Autonomic Electronic Institutions' Self-Adaptation in Heterogeneous Agent Societies

Eva Bou[1] and Maite López-Sánchez[2] and J. A. Rodríguez-Aguilar[1] and Jaime Simão Sichman[3]

[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
[3] LTI, Intelligent Techniques Laboratory, Computer Engineering Department, University of São Paulo, Brazil, email: jaime.sichman@poli.usp.br

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 rules to comply with their goals when regulating agent societies composed of varying populations of self-interested 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 populated by heterogeneous agents. Within this setting, we demonstrate statistically that an AEI is able to adapt to different heterogeneous agent populations.

1 Introduction

Electronic institutions (EIs) [1] have been proved to be valuable to regulate open agent systems. The idea behind EIs is to mirror the role traditional institutions play in the establishment of "the rules of the game" –a set of conventions that establish what agents are permitted and forbidden to do and under what circumstances. According to North [2] human institutions are not static; they may evolve over time by altering, eliminating or incorporating norms. Thereby, Bou et al. [3] have defined Autonomic Electronic Institutions (AEIs) as EIs with autonomic capabilities. The vision of autonomic computing [4] constitutes an approximation to computing systems with a minimal human interference. One main challenge of autonomic computing is the development of adaptive algorithms that allow a system to deal with changing, heterogeneous and open environments. In this way, AEIs use learning techniques to be capable of dynamically adapting their rules to social changes –changes in the agent's behaviours.

Notice that adaptation of Multi-Agent Systems (MAS) has been usually envisioned as an agent capability, where the agents composing the system learn how to adapt to the whole system in order to accomplish some organizational goals. It has been largely approached either as a coordination problem, a reorganization problem, and an organization formation problem [5] [6] [7] [8]. However, in AEIs the adaptation is addressed as a system capability: an AEI self-adapts to social changes. To the best of our knowledge, adaptation in EIs have been only previously approached by Sierra et al. [9]. However, they search the best population of agents that helps achieve some institutional goals instead of adapting the institution itself.

In this paper we focus in AEIs that use adaptive algorithms in order to adapt its regulations to comply with institutional goals despite varying agent's behaviours. We present a Case-Based Reasoning (CBR) approach that allows an AEI to self-adapt its regulations for any heterogeneous agent society. As heterogeneous agent societies we refer to agent populations where agents behave in different ways. Our CBR approach helps an AEI identify agent populations that behave similarly and subsequently retrieve the rules that best regulate them. We present a traffic case study to empirically test the proposed CBR approach. A statistical analysis of results allows us to conclude that an AEI is able to adapt to heterogeneous agent populations.

The paper is organized as follows. First, section 2 presents an overview of the related work. In section 3 we recall the notion of autonomic electronic institution as introduced by Bou et al. [3]. Section 4 describes the learning model that we propose and how an AEI uses CBR. Section 5 describes the case study employed as a scenario wherein we have tested AEI's adaptation. Section 6 provides some empirical results. Finally, section 7 draws some conclusions and sets paths to future research.

2 Related work

Within the area of multi-agent systems, system adaptation has been usually envisioned from an agent point of view where agents learn how to reorganise or coordinate themselves. Along this direction, Hübner et al. [10] propose a model for controlling adaptation by using the $\mathcal{M}$OISE+ organization model. Gâteau et al. [11] propose $\mathcal{M}\text{OISE}^{Inst}$ as an extension to $\mathcal{M}$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. Vecht et al. [12] also take an agent point of view, where agents must coordinate to accomplish some organizational goals. They introduce adjustable autonomy as a concept that allows dynamically switching between coordination types. They propose a way to implement adjustable autonomy in the agents in order to achieve the dynamic coordination. 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. Our approach is centered on an organizational point of view, since the institution adapts its regulations to accomplish its institutional goals. The most similar work to ours is [13], where Hoogendoorn addresses the dynamical adaptation of organizations to environment changes by translating the organizational model into a max flow network. Therefore, his purposes differ from ours because they only focus on adapting to environment fluctuation.

Regarding the traffic domain, it has been previously used in MAS research [14] [15]. For example, Bazzan et al. [16] discuss the effects of integrating co-evolution in traffic networks. They design the control to achieve the system goal via decentralized traffic lights (modeled as agents). In this framework, they test the use of different strategies when both road agents and traffic light agents adapt, each one having its own goal.

Additionally, Case-Based Reasoning has been applied before in MAS where agents use different CBR approaches to individual learning and to cooperative learning for distributed systems. For example, Ros and Veloso [17] propose a case-based coordination mechanism where they use a case-based reasoning approach to coordinate a multi-robot system.

3 Autonomic Electronic Institutions

Basically, an EI [1] is composed of three componets: a dialogical framework (DF) that represents the context of interaction between agents; a performative structure (PS) that defines the activities among the agents; and a set of norms defining the consequences of agent's actions. In general, an EI [1] involves different groups of agents playing different roles within scenes in a so called performative structure. Each scene is composed of a coordination protocol along with the specification of the roles that can take part in the scene. Nonetheless, the current definition of EI does not support the adaptation at runtime in the face of a changing environment and changing agents behaviour. In [3], Bou et al. extend the notion of EI to define *autonomic electronic institutions* (AEIs), namely EIs capable of dynamically adapting their rules under changing circumstances. They incorporate the notions of institutional goals and self-configuration to the definition of an EI to support self-adaptation, in the sense of regulation adaptation. Next, we just provide an intuitive idea about the elements of an AEI, which is defined as a tuple: $\langle PS, N, DF, G, P_i, P_e, P_a, V, \delta, \gamma \rangle$ (further details can be found in [3]).

The main objective of an AEI is to accomplish its institutional goals (G). For this purpose, an AEI will adapt its rules to the agents it regulates. The institution can observe some information to assess whether its goals are accomplished or not: the environment where agents interact (P_e), the institutional state of the agents participating in the institution ($P_a = \langle a_1, \ldots, a_n \rangle$), and its own state ($P_i$). An AEI is only aware of the *institutional (social) state* of a participating agent (a_i) because an AEI has no access whatsoever to the inner state of any agent. Therefore, an AEI characterizes each agent by his institutional state $a_i = \langle a_{i_1}, \ldots, a_{i_m} \rangle$ where $a_{i_j} \in \mathbb{R},\ 1 \leq j \leq m$. ¿From all this observable information an AEI obtains a set of reference values (V) required to determine the fulfillment of goals. Formally, these reference values are defined as a vector $V = \langle v_1, \ldots, v_q \rangle$, where each v_j results from applying a function h_j upon the information observed by the institution ($v_j = h_j(P_a, P_e, P_i),\ 1 \leq j \leq q$).

The goals of an AEI are fixed and are defined as a finite set of constraints. Formally, institutional goals are defined as $G = \{c_1, ..., c_p\}$ where each c_i is defined as an expression $g_i(V) \lhd [m_i, M_i]$ where $m_i, M_i \in \mathbb{R}$, and $\lhd$ stands for either $\in$ or $\notin$. Additionally, g_i is a function over the reference values. Thus, each goal is a constraint upon the reference values where each pair $(m_i,\ M_i)$ defines an interval associated to some constraint c_i. The institution achieves its goals if all $g_i(V)$ values satisfy their corresponding constraints of belonging (at 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 [3] for further details).

Finally, *norm transition function* (δ) and *PS transition function* (γ) are the mechanisms to support the adaptation of an AEI to the agent populations it regulates. ¿From all components that an AEI uses to constraint agents' behaviours, and AEI only adapts at runtime norms and roles. Norms are employed to constrain agents' behaviors and to assess the consequences of their actions within the scope of the institution. An AEI does not create or eliminate norms to adapt, but the adaptation of norms is made by using parameters values. Each norm $N_i \in N$ $(i = 1, \ldots, n)$ has a set of parameters $\langle p^N_{i,1}, \ldots, p^N_{i,m_i} \rangle \in \mathbb{R}^{m_i}$. The *norm transition function* (δ) changes the values norms' parameters to adapt. On the other hand, each scene in the performative structure, $S_i \in PS$, $i = 1, \ldots, t$, is defined 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. Notice that an AEI does not create or eliminate roles to adapt. Thus, adapting a PS involves changing the values of these parameters, whose values are changed by the *PS transition function* (γ).

Since both *norm transition function* (δ) and *PS transition function* may be difficult to define mathematically, we propose to use learning methods to learn them. Next section details the learning model used to adapt an AEI by changing the values of the norms' parameters and the PS' parameters.

4 Learning Model

With the aim of adapting AEI's regulations to any population at run-time, we propose to learn the *norm transition function* (δ) and the *PS transition function* (γ) in two different steps in an overall learning process. As Figure 1 shows, first learning step corresponds to learning the best parameters for a collection of predefined agent populations. Learning is performed using an evolutionary approach. Each individual (I_i) represents a specific AEI's parameter configuration. For a given population of agents (A), a genetic 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. Our AEI learns the best parameters for a collection of predefined agent populations by repeating a genetic algorithm

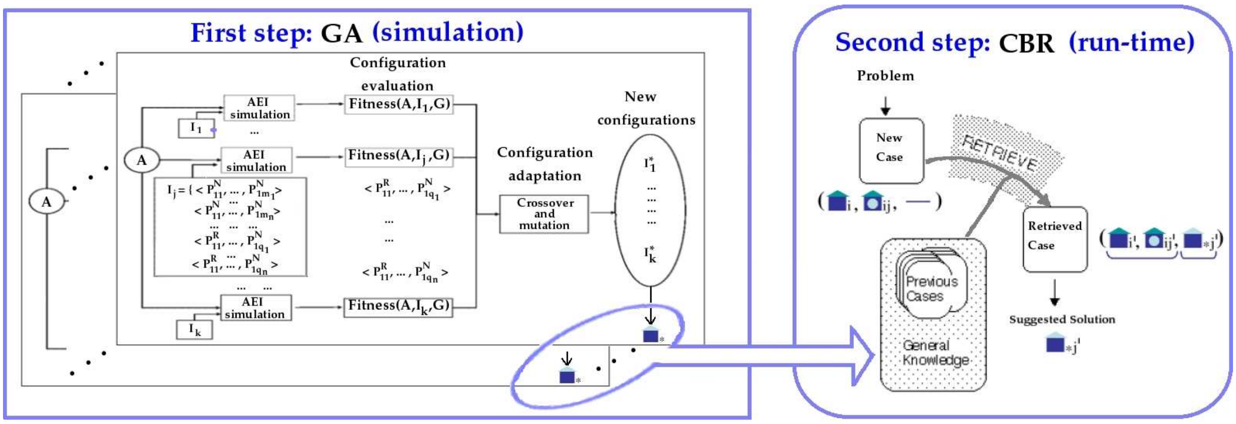

Fig. 1. Learning Model in two steps.

for each population. These best parameters will be used by subsequent step in the learning process.

As a second learning step, we propose to use a Case-Based Reasoning (CBR) approach, as shown in Figure 1. CBR [18] is based on learning from experience. Its basic idea is to search in the system's experience for similar situations - called cases. 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. We assume that agent populations that behave in similar way may require similar solutions (i.e. regulations). Therefore, we expect the CBR will allow an AEI to solve situations at run-time that are similar to the ones learned during the first learning step. Thus, whenever goals are not achieved, our AEI uses CBR to retrieve a solution (regulation parameters) from the most similar situation in the knowledge base. The knowledge base is generated by simulating the same agent populations used in first learning step. So that the cases' solution correspond to the best parameters learnt at first learning step for each population.

In this work we focus on the second learning step, namely how to adapt the parameters to any population at run-time. The way the first learning step is carried out is fully detailed by Bou et al. in [3].

4.1 Applying CBR

In this section we describe the representation of cases we propose to be used by an AEI. We also describe how to compute the similarity function used for comparing two cases. We assume that the set of all cases approximates both the *norm transition function* (δ) and the *PS transition function* (γ).

Intuitively, a case represents how an AEI using some parameters' values for regulating a given population of agents should change its regulations (to the best parameters' values). However, notice that an institution can only use observable information for representing cases because it has no access to the inner characterization of the agents that populate it. Therefore, the agent population can not be explicitly represented in the cases. With all of this in mind, we differentiate three main features to represent cases that we define as a tuple (N^p,PS^p,V,pop,N^{p*},PS^{p*}):

– **AEI parameters' values: ($\mathbf{N^p}$,$\mathbf{PS^p}$).** They represent the parameters' values of the institution, namely norm parameters' values and performative structure parameters' values that it uses for regulating agents.
  - $\mathbf{N^p}$ stands for the current norm parameters' values; and
  - $\mathbf{PS^p}$ stands for the current performative structure parameters' values.
– **Runtime behaviour: (V,pop).** They represent the global behaviour of the institution at run-time for a given agent population when the institution uses the previous *AEI parameters' values*. In other words, the effect of the parameters in the behaviour of agents at run-time.
  - **V** stands for the current set of reference values; and
  - **pop** stands for statistical data that characterises the behaviour of the agents' population at run-time[1].
– **Best AEI parameters' values: ($\mathbf{N^{p^*}}$,$\mathbf{PS^{p^*}}$).** They represent the learned parameters' values of the institution for the previous agent population. In other words: the solution (learning by simulation at first learning step). 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.
  - $\mathbf{N^{p^*}}$: represents the best values for the norm parameters given the current norm parameters values (N^p) and the runtime behaviour (V,pop); and
  - $\mathbf{PS^{p^*}}$: represents the best values for the performative structure parameters given the current performative structure parameters values (PS^p) and the runtime behaviour (V,pop).

In order to compare two cases, we use an aggregated distance function to compute the degree of similarity $S(C^i, C^j)$ between a new case C^i and a case C^j in the case base:

$$w_1 \cdot S_{AEI}(C^i, C^j) + w_2 \cdot S_V(C^i, C^j) + w_3 \cdot S_{pop}(C^i, C^j) \tag{1}$$

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. The distance between the values of an attribute is computed as:

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

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

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

5 Case Study: Traffic Control

This section introduces the case study we use to test our learning model. As case study we have extended the Traffic Regulation Authority presented in [3] as an Autonomic Electronic Institution. The environment is modeled as a 2-lane road junction where no traffic signals are considered. The case study considers the performative structure to be a single traffic scene with two agent roles: one institutional role played by police agents (that detect norm violations); and one external role played by car 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. There are 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 40 points each so that norm offenses cause points reduction. Moreover, the institution forbids external agents to drive without points in their accounts. In our setting, the environment is populated with 10 cars interacting inside the traffic scene and moving along the road network. During each discrete simulation, the institution replaces these cars running out of points by new cars so that the cars' population is kept constant.

Car agents only coordinate by following the traffic norms imposed by the AEI. Cars do not have learning skills. They just move based on their random trajectories and the probability of complying with a norm (based on function 3). Each car is modeled by their *agent norm compliance parameters*: $\langle fulfill\_prob$, h_p, $inc\_prob\rangle$; where $fulfill\_prob \in [0, 1]$ stands for the probability of complying with norms that is initially assigned to each agent; $h\_p \in \mathbb{N}$ stands for the fine threshold that causes an agent to consider a fine to be high enough to reconsider 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 (h_p). Thus, 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:

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

Notice that to use the norm compliance parameters to model cars allows us to generate homogeneous and heterogeneous agent populations. We refer to homogeneous agent population when agents in this population have the same compliance parameter values. Thus, all agents from homogeneous populations decide to comply with norms based in the same probability and therefore behave in a similar way. On the other hand, if in a population there are cars with different norm compliance parameters we consider it as an heterogeneous agent population, because agents have different behaviours.

The reference values the institution can observe along time are $V = \langle col,\ off,\ crash,\ block,\ expel,\ police\rangle$ 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 [2], *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.

On the other hand, the institution has four conflicting 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. Goal satisfaction is measured by combining the degree of satisfaction of these four institutional goals. They are combined in a weighted addition function, with weights 0.4, 0.4, 0.1, and 0.1 respectively. Thus, the first two goals are considered to be most important. 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.

Following the tuple case definition introduced in section 4.1, $(N^p, PS^p, V, pop, N^{p*}, PS^{p*})$, a case C^i in the traffic scenario is defined as follows:

- $\mathrm{N}^p = (fine_{right},\ fine_{front})$ are the values of both norms' parameters;
- $\mathrm{PS}^p = (police)$ is the value of the performative structure parameter;
- $\mathrm{V} = (col,\ crash,\ off,\ block,\ expel)$ 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 $(0 \leq t_w \leq t_{now})$;
- $\mathrm{N}^{p*} = (fine^*_{right},\ fine^*_{front})$ are the best values for both norms' parameters;
- $\mathrm{PS}^{p*} = (police^*)$ is the best value for the parameter of the performative structure.

[2] 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.

To compute the degree of similarity between two cases, we use aggregated distance function (1), $S(C^i, C^j)$. 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, $fine_{front}$ and $fine_{right}$ values are in the interval $[0, 15]$, and *police* values are in the interval $[0, 1]$. Although 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 *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 $|attr^i - attr^j| > max(attr) - min(attr)$.

Table 1. Homogeneous agent populations employed to generate the case base in setting 1.

Populations	Pop1	Pop2	Pop3	Pop4	Pop5	Pop6	Pop7
$fulfill\_prob$	0.5	0.5	0.5	0.5	0.5	0.5	0.5
h_p	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	8	13	12	13	14	15
$fine^*_{front}$	3	13	8	13	11	13	15
$police^*$	1	1	1	1	1	1	1

5.1 Case Base generation

As mentioned in section 4, (during training period) an AEI generates an initial base of cases from simulations of a set of prototypical populations. In this work, our aim is to test if the proposed CBR approach is suitable to help the AEI to adapt its parameters to heterogeneous agent populations. Notice the importance of the case base, because the adaptation of an AEI largely depends on the cases it uses. Notice that an AEI needs to retrieve a similar case whose values, when applied to the current population, help the AEI fulfill its institutional goals. If the AEI has not any good, similar cases in the case base, it would be not able to adapt its parameters to the population. Therefore, the more representative a case is (i.e., the more coverage it has), the more useful it becomes for solving new cases. Case coverage is related to similarity but also to generality. In order to study if case coverage is affected by population heterogeneity, we have considered two ways of generating the case base. First, we want to test how well the AEI adapts to heterogeneous populations when it uses a case base whose cases have been created from homogeneous agent populations. Moreover, we want to test the AEI adaptation performance to heterogeneous agent populations when heterogeneous populations are also used for creating the cases, so that cases will be more similar.

The idea is to compare the adaptation of the AEI to different heterogeneous populations when it uses each case base. Thus, we have considered two settings, one for each case base.

Setting 1 This setting has a case base generated from homogeneous agent populations. Table 1 shows the features of the seven homogeneous populations (Pop) we have considered to generate this case base. They are characterized by the norm compliance parameters of the agents, being $fulfill\_prob = 0.5$ and $inc\_prob = 0.4$ for all of them, whereas h_p varies from 0 to 14. Table 1 also shows the best AEI parameters' values (N^*, PS^*) the institution has learned by using genetic algorithms for each population ($fine^*_{right}$, $fine^*_{front}$ and $police^*$).

In order to create the case base we have combined all values of the AEI parameters' values. We have considered to cover the range of all possible parameters' values ($fine_{front}$ and $fine_{right}$ values are in the interval $[0, 15]$, and $police$ values are in the interval $[0.8, 1]$). Thus, we have used $fine_{right} \in \{0, 3, 6, 9, 12, 15\}$, $fine_{front} \in \{0, 3, 6, 9, 12, 15\}$, and $police \in \{0.8, 0.9, 1\}$. Thus, overall we have obtained 108 different AEI parameters' values. To create cases for this case base, we have simulated each population in Table 1 with all 108 AEI parameters' values. 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 generated 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.

Setting 2 Here we have combined homogeneous and heterogeneous agent populations to generate the case base. In order to create heterogeneous agent populations, we have combined agents from different homogeneous populations in Table 1. To create cases for this case base, we have considered two types of heterogeneous populations. The first one has two kinds of agents that are equally distributed (50 percent each, i.e., 5 cars from a certain homogeneous populations and 5 cars from another one). The second heterogeneous population is composed of seventy percent agents of one kind and a thirty percent of another one. With the aim of covering the range of all h_p values, the agents we have combined to generate both types of heterogeneous populations are agents from populations Pop2, Pop4, Pop6, Pop7 in Table 1 (with h_p values 3,8,12 and 14 respectively). Additionally, we have also generated cases for this case base from these homogeneous populations. Likewise the first setting, to create the case base we have simulated each population (homogeneous and heterogeneous) with the 108 different AEI parameters' values, generating 3456 cases overall. Cases from each 2000 ticks have also been created likewise in the first setting.

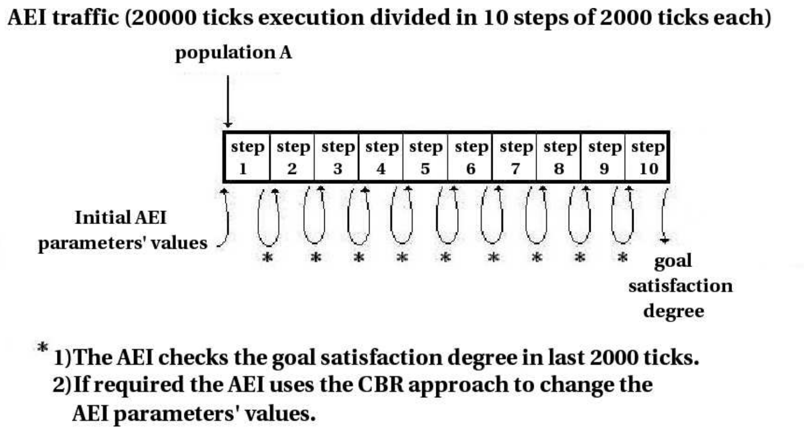

Fig. 2. Scheme of an AEI traffic experiment.

6 Empirical Evaluation

We have designed an experiment to test if an AEI is able to self-configure at run-time for different heterogeneous agent populations by using the proposed CBR approach. Figure 2 shows a scheme of an experiment using the case study in section 5. We run this experiment several times so that each experiment corresponds to the simulation of a different heterogeneous population. As Figure 2 shows, each experiment is composed of 20000 ticks that we divide into 10 steps. At each step (every 2000 ticks), the AEI checks the fulfillment of the institutional goals by means of its goal satisfaction degree. At the end of each step the AEI uses the CBR approach to change its parameters' values, if required.

In the context of CBR, the AEI generates each new problem case by considering its current configuration values and the population's runtime behaviour at the end of the current step. As mentioned above, after each step, the AEI decides if it needs to retrieve a case or not. The decision is based on the goal satisfaction and a threshold value. This threshold is computed as a desired goal satisfaction value (G^*) with some deviation (ϵ). In our experiments, we have set $\epsilon = 0.04$ and $G^* = 0.67$, which corresponds to the minimum of the best goal satisfaction degrees for the populations in Table 1. Therefore, we set the threshold value to 0.63. If the goal satisfaction is greater than the threshold, the AEI keeps the same parameters during 2000 more ticks (that is, until the next step is reached). Otherwise (when the goal satisfaction is lower than the threshold) the AEI launches its CBR engine to retrieve the most similar case from its case base so that the AEI can use its solution (i.e., best parameters' values) at the next step.

In order to test if the AEI is able to self-adapt its regulations at run-time, we start each experiment with parameter values that prevent the institution from accomplishing its goals. Thus, for all experiments, the AEI starts with (0,0,0.8) parameters, causing no agent to follow the norms. These parameters correspond to no fine for both right-hand side and front priority norms, and a deployment of 80% of police agents. Thus, we expect the AEI to start with a low goal satisfaction degree. However, we also expect the AEI to be able to progress towards its goals in a few steps by retrieving a similar case whose parameters help the AEI to fulfill its institutional goals. Notice that if the AEI does not manage to retrieve a case whose parameters help it obtain a goal satisfaction higher than 0.63, the AEI will not adapt. Therefore, an AEI keeps retrieving cases trying to adapt better to each population it regulates. Thus, it is natural to think of computing the number of steps an AEI needs to adapt (i.e., to achieve its goals), namely to reach a goal satisfaction higher than the threshold value 0.63. At this aim, we compute the step at which an AEI achieves its goals for each experiment. We consider that an experiment is successful at some step if the goal satisfaction degree during the last 2000 ticks is equal or greater than the threshold value. Otherwise, we consider that the experiment is not successful at this step. That is, we consider that an experiment is successful at a specific step if the AEI fulfills its institutional goals during that step.

The experiment we have designed (see Figure 2) helps us testing if an AEI can retrieve a case out of the case base in order to adapt its parameters to different heterogeneous agent populations at run-time. To test our approach we have used different heterogeneous populations. As before, to create heterogeneous populations we have combined two different cars' behaviours. Cars to be combined are characterized by their norm compliance parameters, being $fulfill\_prob = 0.5$ and $inc\_prob = 0.4$ for all of them, whereas h_p varies from 0 to 14. In this work, trying to cover a wide range of heterogeneous populations, we have performed experiments combining cars with $h\_p \in \{2, 4, 8, 10, 12, 14\}$ in percentages of 50, 60, 70 and 80 percent with cars with $h\_p \in \{5, 9, 13\}$ in percentages of 50, 40, 30 and 20 percent respectively. We have used the heterogeneous populations resulting of all these combinations to test both experimental settings 1 and 2. In order to obtain statistical results we have run each experiment 50 times for each heterogeneous agent population. Thus, overall we have performed a total of 3600 experiments for each setting.

Table 2. Number of successful at each step.

	Steps	2	3	4	5	6	7	8	9	10	No adaptation
Setting 1	**First time successful**	**1612**	**908**	**276**	**157**	**119**	**67**	**56**	**44**	**52**	**309**
	Percentage	44.8	25.2	7.7	4.3	3.3	1.9	1.6	1.2	1.4	8.6
	Accumulated percentage	44.8	70	77.7	82	85.3	87.2	88.8	90	91.4	
Setting 2	**First time successful**	**2156**	**900**	**261**	**103**	**60**	**40**	**31**	**16**	**7**	**26**
	Percentage	59.9	25	7.3	2.9	1.7	1	0.9	0.4	0.2	0.7
	Accumulated percentage	59.9	89.4	92.2	95.1	96.8	97.8	98.7	99.1	99.3	

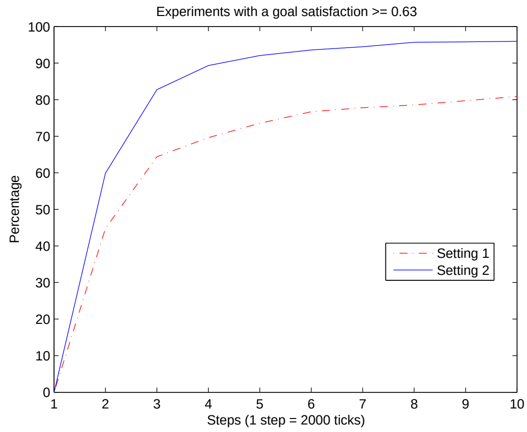

Fig. 3. Percentage of successful experiments.

6.1 Results

Setting 1 In these experiments, although we use heterogeneous populations, the AEI retrieves cases from a case base created from homogeneous populations. The idea is to test how well the AEI adapts to heterogeneous agent populations when it uses cases created from information of homogeneous agent populations.

Figure 3 shows the percentage of successful experiments at each step from the total of 3600 experiments. In Figure 3 we can see how the curve of the percentage of successful experiments corresponding to setting 1 stabilizes around 70 percent at step 5. Thus, we can state that, in these experiments, after five steps (10000 ticks) our AEI manages to adapt its parameters to approximately 70 percent of heterogeneous populations when using a case base created from homogeneous populations.

In order to analyse how many steps the AEI needs to adapt, we have computed for each experiment at which step the experiment starts to succeed (that is, the first step for which the AEI has obtained a goal satisfaction $>= 0.63$). Table 2 shows the number of experiments of this setting that were successful for the first time at each step, together with the corresponding percentage they represent and the accumulated percentage. We can see that, although at step three (6000 ticks) the AEI has adapted to a 70 percent of experiments, for 8.6 percent of the experiments the AEI did not manage to adapt its parameters (i.e., at ten steps the AEI obtained a goal satisfaction lower than 0.63).

Setting 2 Like in the previous setting, we have run a total of 3600 experiments using heterogeneous populations. However, in these experiments the AEI retrieves cases from a case base created from both homogeneous and heterogeneous populations.

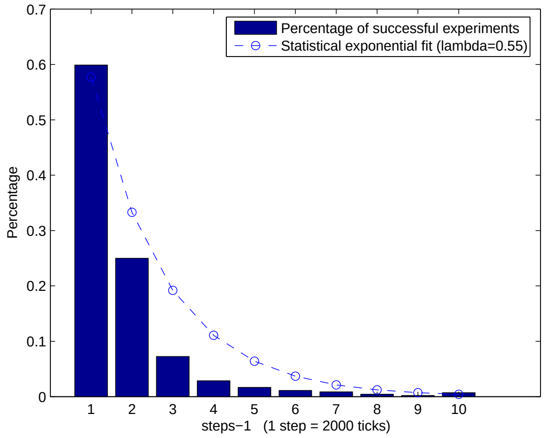

Fig. 4. Percentage of experiments initially successful.

Figure 3 shows the percentage of successful experiments at each step. We can see how the curve of the percentage of successful experiments corresponding to setting 2 stabilizes around 90 percent at step 5. Comparing both curves, we can state that our AEI adapts better its parameters to heterogeneous populations when using a case base created from homogeneous and heterogeneous populations. That is because in this setting cases are more representative of heterogeneous populations than the previous setting's.

In the same way that in the previous setting, we have computed the required number of steps to reach adaptation for a given initial heterogeneous population of agents. The number of experiments of this setting that have been successful for the first time at each step are shown in Table 2, together with the corresponding percentage they represent and the accumulated percentage. In order to obtain statistical results, we have performed the chi-square test to test if the data (the percentage of experiments that have been successful for first time at each step) follows an exponential distribution. The chi-square test allows us to say that our data follows an exponential distribution with lambda = 0.55 (chi-square value=6.78, with a significance level of 0.01 and 4 degrees of freedom). Figure 4 shows the percentage of experiments that have started to succeed at each step and the fitted exponential[3]. Finally, we have computed the accumulated function of the fitted exponential in order to compute the number of steps an AEI needs to adapt to heterogeneous agent populations when using cases from homogeneous and heterogeneous populations. We have found that an AEI needs 6 steps (12000 ticks) to adapt for the very first time to the 95 percent of population -in the statistical sense- (significance level of 0.01).

[3] Notice that we have performed a correction in the steps because in first step there is a zero percentage.

7 Conclusions

This paper presents a Case-Base Reasoning approach which allows an AEI to self-adapt at run-time. We have tested the retrieval process of the approach on a case study, where the AEI learns two traffic norms and the number of institutional agents it requires in order to adapt the norms and the performative structure to heterogeneous agent populations. Due to the importance of the base of cases to the adaptation, we have tested the approach in two settings, one for each case base. As to the first setting, we have used homogeneous agent populations to create the cases, whereas for the second setting we have used both homogeneous and heterogeneous agent populations. Since we have run experiments with heterogeneous agent populations, the case base of the second setting is more suitable for the AEI adaptation because cases are more representative. We have performed statistical analysis about the time (in steps of 2000 ticks) an AEI needs to adapt its parameters to heterogeneous populations. The results in this paper look promising since they show that our traffic AEI can indeed adapt to heterogeneous populations. However, the degree of adaptation depends on the case base an AEI uses. The adaptation to heterogeneous populations is better when the cases the AEI uses are generated from homogeneous and heterogeneous agent populations than when they are only generated from homogeneous agent populations. Statistical results show that an AEI needs 6 steps (12000 ticks) to adapt to the 95 percent of population (significance level of 0.01), when it uses cases generated from homogeneous and heterogeneous agent populations. Nevertheless, it is important to notice that distribution percentage does not need to be equal in order for a case to be representative. In fact, we have generated a case base with distributions 50%-50% and 70%-30% and tested with distributions 50%-50%, 60%-40%, 70%-30% and 80%-20%. Therefore, our experimental results show that there is no need for the case base to include all possible distributions in order to provide useful cases.

As future work, we plan to develop a more complex traffic network. Additionally, we are interested in studying how institutional parameters and agent strategies may co-evolve. Nevertheless, this will require to extend the agents so that they become able to adapt to institutional changes. We are also interested in modifying the approach to include institutional goals in the definition of the cases in order to allow the AEI to change its goals at run-time.

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 Generalitat de Catalunya under the grant 2005-SGR-00093 and 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. Esteva, M.: Electronic Institutions: from specification to development. IIIA PhD Monography. Vol. 19 (2003)
2. North, D.: Institutions, Institutional Change and Economics Perfomance. Cambridge U. P. (1990)
3. 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
4. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. IEEE Computer **36**(1) (2003) 41–50
5. Norman, T.J., Preece, A., Chalmers, S., Jennings, N.R., Luck, M., Dang, V., Nguyen, T., Deora, V., Shao, J., Gray, A., Fiddian, N.: Conoise: Agent-based formation of virtual organisations. Research and Development in Intelligent SystemsXX: Proceedings of AI2003, the Twentythird SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (2003) 353–366 Best Paper Award at AI-2003, ©Springer Verlag.
6. Excelente-Toledo, C.B., Jennings, N.R.: The dynamic selection of coordination mechanisms. Autonomous Agents and Multi-Agent Systems **9**(1-2) (2004) 55–85
7. Verhagen, H.: Norm Autonomous Agents. PhD thesis, (Stockholm University)
8. Sen, S., Airiau, S.: Emergence of norms through social learning. In: IJCAI. (2007) 1507–1512
9. Sierra, C., Sabater, J., Agustí, J., Garcia, P.: Integrating evolutionary computing and the sadde methodology. In: AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems, New York, NY, USA, ACM Press (2003) 1116–1117
10. 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
11. 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
12. van der Vecht, B., Dignum, F., Meyer, J.J.C., Neef, M.: A dynamic coordination mechanism using adjustable autonomy. In: Coordination, Organization, Institutions and Norms in agent systems (COIN@Durham'07). Co-held in Multi-Agent Logics, Languages, and Organisations Federated Workshop. (2007)
13. Hoogendoorn, M.: Adaptation of organizational models for multi-agent systems based on max flow networks. In Veloso, M.M., ed.: IJCAI. (2007) 1321–1326
14. 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
15. Camurri, M., Mamei, M., Zambonelli, F.: Urban traffic control with co-fields. In: Proc. of E4MAS Workshop at AAMAS 2006. (2006) 11–25
16. Bazzan, A.L.C., de Oliveira, D., Klügl, F., Nagel, K.: Effects of co-evolution in a complex traffic network. In: Proceedings of the AAMAS 2007 Workshop on Adaptive and Learning Agents (ALAg-07). (2007)
17. Ros, R., Veloso, M.: Executing Multi-Robot Cases through a Single Coordinator. In: Proc. of Autonomous Agents and Multiagent Systems. (2007) 1264–1266
18. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Commun. **7**(1) (1994) 39–59