Self-adaptation in Autonomic Electronic Institutions through Case-Based Reasoning

Eva Bou[1] and Maite López-Sánchez[2] and J. A. Rodríguez-Aguilar[1]

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

[2] WAI, Volume Visualization and Artificial Intelligence, MAiA Dept., Universitat de Barcelona, email: 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. We present a self-adaptation model based on Case-Based Reasoning (CBR) that allows an AEI to yield a dynamical answer to changing circumstances. In order to demonstrate adaptation empirically, we consider a traffic control scenario where the agent population changes. Within this setting, we demonstrate statistically that an AEI is able to adapt at run-time by means of CBR.

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 autonomic systems 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 Institution (AEI) 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 [5] 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 and statistical 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.

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 can be fully characterized by his institutional state $P_a = \langle a_{i_1}, \ldots, a_{i_m} \rangle$ where $a_{i_j} \in \mathbb{R},\ 1 \leq j \leq m$.

Formally, we define the goals of an AEI as a finite set of constraints $G = \{c_1, ..., c_p\}$ where each c_i is defined as an expression $g_i(V) \lhd [m_i, M_i]$ where $m_i, M_i \in \mathbb{R}$, $\lhd$ stands for either $\in$ or $\notin$. Additionally, g_i is a function over the reference values $V = \langle v_1, \ldots, v_q \rangle$, 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. Thus, the institution achieves its goals if all $g_i(V)$ values satisfy their corresponding constraints of belonging (at

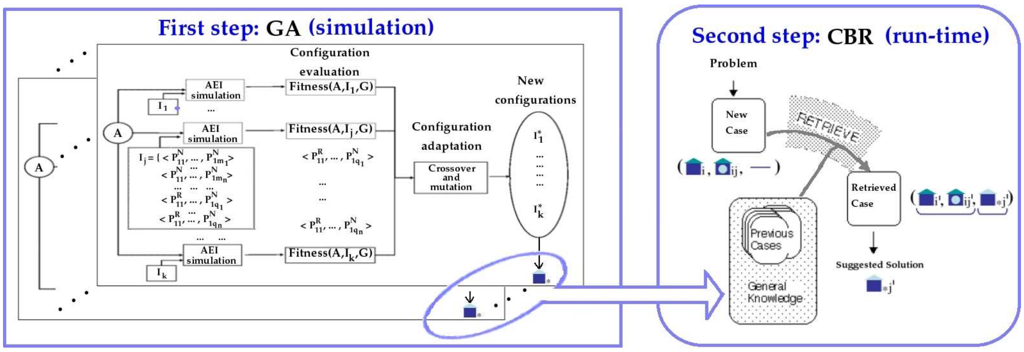

Fig. 1. Learning Model in two steps.

list to a certain degree) to their associated intervals. This is measured by means of a satisfaction function that computes the goal satisfaction degree (see [5] 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 $N_i \in N$, $i = 1, \ldots, n$, has a set of parameters $\langle p^N_{i,1}, \ldots, p^N_{i,m_i} \rangle \in \mathbb{R}^{m_i}$. In fact, these parameters correspond to the variables in the *norm transition function* that will allow the institution to adapt. 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, $S_i \in PS$, $i = 1, \ldots, t$, as having a set of parameters $\langle p^R_{i,1}, ..., p^R_{i,q_i} \rangle \in \mathbb{N}^{q_i}$ where $p^R_{i,j}$ 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.

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 [5] 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 ($I_1, .., I_k$) in search for the ones that lead the AEI to best accomplish its goals (G) for this population of agents. Afterwards, we propose to use a Case-Based Reasoning (CBR) approach as a second step because it allows the AEI to solve situations that have been learned previously. We assume that similar agent populations behave in similar way, causing similar situations that may require similar solutions. Thus, at a second step an AEI 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) [6] 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 AEIs, we differentiate the following main features to be considered to represent cases:

- **AEI parameters' values.** They represent the parameters' values of some institution, namely the norm parameters' values and the performative structure parameters' values that an AEI 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 *AEI parameters' values.*
- **Best AEI 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,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 runtime[1];

– (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 approximates 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 similarity 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 similarity of the AEI parameters' values (N^p, PS^p), s_V and s_pop correspond to the similarity 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 similarity functions are computed as the similarity average of their attributes. To assess the similarity 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.

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 similarity function S.

[1] Notice that this data corresponds to reference values.

We use 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 similarity s_AEI for all cases with the same values of AEI parameters' values.

4 Case Study: Traffic Control

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 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): $\langle fulfill\_prob$, $high\_punishment$, $inc\_prob\rangle$; where $fulfill\_prob \in [0,1]$ stands for the probability of complying with norms that is initially assigned to each agent; $high\_punishment \in \mathbb{N}$ stands for the fine threshold that causes an agent to consider a fine to be high enough to reconsider the norm compliance; and $inc\_prob \in [0,1]$ stands for the probability increment that is added to $fulfill\_prob$ when the fine norm is greater than the fine threshold (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:

$$prob = \begin{cases} police \cdot fulfill\_prob & fine \leq high\_punishment \\ police \cdot (fulfill\_prob + inc\_prob) & fine > high\_punishment \end{cases} \quad (3)$$

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

Finally, 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.

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 [5]. 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	Pop1	Pop2	Pop3	Pop4	Pop5	Pop6	Pop7
$fulfill\_prob$	0.5	0.5	0.5	0.5	0.5	0.5	0.5
$high\_punishment$	0	3	5	8	10	12	14
inc_prob	0.4	0.4	0.4	0.4	0.4	0.4	0.4
$fine^*_{right}$	2	5	8	11	13	14	15
$fine^*_{front}$	1	4	6	9	12	13	15
$police^*$	1	1	1	1	1	1	1

5.1 Case Base

As mentioned in section 3, (during training period) an AEI generates an initial base of cases from simulations of a set of prototypical populations. 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. Table 1 also shows the best AEI parameters' values (N^*, PS^*) for each one, that is the $fine^*_{right}$, $fine^*_{front}$ and $police^*$ values the institution has learned by using genetic algorithms for each population.

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 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.

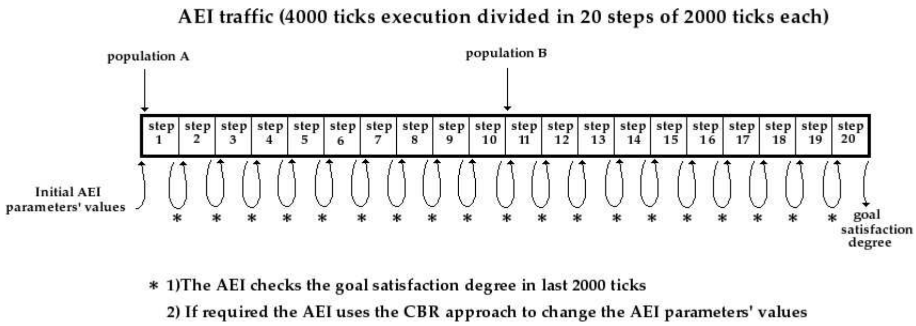

Fig. 2. Scheme of an AEI traffic experiment.

5.2 Similarity function

We use the aggregated similarity 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 similarity to be 1 when $|attr^i - attr^j| > max(attr) - min(attr)$.

5.3 Retrieving

We have designed an experiment to test the retrieval process. 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. Additionally, we want to test if the CBR approach helps the AEI to adapt its parameters when a change of agent population occurs at run-time. Figure 2 shows an scheme of an experiment using the traffic simulator. As Figure 2 shows, each experiment is composed of 40000 ticks which we divide in 20 steps. At each step (every 2000 ticks) the AEI checks its goal satisfaction degree and, if required, changes its parameters' values. Although this allows us to change the population of agents at any step, we have run the experiments changing only once the population of agents. We start the traffic simulator with a certain population of agents which remains during first 10 steps, at step 11 (tick 20001) we always change the agent population to Pop7 (see Table 1) which remains until last step. Thus, we are simulating a run-time change of population.

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,

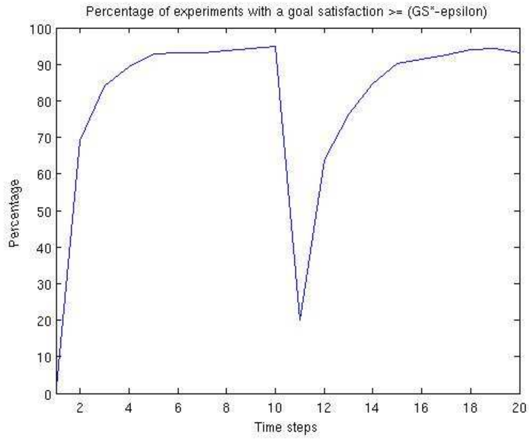

Fig. 3. Percentage of correct experiments.

we expect the AEI to start with a low goal satisfaction degree (caused by the parameters it is using since no population will follow norms) and to be able to retrieve a similar case whose parameters will increase its goal satisfaction degree. We also expect that in step 11 the AEI will obtain a low goal satisfaction degree (caused by the change of agent population) and it will be able to increase the goal satisfaction degree again by retrieving a case with more suitable parameters.

As mentioned above, when each step finishes the AEI decides, based on the goal satisfaction of last 2000 ticks, 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 2000 ticks in next step. Otherwise (when the goal satisfaction is lower than the threshold) the AEI launches its CBR engine to retrieve a case of the case base in order to adapt its parameters. 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 populations in Table 1. The problem case is generated from the values in last 2000 ticks in the same way as when creating the cases. The CBR system retrieves the most similar case so that the AEI uses its solution (i.e., the best paraemters' values) for next step. 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 agent 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. Seven of them are the ones used for generating cases[2] (with $high\_punishment \in \{0, 3, 5, 8, 10, 12, 14\}$) whereas the AEI has no prior cases

[2] Notice that using the same agent population does not imply use the same case because the runtime behaviour may result in different reference values.

about the remaining eight populations (with *high_punishment* $\in \{1, 2, 4, 6, 7, 9, 11, 13\}$).

Table 2. Number of experiments initially stabilized in first 10 steps.

Steps	1	2	3	4	5	6	7	8	9	10	Not stabilized
Stabilized	0	518	153	36	19	9	5	2	1	1	6
Percentage stabilized	0	69	20.4	4.8	2.5	1.2	0.7	0.3	0.1	0.1	0.8

Table 3. Number of experiments initially stabilized in last 10 steps.

Steps	11	12	13	14	15	16	17	18	19	20	Not stabilized
Stabilized	157	332	102	70	46	15	15	8	4	1	0
Percentage stabilized	20.9	44.2	13.6	9.3	6.1	2	2	1	0.5	0.1	0

In order to obtain statistical results we have run each experiment 50 times for each agent population. Thus, overall we have performed 750 experiments. Figure 3 shows the percentage of correct experiments at each step for all experiments. We consider that an experiment is correct when the goal satisfaction degree value in last 2000 ticks (i.e., last step) is equal or greater than the threshold (G^*-*epsilon* $= 0.62$). In Figure 3 we can see how the percentage of correct experiments starts to stabilize around 90 percent between steps 5 and 6. Due to the change of population, the percentage drops drastically at step 11, but it is able to recover and it stabilizes again between steps 15 and 16. Thus, we can state that in six steps our AEI is able to adapt its parameters to approximately 90 percent of experiments. We also can state that when the change of population occurs in six steps our AEI is also able to adapt to the new population at approximately 90 percent of experiments.

In order to analyse statistically how many steps the AEI needs to stabilize we have separated first 10 steps from last 10 steps. That is, we compute separately the initial stabilization and the stabilization when the population changes. Focusing on first 10 steps, our aim is to compute the required number of steps to reach stabilization for a given initial population of agents. For each experiment we have computed the first step for which the AEI has obtained a goal satisfaction equal or greather than the threshold (that is, it has stabilized for the first time). Table 2 shows the number of experiments that have stabilized for the first time at each step, together with the corresponding percentage they represent. Notice that not stabilizing a population means that the AEI is not able to retrieve a case whose parameters, when applied to the current population, yield to an AEI's satisfaction degree higher than the fixed threshold. Therefore, the AEI

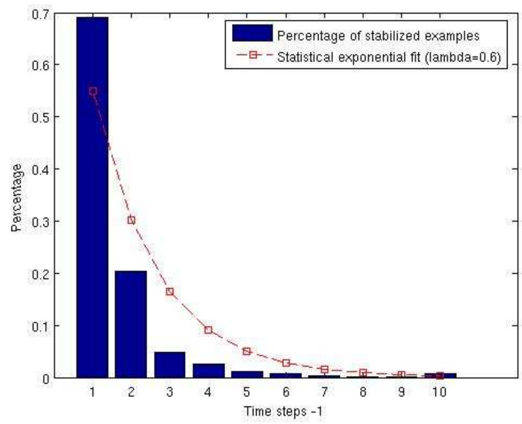

Fig. 4. Percentage of experiments initially stabilized in first 10 steps.

keeps retrieving cases to try to adapt better to the given population. Notice that there are some experiments which have not been stabilized at any of the first 10 steps. We have performed the chi-square test to test if our data (the percentage of experiments initially stabilized) follows an exponential distribution. The chi-square test allows us to say that our data follows an exponential distribution with lambda = 0.6 (chi-square value=9.73, with a significance level of 0.01 and 4 degrees of freedom). Figure 4 shows the stabilized percentage and the fitted exponential.[3] Finally, in order to compute the number of steps the AEI needs to stabilize, we have computed the accumulate function of the fitted exponential and have found that at step 5 the AEI stabilizes for the first time the 95 percent of population -in the statistical sense- (significance level of 0.01).

Analogously, in order to compute the number of steps the AEI needs to stabilize after an agent population change, we have used the same statistical test with the data at steps from 11 to 20. Table 3 shows how many experiments the AEI has initially stabilized in second 10 steps, toghether with the percentage of stabilized experiments. Notice that at step 11 there is an initial percentage of experiments stabilized. The change of agent population does not affect these experiments the AEI does not require to perform any adaptation for them. Thus, we have not used these experiments as data for our statistical test. The results of the chi-square test applied to the percentage of experiments initially stabilized in steps from 12 to 20 (in Table 3) point out that our data follows an exponential distribution with lambda=0.55 (chi-square value=4,99, with a significance level of 0.01 and 8 degrees of freedom). The accumulate function of the fitted

[3] Notice that we have performed a correction in the steps because in first step there is a zero percentage (this is due to the experimental setting: during first step the AEI only observes and its initial parameters have values that prevent any population of fulfilling its goals).

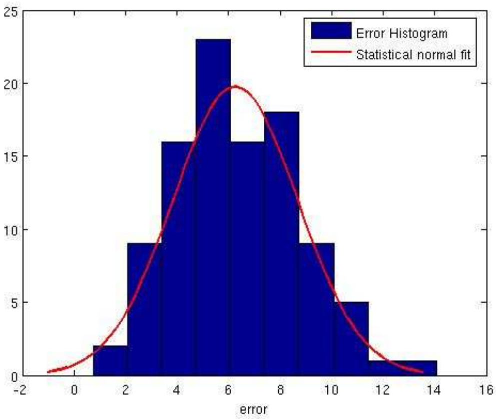

Fig. 5. Histogram of the error computed in 100 subsets of 30 experiments.

exponential shows that the AEI needs 6 steps (step 16) to stabilize for the first time the 95 percent of population -in the statistical sense- when there is an agent population change to Pop7 (significance level of 0.01).

In Figure 3 we can see that none of the steps reaches the 100 percent of correct experiments. That is, the AEI performs an error because there is always a percentage of experiments where the goal satisfaction degree is less than the threshold. In order to analyse statistically this error we have used the frequentist classical statistical method of replicated measurements [7]. We have computed replicated measurements of the error using our previous 750 experiments. From the 750 experiments we have chosen 100 times a random subset of 30 experiments. We have calculated the error in each subset of 30 experiments. The error is computed as the percentage of experiments with error at steps 6,7,8,9,10,17,18,19 and 20 (steps where we have seen that the AEI is stabilized). Figure 5 shows the histogram results of the error computed in the 100 subsets. Figure 5 also shows the fitted normal distribution of the data. The mean of the error in the data is 6.3, however the fitted normal distribution allows us expect that the error on population -in the statistical sense- will be in the range [1.5 11.01] (with a significance level of 0.05). In other words, with 97.5 percent confidence the maximum error is 11.01. In conclusion, our traffic AEI is able to stabilize in few steps, however once it has stabilized it does a percentage of error (i.e., it obtains a goal satisfaction degree less than the fixed threshold).

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, Hübner et al. [8] propose a model for controlling adaptation by using the $\mathcal{M}$OISE+ organization model, and Gâteau et al. [9] propose $\mathcal{M}\text{OISE}^{Inst}$ as an extension of $\mathcal{M}$OISE+ as an institution organization specification of the rights and duties of agents' roles. In both models agents adapt their MAS organization to both environmental changes and their own goals. In [10] Gasser and Ishida present a general distributed problem-solving model which can reorganize its architecture; and Horling et al. [11] 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. In our approach there is indeed a group of internal agents who can punish external agents but the reorganization is carried out by the institution, instead of by the agents.

On the other hand, it has been long stated [12] 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 [13] instead of at run-time as proposed in this paper. In this manner, Fitoussi and Tennenholtz [13] select norms at design stages by proposing the notions of minimality and simplicity as selecting criteria. They study two basic settings, which include Automated-Guided-Vehicles (AGV) with traffic laws, by assuming an environment that consists of (two) agents and a set of strategies available to (each of) them. From this set, agents devise the appropriate ones in order to reach their assigned goals without violating social laws, which must be respected. Our approach differs from it because we do not select norms at design stages. Previously, Sierra et al. [14] used evolutionary programming techniques in the SADDE methodology to tune the parameters of the agent populations that best accomplished the global properties specified at design stages by the electronic institution. Their approach differs from our approach because they search the best population of agents by a desired institution and we adapt the institution to the population of agents.

Regarding the traffic domain, MAS has been previously applied to it [15] [16], [17]. For example, Camurri et al. [18] propose two field-based mechanisms to control cars and traffic-lights in order to manage to avoid deadlocks and congestion. Traffic has been also widely studied outside the scope of MAS, for example, the preliminary work by [19] used Strongly Typed Genetic Programming (STGP) to control the timings of traffic signals within a network of orthogonal intersections. Their evaluation function computed the overall delay.

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 [20, 21]. For example, Ros and Veloso [22] propose a case-based coordination mechanism where they use a case-based reasoning approach to coordinate a multi-robot system.

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 similarity 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. We have done statistical analysis about the time (in steps of 2000 ticks) the AEI needs to adapt its parameters to a high percentage of populations. We also have computed a statistical measure of the error. Preliminary results in this paper are promising, they show that our traffic AEI can adapt to a new population in five steps. They also show that our traffic AEI is able to adapt its parameters when a change of populations occurs. Results also show our traffic AEI does a percentage of error, however the maximum statistical error is low (around 11%). We plan to continue our experiments on the retrieval process by changing more populations between steps and using heterogeneous populations. We also plan to continue on finishing the learning by focusing our work in the other CBR processes (e.g., revise and retain). 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. The authors want to thank Pere García and Lluis Godo for their comments and advice. 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. Research partially supported by the Spanish project "Agreement Technologies" (CONSOLIDER CSD2007-0022, INGENIO 2010). 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. Agenlink.org (2005)
2. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. IEEE Computer **36**(1) (2003) 41–50
3. Esteva, M.: Electronic Institutions: from specification to development. IIIA PhD Monography. Vol. 19 (2003)
4. Bou, E., López-Sánchez, M., Rodríguez-Aguilar, J.A.: Towards self-configuration in autonomic electronic institutions. In: COIN 2006 Workshops. Number LNAI 4386, Springer (2007) 220–235
5. Bou, E., López-Sánchez, M., Rodríguez-Aguilar, J.A.: Adaptation of autonomic electronic institutions through norms and institutional agents. In: Engineering Societies in the Agents World. Number LNAI 4457, Springer (2006) 300–319
6. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Commun. **7**(1) (1994) 39–59
7. Cohen, P.R.: Empirical methods for artificial intelligence. IEEE Expert: Intelligent Systems and Their Applications **11**(6) (1996) 88

8. Hübner, J.F., Sichman, J.S., Boissier, O.: Using the $\mathcal{M}$oise+ for a cooperative framework of mas reorganisation. In: LNAI - Proc. of the 17th Brazilian Symposium on Artificial Intelligence (SBIA'04). Volume 3171., Springer (2004) 506–515
9. Gâteau, B., Boissier, O., Khadraoui, D., Dubois, E.: Moiseinst: An organizational model for specifying rights and duties of autonomous agents. In Gleizes, M.P., Kaminka, G.A., Nowé, A., Ossowski, S., Tuyls, K., Verbeeck, K., eds.: EUMAS, Koninklijke Vlaamse Academie van Belie voor Wetenschappen en Kunsten (2005) 484–485
10. Gasser, L., Ishida, T.: A dynamic organizational architecture for adaptive problem solving. In: Proc. of AAAI-91, Anaheim, CA (1991) 185–190
11. Horling, B., Benyo, B., Lesser, V.: Using Self-Diagnosis to Adapt Organizational Structures. Proceedings of the 5th International Conference on Autonomous Agents (2001) 529–536
12. Conte, R., Falcone, R., Sartor, G.: Agents and norms: How to fill the gap? Artificial Intelligence and Law (7) (1999) 1–15
13. Fitoussi, D., Tennenholtz, M.: Choosing social laws for multi-agent systems: Minimality and simplicity. Artificial Intelligence **119**(1-2) (2000) 61–101
14. Sierra, C., Sabater, J., Agustí-Cullell, J., García, P.: Integrating evolutionary computing and the sadde methodology (2003)
15. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K.: Mason: A new multi-agent simulation toolkit. In: Proceedings of the 2004 SwarmFest Workshop. (2004) 8
16. Dresner, K., Stone, P.: Multiagent traffic management: An improved intersection control mechanism. In: The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, New York, NY, ACM Press (2005) 471–477
17. Doniec, A., Espié, S., Mandiau, R., Piechowiak, S.: Dealing with multi-agent coordination by anticipation: Application to the traffic simulation at junctions. In: EUMAS. (2005) 478–479
18. Camurri, M., Mamei, M., Zambonelli, F.: Urban traffic control with co-fields. In: Proc. of E4MAS Workshop at AAMAS 2006. (2006) 11–25
19. Montana, D.J., Czerwinski, S.: Evolving control laws for a network of traffic signals. In: Genetic Programming 1996: Proceedings of the First Annual Conference, Stanford University, CA, USA, MIT Press (1996) 333–338
20. Plaza, E., Ontañón, S.: Cooperative multiagent learning. Adaptative Agents and Multi-Agent Systems **LNAI 2636** (2003) 1–17
21. Plaza, E., Arcos, J.L., Martín, F.J.: Cooperative case-based reasoning. In: ECAI '96: Selected papers from the Workshop on Distributed Artificial Intelligence Meets Machine Learning, Learning in Multi-Agent Environments, London, UK, Springer-Verlag (1997) 180–201
22. Ros, R., Veloso, M.: Executing Multi-Robot Cases through a Single Coordinator. In: Proc. of Autonomous Agents and Multiagent Systems. (2007) 1264–1266