

Using Case-Based Reasoning in Autonomic Electronic Institutions

Eva Bou¹, Maite López-Sánchez², and J.A. Rodríguez-Aguilar¹

¹ IIIA, Artificial Intelligence Research Institute
CSIC, Spanish National Research Council, Campus UAB 08193 Bellaterra, Spain
{[ebm](mailto:ebm@iiia.csic.es),[jar](mailto:jar@iiia.csic.es)}@iiia.csic.es

² WAI, Volume Visualization and Artificial Intelligence, MAiA Dept.,
Universitat de Barcelona
maite@maia.ub.es

Abstract. Electronic institutions (EIs) define the rules of the game in agent societies by fixing what agents are permitted and forbidden to do and under what circumstances. Autonomic Electronic Institutions (AEIs) adapt their regulations to comply with their goals despite coping with varying populations of self-interested external agents. This paper presents a self-adaptation model based on Case-Based Reasoning (CBR) that allows an AEI to yield a dynamical answer to changing circumstances.

1 Introduction

The growing complexity of advanced information systems in the recent years, characterized by being distributed, open and dynamical, has given rise to interest in the development of systems capable of self-management. Such systems are known as self-* systems [1], where the * sign indicates a variety of properties: self-organization, self-configuration, self-diagnosis, self-repair, etc. A particular approximation to the construction of self-* systems is represented by the vision of autonomic computing [2], which constitutes an approximation to computing systems with a minimal human interference. Some of the many characteristics of an autonomic system are: it must configure and reconfigure itself automatically under changing (and unpredictable) conditions; it must aim at optimizing its inner workings, monitoring its components and adjusting its processing in order to achieve its goals; it must be able to diagnose the causes of its eventual malfunctions and repair itself; and it must act in accordance to and operate into a heterogeneous and open environment.

Electronic Institutions (EIs) [3] have been proved to be valuable to regulate open agent systems. EIs define the rules of the game by fixing what agents are permitted and forbidden to do and under what circumstances. We have defined Autonomic Electronic Institutions (AEIs) as an EI with autonomic capabilities that allows it to adapt its regulations to comply with institutional goals despite varying agent's behaviours [4]. Thus, an AEI has to self-configure its regulations to accomplish its institutional goals. In previous work [4] we have learned

those regulations that best accomplished the institutional goals for a collection of simulated agent populations. This paper extends that work with a Case-Based Reasoning (CBR) approach that allows an AEI to self-configure its regulations for any agent population. Since our hypothesis is that populations that behave similarly can be regulated in a similar manner, the CBR approach helps us identify populations that behave similarly and subsequently retrieve the “control” parameters for an AEI to regulate it.

The paper is organized as follows. In section 2 we describe the notion of autonomic electronic institutions. Section 3 details the learning model that we propose and how an AEI uses CBR. Section 4 describes the case study employed as a scenario wherein we have tested our model. Section 5 provides some empirical results. Finally, section 6 summarizes some conclusions and related work and outlines paths to future research.

2 Autonomic Electronic Institutions

In general, an EI [3] involves different groups of agents playing different roles within scenes in a performative structure. Each scene is composed of a coordination protocol along with the specification of the roles that can take part in the scene.

According to [3] an EI is solely composed of: a dialogic framework (DF) establishing the common language and ontology to be employed by participating agents; a performative structure (PS) defining its activities along with their relationships; and a set of norms (N) defining the consequences of agents’ actions. We have extended the notion of EI to support self-configuration, in the sense of regulation adaptation. In this manner in [4] we incorporate notions of institutional goals and regulation configuration to define an *autonomic electronic institution* (AEI) as a tuple: $\langle PS, N, DF, G, P_i, P_e, P_a, V, \delta, \gamma \rangle$. Next, we only provide an intuitive idea about the elements of an AEI (further details can be found in [4]).

We assume that the main objective of an AEI is to accomplish its institutional goals (G). For this purpose, an AEI will adapt. We assume that the institution can observe the environment where agents interact (P_e), the institutional state of the agents participating in the institution (P_a), and its own state (P_i) to assess whether its goals are accomplished or not. Since an AEI has no access whatsoever to the inner state of any participating agent, only the *institutional (social) state* of an agent (P_a) can change. Therefore, each agent (a_i) can be fully characterized by his institutional state $P_{a_i} = \langle a_{i_1}, \dots, a_{i_m} \rangle$ where $a_{i_j} \in \mathbb{R}$, $1 \leq j \leq m$ is an observable value of agent a_i . Taking the traffic as an example of an AEI, the speed of a car could be an example of an observable value of an agent; the number of lanes could be an example of an observable value of the environment; and the number of polices the institution uses to control the cars could be an example of an observable value of the state of the institution.

Formally, we define the goals of an AEI as a finite set of constraints $G = \{c_1, \dots, c_p\}$ where each c_i is defined as an expression $g_i(V) \triangleleft [m_i, M_i]$ where $m_i, M_i \in \mathbb{R}$, \triangleleft stands for either \in or \notin . Additionally, g_i is a function over

the reference values $V = \langle v_1, \dots, v_q \rangle$, where each v_j results from applying a function h_j upon the agents' properties, the environmental properties and/or the institutional properties; $v_j = h_j(P_a, P_e, P_i)$, $1 \leq j \leq q$. In this manner, each goal is a constraint upon the reference values where each pair m_i and M_i defines an interval associated to the constraint. Continuing with the traffic example, an example of an institutional goal could be to minimize the number of accidents. Thus, the institution achieves its goals if all $g_i(V)$ values satisfy their corresponding constraints of belonging (at least to a certain degree) to their associated intervals. This is measured by means of a satisfaction function that computes the goal satisfaction degree (see [4] for further details).

The AEI definition includes the mechanisms to support the adaptation with the *normative transition function* (δ), and with the *PS transition function* (γ). An AEI employs norms to constrain agents' behaviors and to assess the consequences of their actions within the scope of the institution. We focus on norms describing prohibitions parametrically. So that each norm $N_i \in N$, $i = 1, \dots, n$, has a set of parameters $\langle p_{i,1}^N, \dots, p_{i,m_i}^N \rangle \in \mathbb{R}^{m_i}$. In fact, this parameters correspond to the variables in the *norm transition function* that will allow the institution to adapt. Continuing with the same traffic example, an example of a norm could be to stop always before to enter in an intersection and it norm can be parametrized by an associated fine applied if a car does not fulfill it. Notice that our AEI can not learn new norms, it only can adapt its norms by changing their parameters. On the other hand, adapting a PS involves the definition of a set of parameters whose values will be changed by the PS transition function. We define each scene in the performative structure, $S_i \in PS$, $i = 1, \dots, t$, as having a set of parameters $\langle p_{i,1}^R, \dots, p_{i,q_i}^R \rangle \in \mathbb{N}^{q_i}$ where $p_{i,j}^R$ stands for the number of agents playing role r_j in scene S_i . Thus, changing the values of these parameters means changing the performative structure.

The AEI definition includes the mechanisms to support the adaptation with the *normative transition function* (δ), and with the *PS transition function* (γ). We propose to use learning methods to learnt the *normative transition function* (δ), and the *PS transition function* (γ). Next section details the learning model used to adapt the AEI by changing those parameters.

3 Learning Model

Our aim is that at run-time an AEI could adapt its regulations to any population. We propose to learn the *norm transition function* (δ) and the *PS transition function* (γ) in two different steps in an overall learning process. In previous work [4] we have approached the first learning step, which corresponds to learn the best parameters for a set of predefined populations. In this work we focus on the second learning step: how to adapt the parameters to any population. As shown in Figure 1, in an initial step our AEI learns by simulation the best parameters for a collection of different agent populations. For each population of agents (A), the algorithm explores the space of parameter values (I_1, \dots, I_k) in search for the ones that lead the AEI to best accomplish its goals (G) for this population of

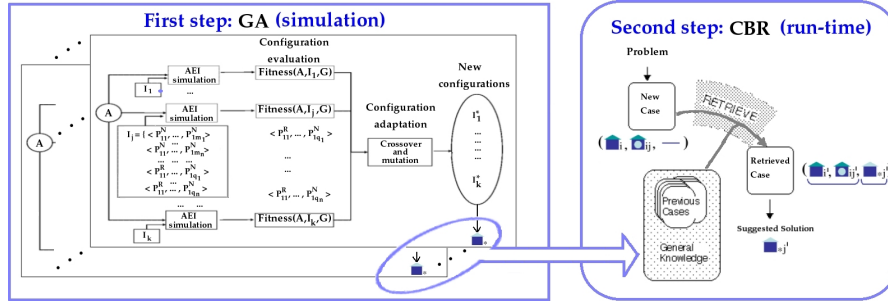


Fig. 1. Learning Model in two steps

agents. Afterwards, we propose to use a Case-Based Reasoning (CBR) approach as a second step because it allows the AIE to solve situations that have been learned previously. We assume that agent populations that behave in similar way caused similar situations that may require similar solutions. Thus, at a second step an AIE identifies, in run-time, those situations for which its goals are not accomplished and uses CBR to retrieve a solution (regulation parameters) from the most similar situation in the knowledge base.

3.1 Applying CBR

Case Based Reasoning (CBR) [5] is based on learning from experience. The idea is to search in the experience (memory) of the system for similar situations, called cases, and using the corresponding solution to solve the current problem. In general, a new problem in a CBR system is solved by retrieving similar cases, reusing the case solution, revising the reused solution, and retaining the new experience. In this work we focus our attention in the first step of the CBR cycle, namely the retrieve process. Nevertheless, before addressing it, it is necessary to choose a representation for cases.

Case Definition. The representation of cases is central to any CBR system. Cases must be represented based on the knowledge of the problem domain in order to choose the main features that better describe the case and thus that better help the processes involved in the CBR cycle. As to AIEs, we differentiate the following main features to be considered to represent cases:

- **AIE parameters’ values.** They represent the parameters’ values of some institution, namely the norm parameters’ values and the performative structure parameters’ values that an AIE uses for regulating agents.
- **Runtime behaviour.** They represent the global behaviour of the institution at runtime for some agent population when the institution uses the *AIE parameters’ values*.
- **Best AIE parameters’ values.** They represent the learned parameters’ values of the institution for the previous agent population. In other words:

the solution. Thus, they correspond to the parameters that the institution must apply in order to accomplish its institutional goals given both previous AEI parameters' values and runtime behaviour.

More precisely, regarding AEIs, we propose the definition of a case as a tuple $(N^p, PS^p, V, \text{pop}, N^{p*}, PS^{p*})$, where:

- (N^p, PS^p) stands for the AEI parameters' values:
 - N^p stands for the current norm parameters' values;
 - PS^p stands for the current performative structure parameters' values;
- (V, pop) stands for the runtime behaviour:
 - V stands for the current set of reference values;
 - pop stands for statistic data that characterises the behaviour of the agents' population at runtime¹;
- (N^{p*}, PS^{p*}) stands for the best AEI parameters' values:
 - N^{p*} : represents the best values for the norm parameters given the current norm parameters values (N^p) and the runtime behaviour (V, pop) ; and
 - PS^{p*} : represents the best values for the performative structure parameters given the current performative structure parameters values (PS^p) and the runtime behaviour (V, pop) .

Thus, a case represents how an AEI (using N^p as norm values and PS^p as performative structure values) regulating a population of agents (showing the runtime behaviour described by pop and V) should change its regulations (to the N^{p*} and the PS^{p*} values). Notice that each case is an entry of the *normative transition function* (δ) and the *PS transition function* (γ). That is, the set of all cases approximate both transition functions.

Similarity Function. In order to compare two cases we must define an appropriate similarity function based on our representation of cases. We use aggregated distance function to compute the degree of similarity between a new case C^i and a case C^j in the case base:

$$S(C^i, C^j) = w_1 \cdot s_AEI(C^i, C^j) + w_2 \cdot s_V(C^i, C^j) + w_3 \cdot s_pop(C^i, C^j) \quad (1)$$

where s_AEI corresponds to the distance of the AEI parameters' values (N^p , PS^p), s_V and s_pop correspond to the distance of the runtime behaviour (V, pop) , and $w_1, w_2, w_3 \leq 0$ are weighting factors such that $w_1 + w_2 + w_3 = 1$. The s_AEI , s_V and s_pop distance functions are computed as the distance average of their attributes. To assess the distance between the values of an attribute we use:

$$sim(attr^i, attr^j) = \frac{|attr^i - attr^j|}{max(attr) - min(attr)} \quad (2)$$

where $min(attr)$ and $max(attr)$ correspond to the limits of the interval of values of the attribute considered in the domain.

¹ Notice that this data corresponds to reference values.

The Retrieval Process. In order to retrieve the most similar case to the problem case C^i without comparing all cases in the case base, we propose to perform this process in two steps:

1. Compare the AEI parameters' values, (N^p, PS^p) , of the problem case C^i with the collection of all the AEI parameters' values in the case base using s_{AEI} and select the set of AEI parameters' values that best match.
2. Access the set of examples in the case base with these AEI parameters' values. Afterwards, we compare case C^i with these examples and select the case that best matches it based on distance function S .²

We use the first step with the idea that the most similar case must have similar AEI values because the runtime behaviour depends a lot of the AEI parameters' values. In fact, this is our hypothesis since we want to change the AEI parameters' values to change in some way the population behaviour and thus modify the runtime behaviour in order to achieve the institutional goals. The first step makes easy and fast the access to the most similar cases because we concentrate on only comparing the cases with similar AEI parameters' values. Thus, we do not need to compare all the cases of the case base. Moreover, we only need to compute once the distance function s_{AEI} for all cases with the same values of AEI parameters' values.

4 Case Study: Traffic Control

In order to test our model, we have considered and implemented the Traffic Regulation Authority as an Autonomic Electronic Institution, and cars moving along the road network as external agents interacting inside a traffic scene. Getting into more detail, we focus on a two-road junction where no traffic signals are considered. Therefore, cars must only coordinate by following the traffic norms imposed by the AEI. Our case study considers the performative structure to be a single traffic scene with two agent roles: one institutional role played by police agents; and one external role played by car agents.

We assume institutional agents to be in charge of detecting norm violations so that we will refer to them as police agents. The performative structure is parametrized by the number of agents playing the police role. Each police agent is able to detect only a portion of the total number of norm violations that car agents actually do. Norms within this normative environment are related to actions performed by cars. We consider two priority norms: the 'right hand-side priority norm', that prevents a car reaching the junction to move forward or to turn left whenever there is another car on its right; and the 'front priority norm', that applies when two cars reaching the junction are located on opposite lines, and one of them intends to turn left. Additionally, norms are parametrized by the associated penalties that are imposed to those cars refusing or failing

² Notice that we use a distance function as similarity function where low values imply high similarity.

to follow them. Cars do have a limited amount of points so that norm offenses cause points reduction. The institution forbids external agents to drive without points in their accounts.

In this work we focus on homogeneous populations where all agents in the population share the same behaviour. We propose to model each population based on three parameters (henceforth referred to as agent norm compliance parameters): $\langle \textit{fulfill_prob}, \textit{high_punishment}, \textit{inc_prob} \rangle$; where $\textit{fulfill_prob} \in [0, 1]$ stands for the probability of complying with norms that is initially assigned to each agent; $\textit{high_punishment} \in \mathbb{N}$ stands for the fine threshold that causes an agent to consider a fine to be high enough to reconsider the norm compliance; and $\textit{inc_prob} \in [0, 1]$ stands for the probability increment that is added to $\textit{fulfill_prob}$ when the fine norm is greater than the fine threshold ($\textit{high_punishment}$). Car agents decide whether to comply with a norm based on their norm compliance parameters along with the percentage (between 0 and 1) of police agents that the traffic authority has deployed on the traffic environment. To summarise, agents decide whether they keep on moving –regardless of violating norms– or they stop –in order to comply with norms– based on a probability that is computed as:

$$\textit{prob} = \begin{cases} \textit{police} \cdot \textit{fulfill_prob} & \textit{fine} \leq \textit{high_punishment} \\ \textit{police} \cdot (\textit{fulfill_prob} + \textit{inc_prob}) & \textit{fine} > \textit{high_punishment} \end{cases} \quad (3)$$

The institution can observe the external agents’ institutional properties (P_a) along time. Considering our road junction case study, we identify different reference values, $V = \langle \textit{col}, \textit{off}, \textit{crash}, \textit{block}, \textit{expel}, \textit{police} \rangle$ where \textit{col} indicates total number of collisions for the last t_w ticks ($0 \leq t_w \leq t_{now}$), \textit{off} indicates the total number of offenses accumulated by all agents for the last t_w ticks, \textit{crash} counts the number of cars involved in accidents for the last t_w ticks, \textit{block} describes how many cars have been blocked by other cars for the last t_w ticks, \textit{expel} indicates the number of cars that have been expelled out of the environment due to running out of points for the last t_w ticks, and finally, \textit{police} indicates the percentage of police agents that the institution deploys in order to control the traffic environment.

The institution tries to accomplish its institutional goals by specifying the penalties of both priority norms and by specifying how many police agents should be deployed in the traffic scene. In this work we focus on four institutional goals: (i) minimize the number of collisions; (ii) minimize the number of offenses; (iii) minimize the number of expelled cars; (iv) and minimize the percentage of police agents to deploy to control the traffic environment. Notice, though, that these offences do not refer to offences detected by police agents but to the real offences that have been actually carried out by car agents.

5 Empirical Evaluation

As a proof of concept of our proposal in section 3, we extend the experimental setting for the traffic case study employed in [4]. The environment is modeled as

a 2-lane road junction and populated with 10 homogeneous cars (endowed with 40 points each). Cars correspond to external agents without learning skills. They just move based on their random trajectories and the probability of complying with a norm (based on the function defined in (3)). During each discrete simulation, the institution replaces those cars running out of points by new cars, so that the cars' population is kept constant.

The four institutional goals, related to the *col*, *off*, *expel* and *police* reference values, are combined in a weighted addition, with weights 0.4, 0.4, 0.1 and 0.1 respectively. Thus, the first two goals are considered to be more important. The goal satisfaction is measured by combining the degree of satisfaction of these four institutional goals.

Table 1. Agent populations employed to generate the case base

Populations	Pop. 1	Pop. 2	Pop. 3	Pop. 4	Pop. 5	Pop. 6	Pop. 7
<i>fulfill_prob</i>	0.5	0.5	0.5	0.5	0.5	0.5	0.5
<i>high_punishment</i>	0	3	5	8	10	12	14
<i>inc_prob</i>	0.4	0.4	0.4	0.4	0.4	0.4	0.4
<i>fine_{right}*</i>	2	5	8	11	13	14	15
<i>fine_{front}*</i>	1	4	6	9	12	13	15
<i>police*</i>	1	1	1	1	1	1	1

As mentioned in section 3, (during training period) an AEI generates an initial base of cases from simulations of a set of prototypical populations. Following the tuple case definition introduced in section 3.1, $(N^P, PS^P, V, pop, N^{P*}, PS^{P*})$, we define a case C^i in this scenario as follows:

- $N^P = (fine_{right}, fine_{front})$ are the values of both norms' parameters;
- $PS^P = (police)$ is the value of the performative structure parameter;
- $V = (col, crash, off, block, expel)$ are the reference values;
- $pop = (mean_off, median_off, mean_frequency_off, median_frequency_off)$ contains the mean number of offenses, the median number of offenses, the mean of the frequency of offenses, and the median of the frequency of offenses carried out by agents for the last t_w ticks ($0 \leq t_w \leq t_{now}$);
- $N^{P*} = (fine_{right}^*, fine_{front}^*)$ are the best values for both norms' parameters;
- $PS^{P*} = (police^*)$ is the best value for the parameter of the performative structure.

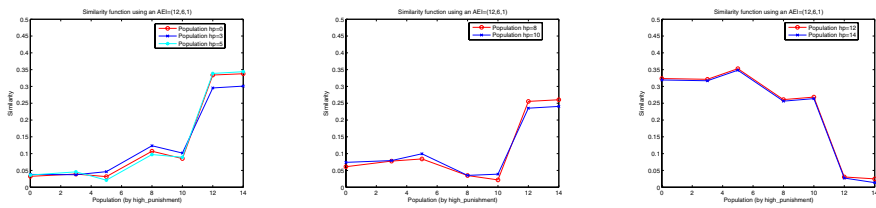
Table 1 shows the seven populations we have considered to generate the case base. They are characterized by their norm compliance parameters, being *fulfill_prob* = 0.5 and *inc_prob* = 0.4 for all of them, whereas *high_punishment* varies from 0 to 14. The *fine_{right}**, *fine_{front}** and *police** values in Table 1 are taken to be the best AEI parameters' values (N^{P*}, PS^{P*}) .

5.1 Similarity Function

We use the aggregated distance function defined in (1) to compute the degree of similarity between two cases. We have set the weights as follows: $w_1 = 0.1$, $w_2 = 0.5$, and $w_3 = 0.4$. Regarding the attributes of the AEI parameters' values, the $fine_{front}$ and $fine_{right}$ values are in the interval $[0, 15]$, and the $police$ values are in the interval $[0, 1]$. However, the attributes of the runtime behaviour have not known limited values. We have established limits based on the values of the initial generated cases. Thus, we have established that the col values are in the interval $[0, 300]$, $crash \in [0, 400]$, $off \in [0, 500]$, $block \in [0, 200]$, $expel \in [0, 900]$, $mean_off \in [0, 30]$, $median_off \in [0, 30]$, $mean_frequency_off \in [0, 2]$, and $median_frequency_off \in [0, 2]$. Since the values of these attributes can be out of the proposed interval, we force distance to be 1 when $|attr^i - attr^j| > \max(attr) - \min(attr)$.

First of all, we have tested whether the distance function and the weights selected are suitable for the traffic domain. For this purpose, we have generated a little case base of only seven cases by simulating each population in Table 1. In order to create this case base, all seven populations have been run with the same AEI parameters: $fine_{right} = 12$, $fine_{front} = 6$ and $police = 1$. Afterwards, in order to test the distance function, we have created seven new cases simulating another time each population in Table 1 using the very same AEI parameters' values and have compared each one with the seven cases in the case base. Notice that two simulations of the same population using the very same AEI parameters' values do not create the very same case, because the runtime behaviour in both simulations may be similar but not exactly the same.

Figure 2 shows the results of testing similarities for the seven new cases with the seven ones in the base case. These seven new cases could be grouped by the population behaviour regarding the norm compliance. Since population of first three cases have an $high_punishment$ lower than both norms' fines, cars fulfill both norms (with probability 0.9). However, populations with $high_punishment$ 8 and 10 fulfill the right norm with probability 0.9 and the front norm with



(a) Three populations fulfill two norms with probability 0.9.

(b) Two populations fulfill the right norm with probability 0.9 and the front norm with probability 0.5.

(c) Two populations fulfill two norms with probability 0.5.

Fig. 2. Distance between populations when the AEI uses the same parameters values ($fine_{right} = 12$, $fine_{front} = 6$ and $police = 1$)

probability 0.5. Whereas, populations with *high_punishment* 12 and 14 fulfill both norms with probability 0.5. Figure 2 shows three charts corresponding to cases grouping by this behaviour. Thus, chart 2(a) shows the distance for the three first cases whose cars fulfill both norms with probability 0.9. We can see how these three cases are similar when compared with the seven cases in the case base, and also that the distance among them is less than with respect to other cases. Chart 2(b) shows the distance for cases using populations with *high_punishment* 8 and 10 whose cars fulfill the right norm with probability 0.9 and the front norm with probability 0.5. Chart 2(c) shows distance for cases using populations with *high_punishment* 12 and 14 whose cars fulfill both norms with probability 0.5. In the three charts we can see how distances are similar among cases created with populations that have similar behaviour. This figure also shows that if two different populations regulated by the very same norms behave in very similar manner, an AEI cannot differentiate them. This effect is because the AEI can only observe the external behaviour of populations. In any case, these results allow us to conclude that the proposed distance function is suitable. Next step is to test at run-time the proposed CBR approach.

5.2 Case Base

With the aim that at run-time the AEI could adapt its regulations to any population, we create a case base using populations in Table 1 and the corresponding best AEI parameters' values. In order to create the case base we have considered as AEI parameters' values $fine_{right} \in \{0, 3, 6, 9, 12, 15\}$, $fine_{front} \in \{0, 3, 6, 9, 12, 15\}$, and $police \in \{0.8, 0.9, 1\}$. Overall we have considered 108 different AEI parameters' values, as the result of combining $fine_{right}$, $fine_{front}$, and $police$ values. To create cases for our case base, we have simulated each population in Table 1 with all 108 AEI parameters' values, so we have generated a total of 756 cases for the seven agent populations. To create each case, we have simulated the traffic model during 2000 ticks. Once finished the simulation, we generate a case by saving the AEI parameters' values (N^p, PS^p) used in this simulation, the runtime behaviour for the 2000 ticks (V, pop) , and the best AEI parameters' values (N^{p*}, PS^{p*}) corresponding to the population used in this simulation.

5.3 Retrieving

We have designed an experiment to test the retrieval process and therefore our approach. That is, we want to test if at run-time the AEI is able to self-configure its parameters for different agent populations by using the proposed CBR approach. Since we are testing our approach and we are not interested in efficiency issues, we employ the traffic simulator to recreate a run-time execution. We launch simulations of 2000 ticks during 20 times, namely steps (overall 40000 ticks). At each step, once the simulation finishes, we check the goal satisfaction degree and change the AEI parameters' values using the CBR approach when required. Although this allows us to change the population of agents at any step

we have run the experiments using the same population in 20 simulations. For all experiments, the AEI starts with (0,0,0.8) parameters, that correspond to no fine for both norms and a deployment of 80% of police agents. Thus, we expect that the AEI starts with a low goal satisfaction degree (caused by the parameters it is using) and it will be able to retrieve a similar case with whose parameters that do increase the goal satisfaction degree.

At each step, we launch a simulation with a certain population of agents and when the simulation finishes, the AEI decides, based on the goal satisfaction, if it has to retrieve a case or not. If the goal satisfaction is greater than a threshold the AEI continues with the same parameters for a new simulation in the next step. Otherwise (when the goal satisfaction is lower than the threshold) we launch the CBR engine to retrieve a case of the case base (see section 5.2) in order to adapt the AEI parameters, namely to adapt the institution, its regulation. The threshold is computed as a desired goal satisfaction value G^* minus an epsilon value ϵ . In our experiments, we have set $\epsilon = 0.03$ and $G^* = 0.65$, which corresponds to the minimum of the best goal satisfaction degrees for our populations. The problem case is generated from the AEI parameters' values used in the last simulation and the runtime behaviour in the last 2000 ticks. The CBR system retrieves the most similar case and uses the best AEI parameters' values of the retrieved case for next simulation. Thus, the goal satisfaction degree can be computed again to check if it is necessary to define a new problem case.

We have used fifteen different populations to test our approach. Each population is characterized by their norm compliance parameters, being *fulfill_prob* = 0.5 and *inc_prob* = 0.4 for all of them, whereas *high_punishment* varies from 0 to 14. Notice that seven of them are the ones used for generating cases³ (when *high_punishment* \in {0, 3, 5, 8, 10, 12, 14}) whereas the AEI has no prior cases about of the other eight populations (when *high_punishment* \in {1, 2, 4, 6, 7, 9, 11, 13}). Figure 3 shows the results for fifteen populations, where each chart shows five populations. Each population is run three times. Thus, overall we have performed 45 experiments. For each experiment, the figure shows the goal satisfaction every 2000 ticks during 20 steps. On chart 3(a) we can see that at initial step the goal satisfaction is low (around 0.2) and how the AEI quickly rises it up and maintains it constant during the rest of steps (between 0.6 and 0.7). On chart 3(a) we can see how the goal satisfaction degree starts at 0.2 and quickly rises up to 0.6 – 0.7 with the initial case retrievals. This effect repeats on charts 3(b) and 3(c) on figure 3. That is, the AEI is able to adapt quickly its parameters in all experiments. However, we observe that for some populations (when *high_punishment* is 6, 10 and 12) the goal satisfaction does not remain constant. In particular, the goal satisfaction for one of the populations with *high_punishment* = 6 goes down three times (steps 8, 10 and 11) to values close to 0.2. These oscillations happen because given a population regulated by the very same AEI parameters' values there is a variability on the behaviour in different simulations, that causes a variability in goal satisfaction. Thus, it

³ Notice that use the same population does not mean use the same case because the runtime behaviour may be similar in both cases but not exactly the same.

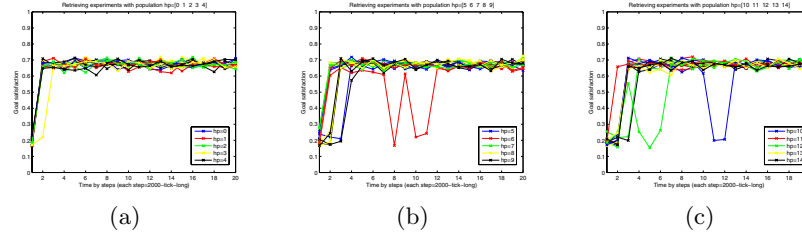


Fig. 3. Goal satisfaction for fifteen populations. (a) Populations with *high_punishment* $\in \{0, 1, 2, 3, 4\}$; (b) Populations with *high_punishment* $\in \{5, 6, 7, 8, 9\}$; and (c) Populations with *high_punishment* $\in \{10, 11, 12, 13, 14\}$.

sometimes occurs that because of this variability the goal satisfaction drops below the threshold and causes to restart the retrieval process. After this, the AEI stabilizes quickly again the goal satisfaction degree.

In order to estimate the error caused by these oscillations we have computed the percentage of simulations with a goal satisfaction greater than the threshold (0.62). At first step all experiments have a goal satisfaction less than the threshold. At second step a 52% of experiments (23 of 45) have a goal satisfaction greater than it. The percentage goes up to 89% (40 of 45) at third step and to 95% to the fourth. That is, in our experiments, the AEI needs four simulations to adapt itself in a correct manner to a 95% of new cases. At the rest of simulations (from simulation 5 to simulation 20) the average of the percentage of experiments with a goal satisfaction greater than the threshold is around 98%. That is, there is an error around the 2% caused by the oscillations. In any case, we can conclude that the AEI is able to adapt to the populations, that is with the initial cases retrievals the AEI is able to adapt its parameters to accomplish its goals for each population.

6 Discussion and Future Work

Within the area of Multi-Agent Systems, adaptation has been usually envisioned as an agent capability where agents learn how to reorganise themselves. Along this direction, in [6] Gasser and Ishida present a general distributed problem-solving model which can reorganize its architecture; and Horling et al. [7] propose an approach where the members adapt their own organizational structures at runtime. The fact that adaptation is carried out by the agents composing the MAS is the most significant difference with the approach presented in this paper. On the other hand, it has been long stated [8] that agents working in a common society need norms to avoid and solve conflicts, make agreements, reduce complexity, or to achieve a social order. Most research in this area consider norm configuration at design time [9] instead of at run-time as proposed in this paper. Regarding the traffic domain, MAS has been previously applied to it. For example, Camurri et al. [10] propose two field-based mechanisms to control

cars and traffic-lights in order to manage to avoid deadlocks and congestion. Additionally, Case-Based Reasoning has been applied before in multi-agent systems where agents use different CBR approaches to individual learning and to cooperative learning for distributed systems [11,12].

This paper presents a Case-Base Reasoning approach as an extension of previous work which allows an AEI to self-configure its regulations. We have presented the initial step towards a Case-Based Reasoning system, centering our work on the retrieval and usage processes. We have proposed a case description and the distance function to be used by a generic AEI. We have tested the retrieval process of our approach in the traffic AEI case study, where the AEI learns two traffic norms and the number of institutional agents in order to adapt the norms and the performative structure to dynamical changes of agent populations.

Preliminary results in this paper are promising but they show some oscillations of the goal satisfaction degrees for some populations. Although, the computed error is low (around 2%), currently we are tuning the function used to compute the goal satisfaction and the threshold value in order to reduce the error and do it less sensitive to the variability. Once solved this, we plan to continue our experiments on the retrieval process by changing the populations between simulations. We also plan to continue on finishing the learning by focusing our work in the other CBR processes. As future work, and since this basically represents a centralized scenario, we plan to develop a more complex traffic network, allowing us to propose a decentralized approach where different areas (i.e., junctions) are regulated by a distributed institution.

Acknowledgments. This work was partially funded by the Spanish Education and Science Ministry as part of the IEA (TIN2006-15662-C02-01) and the 2006-5-0I-099 projects. The first author enjoys an FPI grant (BES-2004-4335) from the Spanish Education and Science Ministry.

References

1. Luck, M., McBurney, P., Shehory, O., Willmott, S.: Agentlink Roadmap (2005), Agenlink.org
2. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. *IEEE Computer* 36(1), 41–50 (2003)
3. Esteva, M.: Electronic Institutions: From specification to development. IIIA, Ph.D. Monography 19 (2003)
4. Bou, E., López-Sánchez, M., Rodríguez-Aguilar, J.A.: Towards self-configuration in autonomic electronic institutions. In: Noriega, P., Vázquez-Salceda, J., Boella, G., Boissier, O., Dignum, V., Fornara, N., Matson, E. (eds.) COIN 2006. LNCS (LNAI), vol. 4386, pp. 220–235. Springer, Heidelberg (2007)
5. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun* 7(1), 39–59 (1994)
6. Gasser, L., Ishida, T.: A dynamic organizational architecture for adaptive problem solving. In: Proc. of AAAI-91, Anaheim, CA, pp. 185–190 (1991)
7. Horling, B., Benyo, B., Lesser, V.: Using Self-Diagnosis to Adapt Organizational Structures. In: Proceedings of the 5th International Conference on Autonomous Agents, pp. 529–536 (2001)

8. Conte, R., Falcone, R., Sartor, G.: Agents and norms: How to fill the gap? *Artificial Intelligence and Law* (7), 1–15 (1999)
9. Fitoussi, D., Tennenholtz, M.: Choosing social laws for multi-agent systems: Minimality and simplicity. *Artificial Intelligence* 119(1-2), 61–101 (2000)
10. Camurri, M., Mamei, M., Zambonelli, F.: Urban traffic control with co-fields. In: *Proc. of E4MAS Workshop at AAMAS*, vol. 2006, pp. 11–25 (2006)
11. Plaza, E., Ontañón, S.: Cooperative multiagent learning. In: Alonso, E., Kudenko, D., Kazakov, D. (eds.) *AAMAS 2000 and AAMAS 2002*. LNCS (LNAI), vol. 2636, pp. 1–17. Springer, Heidelberg (2003)
12. Ros, R., Veloso, M.: Executing Multi-Robot Cases through a Single Coordinator. In: *Proc. of Autonomous Agents and Multiagent Systems*, pp. 1264–1266 (2007)