
- 11
- 12

Couleur de police : Automatique

Soulignement : (Aucun)

Couleur de soulignement : Automatique

Effets

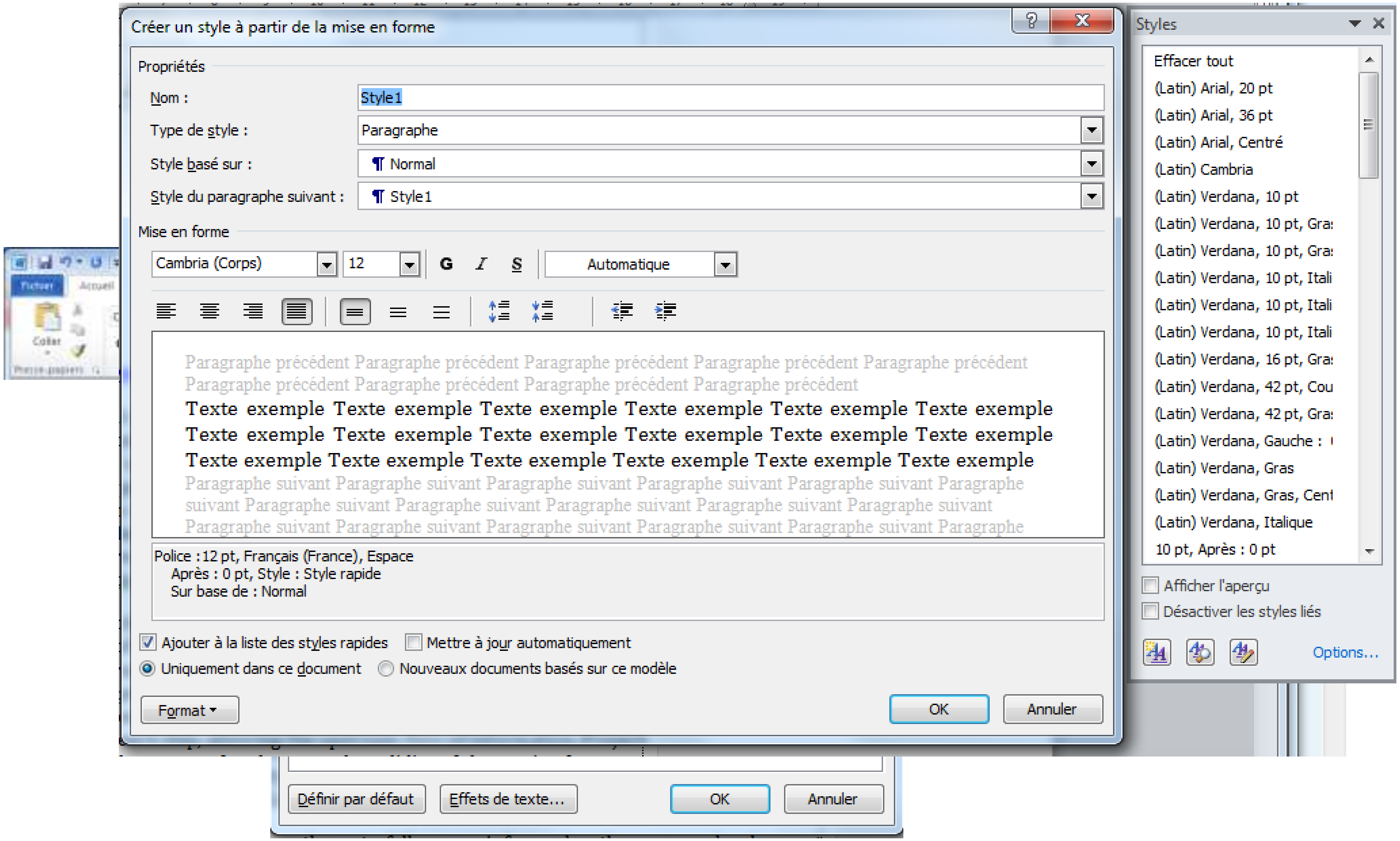
☐ Barré
☐ Barré double
☐ Exposant
☐ Indice
☐ Petites majuscules
☐ Majuscules
☐ Masqué

Aperçu

+Corps

Police de thème de corps. Le thème du document actif définit la police qui sera utilisée.

Example MS Word



Créer un style à partir de la mise en forme

Propriétés

Nom : Style1

Type de style : Paragraphe

Style basé sur : ¶ Normal

Style du paragraphe suivant : ¶ Style1

Mise en forme

Cambria (Corps) | 12 | Automatique

Paragraphe précédent Paragraphe précédent Paragraphe précédent Paragraphe précédent Paragraphe précédent Paragraphe précédent Paragraphe précédent Paragraphe précédent Paragraphe précédent

Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple Texte exemple

Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe suivant Paragraphe

Police :12 pt, Français (France), Espace
Après : 0 pt, Style : Style rapide
Sur base de : Normal

☑ Ajouter à la liste des styles rapides ☐ Mettre à jour automatiquement

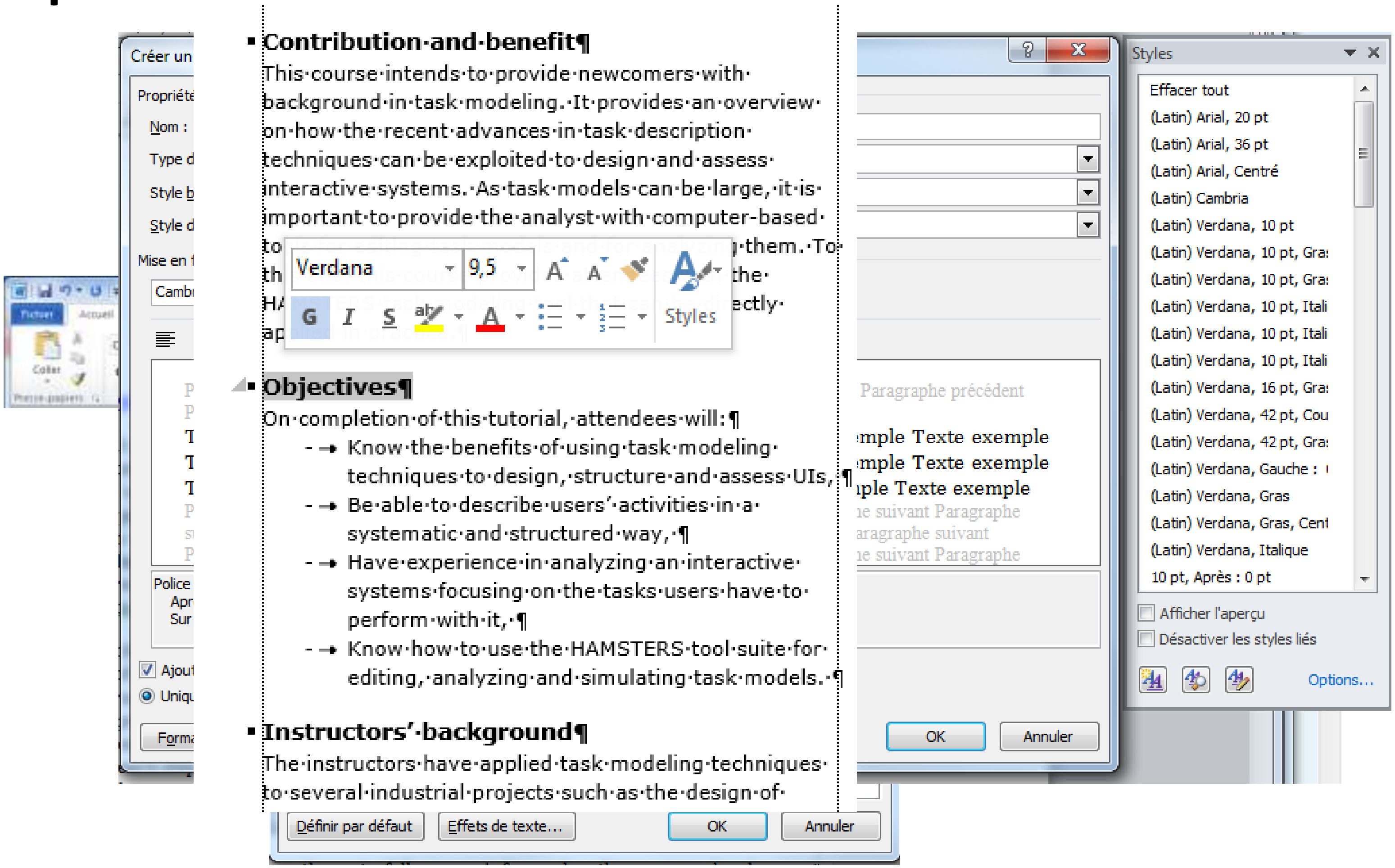
● Uniquement dans ce document ○ Nouveaux documents basés sur ce modèle

Styles

Effacer tout
(Latin) Arial, 20 pt
(Latin) Arial, 36 pt
(Latin) Arial, Centré
(Latin) Cambria
(Latin) Verdana, 10 pt
(Latin) Verdana, 10 pt, Gra
(Latin) Verdana, 10 pt, Gra
(Latin) Verdana, 10 pt, Itali
(Latin) Verdana, 10 pt, Itali
(Latin) Verdana, 10 pt, Itali
(Latin) Verdana, 16 pt, Gra
(Latin) Verdana, 42 pt, Cou
(Latin) Verdana, 42 pt, Gra
(Latin) Verdana, Gauche :
(Latin) Verdana, Gras
(Latin) Verdana, Gras, Cent
(Latin) Verdana, Italique
10 pt, Après : 0 pt

☐ Afficher l'aperçu
☐ Désactiver les styles liés

Example MS Word



Contribution and benefit

This course intends to provide newcomers with background in task modeling. It provides an overview on how the recent advances in task description techniques can be exploited to design and assess interactive systems. As task models can be large, it is important to provide the analyst with computer-based to … them. To th … the HA … ectly ap …

Objectives

On completion of this tutorial, attendees will:

- Know the benefits of using task modeling techniques to design, structure and assess UIs,
- Be able to describe users' activities in a systematic and structured way,
- Have experience in analyzing an interactive systems focusing on the tasks users have to perform with it,
- Know how to use the HAMSTERS tool suite for editing, analyzing and simulating task models.

Instructors' background

The instructors have applied task modeling techniques to several industrial projects such as the design of

Various types of task models

- Existing (extant) System
  - One or many systems
  - One or many operators organization
- Envisioned System
  - One or many systems
  - One or many operators organization
- Variation of models (co-evolution) as tasks are performed on a given system – task modeling is a moving target

Carroll, J. M., Kellogg, W. A. and Rosson, M. B. 1991. **The task-artifact cycle. Designing interaction: Psychology at the human-computer interface**.
Carrol, J. M. Cambridge, England, Cambridge University Press: 74-102.

Summary

- Main principles of task modeling
- A notation for describing user tasks
- A tool to create, edit and simulate task models
- A simple technique to describe and analyze user tasks

Advanced design and fine tuning of interaction supported by task models

- Structuring mechanisms in task models and in HAMSTERS
- Industrial case study: UI for satellite platform monitoring and control
- Modeling process
  - Performance analysis
  - Exploiting task model for re-design/automation design/task migration
  - Task-based design of interactive systems
- Advanced practice of HAMSTERS tool and exercises
- Beyond (task-based UI design, human error, multiuser, training)
- Limitations (UX, co-execution, interface generation, …)

Thank you for your attention

- Acknowledgements
- Thanks to Racim Fahssi who is in charge of developing the edition module of the HAMSTERS tool
- Thanks to the ICS (Interactive Critical Systems) team for helping us with the testing of the tool