M'SILA University Faculty of Mathematics and Computer Science

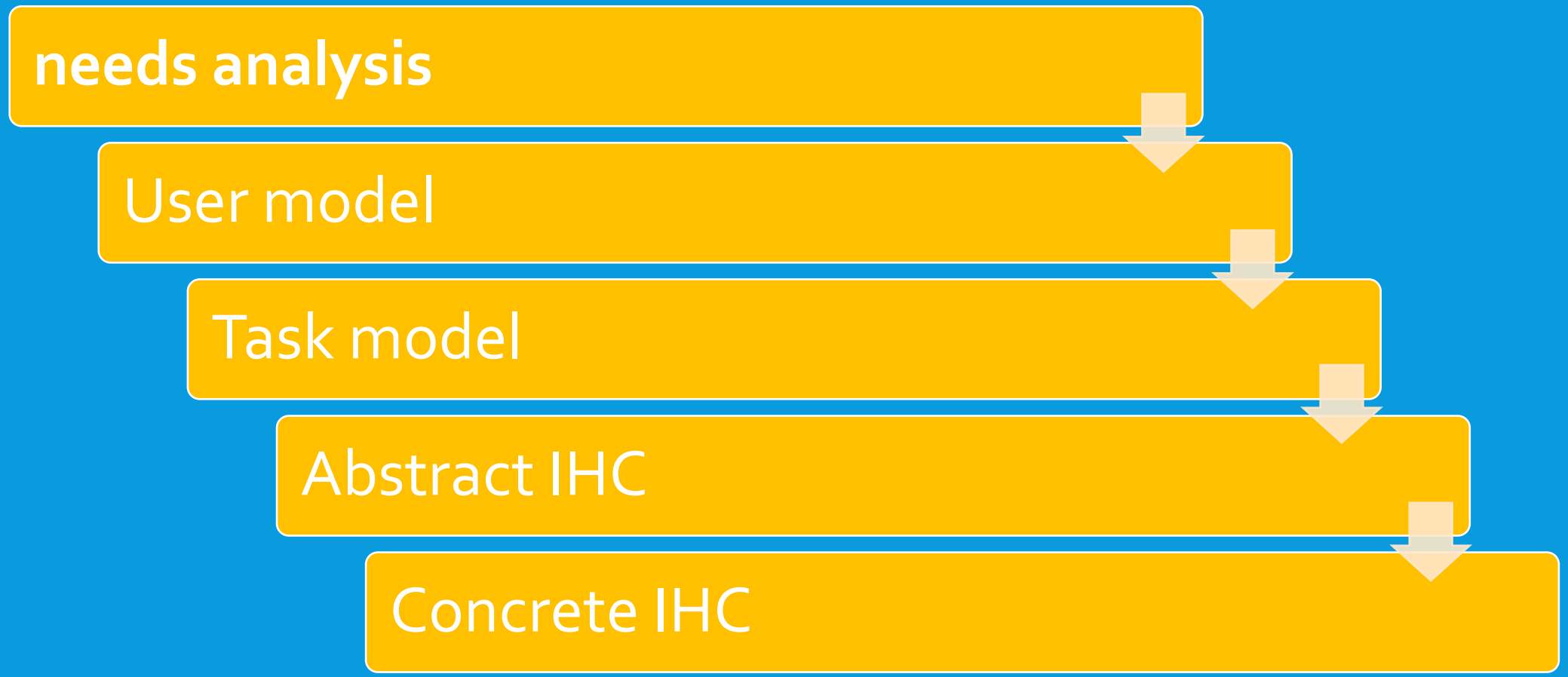
Department of Computer Science

TASK MODEL

Academic Year 2024-2025

Prepared and presented by SAOUDI Lalia

IHM DEVELOPMENT PROCESS



TASK ANALYSIS

- Task analysis involves collecting information on how users accomplish an activity.
- The information is obtained through information gathering methods:
  - Interviews
  - Observation
  - Reading reports
  - …etc

WHAT IS A TASK MODEL?

- Task models are logical descriptions of users' activities (interacting with the considered application) to achieve their goals,
- They are useful for designing, analysing, and assessing interactive software applications.

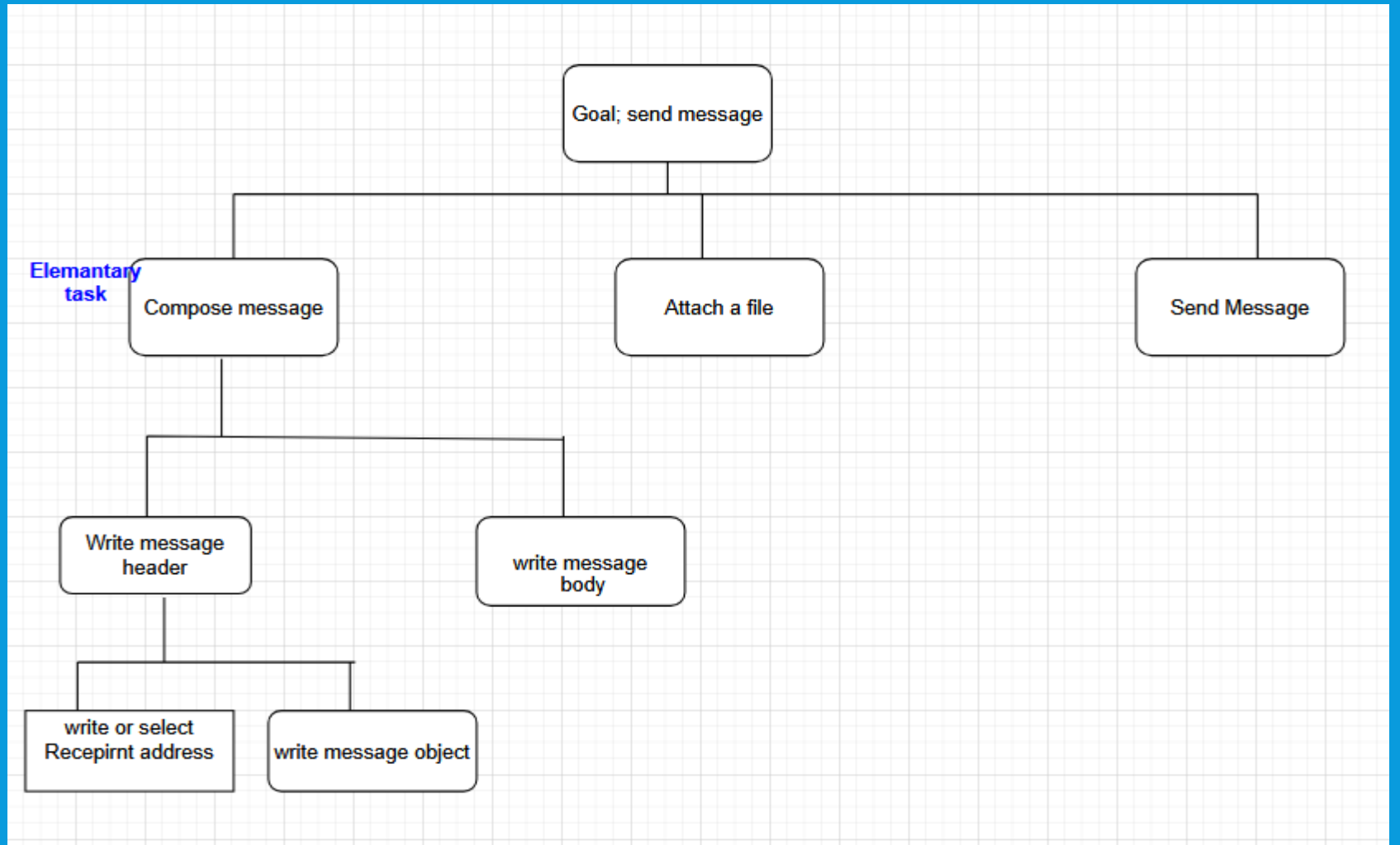
TASK MODELLING

- Task modelling describes how to use a system to achieve a goal, and it is represented as a sequence of commands.
- There are many notations for task models, such as CTT, HTA, MAD
- Use cases and UML sequence diagrams can also be used.

WHAT IS A TASK?

- A task consists of:
- **Goals**: State to be achieved (e.g.: write an email)
- **Procedure: is a set of subtasks linked by:**
  - **Composition relations and**
  - **Temporal relations**
- **Subtasks**
  - an elementary task is a task that can be broken down into physical actions
- **Actions**
  - Simple tasks, which do not have control structures, involve an operation on an I/O device that causes a change of state on the device.
- **Procedural level: is defined by an action or by a list of elementary tasks structured by logical-temporal relations (sequence, parallel, alternative, loop)**

WHAT IS A TASK?: EXAMPLE



TASK MODEL

- Task models are tree structures where the nodes are goals and the subtrees are the procedures to achieve those goals.
- The nodes can be decorated by:
  - the concepts in the domain (referenced objects, i.e. parameters, global variables …) ;
  - the preconditions and postconditions ;
  - the frequency ;
  - the complexity ;
  - the criticality (level of danger, irrevocability) ;
  - the time constraints (maximum duration) ;
  - the actor responsible for executing the task (user and/or system) ;
  - any other relevant information

TASK MODEL

- There are many notations for modelling tasks, we present the following models:
  - HTA: Hierarchical Task Analysis
  - MAD: Analytic Method for Task Description
  - CTT: Concurrent Task Trees

- Use cases and UML sequence diagrams can also be used

HTA: HIERARCHICAL TASK ANALYSIS

- **What is HTA?**
  - Hierarchical Task Analysis is a systematic method for describing how work is organized to achieve a goal. It breaks down tasks into a hierarchy of goals, sub-tasks, operations, and plans.

Core Components of HTA

- **Goal**: The overall objective to be achieved
- **Task**: Main activities required to achieve the goal
- **Sub-task**: Smaller components of main tasks
- **Operations:** Basic actions that can't be broken down further
- **Plans:** Rules specifying when and under what conditions sub-tasks are performed

HTA: HIERARCHICAL TASK ANALYSIS

- **Principle of the HTA model**
  - Hierarchically decompose tasks into sub-tasks down to actions.
  - For each task, associate a plan that defines how the sub-tasks unfold
  - The plan defines the structuring of the decomposition to describe the temporal and conditional relationships between the sub-tasks

  text 0. Main Goal
  - Plan 0: [Sequence instructions]
  - 1. First main task
  - Plan 1: [Sub-task sequence]
  - 1.1. First sub-task
  - 1.2. Second sub-task
  - 2. Second main task
  - etc.

HTA: HIERARCHICAL TASK ANALYSIS : EXAMPLE

0. Ride elevator to destination floor
   Plan 0: Do 1, then 2, then 3, then 4.

   1. Call elevator
      Plan 1: Do 1.1, then 1.2.
      1.1. Determine direction needed
      1.2. Press call button
           Plan1.2: Do either 12.1 or 1.2.2.

           1.2.1. Press "Up" button
           1.2.2. Press "Down" button

   2. Enter and select floor
      Plan 2: Do 2.1, then 2.2,

      2.1. Wait for and enter elevator
      2.2. Select destination floor

3. Ride and exit
   Plan 3: Do 3.1, then 3.2.
   3.1. Monitor elevator progress
   3.2. Exit at destination

NON-GRAPHICAL HTA : EXAMPLE

- 0. Perform online purchase
- 1. Login
  - 1.1 Select login screen
  - 1.2 Enter user name
  - 1.3 Enter password
- 2. choose object
  - 2.2 Browse listing
  - 2.3 choose product
- 3. Pay
  - 3.1 choose pay screen
  - Select payment method
  - 3.3 Enter payment card details
- Plan 0:1 then 2 as many times as needed then 3

THE MAD (MODEL FOR ANALYSIS & DESIGN)

- **Definition**
- MAD is "a model-based approach that structures the analysis and design of interactive systems around four interconnected models, focusing on the correspondence between user perspectives and system capabilities to ensure effective and accessible interaction."

THE MAD (MODEL FOR ANALYSIS & DESIGN)

- **The Four Core Models of MAD**
- **MAD describes an interactive system through four interconnected perspectives:**
- **1. User's Model (M-U)**
  - What it is: The user's mental model of how the system works
  - Content: User's beliefs, expectations, and understanding of system capabilities
  - IHC Focus: How users with different abilities and experiences perceive the system
- **2. System's Model (M-S)**
  - What it is: The actual implemented system and its capabilities
  - Content: Technical specifications, functionality, and internal processes
  - IHC Focus: What the system technically supports, including accessibility features

THE MAD (MODEL FOR ANALYSIS & DESIGN)

- The Four Core Models of MAD
- **3. User's Task Model (M-UT)**
  - What it is: The procedures and steps the user follows to accomplish goals
  - Content: User's goals, tasks, methods, and strategies
  - IHC Focus: How users with different capabilities approach tasks
- **4. System's Task Model (M-ST)**
  - What it is: The procedures and processes the system executes
  - Content: System workflows, state changes, and responses
  - IHC Focus: How the system supports multiple interaction pathways

THE MAD EXAMPLE (MODEL FOR ANALYSIS & DESIGN)

Scenario: User wants to leave a voice message

- 1. User's Model (M-U)
  - "I can press and hold a button to record my message"
  - "I'll hear a beep when recording starts"
  - "I can release the button when I'm done"
  - "The system will save my message automatically"
- 2. System's Model (M-S)
  - Touchscreen with pressure-sensitive areas
  - Audio input/output capabilities
  - File storage system for recordings
  - Timer functions for maximum recording length

THE MAD EXAMPLE (MODEL FOR ANALYSIS & DESIGN)

Scenario: User wants to leave a voice message

- User's Task Model (M-UT)
  - **textGoal**: Leave a clear voice message
  - **Plan**:
    - 1. Locate and press record button
    - 2. Wait for audio cue
    - 3. Speak message
    - 4. Release button
    - 5. Receive confirmation

THE MAD EXAMPLE (MODEL FOR ANALYSIS & DESIGN)

4. System's Task Model (M-ST)

- textGoal: Capture and store audio message
- Plan:
  - 1. Detect button press
  - 2. Play start tone
  - 3. Capture audio input
  - 4. Monitor for button release OR 60-second timeout
  - 5. Save audio file
  - 6. Play confirmation tone

CTT (CONCURRENT TASK TREE)

- CTT (Concurrent Task Tree) offers a significant number of operators, each with a precise meaning, allowing for the description of many temporal relationships, such as concurrency, interruption, deactivation, iteration, etc. This enables the designer to concisely represent the possible evolutions of a user session.

- A task model, described with this model, is constructed in three phases:
  - A hierarchical decomposition of tasks represented in the form of a tree;
  - An identification of the temporal links that connect these tasks;
  - An identification of the associated objects and actions.

CTT (CONCURRENT TASK TREE) : Core Concepts of CTT

- **1. Basic Elements**
  - ***Tasks***: Units of activity (represented as tree nodes)
  - ***Temporal Operators***: Define how tasks are related in time
  - ***Task Categories:*** Different types of tasks
  - ***Roles:*** Who performs the tasks (crucial for IHC)

CTT (CONCURRENT TASK TREE) : Core Concepts of CTT

2.Task Categories

- [User] - Performed entirely by the user
- |Application| - Performed entirely by the application
- [Interaction] - Involves user-system interaction
- {Abstract} - High-level task that decomposes further

CTT (CONCURRENT TASK TREE) : Core Concepts of CTT

3. Temporal Operators (Core of CTT)

- CTT (Concurrent Task Tree) proposes the following operators:
- T1 >> T2: sequence;
- T1 [data]>> T2: sequence & information transmission;
- T1 [] T2: disjunction;
- T*: closure (repetition from zero to as many times as desired);
- Tn: repetition;
- [T]: option;
- T1 [> T2: definitive interruption;
- T1 |> T2: interruption with resumption;
- T1 ||| T2: parallelism.

CTT (CONCURRENT TASK TREE) : Core Concepts of CTT

3. Temporal Operators (Core of CTT)

- 1 [] : . Choice Operator:
  - Meaning: The user can choose which task to perform.
  - Ex : LogIn()
    - ├──────── with Google account
    - └──────── with Email account
- 2.||| Concurrent Operator
  - Meaning: Tasks can be performed in any order, concurrently, or interleaved.
  - Ex ; SetupProfile |||
  - ├──────── EnterPersonalInfo
  - ├──────── SetAccessibilityPreferences
  - ├──────── UploadProfilePhoto

CTT (CONCURRENT TASK TREE) : Core Concepts of CTT

3. Temporal Operators (Core of CTT)

- **3. |=| Order Independence**
  - **Meaning: Both tasks must be completed, but order doesn't matter.**
- **4. >> Enabling**
  - **Meaning: The first task must be completed before the second can start.**

- 5. []>> : Enabling with Information Passing
  - Meaning: First task enables second and passes information to it.
  - Ex :SearchProducts []>> DisplayResults
- 6 [> . Disabling
  - Meaning: The first task is interrupted and terminated by the second.

CTT (CONCURRENT TASK TREE) : Core Concepts of CTT

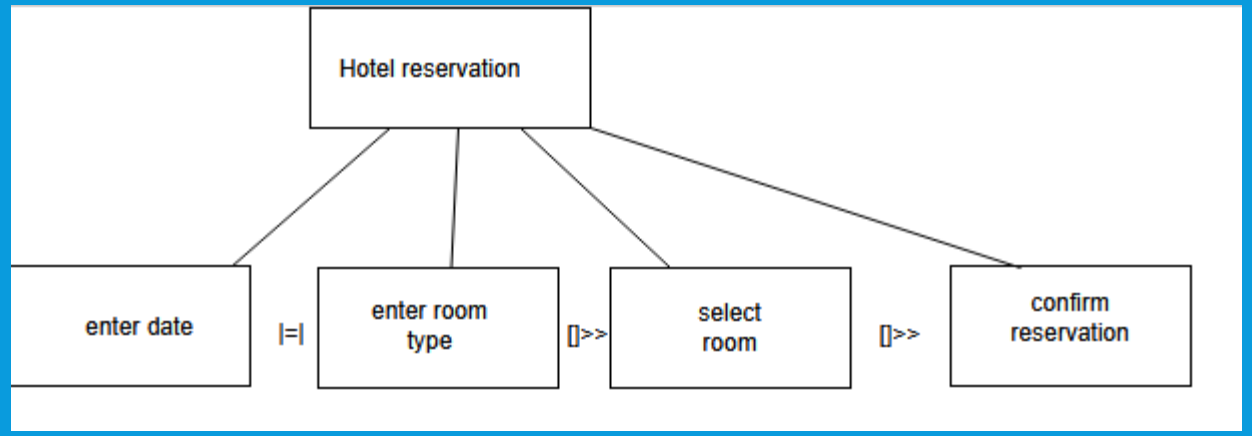
7 |>. Suspend/Resume

- Meaning: The first task can be suspended by the second and later resumed.
- EX : FillLongForm |> GetHelp

- 8. * Iteration
  - Meaning: The task is repeated multiple times.
  - Ex : Search room*
- 9. [ ]. Optional
  - Meaning: The task is not required for completion.
  - CompleteRegistration >> [ProvideFeedback]
- 10. . (n)* Finite Iteration
  - Meaning: The task is repeated a specific number of times.

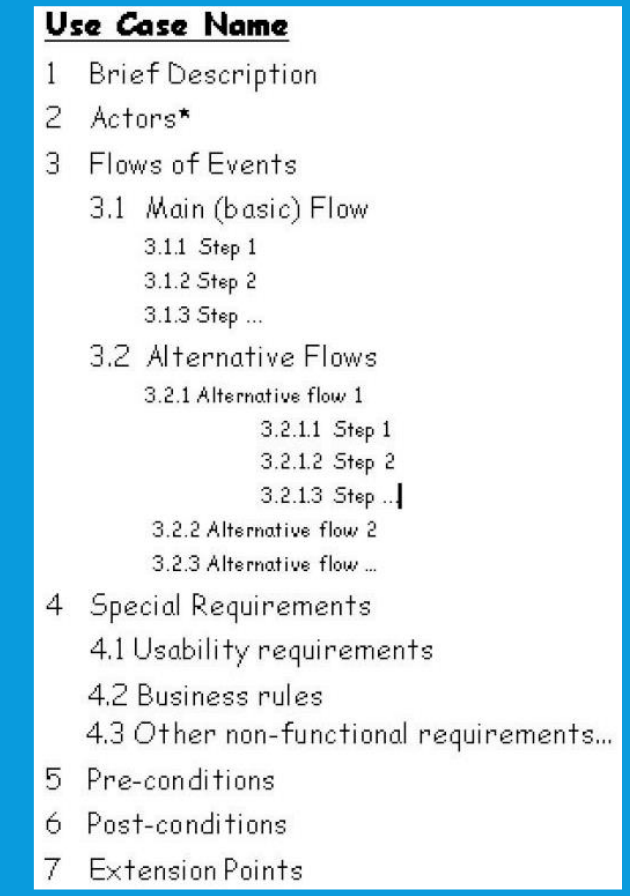
CTT (CONCURRENT TASK TREE)

- Example: the model above represents a CTT modelling of a task to consult information about a student.



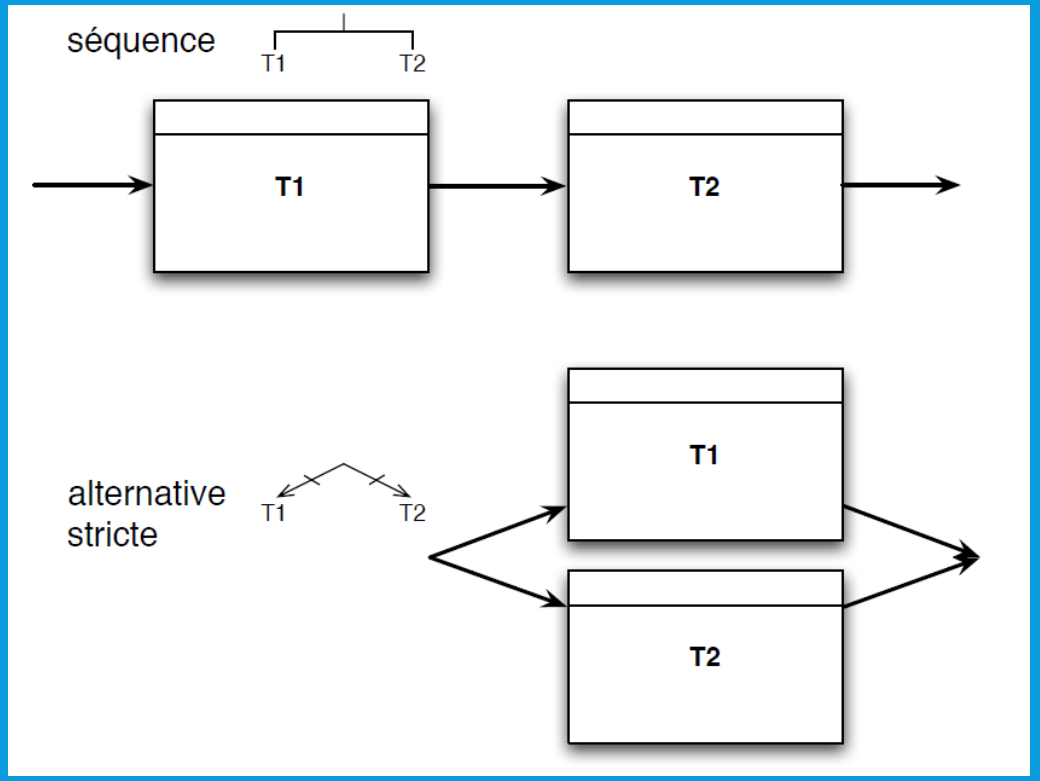
UML USE CASE

UML use case models specify the functional requirements from the actors' perspective.



FROM THE TASK MODEL TO THE ABSTRACT UI

- The task tree is segmented into workspaces.
- The relationships between tasks allow for the inference of sequences between workspaces.



FROM THE TASK MODEL TO THE ABSTRACT UI

- It is necessary to verify the conformity of the sequences with the task model (semantic distance).
- It is essential to annotate the workspaces with relevant domain concepts.
- Workspaces can be grouped to shorten interaction trajectories.
- It is necessary to ensure that frequent tasks are quickly accessible.
- Public utility services such as cut/paste, undo/redo, relevant default values, help... should be added (and must be included).

FROM THE ABSTRACT UI TO THE CONCRETE UI.

- It is important to keep in mind that the goal is to reduce execution and evaluation distances.
- To achieve this, it is necessary to:
  - check compliance with the abstract HMI;
  - ensure the visibility of relevant concepts;
  - highlight the changes in state;
  - group information that has a logical connection.