Incorporating Temporal Constraints in the Planning Task of a Hybrid Intelligent IDS

Álvaro Herrero[1], Martí Navarro[2], Vicente Julián[2] and Emilio Corchado[3]

[1] Department of Civil Engineering, **University of Burgos** (Spain)
[2] Departamento de Sistemas Informáticos y Computación, **Universidad Politécnica de Valencia** (Spain)
[3] Departamento de Informática y Automática, **Universidad de Salamanca** (Spain)

5th Int. Conference on Hybrid Artificial Intelligence Systems

San Sebastián (Spain) - June 2010

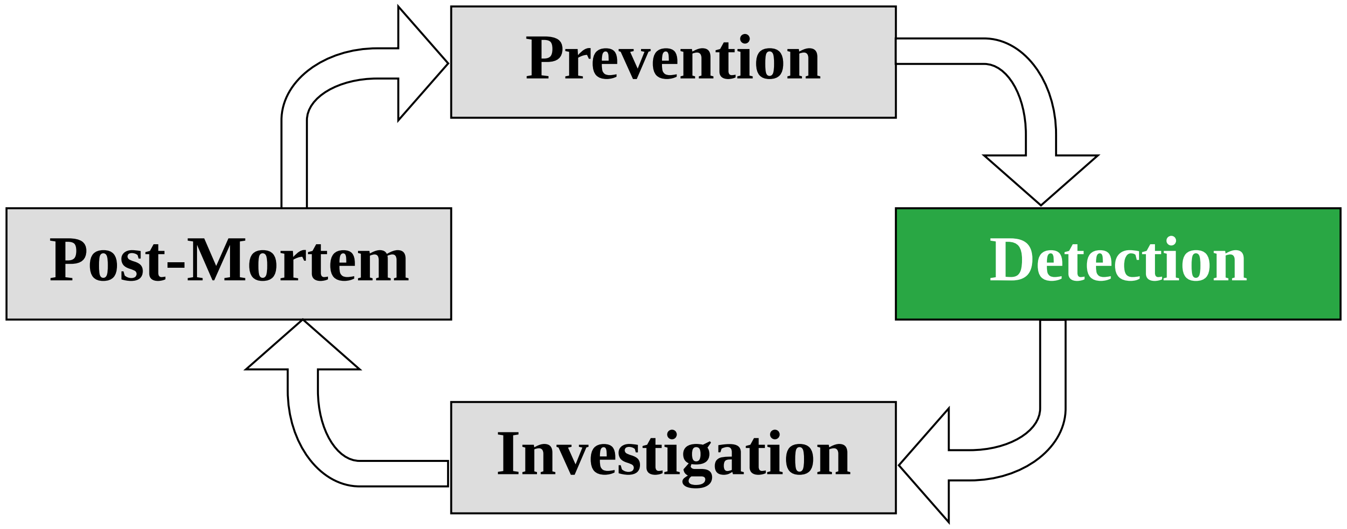
HAIS'10

Outline

- Computer Security
- Intrusion Detection Systems (IDS's)
- MOVICAB-IDS
  - Overview
  - Projection Technique - CMLHL
  - Agents
  - Sample Visualizations
- Real-Time Extension
- Conclusions

Computer Security

A Security Management Model



This work addresses the detection component, that identifies security breaches that are or can be exploited

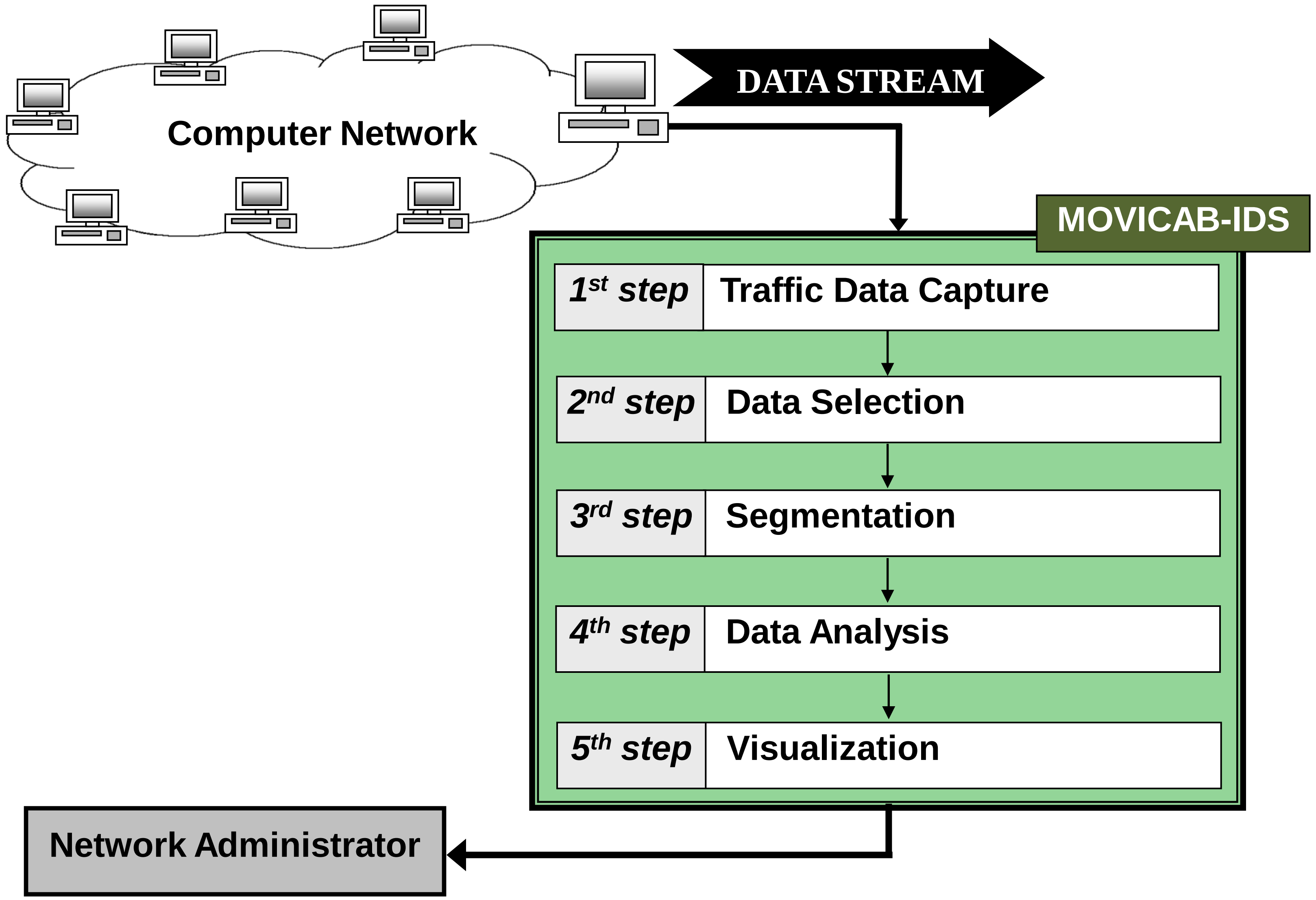
Intrusion Detection Systems

- New **security failures** are discovered everyday and there is a growing number of bad-intentioned people trying to take advantage of such failures.
- New **network security tools** are being developed. Firewalls are the most widely used but **Intrusion Detection Systems** (IDSs) are becoming more popular.
- IDSs monitor the activity of the network in order to identify intrusive events and can also take actions to abort these risky events.

MOVICAB-IDS (I)

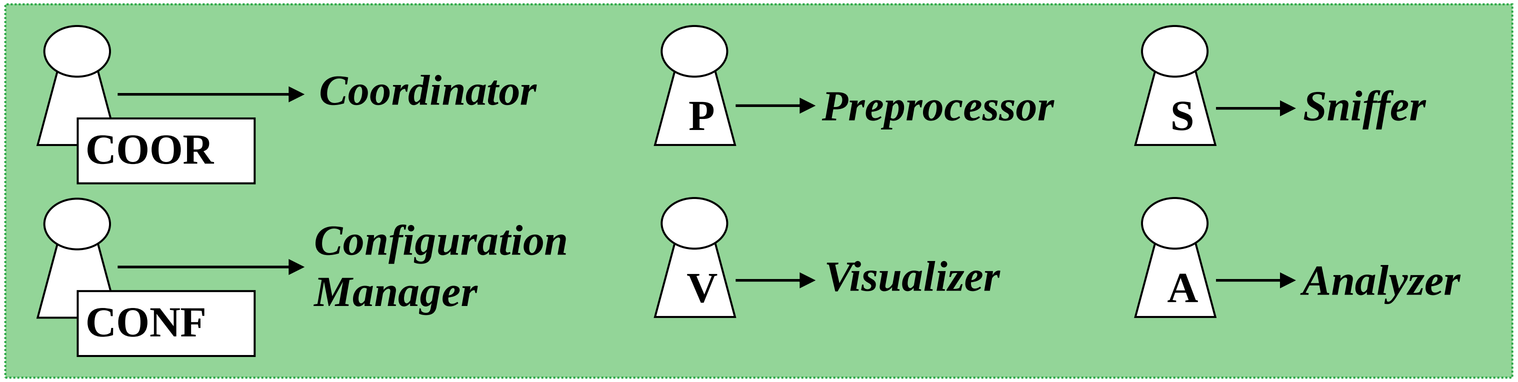
- The **Mobile Visualization Connectionist Agent-Based IDS** (MOVICAB-IDS) has been designed to detect anomalous situations taking place in a computer network.
- The proposed MAS incorporates:
  - reactive agents.
  - deliberative (CBR-BDI) agents.
- The extended version of the Gaia methodology has been applied.

MOVICAB-IDS (II)



MOVICAB-IDS (III)

- The **CBR-BDI agents** use CBR systems as a reasoning mechanism, which allows them:
  - to learn from initial knowledge.
  - to interact autonomously with the environment, users and other agents within the system.
- Six agents have been developed in this study:



Projection Technique – *CMLHL I*

- Based on Maximum Likelihood Hebbian Learning (**MLHL**) [Corchado *et al*., 2004].
- It has been applied in other research fields (*i.e.* Artificial Vision [Corchado *et al.*, 2003])
- **Lateral Connections** take into account the relation between the distances among the output neurons. They are derived from the Rectified Gaussian Distribution [Seung *et al., 1998*].
- It gives a more sparse representation.

Projection Technique – *CMLHL II*

- Feedforward:

$$y_i = \sum_{j=1}^{N} W_{ij} x_j , \forall_i$$

- Lateral connections:

$$y_i(t+1) = \left[y_i(t) + \tau \left(b - A\mathbf{y} \right) \right]^+$$

- Feedback:

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i$$

- Weight Update:

$$\Delta W_{ij} = \eta . y_i . sign\left(e_j \right) | e_j |^{p-1}$$

Agents (I)

Sniffer

- It captures the traffic data (continuous traffic flow) and splits it into segments to send it through the network for further process.

Preprocessor

- The split segments are preprocessed in order to apply subsequent analysis. Then, an analysis for this data is requested.

Configuration Manager

- The processes of data capture, split, preprocess and analysis depends on the values of several parameters. This information is managed by this agent.

Agents (II)

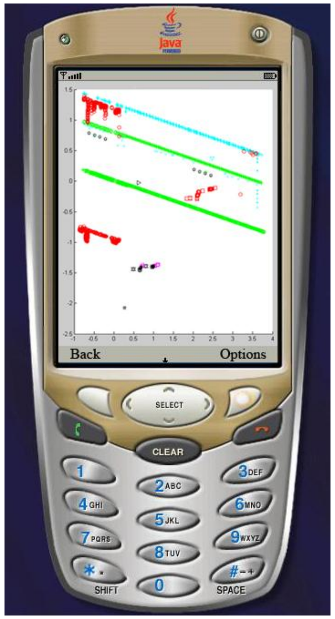
Analyzer

- This CBR-BDI agent has got embedded a connectionist model within the adaptation stage of its CBR system that helps to analyze preprocessed traffic data.
- The connectionist model is the previously introduced Cooperative Maximum Likelihood Hebbian Learning (CMLHL).
- This agent generates a solution (or achieve its goals) by retrieving a case and analyzing the new one using a CMLHL network.
- Learning and exploitation modes.

Agents (III)

Visualizer

- This is an interface agent. The analyzed data is presented to the network manager by means of a functional and mobile visualization interface:



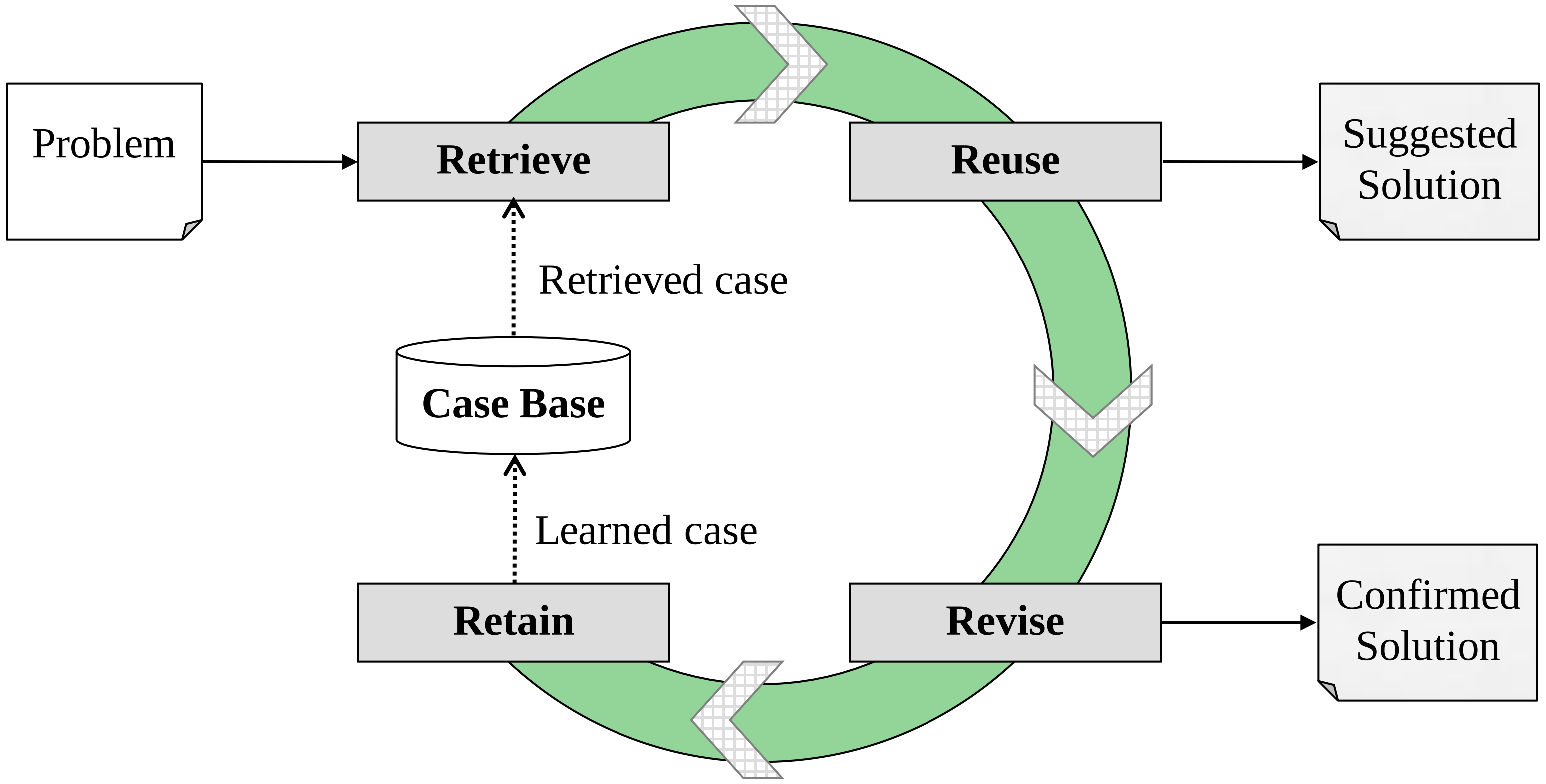
Coordinator (I)

- This CBP agent is in charge of sharing the analysis work out among the Analyzer agents. The preprocessed data must be dynamically and optimally assigned (taking into account the resources and demands).

Agents (IV)

Coordinator (II)

- Case-Based Reasoning



Agents (V)

Coordinator (III)

- The Coordinator agent plans to allocate an analysis to one of the available Analyzer agents based on the following criteria:
  - **Location**. Prioritise closer Analyzer agents (located in the same network segment).
  - **Available resources** of the computer where each Analyzer agent is running.
  - **Analysis demands**. Amount and volume of data to be analysed.
  - **Analyser agents behaviour**. Rather "learning" or "exploitation" mode.

Agents (VI)

Coordinator (IV)

- Case representation:

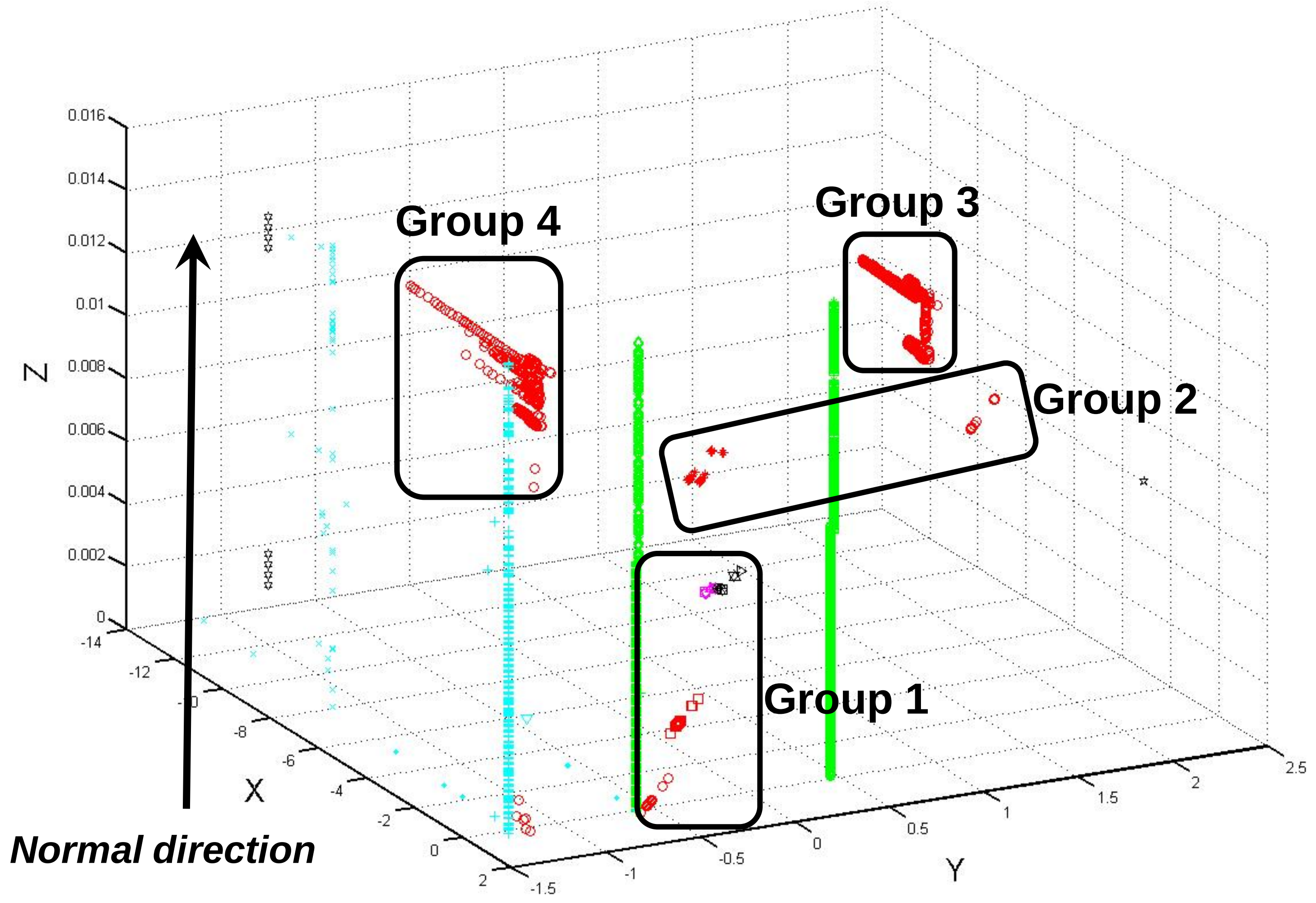
Class	Feature	Type	Description
P	#packets	Integer	Total number of packets contained in the dataset to be analysed.
P	Analyzers / location	Array	An array (of variable length depending on the number of available Analyzer agents) indicating the network segment where the Analyzer agent is located.
P	Analyzers / features	Array	An array (of variable length depending on the number of available Analyzer agents) containing information about the resources, their availability, and pending tasks.
P	Analyzers / failures	Array	An array (of variable length depending on the number of available Analyzer agents) containing information about the number of times each Analyzer agent has stopped working in the recent past (execution failures).
S	Analyzers / plans	Array	An array (of variable length depending on the number of available Analyzer agents) containing the analyses assigned to each Analyzer agent.

Sample Visualizations (I)

- Data sets (packets over UDP):
  - **Timestamp**: the time when the packet was sent.
  - **Source Port**: the port of the source host from where the packet was sent.
  - **Destination Port**: the port of the destination host to where the packet was sent.
  - **Size**: total packet size (in Bytes).
  - **Protocol**: all the protocols have been codified.
- In addition to the "*anomalous*" packets, the data sets include traffic related to other protocols in the network (*normal traffic*).

Sample Visualizations (II)

CMLHL 3-D projection



Real-Time Extension (I)

- Less than 15 minutes are required by some distributed and coordinated attacks to cease normal functions of a large portion of Internet.
- Response time is a critical issue for most of the security infrastructure components of an organization. The importance of a fast, predictable and smart response increases in the case of IDSs.
- For this reasons, the original formulation of MOVICAB-IDS has been upgraded by modifying the Coordinator agent.

Real-Time Extension (II)

Temporal Bounded CBR [TB-CBR] (I)

- The phases of the TB-CBR cycle are grouped in 2 stages according to their function within the reasoning process of an agent with real-time constraints:
  - **Learning stage**: revise and retain phases.
  - **Deliberative stage**: retrieve and reuse phases.
- Each phase will schedule its own execution time to support the designer in the time distribution among the TB-CBR phases.
- These stages can incorporate an anytime algorithm.

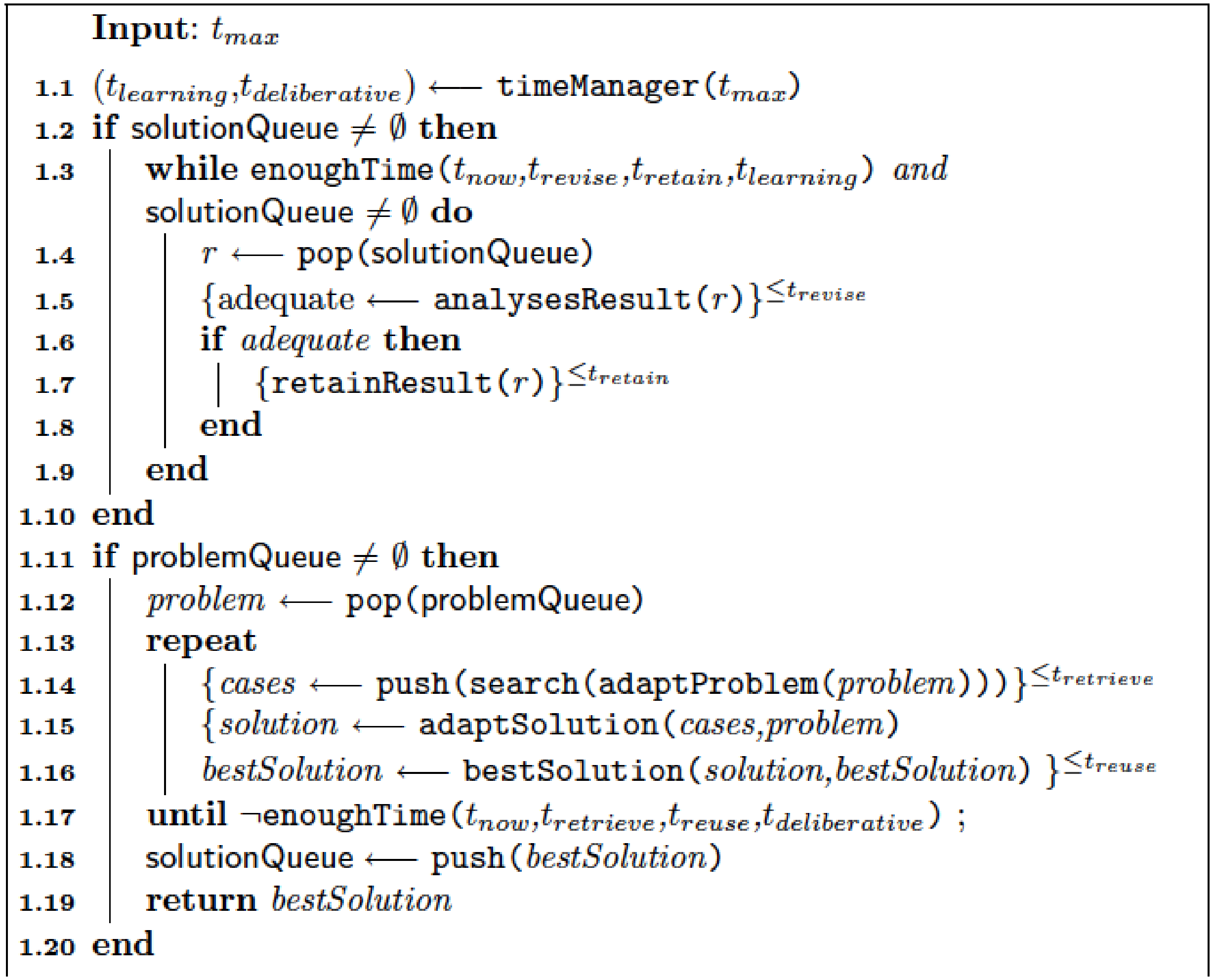
Real-Time Extension (III)

Temporal Bounded CBR [TB-CBR] (II)

- **Learning stage**
  - Entails checking whether previous cases are awaiting for revision and could be stored in the case base.
  - The solutions provided by the TB-CBR are stored in a solution list at the end of the deliberative stage. When there is sufficient time, the learning stage is implemented for cases where solution feedback has recently been received.
- **Deliberative stage**
  - At the end of each iteration in the deliberative stage, the TB-CBR method provides a solution which may be improved in subsequent iterations if there is any remaining time.

Real-Time Extension (IV)

Temporal Bounded CBR [TB-CBR] (III)



```
Input: $t_{max}$
($t_{learning}$,$t_{deliberative}$) $\longleftarrow$ timeManager($t_{max}$)
if solutionQueue $\neq \emptyset$ then
    while enoughTime($t_{now}$,$t_{revise}$,$t_{retain}$,$t_{learning}$) and
    solutionQueue $\neq \emptyset$ do
        $r \longleftarrow$ pop(solutionQueue)
        {adequate $\longleftarrow$ analysesResult($r$)}$^{\leq t_{revise}}$
        if adequate then
            {retainResult($r$)}$^{\leq t_{retain}}$
        end
    end
end
if problemQueue $\neq \emptyset$ then
    problem $\longleftarrow$ pop(problemQueue)
    repeat
        {cases $\longleftarrow$ push(search(adaptProblem(problem)))}$^{\leq t_{retrieve}}$
        {solution $\longleftarrow$ adaptSolution(cases,problem)
        bestSolution $\longleftarrow$ bestSolution(solution,bestSolution) }$^{\leq t_{reuse}}$
    until $\neg$enoughTime($t_{now}$,$t_{retrieve}$,$t_{reuse}$,$t_{deliberative}$) ;
    solutionQueue $\longleftarrow$ push(bestSolution)
    return bestSolution
end
```

Real-Time Extension (V)

Temporal Bounded CBR [TB-CBR] (IV)

- Temporal restrictions:

$$t_{\max} \geq t_{cognitiveTask}$$

$$t_{cognitiveTask} \geq t_{learning} + t_{deliberative}$$

$$t_{learning} \geq (t_{revise} + t_{retain}) * n$$

$$t_{deliberative} \geq (t_{retrieve} + t_{reuse}) * m$$

Where:

$t_{cognitiveTask}$	maximum time available to provide a response.
$t_{deliberative}$ $t_{learning}$	total execution times of each deliberative stage.
t_x	execution time of phase *x*.
n, m	number of iterations of each stage.

Real-Time Extension (VI)

TB Coordinator Agent (I)

- The 4 phases of the TB-CBP cycle are re-defined to comply with the temporal constraints:
  - Revise. Two-fold analysis:
    - Planning failures are identified by finding under-exploited resources.
    - Execution failures are detected when communication with Analyzer agents has been interrupted.
  - Retain. When a plan is adopted, the Coordinator agent stores a new case containing the dataset-descriptor and the solution.
- This learning stage is executed if the agent has the plans from previous executions stored in the *solutionQueue.*

Real-Time Extension (VII)

TB Coordinator Agent (II)

- The deliberative stage is launched when there is a new network segment to be analysed (in the *problemQueue*):
  - (Plan) Retrieve: the Coordinator agent knows how long it takes to get the first solution. After that, if there is some extra time to plan the analyses, the agent will attempt to improve the first plan within the available time by continuing searching previously stored plans.
  - Reuse: the Coordinator agent knows when it will finish the adaptation of the cases and when the Analyzer agents will finish their tasks. So, it also knows the available time to continue building the plan.

Real-Time Extension (VIII)

Performance

- The CPU utilization and average execution time have been checked in a set of tests.
- Analysis requests were generated each 3 seconds. 2 ms were allocated for planning. Results after 100 executions:

	CPU utilization	**Analysis fulfilled on time**	**Average execution time**
TB-CBP	97 %	98.2 %	1.6 ms
CBP	72 %	61.5 %	2.4 ms

Conclusions

- The main advantage of using the TB-CBP with regard to using a CBP without temporal constraints is to ensure a system response on time.
- The use of TB-CBP allows the distribution of the analysis to the Analyzer agents taking into account the available time to perform this task.
- On the other hand, the application of TB-CBP improves the CPU utilization and minimizes the average execution time of the analyses as it has been checked in a set of tests.

The End

University of Burgos

- ✓ Thank you !
- ✓ Questions

ahcosio@ubu.es
escorchado@usal.es
{mnavarro,vinglada}@dsic.upv.es