Full OCM model for the ATC

Andrea Lorenzani

Abstract

In this seminar I'll show you:

- Two (similar) formal model of the cognitive processes involved in a simplified Air Traffic Control task

- The role of these models in the investigation, detection and prevention of human error in safety-critical systems

Cognitive process

"Cognitive process is the manipulation of events, concepts, images, thoughts or other symbolic material in the mind. The cognitive process is the higher mental processes of reasoning, planning and problem solving."

Why is it important?

The relevance of the absence of errors is high, mainly in safety-critical systems, in which any error can bring injury, also the loss of life of the person connected to the system.

ATC is an example of these systems in which an human error can lead to an air crash with the death of many people.

Introduction

Existing models of human error do not provide a precise specifications of the conditions leading to error or the mechanism responsible for error.

For the model of cognition we use a simplified ATC system and we take into account some psychological theories of human error, starting from the HCI of the ATC.

Introduction (2)

The purpose of the cognitive model is to identify the main cognitive processes and how the operator's attention moves from one process to another.

The model is hierarchical and considers both memory-based and rule-based processes.

Semplifications

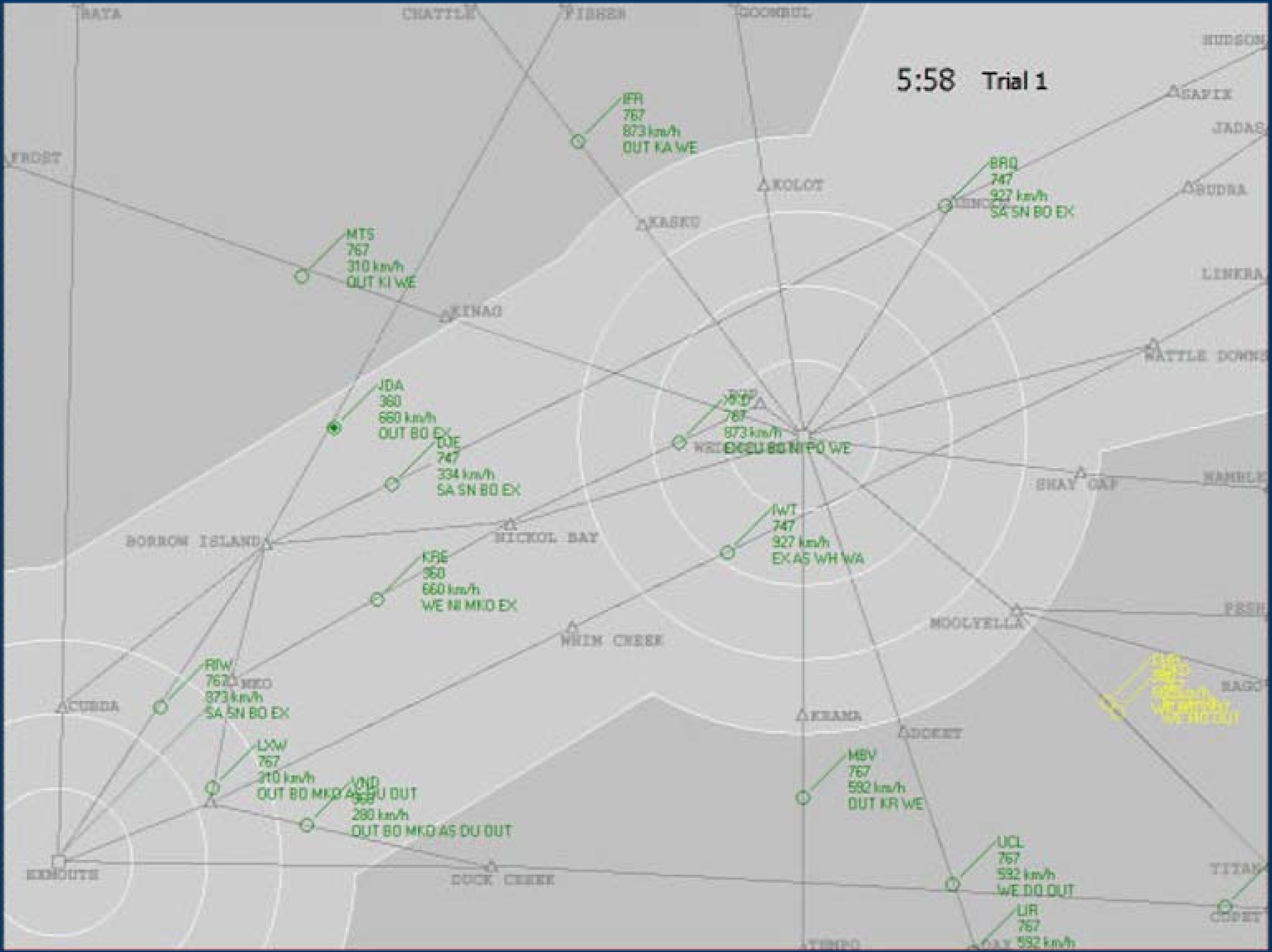
- Aircrafts move only in two dimensions
- Aircrafts travel only on path ('flight paths')
- Aircrafts change the speed instantly
- Aircrafts change the course instantly
- Operators may change only speeds
- Pilots respond always to the instructions

Terminology

- Aircrafts fly along straight-line segments, called *flight paths*, between *waypoints* within a fixed *sector* of airspace.

- The operator *scan* a representation of this sector in his *screen*, which includes the HCI for the ATC, and his main task is to ensure that aircrafts remain separated by no less than a minimum distance. Failure of this requirement is called *separation violation*.

ATC Screenshot

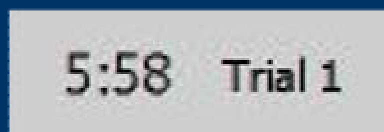


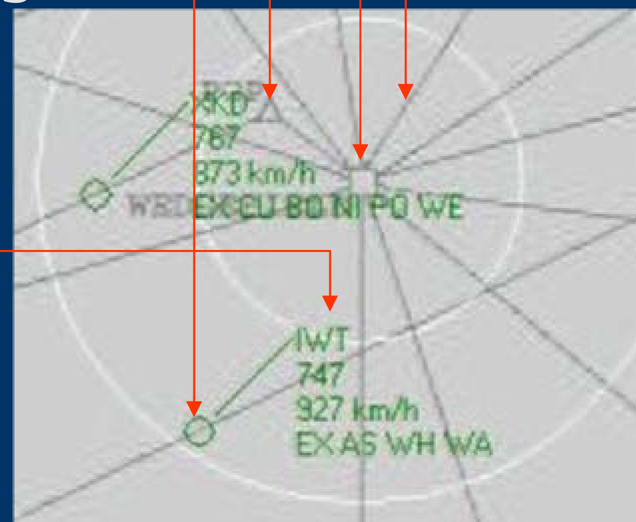
ATC Human-Computer Interface

- HCI visualises the state of the underlying simulation
- HCI provides a basic range of operations for acting on it (selecting an aircraft and changing its speed)
- Operator acts only to avoid separation violation, he has not to ensure efficiency and minimal delay
- The operator is aided by a simulation timer
- The display is updated at short intervals
- Separation violation is indicated with alarm and colour

The display

- Flight paths are shown in grey lines
- Aircrafts are rapresented by circles
- Airports are shown as squares
- Waypoints are shown as triangles
- Details of each aircraft are shown on labels attached to the aircraft symbol
- Simulation timer is in the top right corner





User Interface (UI) functionality

The operator is responsible only for changes to aircraft speed, so the simulator is fairly simple. The UI provides the operator with 2 functions:

- Selecting a single aircraft
- Changing the speed of the selected aircraft

UI functionality (2)

To select an Aircraft the operator moves the cursor on it and click the left button: a dot appears in the center of the circle

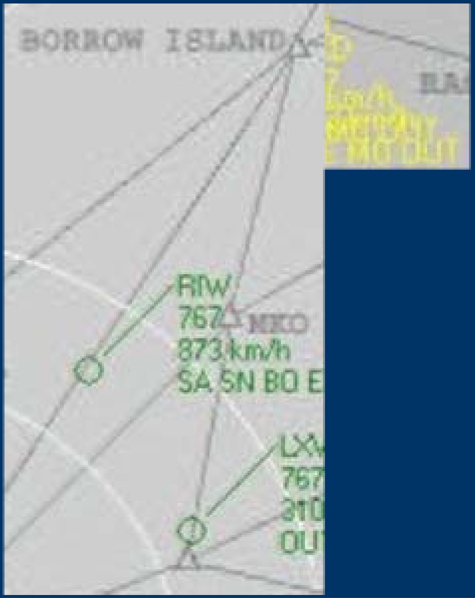
To change the speed the operator has to:

1. Open the menu by clicking on the right mouse button
2. Navigating the speed menu
3. Selecting a speed by clicking the left mouse button

Task terminology

A *conflict* is defined as a separation violation that will occur if two aircraft will have their speeds left unchanged. There are 2 types of conflicts:

- *Overtaking conflict*
- *Convergence conflict*



Task terminology (2)

- A *problem* is a pair of aircraft to which the controller pays attention as possibly being in conflict
- An *episode* refers to a problem as it develops over time.
- At any time only one episode can be *active*, that is the operator has his attention on it.
- The operator can take one ore many *corrective actions* to resolve a problem.

Background

- There are considerable progress in understanding the task conditions that lead to human error
- Traditional approaches do not allow precise formulations of error to be developed
- The cognitive model presented is based on new psychological theories
- The model is given using statecharts, it is memory-based and rule-based

Memory

“is an organism's ability to store, retain, and subsequently recall information. There are several ways to classify memories, based on duration, nature and retrieval of information. From an information processing perspective there are three main stages in the formation and retrieval of memory:

- *Encoding or registration (processing and combining of received information)*
- *Storage (creation of a permanent record of the encoded information)*
- *Retrieval or recall (calling back the stored information in response to some cue for use in a process or activity)”*

Memory (2)

“A basic and generally accepted classification of memory is based on the duration of memory retention, and identifies three distinct types of memory: sensory memory, short term memory and long term memory.”

We take into account only Short Term Memory and Long Term (Episodic) Memory

Short Term Memory (STM)

- Temporarily records information regarding current problem, including active problem.
- Contains a truncated version of the data relations and priority of problems
- Recall is determined by recency
- Has a very limited capacity
- Has a very fast access

Long Term Memory (LTM)

- Records episodes that the operator experiences
- Is cued by the info presented on the screen
- It is possible to retrieve a number of different types of knowledge from this:
  - Semantic knowledge (abstracting from similar episodes)
  - Episodic knowledge (info of a specific episode)
- The infos are influenced by many factors (cues used, frequency and recency of episodes...)

Cognitive data relation

Data of a specific episode stored in Long Term Memory is modelled as tuples with the following information:

- *Aircrafts attributes*
- *Context* (time, position...)
- *Classification*
  - *Conflict*
  - *Non-conflict*
- *(Projected) Time of violation*

Cognitive data relation (2)

- *Priority*
  - *IS_HIGHEST*
- *Decision (corrective actions)*
- *Decision stored ?*
  - *DECISION_STORED*
- *Windows of opportunity:*
  - *Inside Window*
  - *Outside Window*
  - *Must Act Now*
- *Behaving as expected?*
  - *AS_EXPECTED*

Cognitive data relation (3)

At any time a data relation may contain incomplete information.

An example of tuple of cognitive data is the following:

(({JDA, 360, 660km/h}, {DJE, 747, 334km/h}), "Approaching Borrow Island en-route to Exmouth airport", conflict, 5:58+10, Is Highest, ((JDA, 330km/h, ?, now), (DJE, 860km/h, ?, now)), decisionStored, Inside Window, AsExpected)

Cognitive model

- Identifies main cognitive processes used by operators
- Describes the flow of control through these processes
- Each state represent an abstract cognitive process
- Described with UML-like notation (statechart)
- Uninterruptable process is written as “action”
- Predicates for the cognitive data relations are used

Operator Choice Model

Scanning: The operator searches the interface for a certain property.

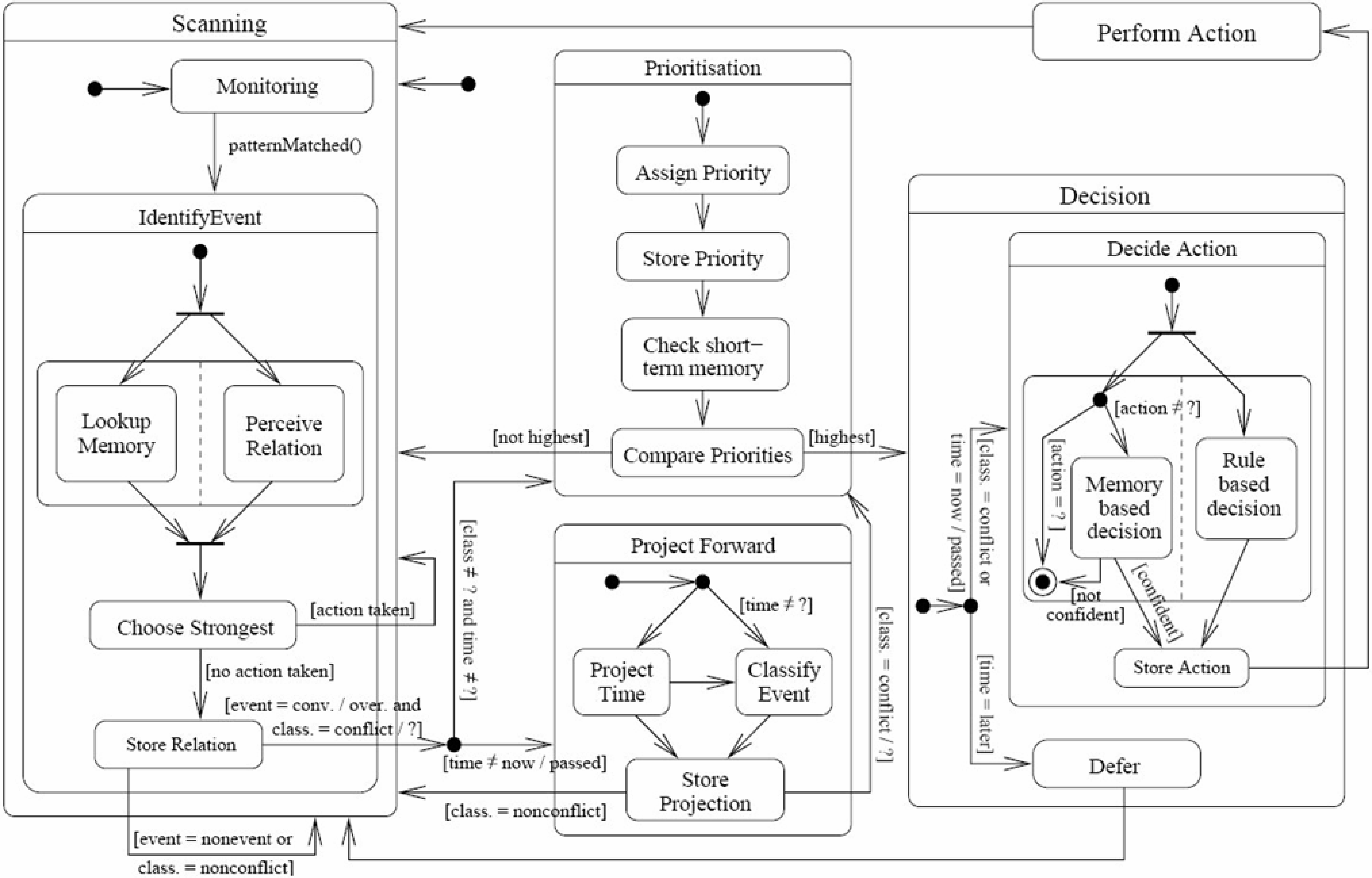
Identification: The operator identifies part of the interface that may represent the property.

Classification: The operator

- assesses whether the property is in need of further interest;
- if so, gives some form of priority to the property.

Decision on how to resolve the situation.

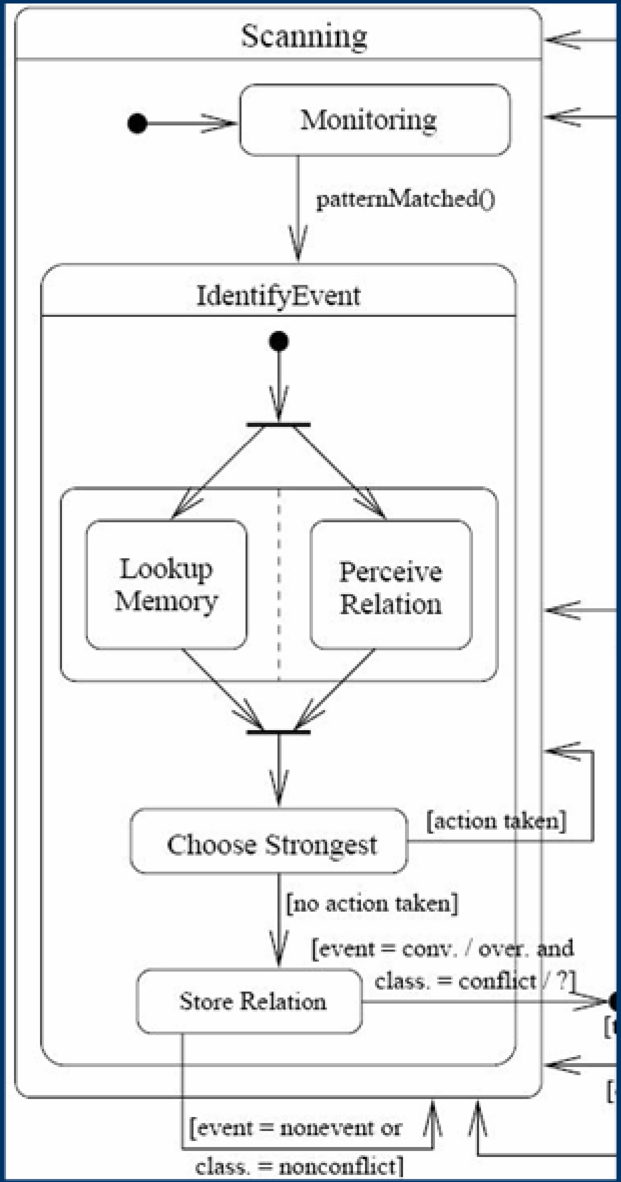
Action to be performed as a series of interaction with the interface.



Scanning
Monitoring
patternMatched()
IdentifyEvent
Lookup Memory
Perceive Relation
Choose Strongest
[action taken]
[no action taken]
[event = conv. / over. and class. = conflict / ?]
Store Relation
[event = nonevent or class. = nonconflict]
[time ≠ now / passed]
[class ≠ ? and time ≠ ?]
[class. = nonconflict]
Prioritisation
Assign Priority
Store Priority
Check short-term memory
Compare Priorities
[not highest]
[highest]
Project Forward
[time ≠ ?]
Project Time
Classify Event
Store Projection
[class. = conflict / ?]
Perform Action
Decision
Decide Action
[action ≠ ?]
[action = ?]
Memory based decision
Rule based decision
[not confident]
[confident]
Store Action
[class. = conflict or time = now / passed]
[time = later]
Defer

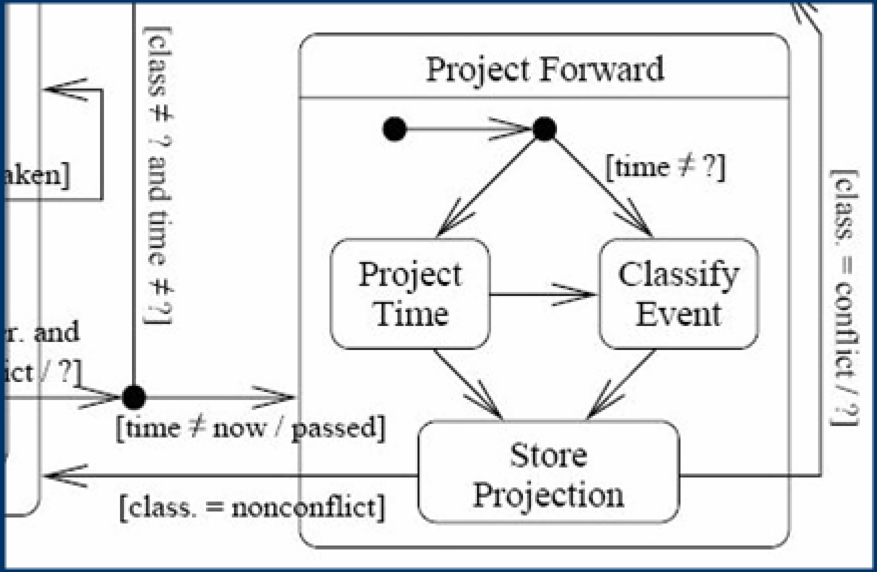
Scanning

- Monitoring the HCI until the operator matches an event geometry
- Retrieve a matching relation from memory and at the same time produce a new relation
- Choose the best relation
- If no action is previously taken, store the event relation in memory



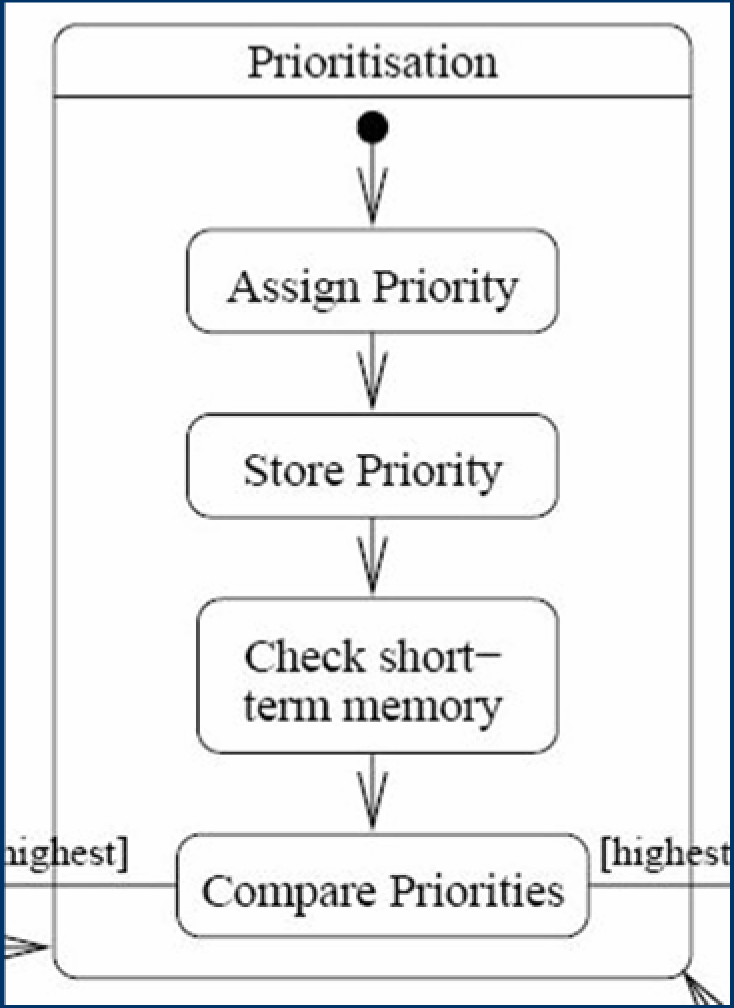
Project Forward

- Estimate the time at which action must be taken to resolve and classify the event
- Projection is skipped if immediate action is required



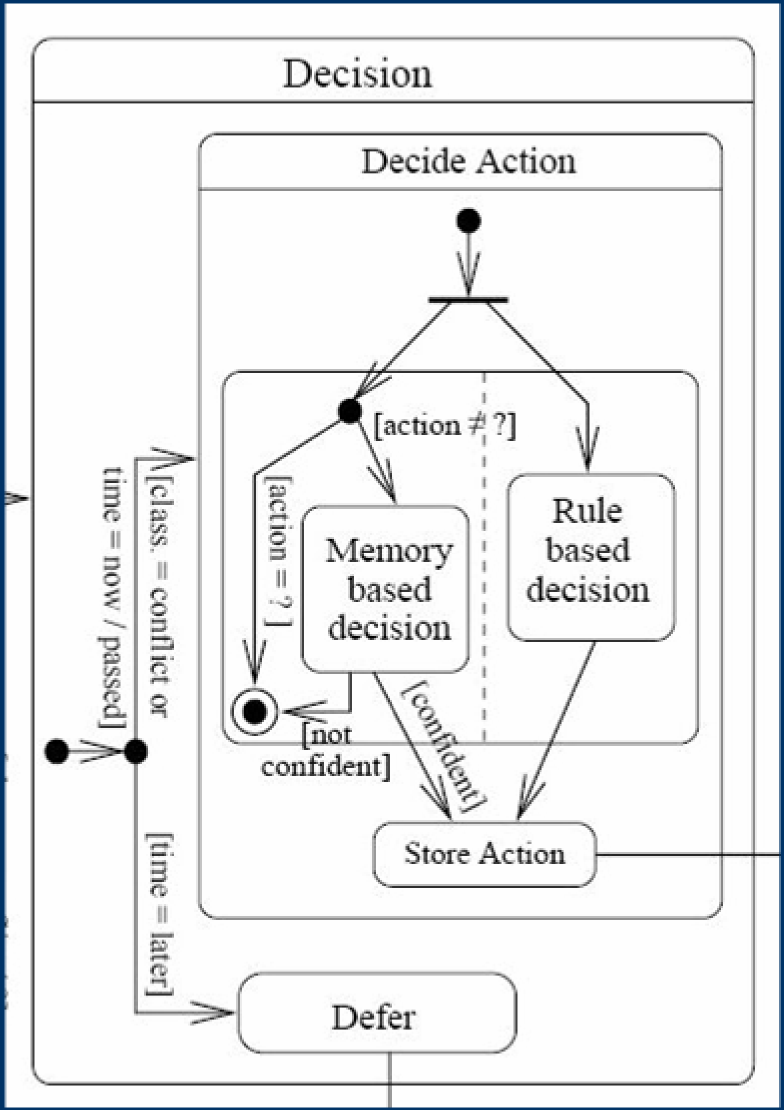
Priorisation

- Operator assign priority to check if it is the case to perform a corrective action
- This check is made in the Short Term Memory, where are stored the other data relation of the more recent scans
- If this relation has the highest priority the operator go to decision



Decision

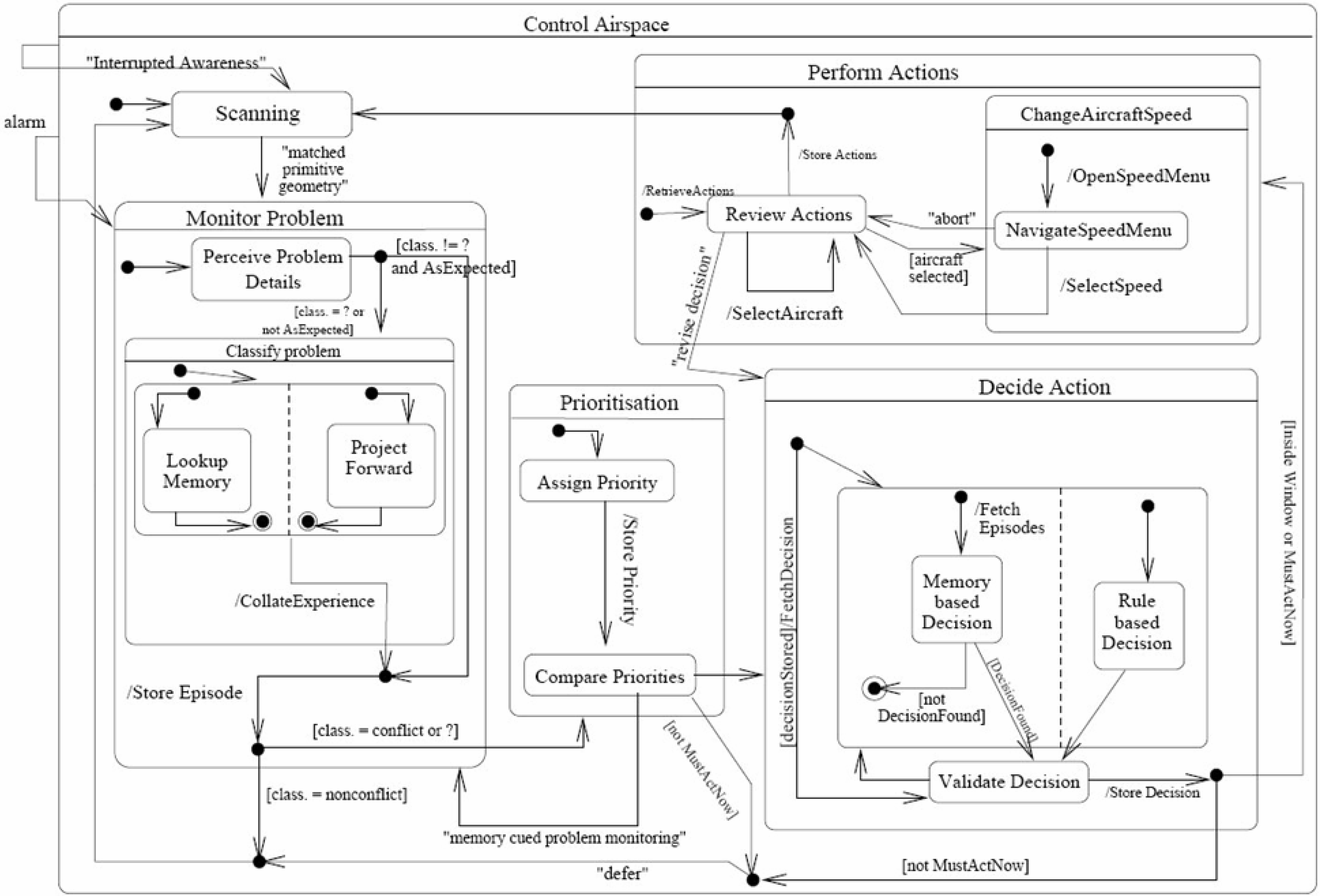
- If the action is not urgent the operator may defer the decision and return to scan
- At the end the decision is stored with event relation and operator proceeds to perform the action



Perform Action

- Operator performs the action choosen through the interation with ATC HCI

- Once the action is performed he returns to scan



Control Airspace
"Interrupted Awareness"
alarm
Scanning
"matched primitive geometry"
Monitor Problem
Perceive Problem Details
[class. != ? and AsExpected]
[class. = ? or not AsExpected]
Classify problem
Lookup Memory
Project Forward
/CollateExperience
/Store Episode
[class. = conflict or ?]
[class. = nonconflict]
Prioritisation
Assign Priority
/Store Priority
Compare Priorities
"memory cued problem monitoring"
[not MustActNow]
"defer"
Perform Actions
/Store Actions
/RetrieveActions
Review Actions
/SelectAircraft
ChangeAircraftSpeed
/OpenSpeedMenu
"abort"
NavigateSpeedMenu
[aircraft selected]
/SelectSpeed
"revise decision"
Decide Action
/Fetch Episodes
Memory based Decision
Rule based Decision
[not DecisionFound]
[DecisionFound]
[decisionStored]/FetchDecision
Validate Decision
/Store Decision
[Inside Window or MustActNow]
[not MustActNow]

Scanning

- In this process operator tries to identify problems looking for “patterns” (*primitive geometries*)

- There can be differences between operators during this process (different experiences)

Monitoring

- Used to gather infos about problem being attended and classify it.
- Operator encodes infos from display and check episodic memory for the same problem (if it is successful he uses info from memory)
- The importance of the infos may differ
- Speed and confidence linked to recency and frequency
- Infos can continue coming from display, to cue other infos from memory

Monitoring (2)

- Projection is used to estimate the time and position of the conflict, it is slow and difficult
- Novices use mainly Projection Forward
- Experts uses mainly Lookup Memory
- Collate the infos and store in episodic memory

Priorisation

- Compares priority in Short Term Memory
- The problem with highest priority go directly to Monitor Problem
- If the problem is not MustActNow the operator may defer the decision

Decision

- If it is made yet, the decision is skipped and it is directly revalidated
- Otherwise the operator check his memory for any previous episode
- A confidence level is associated to this
- Validate is like Project Forward and may lead to formulation of ‘windows of opportunity’

Other Transitions

- Alarm
  - Go to ‘monitoring’ the indicated problem
- Interrupted awarness
  - The operator ‘forgets’ what he was doing
  - He restarts to Scanning

Types of errors

- Perception
  - Relates to Monitoring Task
- Action performance
  - Relates to Perform Action task
- Generation of actions
  - Involves all the remaining cognitive tasks

We can have error in any cognitive task!!

Base error rate

- We must consider each different type of error individually
- We associate a base error rate for every different type of error: this is the probability that an output associated to that error mode is made if all external factors are ignored
- External factors have a multiplicative effect on the base error rate: for every error mode the multiplicative effect changes (may also decrease the error rate)

Effect Matrix

External Factor	Cognitive Task: Lookup memory	Cognitive Task: Classify event	Cognitive Task: Rule-based decision
Workload ('pressure')	$\uparrow$w = $\uparrow$error $\downarrow$w = $\downarrow$error	$\uparrow$w = $\uparrow$error $\downarrow$w = $\downarrow$error	$\uparrow$w = $\uparrow$error $\downarrow$w = $\downarrow$error
Fatigue	$\uparrow$f = $\uparrow$error $\downarrow$f = $\downarrow$error	$\uparrow$f = $\uparrow$error $\downarrow$f = $\downarrow$error	$\uparrow$f = $\uparrow$error $\downarrow$f = $\downarrow$error
Frequency (of event)	$\uparrow$f = $\downarrow$error $\downarrow$f = $\uparrow$error	$\uparrow$f = $\downarrow$error $\downarrow$f = $\uparrow$error	no effect

List of external factors

- Task demands (workload, duration, attention required…)
- Instructions and procedures (accuracy, clarity, ease of use…)
- Enviroment (temperature, noise,movement restriction…)
- Stresses (workload, fatigue, monotony, distractions…)
- Individual (capacities, experience, skills…)

Design interventions

- Provide guidance for how to design the user interface such that particular operator errors may be diminished
- Can draw “pattern” with context, solution of the problem and examples
- Can be targetted at each individual cognitive task to reduce the base error rate
- Can involve changes to HCI design or to the cognitive model

Design interventions (2)

- Can be targetted also at each individual external factor for reducing the multiplier values
- In that case they are commonly not HCI related
- It is better to make interventions that improve the multiplier values of the major number of external factors