A

Robust regulation adaptation in multi-agent systems

JORDI CAMPOS, Universitat de Barcelona
MAITE LOPEZ-SANCHEZ, Universitat de Barcelona
MARIA SALAMÓ, Universitat de Barcelona
PEDRO AVILA, Universitat de Barcelona
JUAN A. RODRIGUEZ-AGUILAR, Artificial Intelligence Research Institute (IIIA-CSIC)

Adaptive organisation-centred multi-agent systems can dynamically modify their organisational components to better accomplish their goals. Our research line proposes an abstract distributed architecture (2-LAMA) to endow an organisation with adaptation capabilities. This paper focuses on regulation-adaptation based on a machine learning approach, in which adaptation is learned by applying a tailored case-based reasoning method. We evaluate the robustness of the system when it is populated by non-compliant agents. The evaluation is performed in a peer-to-peer sharing network scenario. Results show that our proposal significantly increases system performance and can cope with regulation violators without incorporating any specific regulation-compliance enforcement mechanism.

Categories and Subject Descriptors: I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms: Design, Performance

Additional Key Words and Phrases: Machine Learning, Organisation Centred MAS, Regulation, Adaptation, Case-Based Reasoning

ACM Reference Format:
Campos, J., Lopez-Sanchez, M., Salamó, M., Avila, P., Rodríguez-Aguilar, J. A. 2012. Robust regulation adaptation in multi-agent systems. ACM Trans. Autonom. Adapt. Syst. V, N, Article A (March 2012), 27 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

The expansion of the Internet has caused a shift from stand-alone and single user applications to distributed and open systems composed of networked software (SW) components. These components may collaborate to achieve common (or system) goals, or compete to accomplish individual goals. Some components may even decide not to behave properly to better accomplish their individual goals. In general, goal achievement requires a coordination model that should be continuously adapted to cope with changing and unexpected conditions.

Multi-agent systems (MAS) [Wooldridge 2009] provide a powerful paradigm to engineer this kind of applications, where SW components are modelled as autonomous agents. In particular, organisation-centred MAS approaches (OCMAS) [Ferber et al. 2004] use explicit organisations to structure agents' interactions. Nevertheless, run-time changing situations may vary organisations' ability to fulfil their goals unless

---

Author's addresses: J. Campos, M. Lopez-Sanchez, M. Salamó, and P. Avila, Volume Visualization and Artificial Intelligence Research Group (WAI), Universitat de Barcelona, Gran Via de les Corts Catalanes, 585 - 08007 Barcelona, Spain; J.A. Rodríguez-Aguilar Artificial Intelligence Research Institute (IIIA), Spanish National Research Council (CSIC), Campus UAB, 08193 Bellaterra (Cerdanyola del Valles), Spain.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

they are adapted accordingly [Martin and Barber 2006]. In fact, under changing circumstances, organisational dynamic adaptation can improve system performance despite the overhead and instability inherent to changes. We refer to these systems as *Adaptive* OCMAS (AOCMAS). In [Campos 2011] we proposed a general distributed architecture that allowed to endow an organisation with adaptation capabilities. As for any other MAS, distribution among agents provides robustness, avoids both communication and computational bottle-necks and can deal with local information[1].

Along this line, in this paper we focus on the design and implementation of a general-purpose adaptation mechanism for organisation-centred MAS. More precisely, we propose an adaptation mechanism that can be embedded in the general distributed architecture proposed in [Campos 2011] to found the development of *Adaptive* OCMAS. With this aim, here we consider that the organisation model to endow with adaptation capabilities is composed of: a social structure, some social conventions and organisational goals. In particular, social conventions include regulations[2]. Therefore, adding adaptive capabilities to an organisation will amount to endowing it with an adaptation mechanism that performs the autonomous adaption of its regulations to cope with changing circumstances. To achieve this goal, we offer several contributions in this paper:

— *A formal model of regulation*. Our formal model allows us to encode regulations as norms. Moreover, we also introduce a general notion of norm pattern. Based on this concept, adapting the norms (regulations) of an OCMAS can be regarded as learning the values of the parameters in their norm patterns that lead to the best performance of the OCMAS.
— *An adaptation mechanism for regulations* based on a variation of standard case-based reasoning (CBR) [Riesbeck and Schank 1989]. Our mechanism extends the mechanism described in [Campos et al. 2011], which performs adaptation by means of an incomplete CBR cycle. Thus, our mechanism implements a complete CBR cycle that introduces several improvements with respect to [Campos et al. 2011]: (i) it is more expressive, and hence it enhances the knowledge representation in the case base; (ii) it allows to improve the quality of the knowledge in the case base, thus leading to more accurate reasoning; and (ii) it avoids knowledge degradation in the case base by improving the way knowledge is refined.
— *An empirical evaluation of the robustness of our adaptation mechanism*. This paper is also devoted to studying whether our proposed mechanism can deal with different (and simultaneous) instabilities. In particular, we pay attention to those derived from inherent system dynamics as well as those resulting from the existence of agents that decide not to comply with established regulations (henceforth referred to as *violators*). Thus, we first introduce a behaviour model for violators. Thereafter, we empirically evaluate our adaptation mechanism in the context of a simplified peer-to-peer (P2P) data sharing network scenario[3]. Our empirical results show that we can successfully regulate any agent population even in the presence of misbehaving agents. In fact, our mechanism adapts general regulations for the whole agent population without employing any norm-compliance enforcement mechanism nor direct norm-compliance monitoring, which are both common practice in the literature. Instead, the approach is able to establish regulations whose associated constraints are proportional to the current population behaviour. This is due to non-compliant agents caus-

[1]Information can be considered to be local due to unavailability or privacy issues.
[2]We consider regulations as norms (in their broad sense) or social conventions [Conte et al. 2010] in which the sanctions may not be specified.
[3]Although P2P data sharing constitutes our simulated scenario, we will also refer to a traffic example in order to illustrate the potential applications and generality of our proposal.

ing a deviation in the organisational goals accomplishment, which causes, in turn, the adaption mechanism to further restrict current regulations. Thus, as a result, we have that the more non-compliant agents in the population, the more restrictive the regulations become.

The rest of the paper is structured in eight sections: §2 presents a formalisation of the models in our meta-level approach; §3 illustrates it in the P2P case study; and §4 details the learning adaptation process that is evaluated in §5. Subsequently, §6 introduces more specific background and §7 provides further insights into our work. Finally, §8 exposes the derived conclusions and future work.

2. FORMALISATION

This section sets the foundations for the description of our adaptation mechanism. First, in Section 2.1 we provide background on the general distributed architecture for adaptive OCMAS described in [Campos 2011]. Our adaptation mechanism is aimed at operating over that architecture. Next, in Section 2.2 we introduce a novel, formal model of norms that allows us to encode regulations as norms. As part of this model, we introduce the notion of norm pattern and discuss how norms are used by the general adaptation mechanism introduced in [Campos 2011]. Finally, in Section 2.3 we introduce the model for norm violation that we will employ to test our adaptation mechanism described in Section 4.

2.1. Description of the abstract distributed architecture

The Two-Level Assisted MAS Architecture (2-LAMA [Campos et al. 2011]) is an abstract distributed architecture proposed to endow an organisation with adaptation capabilities. As Figure 1 shows, 2-LAMA considers that domain-level (DL) agents (noted as $ag_1, ..., ag_n$) conduct regular/domain-specific activities. On top of them, assistant agents (noted as $as_1, \dots, as_m$) are in charge of adapting the domain-level organisation (Org_{DL}) while preserving the domain agents' autonomy. Following a division of labour paradigm, instead of increasing the complexity of DL agents, assistants are in charge of reasoning at a higher level of abstraction than DL agents. They consider general information —such as DL organisational goals, organisational description or DL system behaviour— whereas DL agents reason at their local —i.e., individual— level. Therefore, assistants are located at the meta-level (ML) and assumed to be staff (organisational) agents[4]. Hence, assistants are conceptually separated from the DL.

Notice that 2-LAMA is thus based on the notion of meta-level, which was early and successfully introduced in the AI literature as a means of introducing control in a complex system [Corkill and Lesser 1983; Zhang et al. 2009; Martin and Barber 2006]. In particular, 2-LAMA's encapsulation of the adaptation process in a meta-level takes inspiration from the seminal work by Corkill et al. [Corkill and Lesser 1983], where the meta-level was in charge of providing guidelines to guarantee acceptable global behaviour.

Equations 1 to 8 below formally capture the components of the 2-LAMA architecture. Equation 1 shows that each level has its own set of agents, organisation, and environment: $\langle Ag_{ML}, Org_{ML}, Env_{ML}\rangle$ in the meta-level and $\langle Ag_{DL}, Org_{DL}, Env_{DL}\rangle$ in the domain level. Following Equations 2 and 3, there are m agents (henceforth referred to as assistants) in the meta-level and n agents in the domain level. Each assistant in the meta-level assists a cluster (a group) of domain-level agents. Formally, the cluster of assistant as_i is a subset of domain-level agents, namely $cluster_i \subseteq Ag_{DL}$, which is

[4]Although meta-level roles are required to belong to the organisation, DL roles can be enacted by staff agents or external participants.

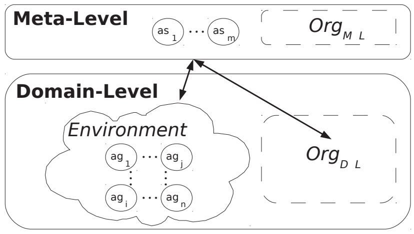


Fig. 1: 2-LAMA architecture, where meta-level (ML) adapts domain-level (DL)

defined according to domain-specific criteria. Each assistant perceives the state of its own cluster to reason about its adaptation.

The equations in 4 define the components of the meta-level organisation Org_{ML} and the domain-level organisation Org_{DL}. Both organisations are composed of a social structure, a set of social conventions and a set of goals. First, each social structure, as defined in Equations 5, contains the set of roles that agents in an organisation play and the set of relationships between roles. Thus, Rol_{ML} and Rel_{ML} stand for roles and relationships at the meta level, whereas Rol_{DL} and Rel_{DL} stand for their counterparts at the domain level. Second, Equation 6 specifies social conventions, which are aimed at regulating agents' actions (though agents should conform to them, we will analyse non-compliance effects). Specifically, the social conventions at either the meta level or domain level are expressed as a pair: a set of norms, which are formally defined later on in Section 2.2; and a set of protocols, specified as UML sequence diagrams. Therefore, Str_{ML} and Str_{DL} stand for the social structures of the meta-level and domain-level, whereas $Conv_{ML}$ and $Conv_{DL}$ stand for their social conventions. Finally, $Goals_{ML}$ and $Goals_{DL}$ in Equations 7 stand for the goals of the meta-level and domain-level respectively. In both cases, the goals express an organisation's design purpose and are formally defined as a function over the environment and agent observable properties $EnvP$ and AgP. In this work we make the assumption that the meta-level and domain level share the very same goals, namely $Goals_{ML}$ and $Goals_{DL}$ are equal. Finally, since assistants gather domain-level information, Equation 8 specifies that the meta-level environment (Env_{ML}) contains its own observable properties ($EnvP_{ML}$) along with the domain-level organisation (Org_{DL}), the observable properties of the domain-level environment (Env_{DL}) and the observable properties of the domain-level agents (AgP_{DL}). The domain-level environment, Env_{DL}, limits to its own observable properties ($EnvP_{DL}$).

$$2LAMA = \langle ML, DL\rangle, ML = \langle Ag_{ML}, Org_{ML}, Env_{ML}\rangle, DL = \langle Ag_{DL}, Org_{DL}, Env_{DL}\rangle \quad (1)$$

$$Ag_{ML} = \{as_1, ..., as_m\},\ |Ag_{ML}| = m \quad \textit{assistants} \quad (2)$$
$$Ag_{DL} = \{ag_1, ..., ag_n\},\ |Ag_{DL}| = n \quad \textit{agents with } AgP_{DL} \textit{ observable properties} \quad (3)$$

$$Org_{ML} = \langle Str_{ML}, Conv_{ML}, Goals_{ML}\rangle,\quad Org_{DL} = \langle Str_{DL}, Conv_{DL}, Goals_{DL}\rangle \quad (4)$$
$$Str_{ML} = \langle Rol_{ML}, Rel_{ML}\rangle,\quad Str_{DL} = \langle Rol_{DL}, Rel_{DL}\rangle, \quad (5)$$
$$SocConv_{ML} = \langle Norms_{ML}, Protocols_{ML}\rangle,\quad SocConv_{DL} = \langle Norms_{DL}, Protocols_{DL}\rangle \quad (6)$$
$$Goals_{ML} = \langle f(EnvP, AgP)\rangle,\quad Goals_{DL} = \langle f'(EnvP, AgP)\rangle \quad (7)$$

$$Env_{ML} = \langle EnvP_{ML}, EnvP_{DL}, AgP_{DL}, Org_{DL}\rangle,\quad Env_{DL} = \langle EnvP_{DL}\rangle \quad (8)$$

2.2. Regulations and their adaptation

Regulations are effective in constraining agents' behaviour. Given certain contexts, constrains can either forbid or bring about specific agent states. An agent state can be expressed in terms of its properties. In general, we will simply consider that an agent state is specified in terms of the values of some variables. Therefore, from a syntactic point of view, we consider regulations as conditional norms (IF-THEN rules) that, given some context, constrain the values that the properties of an agent can take on. Such constraints are specified in terms of: a deontic operator (either a prohibition, obligation, or permission), a constraint operator, and a threshold value. Expression 9 specifies the syntax of a norm. The triple $(var\ limitOp\ limitV)$ stands for the *antecedent* of a norm, and is aimed at capturing the norm's context of application, whereas the triple $deonticOp(var'\ limitOp'\ limitV')$ stands for the consequent of a norm, and is aimed at capturing the actions that trigger if the antecedent holds. In the expression, var and var' stand for variable names, $\mathtt{limitV}$ and $\mathtt{limitV}'$ stand for threshold values for var and var' respectively, $limitOp$ and $limitOp'$ stand for operators, and $deonticOp$ stands for a deontic operator. The types of the variables and thresholds in a norm can be either real, boolean, or a set of labels.

The bottom part of expression 9 illustrates two examples of norms in a traffic domain: the first one limits the speed on a highway to 100 and the second one describes a stop sign obligation. The first norm considers a context described in terms of the location of a car (through variable *carLocation*) and limits the car's actions so that the resulting speed is lower than 100. Similarly, the second norm enforces a car to stop whenever it comes across a stop sign.

$$
\begin{aligned}
&Norm\_Name : \mathtt{IF}\ (var\ limitOp\ \mathtt{limitV})\ \mathtt{THEN}\ deonticOp(var'\ limitOp'\ \mathtt{limitV}')\\
&where\quad deonticOp \in \{Prohibition, Obligation, Permission\}\\
&\qquad\quad limitOp, limitOp' \in \{>, \geq, <, \leq, =, \neq, \in, \notin, \subset, \not\subset, \subseteq, \not\subseteq\}\\
&\qquad\quad \mathtt{type}(var), \mathtt{type}(var'), type(\mathtt{limitV}), type(\mathtt{limitV}') \in \{\Re,\ \mathbb{B},\ \mathtt{Labels}\}
\end{aligned} \tag{9}
$$

$$
\begin{aligned}
&highwaySpeed : \mathtt{IF}\ (carLocation = '\mathtt{highway}')\ \mathtt{THEN}\ Prohibition(carSpeed > 100)\\
&stopSign : \mathtt{IF}\ (carLocation = '\mathtt{StopSign}')\ \mathtt{THEN}\ Obligation(stoppedCar = \mathtt{True})
\end{aligned}
$$

Now it is time to describe the role played by norms in 2-LAMA's organisational adaptation mechanism. Notice that regulation adaptation is part of organisational adaptation, since norms are in turn part of organisations (as follows from Equations 4 and 6). Following expression 10, formally organisational adaptation is performed by function α^O, which transforms the domain-level organisation received as an input. This transformation is based on domain-level observable properties ($EnvP_{DL}$, AgP_{DL}), and the current organisation (Org_{DL}), which includes system goals ($Goals$). Analogously, from a formal point of view, norm adaption is carried out through function α^N in expression 11. This function updates the norms in $Norms_{DL}$ from their current values, the domain-level environment and agent properties, and the organisational goals[5].

Figure 2 outlines 2-LAMA norm adaptation mechanism. Observe that norm adaptation is performed through two stages. During the first stage, each assistant agent as_i computes its *individual norm adaptation function* α_i^N (as shown in expression 12). Each assistant can observe (and summarise) both the environment and agent properties of its cluster. Each assistant as_i then shares summaries and computes its own estimation of the overall values of these observable properties ($EnvP_{DL,i}$, $AgP_{DL,i}$) by combining both local and shared information. Each assistant then computes α_i^N by

[5]Adaptation functions can be defined for each organisational component. Nevertheless we assume that they are independent [Campos et al. 2011] and restrict ourselves to the regulation adaptation process.

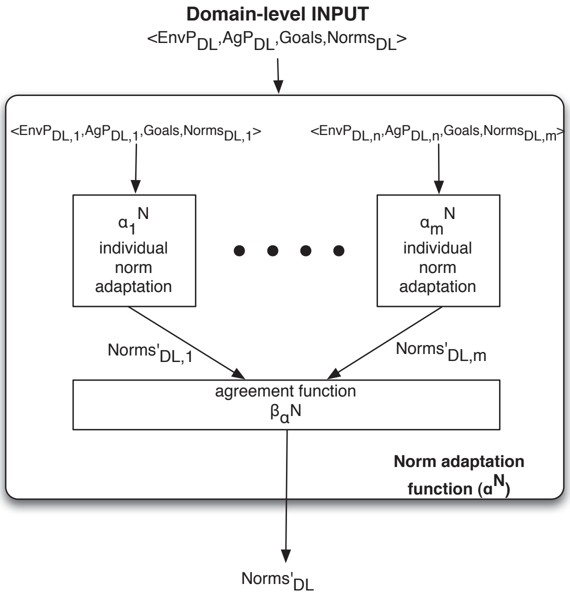


Fig. 2: 2-LAMA norm adaptation mechanism.

using this information, together with goals and current regulations, to generate its own regulation proposal ($Norms'_{DL,i}$). Afterwards, during the second stage, assistants agree on the actual (overall) regulations' update through an *agreement* function that implements the signature in expression 13. Notice, therefore, that the computation of changes in the current norms held by 2-LAMA is distributed between assistants.

In Section 4 we detail a novel way of implementing the individual norm adaptation function (the α_i^N functions in Figure 2) that each assistant computes. More concretely, we propose to implement each individual function as a variation of a standard case-based reasoning system. Furthermore, we propose to implement the agreement function β_{α^N} as a voting mechanism.

$$\textit{organisational adapt.,}\ \alpha^O(EnvP_{DL}, AgP_{DL}, Org_{DL}) = Org'_{DL} \tag{10}$$

$$\textit{norm adaptation func.,}\ \alpha^N(EnvP_{DL}, AgP_{DL}, Goals, Norms_{DL}) = Norms'_{DL} \tag{11}$$

$$\textit{individual norm adapt.,}\ \alpha_i^N(EnvP_{DL,i}, AgP_{DL,i}, Goals, Norms_{DL}) = Norms'_{DL,i} \tag{12}$$

$$\textit{agreement function,}\ \beta_{\alpha^N}(Norms_{DL,1}, \ldots, Norms_{DL,m}) = Norms'_{DL} \tag{13}$$

2.3. Modeling norm compliance and norm violation

Whether agents comply or not with regulations may affect system performance. In order to test the robustness of the norm adaptation approach that we will introduce in the following sections, next we introduce how we model norm compliance and norm violation. The generation of violations occurs through two stages.

First, we choose the number of violating agents out of the population of domain-level agents (Ag_{DL}). With this aim, we employ the global parameter $pVio \in [0, 1]$ as a ratio of violators out of domain-level agents. Then, considering that there are n domain-level agents, we will compute the number of agents that will violate norms, the so-called *violator agents*, and the number of compliant agents following expression 14. In fact, notice that the purpose of this first step is to partition the set of domain-level agents into two sets: *violatorAgents* and *violatorAgents*.

$$\begin{array}{l} |\texttt{normCompliantAgents}| = n \cdot (1 - pVio)\ , \quad |\texttt{violatorAgents}| = n \cdot pVio \\ Ag_{DL} = \texttt{normCompliantAgents} \cup \texttt{violatorAgents} \end{array} \tag{14}$$

Once we have assessed how many agents will violate norms, we must specify their violating behaviour, namely the extent of their violations. We will assume that not all violators are characterised by the same behaviour. Therefore, agents within the *violatorAgents* set will violate regulations differently. Considering that regulations establish limits to specific actions, we model to what extent each agent will exceed/meet these limits. With this aim, each violator agent ag_j counts of a local violation degree parameter, $dVio_j \in [0, 1]$, which sets the extent of its violations. Thus, each violator agent ag_j will combine its violation degree with the limits of the norm to violate to assess the extent of its violation. Formally, Equation 15 speficies how each violator ag_j computes its violation extent for a norm η, noted as $vioLimit_{j,\eta}$, whose consequent is $deonticOp(var'\ limitOp'\ limitV_\eta)$. The expression includes a violation sign, `vioSign`, to make the violation either exceed a maximum regulation limit value (when set to 1) or keep it below a minimum limit value (when set to -1).

$$vioLimit_{j,\eta} = \texttt{limitV}_\eta \cdot (1 + \texttt{vioSign} \cdot dVio_j)\ \ for\ all\ \ j = 1..n\ ,\ \texttt{vioSign} \in \{1, -1\} \tag{15}$$

As an illustration, consider the traffic norm named *highwaySpeed* that limits speed to 100 kilometres per hour in Eq. 9. In this case, the norm's limit threshold is 100, and hence $limitV_{highwaySpeed} = 100$. Recall that violators may exceed this limit differently. For instance, a moderate violator ag_j with $dVio_j = 0.2$ will decide to drive at 120 km/h, since $vioLimit_{j,highwaySpeed} = 100 \cdot (1 + 1 \cdot 0.2) = 120$ km/h. However, an extreme violator ag_k with $dVio_k = 0.9$ will decide to drive at 190 km/h. However, an actual violation depends on other factors such as system status, agent capabilities, or environmental characteristics. For instance, a violating car in a traffic jam or on a rainy day will not be able to drive at its internal violation limit.

Finally, we would like to highlight that our model of norm violation is mainly intended to help in the black-box testing of our norm adaptation mechanism, and hence help validate its operation against varying non-compliant behaviours. Thus, the modeling of agents that reason about how to violate norms for their own benefit is an alternative to our approach.

3. CASE STUDY: P2P SHARING NETWORK

Our case study is a simulation of a *peer-to-peer data sharing network* (hereinafter referred to as P2P) in which a set of computers connected to the Internet (*peers*) contact each other to share some data[6]. The system's performance is evaluated in terms of the time required to spread the datum among all peers (t_{spread}), which may increase due to channel saturation. Saturation occurs when a particular channel has reached its maximum traffic-handling capacity, so additional incoming data cannot be transmitted.

We conceptualise the physical network as the packet-switching net in the bottom part of Figure 3. Each peer network adaptor (n_i) is connected to the router ($r_j, where j > 0$) of its internet service provider (ISP). A cluster is defined for all peers connected to

[6]Please, refer to §7 for a discussion on the appropriateness of the chosen scenario.

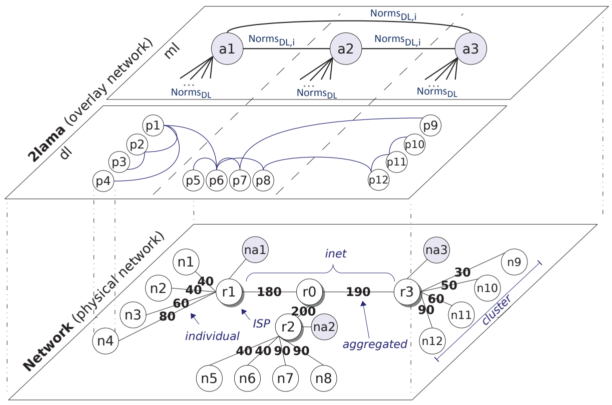


Fig. 3: 2-LAMA over a physical network

the same ISP, which also hosts the assistant's network adaptor (na_j)[7]. The Internet is abstracted as a set of aggregated links among ISPs, so that each one transports all messages from/to each cluster. Thus, bandwidth (BW)[8]resources are shared. As a result, message transmission time depends on the message's length and the channels' actual capacities. This implies that the environment should be dynamic, since peers cannot predict message latencies.

We regard computers (peers) within the P2P system as agents within an OCMAS and specify their model[9] in Equation 16. As shown at the top of Figure 3, the meta-level is composed of a set of assistants (see Eq. 17) and the domain-level is populated by the agents that perform the sharing activity (Eq.18). The meta-level organisation (org_{ML}) defined in Eq. 19 has a single role ($rol_{ML} = \{\texttt{assistant}\}$) and it uses part of a protocol derived from standard BitTorrent protocol ($prot_{ML} = \{\texttt{bitTorrent}'\}$, see §5.1.1). Similarly, domain-level organisation (org_{DL} in Eq. 20) has a single role ($rol_{DL} = \{\texttt{peer}\}$) and uses the same protocol. Both levels share the same organisational goal, which consists in minimising the time required to spread the datum among all agents: $goals_{ML} = goals_{DL} = minimise(t_{spread})$. Domain-level social conventions are specified through two regulations $norm_{DL} = \{NormBW, NormFR\}$ (see Eqs. 21 & 22), which limit the agents' network usage. The first norm prevents peers from making massive use of their bandwidth to send/receive data to/from all other peers. In this Equation 21 $peerPercBW$ is computed as the percentage of nominal bandwidth ($NomBW$ defined below) used to share data and it must be kept under a given norm parameter value ($limitV_{BW}$). On the other hand, second norm limits the number of peers (we refer to them as *friends*) to whom a peer can simultaneously send the data. These regulations are common to DL agents and use the DL environment and agent observable properties that appear in Eqs. 24 and 25. The former specifies environment observable properties ($envP_{DL}$), which are related to the peers' bandwidth and are described in terms of its nominal bandwidth ($NomBW$, i.e. the maximum available), its upload effective bandwidth ($EffUpBW$, i.e. the actual bandwidth consumption), and its actual download effective bandwidth ($EffDnBW$). The latter defines three ob-

[7]Notice, though, that assistants do not represent ISPs, they are just hosted there for a better assistant-peer communication. Instead, an assistant could be hosted by any computer in the network.
[8]The *bandwidth* is the capacity to transfer data over a network link. As a simplification, we assume that upload and download channels have equal BW. Thus, it appears as a single numerical label (corresponding to #data units per time unit) on each link of Fig. 3.
[9]We use upper case to denote the types defined in the formalisation and lower case to denote the instances. Therefore, org_{DL} corresponds to the P2P instantiation of the domain-level organisation in Eq. 4.

servable properties of an agent (agP_{DL}): whether it has the datum ($hasDatum$); which action it is performing (Act); and the number of simultaneous sendings ($numSends$).

$$
\begin{aligned}
2lama_{P2P} &= (ml, dl) \in 2LAMA && (16)\\
ag_{ML} &= \{a_1, ..., a_m\}\ , \quad \text{where } m = 3\ ,\ ag_{ML} \in Ag_{ML} \qquad \textit{assistants} && (17)\\
ag_{DL} &= \{p1 \ldots p_n\}\ , \quad \text{where } n = 12\ ,\ ag_{DL} \in Ag_{DL} \quad \textit{peers} && (18)\\
org_{ML} &= \langle \underbrace{\{\texttt{assistant}\}}_{Rol_{ML} \in Str_{ML}}, \underbrace{\{\texttt{BitTorrent}'\}}_{Protocols_{ML} \in SocConv_{ML}}, \underbrace{\{min(t_{spread})\}}_{Goals} \rangle && (19)\\
org_{DL} &= \langle \underbrace{\{\texttt{peer}\}}_{Rol_{DL} \in Str_{DL}}, \underbrace{\{\texttt{BitTorrent}'\}}_{Protocols_{DL} \in SocConv_{DL}}, \underbrace{\{NormBW, NormFR\}}_{Norm_{DL} \in SocConv_{DL}}, \underbrace{\{min(t_{spread})\}}_{Goals} \rangle && (20)\\
NormBW &: \begin{array}{l} \texttt{IF}\ (peer.Act \in \{receive, serve\})\ \texttt{THEN}\ Prohibition(peerPercBW > \texttt{limitV}_{\texttt{BW}}) \\ where \quad peerPercBW = max(\frac{EffUpBW}{NomBW}, \frac{EffDnBW}{NomBW}) \end{array} && (21)\\
NormFR &: \texttt{IF}\ (peer.Act \in \{serve\})\ \texttt{THEN}\ Prohibition(numSends > \texttt{limitV}_{\texttt{FR}}) && (22)\\
env_{ML} &= \langle envP_{ML}, envP_{DL}, agP_{DL}, org_{DL} \rangle\ , \qquad envP_{ML} = \emptyset && (23)\\
env_{DL} &= envP_{DL} = \langle NomBW, EffUpBW, EffDnBW, t_{spread} \rangle && (24)\\
agP_{DL} &= \langle hasDatum, Act, numSends \rangle \ \ where && (25)\\
& \quad hasDatum = \{1, 0\}, Act = \{no, receive, serve\}, numSends = [0, .., m-1] && (26)
\end{aligned}
$$

4. LEARNING NORM ADAPTATION USING CASE-BASED REASONING

With the aim of achieving system goals, assistants apply their individual norm adaptation function α_i^N (see Eq. 12) to propose new regulations. They do so at run time —at specific time intervals (t_{adapt}) and without bringing the system down. Nevertheless, the computation of this function is complex. The reason is two-fold: first, the actual optimal regulations (i.e., the solution) are unknown; second, the overall system performance depends on the consecutive norm adaptations performed at run time. This is so because dynamic systems require to set different regulations along a single execution (i.e., different system states require different regulations). Since performance is measured at the end of the execution (i.e., when the goal is achieved), it is difficult to assess the actual influence of each regulatory change in the overall performance. In the literature this is usually referred as the *credit assignment problem* [Jones and Goel 2004]. Therefore, we advocate using a machine learning approach to compute α_i^N. Specifically, we propose a variation of a standard case-based reasoning (CBR) method. CBR [Riesbeck and Schank 1989] is based on the assumption that similar problems have similar solutions. Thus, similar system states require similar regulations (provided that the ones experienced were successful). Furthermore, it can deal with very complex forms of knowledge and generalises fast from few (even noisy) training examples. Nevertheless, one may think of other alternatives to norm adaptation, such as, for example, model-based heuristics. In fact, in [Campos et al. 2011] we evaluated an uncompleted CBR with a heuristic approach. Even with the limitations of the previous CBR proposal, it showed a significant improvement on accuracy in comparison to the heuristic proposal.

Concretely, this paper contributes to previous work to (i) extending the previous case representation so as to include a case quality estimation measure; (ii) considering continuous state/action spaces; (iii) refining previous retrieval and reuse phases so as to better exploit the new information contained in the cases; and (iv) including for the first time the revise and retain phases in order to complete the CBR cycle. The revise phase evaluates the effectiveness of a case whereas the retain phase uses the evaluation to enrich the system's knowledge with the newly acquired experience.

Briefly, the main characteristic of a CBR is that it solves new problems (cases) by adapting the solutions of similar past problems, which are stored in the *case base*. Aamodt and Plaza [1994] described the classical CBR cycle in four different phases. As depicted in Figure 4a, a new case (*case_n*) is solved by first *retrieving* the most similar case from the case base, second, *reusing* its solution in *case_n*, *revising* this solution, and finally, *retaining* the new experience (*case_n*) by incorporating it into the case base. Furthermore, Richter [1995] defined the case base as a structure that incorporates further knowledge about the domain, similarity and adaptation functions.

4.1. Tailored case-based reasoning for regulation adaptation

Each assistant computes its α_i^N by applying a tailored CBR cycle. The main differences with classical CBR are that our tailored CBR cycle starts with an empty case base and gathers its experience on-line from the MAS scenario (i.e. the domain-level). As a consequence, our CBR model incorporates heuristic knowledge[10] to tackle the *cold-start* problem –i.e., the lack of previous (off-line) experiences (cases)– and an estimation of the quality of cases for the *credit assignment problem*.

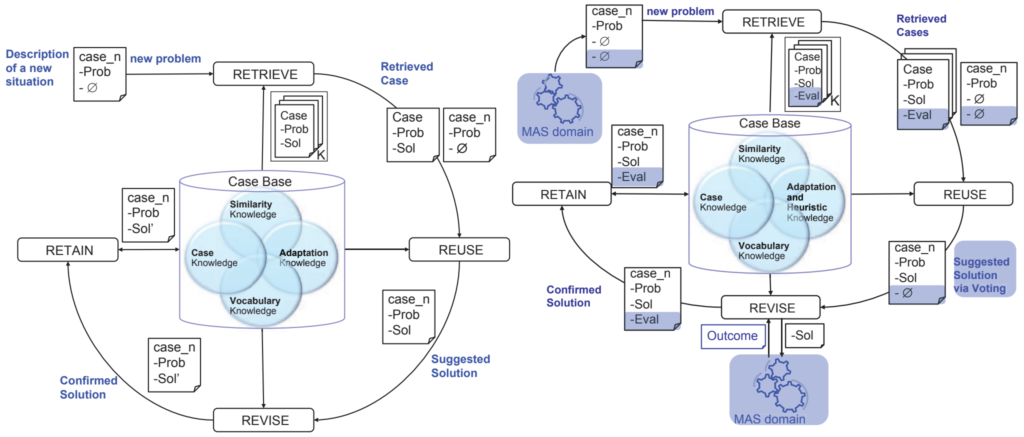
Every time an assistant computes a regulation adaptation (i.e., α_N^i in Eq. 12), it follows our tailored CBR cycle in Figure 4b. Thus, it extracts a new problem description ($case\_n.Prob$) from the scenario. With this new problem, it applies the retrieval phase to look for a set of K most *similar* cases from the case base. Next, the assistant moves to the second phase to *reuse* the retrieved cases to provide a solution to $case\_n$. This solution contains the (new) regulation values ($Norms_{DL,i}$) that the assistant will propose to the other assistants in order to establish the global regulation parameter values $Norms_{DL}$. In this manner, each assistant participates in the agreement process (β_{α^N}, see Eq. 13), which is implemented as a voting process. More specifically, each assistant sends its vote (i.e., the individually proposed regulation value) to the rest of assistants, and the most voted values are the ones finally chosen[11]. As a result, the assistant stores the agreed upon solution in the case ($case\_n.Sol$) and sends the updated regulations to its domain level agents so to be applied in the scenario. After a certain time interval (t_{adapt}), the assistant continues with the third CBR phase to *revise* the outcome of applying the new regulations to the scenario and it introduces the corresponding evaluation ($case\_n.Eval$). Finally, the assistant applies the fourth phase to *retain* $case\_n$ if it is considered representative enough. Next subsections are devoted to provide both the details about the structure of a case in our P2P domain as well as about the phases of our tailored CBR cycle.

4.2. Case description

As Figure 4 details, the case base contains case knowledge, which materializes as a set of cases. A case represents an abstraction of a concrete problem situation. Currently, the proposed case description represents an abstraction of a domain-level state. In classical CBR, a case contains two components: the problem ($Prob$) described as a set of attributes ($ProbAttribs = \{a_1, \cdots, a_f\}$) and its solution ($Sol$) represented as a set of attributes ($SolAttribs = \{s_1, \cdots, s_z\}$). We propose the addition of a third compo-

[10]It is a coded algorithm to adapt regulations that was written by the system designer. This heuristic is fully described in [Campos et al. 2011] and further commented in 4.3.

[11]Although the majority rule agreement mechanism is very simple (and other negotiation mechanisms could be considered instead), it may be worth mentioning that some votes can be blank ballot-papers (see BLANK in 4.2 and 4.4), which are discarded in the voting process. Assistants use a blank ballot paper to let other assistants push for their own interests, and thus, just those assistants in need for a change are actually considered in the decision process. Moreover, if there is a tie, norms are not updated.



(a) Classical CBR (b) Tailored CBR

Fig. 4: Case-Based Reasoning cycle

nent called $Eval$ that represents how well the case solution has been evaluated in the scenario. Hence, the case is defined as a tuple (ProbAttribs, SolAttribs, Eval), where $ProbAttribs$, $SolAttribs$ and $Eval$ are sets of attributes. Figure 5 illustrates a $Case$ in our P2P scenario.

The first component corresponds to the description of the problem, and consists of seven numeric attributes derived from env_{ML} information in Equation 23. Specifically, $ProbAttribs = \{Completeness, SeedBW, LeechBW, SrvBW, RcvBW, RcvEffBW, Waiting\}$, where:

— $Completeness$ expresses the percentage of datum possession (e.g. 100% means all peers have the datum);
— $SeedBW$ and $LeechBW$ correspond to the sum of the nominal bandwidth of peers having and lacking the datum respectively (i.e. seeds and leeches[12]);
— $SrvBW$ and $RcvBW$ define the sum of nominal bandwidths for currently serving seeds and receiving leeches;
— $RcvEffBW$ contains the sum of effective download bandwidths; and,finally,
— $Waiting$ is the number of non-receiving leeches.

Formally:

$$Completeness = \frac{\sum_{i=1}^{n} hasDatum_i}{n}, \qquad SeedBW = \frac{\sum_{i=1}^{n} hasDatum_i \cdot NomBW_i}{\sum_{i=1}^{n} NomBW_i},$$
$$LeechBW = \frac{\sum_{i=1}^{n} (1-hasDatum_i) \cdot NomBW_i}{\sum_{i=1}^{n} NomBW_i}, \quad SrvBW = \frac{\sum_{i=1}^{n} NomBW_i | Act_i = \{serve\}}{\sum_{i=1}^{n} NomBW_i},$$
$$RcvBW = \frac{\sum_{i=1}^{n} NomBW_i | Act_i = \{receive\}}{\sum_{i=1}^{n} NomBW_i}, \quad RcvEffBW = \frac{\sum_{i=1}^{n} EffDnBW_i}{\sum_{i=1}^{n} NomBW_i},$$
$$Waiting = \frac{\sum_{i=1}^{n} \{p_i \in Ag_{DL} | Act_i = \{no\}\}}{n}$$

The solution component (Sol) is described by two numeric attributes $SolAttribs$ $= \{MaxFR, MaxBW\}$ that specify regulation adaptation parameters in Equations 21 and 22. $MaxFR = limitV_{FR}/n$, corresponds to the updated $\texttt{limitV}_{\texttt{FR}}$ (i.e., limit of the number of friends to send the data simultaneously) normalised by the number of peers. $MaxBW = limitV_{BW}/100$ is the updated $\texttt{limitV}_{\texttt{BW}}$ (i.e., limit

[12]The term *seed* is used to refer to a peer who has 100% of the data whereas a *leech* refers to a peer who lacks the data. When a *leech* obtains 100% of the data, that peer by definition becomes a *seed*.

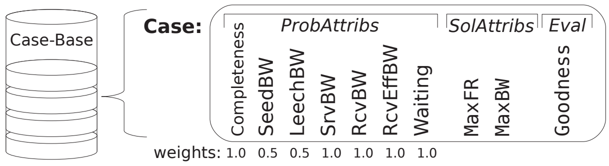


Fig. 5: CBR's case description

of the bandwidth usage for sending/receiving data) expressed as a percentage. Both attributes represent votes in the agreement process β_{α^N} and they may alternatively contain a special value representing a *blank-ballot paper* (BLANK). Thus, an assistant may vote BLANK when no peers in its cluster are waiting for data and other clusters still contain waiting peers. This will allow the interests of the corresponding assistants to prevail.

Finally, the evaluation ($Eval$) corresponds to a numeric $Goodness$ attribute which provides an estimation of how effective the corresponding solution is. Briefly, it considers the increment of $Completeness$ in order to estimate whether the pace of datum spreading is adequate (see §4.5 for computational details).

4.3. Retrieve

The first phase of our tailored CBR cycle (see Figure 4b) is devoted to retrieving the K most similar cases to the new problem (case). We use a K-nearest-neighbour algorithm that traverses the case base computing the similarity of each stored case with the new problem. Given problem descriptions from a new and a stored case ($case\_n.Prob, ret\_case.Prob \in Prob$), the similarity function ($\Theta : Prob \times Prob \rightarrow [0..1]$) in Eq. 27 computes a weighted Euclidean distance between the corresponding problem attributes ($a_i^{case\_n}$,$a_i^{ret\_case} \in ProbAttribs$). Specifically, the distance depends on wether the attribute is numerical or nominal (i.e. categorical), see Eq. 28. The values for the weights (w_i^{Θ}) in Eq. 27 appear in Figure 5 below the case. They were computed using the Proportional Rough Sets (PRS) method [Salamó and López-Sánchez 2011b] which estimates attribute relevance based on Rough Sets Theory[13]. In fact, since proximity and distance are opposite concepts, we use one minus the distance[14] to compute similarity.

$$\Theta(case\_n, ret\_case) = 1 - \sqrt{\frac{\sum_{i \in ProbAttribs} \left(w_i^{\Theta} \cdot dist(a_i^{case\_n}, a_i^{ret\_case})\right)}{\sum_{i \in ProbAttribs} w_i^{\Theta}}} \tag{27}$$

$$dist(a_i^{case\_n}, a_i^{ret\_case}) = \begin{cases} (a_i^{case\_n} - a_i^{ret\_case})^2 & \text{if } a_i \text{ is numerical} \\ 1 & \text{if } a_i \text{ is nominal and } a_i^{case\_n} = a_i^{ret\_case} \\ 0 & \text{if } a_i \text{ is nominal and } a_i^{case\_n} \neq a_i^{ret\_case} \end{cases} \tag{28}$$

Additionally, we apply a *minimum similarity* (MIN_SIM) threshold. This, together with the initial empty case base, may result in failed retrieval. The retrieve phase then obtains a solution by applying heuristic knowledge [Campos et al. 2011]. Thus, when there are no experiences to reuse, the heuristic knowledge is used to align the serving

[13]We compute PRS over a case base that comes from a preliminary CBR test without weights. PRS has also been successfully applied for feature selection in classification problems.
[14]This is inspired by complement computation in Boolean algebra.

and receiving capacities. However, once there exist some previous cases, the use of the heuristic is marginal (our experiments showed that for an average number of 169.7 cases, just 1.3% came from the heuristic).

4.4. Reuse

The reuse phase in Figure 4b employs the solutions of the retrieved cases to provide a solution to the new case ($case\_n$). First, it filters the retrieved cases by applying a *divergence* threshold (`MAX_DIV`) over the standard deviation of their $MaxFR$ solution attribute. Thus, the divergence of a single retrieved case is 0 and it increases proportionally to the variation of the values of the different proposed solutions in the set of retrieved cases. As a result, this process excludes solutions that are too contradictory to provide a good solution for $case\_n$ (again, heuristic knowledge covers the lack of solutions).

ALGORITHM 1: Adaptation in the reuse phase

```
1 def adaptation( retrievedCases, case_n ):
2   valuesFR=∅;  valuesBW=∅;  goodness=∅
3   for each case ret_case ∈ retrievedCases:
4     valuesFR = valuesFR ∪ {ret_case.Sol.MaxFR}
5     valuesBW = valuesBW ∪ {ret_case.Sol.MaxBW}
6     goodness   = goodness ∪ {ret_case.Eval.Goodness}
7   case_n.Sol.MaxFR = wAverage(valuesFR, goodness)
8   case_n.Sol.MaxBW = wAverage(valuesBW, goodness)
```

Secondly, it adapts filtered cases by applying Algorithm 1, which averages the solutions weighted by their individual evaluation. Specifically, line 2 in Algorithm 1 initialises the sets that will contain each solution's attributes (`valuesFR`, `valuesBW`) and its evaluation (`goodness`). Next, it traverses the retrieved cases (see lines 3 to 6) to fill these sets. Finally, in lines 7 and 8, it computes each solution attribute of $case\_n$ by invoking the $wAverage$ function. This function computes a weighted average and is formalised in Equation 29. It receives a set of values (V, i.e. `valuesFR` or `valuesBW`) and their associated weights determined by their goodness (G). Values can contain the blank-ballot paper (`BLANK` introduced in §4.2). Therefore, if the aggregated goodness of blank values is greater than the aggregated goodness of non-blank values, then the result of $wAverage$ is a `BLANK`. Otherwise, the result is a weighted average of non-blank values. Notice that all summations in this formula apply for all pairs of received values and their associated goodness $(v_i, g_i) \in (V, G)$.

$$wAverage(V,G) \;=\; \begin{cases} \mathtt{BLANK} & if\ (\sum(g_i|v_i = \mathtt{BLANK}) > \sum(g_i|v_i \neq \mathtt{BLANK}) \\ \frac{\sum(v_i \cdot g_i|v_i \neq \mathtt{BLANK})}{\sum(g_i|v_i \neq \mathtt{BLANK})} & otherwise \end{cases} \tag{29}$$

Following the CBR cycle depicted in Figure 4b, the reuse phase employs the solutions of the K retrieved cases to provide a solution to the new case, $case\_n$. Our reuse phase starts by checking if the divergence of retrieved solutions is greater than a *maximum trusted divergence* (`MAX_DIV`) threshold. This divergence is computed as the standard deviation of $MaxFR$ solution's attribute[15] of the retrieved cases. Thus, the divergence of a single case is 0 and exceeding `MAX_DIV` means that the solutions of retrieved cases are too contradictory to provide a good solution for $case\_n$. When `MAX_DIV` is exceeded, the heuristic knowledge is invoked to obtain a solution. Once there is a set of slightly divergent retrieved cases it performs an adaptation of the solution of these

[15]We do not consider $MaxBW$ because our experiments reveal that is correlated with $MaxFR$ and in fact changes of $MaxFR$ are more relevant than those on $MaxBW$.

cases. This adaptation can be tackled considering all retrieved solutions but also the differences between the retrieved problems and the current one ($case\_n$). Currently, our *adaptation* function only uses the former as shown in Algorithm 1. Briefly, the *adaptation* function receives a set of retrieved cases (`retrievedCases`) and the current problem (`case_n`) and it returns a solution to $case\_n$ that is the average of retrieved cases' solutions weighted by their evaluation.

4.5. Revise

The regulation adaptation starts periodically (every t_{adapt}), when each assistant performs a CBR cycle. Figure 4(b) shows that the reuse phase provides the assistant with a suggested solution to $case\_n$, based on past experience. Next, as previously explained in section 4.1, each assistant participates in the agreement process (β_{α^N}, see Equation 13) by sending its vote for each regulation ($Norms_{DL,i}$) to the other assistants (see top of Figure 3). Then, each assistant receives other votes and computes the most frequent vote for each regulation ($Norms_{DL}$), discarding blank ballot-papers. Note that if there is a tie, the regulations are not updated. Last, each assistant communicates new regulations to its peers. As a result, new regulations are used during the next adaptation time period (t_{adapt}).

Once the t_{adapt} interval has finished, the revise phase evaluates the results of the applied solution in the scenario (see Figure 4(b)). In the P2P scenario, the current $Goals$ definition (see Eq. 19) is related to the total spread time (t_{spread}), which is unknown until the end of the whole sharing process. Alternatively, we define $Goodness \in [0..1]$ as an evaluation metric that is based on datum possession ($Completeness$, see §4.2). The rationale it follows is: if the increment in $Completeness$ (c_{inc}) along an adaptation interval (t_{adapt}) is large, then a significant number of peers have increased their percentage of the datum, so a short t_{spread} is expected.

Specifically, as shown in Equation 30, this $Goodness$ metric is the $Completeness$ increment (c_{inc}) normalised by the difference between the best conditions possible (max_{inc}) and the worst situation (min_{inc}) of c_{inc} that could be obtained during the adaptation interval. The max_{inc} value refers to the maximum c_{inc} if all seeds serve and leeches also receive at their nominal bandwidth. The min_{inc} value indicates the minimum c_{inc} if all data transmissions are cancelled. Overall, cases with the largest goodness are expected to provide better solutions. Accordingly, the revise phase adds this evaluation to the current problem ($case\_n.Eval.Goodness$) and it can be used later by the reuse phase to weigh different retrieved solutions (see previous §4.4 and Fig. 4b).

$$case\_n.Eval.Goodness = \frac{(c_{inc} - min_{inc})}{(max_{inc} - min_{inc})} \tag{30}$$

4.6. Retain

Retention is the final phase in the CBR cycle, in which the product of the most recent problem-solving episode is incorporated into the system's (in our case, assistant's) knowledge. As described in [Salamó and López-Sánchez 2011a], a case-based reasoning system can adapt its own case base during the reasoning cycle by adding and removing cases. Different retention and forgetting strategies that use a measure of "case goodness" were proposed for evolving correctly the case base of a classifier system. Here, we follow on this vein but introducing variations for a multi-agent system. In the current domain, apart from the inherent dynamics of a MAS, we have introduced self-interested agents that may decide not to comply with established regulations. These two factors lead to the case base changing over time. Moreover, as previously mentioned, the CBR system starts with an empty case base and gathers its experience from the MAS scenario. As a result, the case base needs to increase in size rather than forget cases. Thus, in this paper, we focus on the retention and propose a strategy to update the "goodness" of cases.

In our MAS, each assistant starts the retention strategy once a solution has been evaluated (see the last phase in Figure 4b). In particular, $case\_n$ is stored as a new case when its solution was created using heuristic knowledge or using more than one retrieved case. Notice that these two situations depict that the case base lacks some knowledge. In contrast, when a solution comes from a single similar case, $case\_n$ is not stored and the evaluation of this retrieved case (ret_case) is updated by following an approximation of a Monte Carlo strategy [Sutton and Barto 1998] described in Equation 31. We have defined the goodness as a Markov Decision Process and, in this case, we have approximately solved it by replacing the sum over all states (i.e., each CBR cycle generates a new state) with a Monte Carlo approximation. This way, the new goodness stored in the case base ($ret\_case.Goodness'$) is the previous one ($ret\_case.Goodness$) updated according to current experience ($case\_n.Goodness$) and the given learning rate constant (α_g).

$$ret\_case.Goodness' = \alpha_g \cdot case\_n.Goodness + (1 - \alpha_g) \cdot ret\_case.Goodness \quad (31)$$

Notice that the goodness of a case will be used later for the reuse phase (see Section 4.4). Instead of forgetting cases based on a goodness measure, since the MAS lacks experience, we use it for promoting cases when solving a problem. The higher the goodness of a case, the higher its relevance to choose the solution to the current new problem.

5. EMPIRICAL EVALUATION

To demonstrate that regulation adaptation helps OCMAS, this section evaluates the performance of our CBR proposal in a series of experiments with different coordination models and with the presence of non-compliant agents. Experiments are conducted with our P2P-adaptive OCMAS simulator [Campos et al. 2009a].

5.1. Coordination models and setup

Our simulator considers both agents and network components and allows us to execute different sharing methods with identical initial conditions. Specifically, we evaluate the three coordination models in Table I. Firstly, as the baseline, we take a non-adaptive approach that uses a simplified version of the BitTorrent protocol (see below). This has a flat social structure in which all peers contact each other with unlimited bandwidth usage. The coordination mechanism is fixed and restricts the number of leeches to three that a seed can start serving simultaneously. Secondly, we test a 2-LAMA-RAW model of our suggested architecture, with the configuration[16] shown in Fig. 3. This is non-adaptive and takes fixed regulations that are equivalent to BT ($\texttt{limitV}_{\texttt{BW}} = 100\%$ and $\texttt{limitV}_{\texttt{FR}} = 3$), so that the comparison is fair. Thirdly, 2-LAMA-CBR starts with the same conditions, but assistants use CBR to adapt regulations at run-time. This requires some additional parameters: minimum similarity threshold MIN_SIM=0.65; learning rate $\alpha_g = 0.1$; and adaptation interval $t_{adapt} = 50$ time steps, which is large enough to let new regulations influence agents' behaviour before evaluating their effects. Finally, all messages in all coordination models are of size 1, except for data messages, whose size is always 5000 data units.

5.1.1. Protocols. Nowadays BitTorrent (BT, [BitTorrentInc. 2001]) is one of the most widely used protocols in P2P sharing network scenarios. We have implemented a BT

[16]The overlay network in Fig. 3 has three clusters –each having an assistant–, and numerical labels in the physical network indicate upload and download bandwidths. Different network topologies have been studied in [Campos et al. 2010].

Coordination models	Regulation parameters $limitV_{BW}$	$limitV_{FR}$	Adaptive?
BT	≅100	3	No
2-LAMA-RAW	100	3	No
2-LAMA-CBR	(0..100]	[1..11]	Yes

Table I: Coordination models and their regulation parameters.

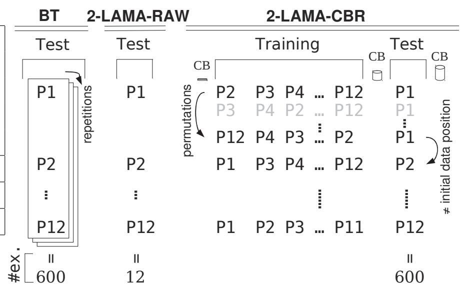


Fig. 6: Performance tests.

simplification [Campos et al. 2009b] where peers share a single-piece datum, the rest is equivalent. Thus, there is a single agent (tracker), which informs peers about other connected peers. Peers without data (leeches) request the data to peers having the datum (seeds), which in response, send a `choke` message. Then, at specific unchoke intervals, seeds choose chocked leeches to send the data (`unchoke`, `request`, `data` messages are then interchanged). Leeches are chosen considering time of request and upload bandwidth (more recent and more bandwidth preferred). On the other hand, our 2-LAMA (both RAW and CBR) protocol varies BT to include our distributed assistance meta-level. Specifically, in 2-LAMA there is no tracker and peers just follow a handshake phase (with `bitfield` messages involved) with a subset of seeds suggested by their assistants. Assistants compute this social structure ($SocStr_{DL}$)[17] based on datum possession and communication latencies that are measured (estimated) by peers at an initial phase (see [Campos et al. 2011] for further details). Afterwards, during the data sharing, seeds inform their assistants when they are completed, which in turn spread this information both at the meta and domain levels. Finally, assistants in 2-LAMA-CBR share messages regarding norms.

5.2. Design of experiments

We test all coordination models by varying the peer that initially has the datum. As Figure 6 shows[18], the number of performed executions depends on the model properties. Thus, BT selects some served peers randomly. Therefore, the results show the average of repeating 50 times the execution with the data in each (of 12) possible initial position (i.e. $12 \times 50 = 600$). In contrast, 2-LAMA-RAW only needs to be executed once in each initial position, since it lacks randomness. However, the CBR approach requires training. Hence, it is trained with executions in which the data is initially located at 11 different positions and tested against the remaining position. Nevertheless, the order in the initial serving peers during training affects learning. Therefore, we tested 50 random permutations out of all possible permutations of the 11 positions (i.e. $11!$). Consequently, the results of 2-LAMA-CBR show the average of executing 50 times each test of initial position (i.e. $12 \times 50 = 600$ executions).

A second set of experiments was used to check the robustness of our 2-LAMA coordination model against non-compliant agents. In these experiments, some DL agents do not fulfil the regulations proposed by the assistant agents —e.g. they simultaneously serve more peers than allowed by the $\mathtt{limitV_{FR}}$ limit. As described in §2.3, we model non-compliant agents by introducing $pVio$ and $dVio$ parameters (see eq. 14 for percent-

[17] Recall that that BT has a flat social structure in which all peers contact among them.
[18] In this figure, each Px, x∈[1,12] stands for an execution where peer px initially has the datum. Moreover, numbers below indicate the number of experiments (#ex).

pVio	dVio 0.3	0.7	1.0
0.3	ex·vSets	ex·vSets	ex·vSets
0.7	ex·vSets	ex·vSets	ex·vSets
1.0	ex·1	ex·1	ex·1

Table II: Robustness tests

pVio	dVio 0.3	0.7	1.0
0.3	120	120	120
0.7	120	120	120
1.0	12	12	12

Table III: 2-LAMA-RAW

pVio	dVio 0.3	0.7	1.0
0.3	6000	6000	6000
0.7	6000	6000	6000
1.0	600	600	600

Table IV: 2-LAMA-CBR

age of violators and eq. 15 for the degree of violation). In order to keep an affordable number of simulations, we consider just four possible situations: norm compliant, median violation, high violation, and full violation agents. These situations have been named specifically in the degree of violation as none, median, high, and full. Concretely, the set up of these values in our experiments corresponds to 0, 0.3, 0.7, and 1, respectively. When $pVio = 0$ or $dVio = 0$ (i.e., agents are norm compliant), the total number of executions corresponds to the $\#ex$ in Fig. 6. Table II shows the number of experiments for non-zero values. When not all 12 peers violate (medium and high violation degrees defined as $0 < pVio < 1$), different sets of violators ($vSets$) can appear. For instance, for $pVio = 0.3$, there are $\binom{12}{4}$ different sets of 4 ($12 \cdot 0.3 \approx 4$) violator peers. Accordingly, the resulting number of experiments is $\#ex \cdot vSets$. We restrict our experiments to $vSets = 10$. Notice that when all peers violate (full violation with a $pVio = 1$ value) permutations do not apply. Tables III and IV show the total number of robustness experiments for our 2-LAMA-RAW and 2-LAMA-CBR coordination models. In particular, 2-LAMA-CBR involves 38,400 test executions ($38400 = 600 \cdot 4 + 6000 \cdot 6$) and their corresponding 422,400 training executions ($422400 = 38400 \cdot 11$, see Figure 6).

5.3. Performance evaluation

This section is on analysing whether the adaptation of regulations over time can improve the performance of non-adaptive coordination models. Performance in our experiments is measured in terms of the total *Time* required to spread the datum among all peers. Furthermore, we measure two additional execution characteristics of the network usage: the *network cost* consumed by all messages (each message cost is computed as its length times the number of links it traverses), and the average number of links traversed by each message (*Hops*). Table V shows the results averaged over the number of executions detailed in §5.2.

If we compare the performance of BT and 2-LAMA coordination models, we see that both 2-LAMA approaches require less *Time* to share the datum. This means that the time invested in communicating with our suggested meta-level is less than the time benefit of having this additional level. In contrast, the *network cost* is higher in 2-LAMA. This shows that, in our approaches, the network is intensively used throughout the execution without achieving saturation, otherwise the time would increase. Having a meta-level means that coordination messages are exchanged among peers and assistants and between assistants. However, the network usage increment is mainly caused by transmitting more data messages (recall they are 5000 data units long whereas the size of any other message is 1). These extra data messages are created because 2-LAMA peers compare data sources by retrieving some data from them: they replace their current data source whenever they find a faster one. Thus, we expect to minimise the network consumption when dealing with more than one piece of data, since peers could compare sources depending on the pieces previously retrieved. Regarding the number of links traversed by messages (*Hops*), our 2-LAMA approaches have more local communications (i.e. intra-clusters) than BT. This is convenient because local messages have lower latencies and costs, since they are usually performed within the same cluster.

	Time	*Network cost*	*Hops*
BT	986.2	206,592.0	3.42
2-LAMA-RAW	793.1	338,448.3	3.22
2-LAMA-CBR	732.6	348,399.6	3.25

Table V: Averaged performance results.

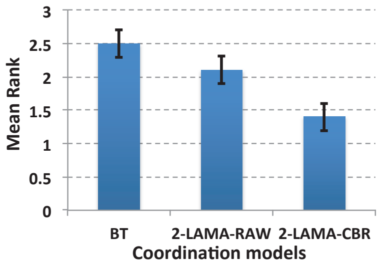


Fig. 7: Performance Nemenyi test.

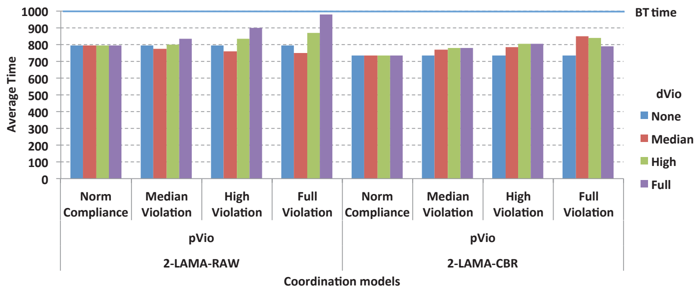
The comparison of 2-LAMA-RAW and 2-LAMA-CBR that is shown in Table V denotes that 2-LAMA-CBR is more accurate in terms of time and uses the network more intensively than 2-LAMA-RAW. However, 2-LAMA-CBR slightly augments the number of links traversed by the messages. This result was expected since the assistants in different clusters exchange more messages, for example, to send their votes for arriving to a consensus solution.

We also analyse whether there are significant differences between approaches [Demšar 2006]. First, we rank alternative algorithms considering all the experiments ($k = 3$ algorithms and $N = 600$ different experiments[19] for each test). The best algorithm's performance is ranked 1, the second best ranked 2, and so on. Then, a method's mean rank is obtained by averaging its ranks across all experiments. Second, we apply the Friedman and Nemenyi tests to analyse statistically significant differences between approaches. The Friedman [1937] test (F_F) is distributed according to the F distribution with $(3 - 1) = 2$ and $(3 - 1) \cdot (600 - 1) = 1198$ degrees of freedom. The critical value of $F(2, 1198) = 4.63$ is computed at the 0.01 critical level. Our experiments obtained the value of $F_F = 363.6$. As this value is higher than 4.63 we can reject the null hypothesis. Once the non-randomness of the results has been checked, we compute the Nemenyi test to find out which approaches are significantly different. We compare k algorithms with a critical value $\alpha = 0.01$, $q_{0.01} = 2.913$ for a two-tailed Nemenyi test and we obtain a critical difference value of $CD = 0.1682$. The Nemenyi results are illustrated in Figure 7, where diamonds represent the mean ranks of each coordination model. Vertical lines across diamonds indicate the 'critical difference'. The performance of two approaches is significantly different with a confidence of 99%, if their corresponding mean ranks differ by at least the critical difference (i.e. vertical lines are not overlapping). Accordingly, Figure 7 reveals that the best mean rank corresponds to 2-LAMA-CBR which is significantly better than 2-LAMA-RAW and BT.

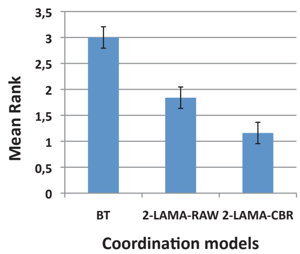
5.4. Robustness when agents violate organisational regulations

Once shown that our adaptive coordination model improves system performance, we analyse its robustness against non-compliant agents. Figure 8a depicts the results of the experiments described in §5.2. The results of the *non-adaptive approach* (2-LAMA-RAW) are on the left and the results for the *adaptive approach* (2-LAMA-CBR) appear on the right. Each part has a set of bars for each *percentage of violators* ($pVio$), which contains a bar for each *degree of violation* ($dVio$). BT is depicted as a constant line at 968.2 ticks (see Table V). We can observe that both

[19]In 2-LAMA-RAW there are 12 experiments and their results have been repeated 50 times to be able to compare them with the rest of the approaches.



(a) Performance



(b) Nemenyi Test

Fig. 8: Robustness results

2-LAMA approaches clearly outperform BT and that, in general, the adaptive approach (2-LAMA-CBR) performs better (i.e. has a lower Time) than the non-adaptive one (2-LAMA-RAW). As expected, it is generally the case that the sharing time increases when the violation rate (a combination of $pVio$ and $dVio$) grows, e.g. see how 2-LAMA-CBR bars clearly increase from $pVio$ $=medium\ violation$ to $pVio\ =\ high\ violation$. Nevertheless, as the alert reader will have noticed, in the extreme situation where the whole agent population deviates completely from the norms, performance improves. The reason is two-fold: firstly, assistants perform norm adaptation based on how the system is performing given current system dynamics rather than based on the actual norm violations; and secondly, we are dealing with a complex dynamic environment where an extensive norm deviation (i.e., when $pVio$ $=full\ violation$) has ceiling effects in the network saturation. Thus, it just happens to be the case that assistants better regulate some system situations than others. Furthermore, 2-LAMA-CBR presents a steadier time variation than 2-LAMA-RAW. For example in the worst situation (i.e., when there is a full violation degree), 2-LAMA-CBR enlarges the time required to spread the datum a maximum of 16% whereas 2-LAMA-RAW increases facing that situation a 24%. Our results denote that 2-LAMA-CBR is robust as it is able to better cope with violation agents.

Finally, we analyse the significance of the differences between these approaches. Again, we apply the Friedman and Nemenyi tests with k=3 algorithms and N=64 different experiments[20]. We compute the mean rank of each algorithm. In the Friedman tests, the critical value is $F(2, 126) = 4.77$ at the 0.01 critical level. We obtained the value of $F_F = 330.8$ so that we could reject the null hypothesis. In the Nemenyi test, with a confidence of 99%, we obtain a critical difference value of $CD = 0.52$. Figure 8b illustrates the results of the Nemenyi tests, where, the best performance again corresponds to the 2-LAMA-CBR. In summary, the results show that the adaptive coordination model outperforms the non-adaptive ones, even when agents violate regulations. In other words, we have empirically proven that our 2-LAMA-CBR is robust enough to changes in agent population behaviour.

6. RELATED WORK

Multi-agent systems (MAS) can be either implicitly or explicitly organised. Agents within Agent Centred MAS (ACMAS) [Serugendo et al. 2006] coordinate by means of

[20]We consider 1 experiment when there are no violators ($pVio = 0$ or $dVio = 0$), 3 experiments when all peers are violators ($pVio = 1$ and $dVio \geq 0.3$) and 60 experiments when there are some violators ($0 < pVio < 1$ and $dVio > 0$ for the $vSets = 10$ violator sets).

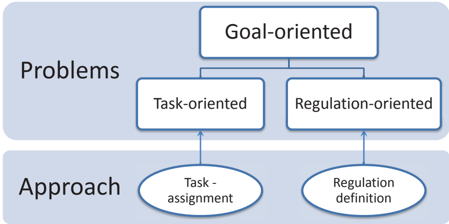


Fig. 9: OCMAS taxonomy of problems and associated solution approaches.

an implicit (i.e., non specified) organisation, whereas Organisation Centred MAS (OCMAS, as described by [Ferber et al. 2004]) use an explicit one, so that the organisation is a first-class entity in the system.

On the one hand, as stated by [Sen and Sen 2010], implicit organisations in ACMAS may change as a consequence of different agent interactions or based on agent observation. Thus, for example, [Mukherjee et al. 2008] show how agents converge to new behaviours by just considering individual utilities. In contrast, [Sen and Airiau 2007] propose organisational changes that may emerge when agents observe other agents' behaviours. On the other hand, in OCMAS, the explicit specification of an organisation -and its regulative structures- favours the reasoning required to guide the adaptation process. Thus, as [Martin and Barber 2006] maintain, "Implementing organizational restructuring at a high level of abstraction first requires developing high-level computational descriptions of the organizational structure. [...] [These] descriptions can be manipulated dynamically to implement organizational restructuring."

Generally, we can regard OCMAS as *goal-oriented* systems (see Figure 9). We distinguish between *task-oriented* and *regulation-oriented* approaches depending on how system goals are achieved. The former identifies tasks that accomplish system goals and decomposes them into subtasks that are subsequently assigned to agents [Dignum 2004; Gomez-Sanz and Pavon 2005; Hübner et al. 2005]. As a consequence, agents receiving task assignments need to somehow assimilate task executions into their individual goals. Alternatively, *regulation-oriented* approaches use regulations to establish limits on agents valid actions with the aim of accomplishing system goals. We take the latter approach, since it assumes individual and system goals to be loosely coupled (or even independent) and preserve agents' autonomy. Consider, for instance, a traffic scenario, where individual goals may be to minimise driving times whereas the system dictates speed limits in order to avoid car accidents.

In general, any fixed organisation may not perform equally well for different runtime situations[21]. Thus, organisational adaptation in OCMAS is a relevant research topic in the OCMAS community. Indeed, it has generated a significant amount of research [Boissier and Gâteau 2007; Sims et al. 2008; Costa and Demazeau 1996; Hübner et al. 2004; Guessoum et al. 2004; Kota et al. 2008]. Just to mention a few, [Carley et al. 2002] explore the resiliency of organisations by studying their performance when key leaders are removed. Similarly, [Dignum et al. 2005] evaluate the VILLA environment by defining a hunter-gatherer community. There, behavioural reorganisation can be modelled by changing agent attributes —e.g., gather power of gatherers. VILLA also considers structural reorganisation strategies through the variation of the number of agents playing a giving role —so for example, hunters and/or gatherers are

[21]This claim is in line with the computational organisational theory, which demonstrates that the best organisation designs are domain and context dependent [Carley 1995].

added/removed by changing the capabilities of other creatures. These research studies have in common that they follow a task-decomposition approach and that they perform organisational reconfiguration [Martin and Barber 2006], in which the structure of the organisation remains the same but the identity of the participants may vary over time. On the contrary, our approach performs organisational restructuring [Martin and Barber 2006], in which the structure of the organisation itself –i.e., key elements within it such as roles or norms– changes over time.

Additionally, [Horling et al. 2001] use self-diagnosing to update their organisational representation by removing methods and interrelationships from agents' conditioned views. Self-diagnosis is based on a causal model of action and coordination faults. In our case, assistants adapt the social structure –i.e., the actual net of relationships– by considering communication latencies and peers that become completed [Campos 2011]. Finally, agents in the system by [Martin and Barber 2006] switch from an existing coordination strategy for decision-making –such as suggest-and-vote or negotiation– to another one that performs better in the current situation. They perform their organisational restructuring by means of "*Decision-Making Frameworks (DMF)* [that become] *a controlling variable in the performance of* [their] *multi-agent system*". Similarly, our norm parametrisation method restructures organisational elements autonomously. Furthermore, as our norms restrict agent interactions, assigning extreme values to norm parameters counts for no-restriction, and thus, addition and deletion of norms are implicitly modelled.

However, fewer OCMAS approaches follow a regulation-oriented perspective in which adaptation is no longer related to tasks. Since the seminal work by [Axelrod 1986], some normative systems have included specific norm-compliance enforcement mechanisms that are articulated through direct sanctions –such as in the work in [Pasquier et al. 2005]–, or by means of social exclusion [Glaser and Morignot 1997; de Pinninck et al. 2008]. Nevertheless, they bring in additional enforcement costs that should be carefully considered. We follow the adaptation approach in autonomic electronic institutions [Bou et al. 2009], which modifies norm penalties according to agents' behavioural changes. As previously mentioned, we claim that regulation adaptation should be robust to different instabilities including non-compliant agents. Therefore, we abstract ourselves from considering specific norm enforcement policies. Instead, our adaptation mechanism learns the best regulation values for the current execution situation. Moreover, our current research focuses on distributing the adaptation responsibility among assistant agents —or, in other words, decentralizing the adaptation process.

If we consider the norm taxonomy in [Savarimuthu and Cranefield 2009], our approach uses a norm adaptation mechanism based on social power, since our assistants are empowered to change regulations. Empowerment is a key concept in the paper by [Artikis et al. 2009], in which agents use an argumentation protocol to decide upon logically grounded norm changes. Their work follows ACMAS approaches, in which norm emergence [Shoham and Tennenholtz 1997; Brooks et al. 2011] and norm adoption [Conte et al. 2010] are attracting most research efforts. In general, robustness in ACMAS approaches is mostly studied in terms of societal topology [Villatoro et al. 2011b] rather than on norm-compliance. Among ACMAS approaches, we should highlight the work by [Grizard et al. 2007], since it focuses on the same peer-to-peer (P2P) sharing network scenario. In their paper, agents adapt local norms by using local information. In fact, locality and other important aspects such as component-based architectures or incentives have been extensively studied in P2P systems [Garlan et al. 1994; Alda and Cremers 2005], and particularly, applied to BitTorrent protocol [Cuevas et al. 2009; Piatek et al. 2007]. Nevertheless, their approach differs from ours in that they follow an agent centered approach, and thus, they cannot reason nor act at an organisational level. This is key if we have in mind that our simplified P2P case-study is not meant to be realistic, but a means to illustrate our normative (organisational) adaptation approach in highly dynamic systems. P2P sharing networks have also been commercially explored. Thus, for example, we have identified P4P [Xie et al. 2008], which also promotes local communications. Nevertheless, it does not rule on network consumption in order to balance net capacity and traffic, which reduces its flexibility.

Norm adapatation has been also tackled formally. [Dastani et al. 2012] present the syntax and operational semantics of a programming mechanism that facilitates the runtime modification of norms. Operational semantics describe the behaviour of a programming language in terms of transitions between program configurations. Thus, it is used to specify when and how the norms may be changed both by external agents or by a normative mechanism. Regarding Machine Learning techniques, they can be used through a number of techniques. Thus, for example, [Savarimuthu et al.] identify prohibition norms in an agent society. They assume norms are already established in an agent society and they provide a method for newcomer agents to learn them. They do so by means of a data mining approach that extracts specific sequences of events. This approach differs from ours in that we learn how the global norms should be in order to improve performance, whilst they discover previously established norms, and therefore, they do not consider any performance criteria. Finally, MASPA, by [Zhang et al. 2009], provides a reference example that presents many commonalities with our approach. There, higher-level agents provide supervisory information to lower-level agents. Agents apply Multi-Agent Reinforcement Learning algorithms, and thus, this provided information is used to guide the exploration of their state-action space. The main difference lies in that their reasoning is task-driven, whereas our meta-level is regulation-oriented and takes its adaptation decisions purely based on goals —which, depending on the problem, may not be directly related to tasks. Additionally, they assume they have control over the development of participant agents and, in particular, that agents will commit to assigned tasks. We plan to deal with open MAS –where agents are developed by third parties–, and therefore, we avoid making such an assumption.

Within 2-LAMA, assistants apply their individual norm adaptation function to propose new regulations. Specifically, they apply a tailored case-based reasoning (CBR) [Riesbeck and Schank 1989]. CBR is a machine learning technique broadly used in many fields. Some examples of its applicability [Watson 1997] are: computer aided diagnosis systems for cancer detection [Golobardes et al. 2002]; knowledge discovery frameworks for textual case-based reasoning [Patterson et al. 2008]; web-based applications [Bittencourt et al. 2009; Gasparetti et al. 2009]; or visual spacial problem solvers [Jim Davies et al. 2008]. CBR has been successful in so many fields due to its simple principle *To solve a new problem based on the previously solved problems*. When using CBR, the need for knowledge acquisition can be limited to establishing how to characterize the cases. CBR provides many advantages, the most important in a MAS is that it is able to deal with multidimensional continuous spaces taking profit of the similarity assumption –i.e., similar problems have similar solutions. The proposal of this paper extends CBR by making it unsupervised: initially, it takes into account heuristic knowledge and, as soon as it starts gaining its own experience, it continuously revises it to refine its acquired knowledge. [Powell et al. 2005] have a similar approach to tackle the lack of supervision in CBR. The main difference is that they apply reinforcement learning to case elicitation.

7. DISCUSSION

So far we have provided positive results about the adaptation of regulations for a simplified P2P data sharing scenario. In fact, as Section § 6 mentions, we focus on scenarios where individual agent goals are loosely aligned with system goals. In our case, individual peers aim at having the data and the system pursues to spread the datum among all peers in the shortest time possible. We argue this ”goal-decoupling” is general enough to include a wide range of complex and dynamic scenarios. We have already provided the traffic scenario example, where car agents pursuit reaching its destination as fast as possible whilst the traffic authority includes traffic rules for avoiding car-accidents or traffic jams. Nevertheless, we can also think of other scenarios such as: markets, where the rules of the market may require to be dynamically adjusted to guarantee some global properties (e.g. avoid electricity shortage or price peaks in an electricity market); robot-soccer competitions, where robot players focus on scoring as much as possible and the referees include soccer rules to guarantee fair play; or resource sharing scenarios, where agents try to maximize their resource usage whereas the system tries to avoid resource extinction.

Regarding the P2P data sharing scenario, we are aware that we are taking an alternative approach to the state of the art in this domain (see §6). As previously stated, we have chosen this case study to illustrate organisational adaptation in highly dynamic domains where system goals cannot be directly associated to individual tasks and subtasks. Our proposed learning solution (and its two-level assisted architecture) may seem too complex to deal with a problem that is usually handled locally. Nevertheless, it allows, by following a division of labour paradigm, to keep domain-level agents focused on their local (i.e., individual) activities, whereas assistants at our meta-level reason at a higher level of abstraction and are in charge of changing global regulations at run-time. This abstraction and global regulation can be seen as a coarse-grained adaptation as opposed to current P2P state of the art, which takes a more fine-grained local approach. However, from an organisational point of view, it is much convenient to have a few general norms that apply to all domain agents in the society rather than having a large number of local regulations. Locality though is a desirable property that is mostly accomplished in our organisational approach by means of a social structure. Assistants in our distributed architecture establish that social structure based on datum possession and communication latencies instead of by means of regulations (see [Campos et al. 2011] for further details).

In our specific case study, coarseness is also induced by the parameter values that the adapted regulations can possibly take. That is, *NormFR* has a $\texttt{limitV}_{FR}$ parameter defined as a natural number and, thus, ± 1 is the smallest change that can be made in the number of communications. Interestingly enough, this is somehow tuned when some of the agents do not comply with the norm (i.e., agents in the population use different values of $\texttt{limitV}_{FR}$). Therefore, deviation in norm compliance has an effect that amounts to having intermediate values in the norm. In fact, norm violation is not the only source of instabilities in our scenario: network saturation is an agent coordination problem that arises during the different phases of the sharing process (in which the proportion of seeds and leeches in the system changes significantly). Our CBR method manages to characterise in its case description these phases and provides different regulations along the sharing process. It is able to do it so even in situations where lowering $\texttt{limitV}_{FR}$ does not imply saturation reduction. For instance, if a peer with a large bandwidth serves data to peers with smaller bandwidths, it may be worse to serve just a few peers, since there may be more probability to saturate their individual links. Overall, despite the specific aspects of the P2P case study, this work proposes an automatic mechanism that can successfully adapt organisational regulations at run-time and that handles instabilities coming from a population of non-compliant agents as well as from a dynamic environment where coordination problems arise.

Another issue that deserves some discussion is our lack of explicit consideration of norm violations when proposing the adaptation of regulations. Punishment –and in general, sanctions– have proven to be effective norm enforcement mechanisms [Becker 1968; Villatoro et al. 2011a]. Nevertheless, imposing sanctions has associated costs that are often obviated but, in fact, they need to be taken into account. Therefore, we claim it is possible not to incur in monitoring costs further than strictly necessary: it is enough to measure system performance (i.e. in terms of peer completion) rather than monitoring the norm compliance of every single action performed by each and all peers in the sharing network. Thus, assistants perform norm adaptation based on how the system is performing given current system dynamics. Obviously, these dynamics implicitly contain the effect of having a specific population of non-compliant agents, but the adaptation decision process does not explicitly consider information about norm-violations nor agent populations. Conversely, our assistants adapt regulations so that they get parameter values that prove to work at specific situations (i.e., cases). Our experiments showed that for highly norm-deviant populations, assistants adapted regulations so that they constrained peer behaviours in a more restrictive manner. Nevertheless, this result may be dependant of the violation model that peers adopt homogeneously, an thus, we leave as future work the study of heterogeneous agent populations. Notice, though that, as in some social conventions, we consider norms as coordination mechanisms and agents may decide to comply with them even in the absence of norm enforcement mechanisms (penalties). In our P2P example, norms help to avoid network saturation, and this benefits agent coordination.

Finally, since our work is based on the architecture introduced in [Campos et al. 2011], this paper has considered a single meta-level. Exploring whether further meta-levels deliver benefits remains an issue for future research.

8. CONCLUSIONS

One of the main challenges in software engineering is to design and to develop mechanisms to endow software systems with self-adaptation capabilities, so that they continue to be effective under changing situations. Our research addresses this issue in the context of organisation-centred multi-agent systems (OCMAS).

In this paper we have focused on endowing the organisation with an adaptation mechanism that performs autonomous adaptation of its regulations to cope with changing circumstances. To this end, we first have defined a formal model that allows to encode regulations as norms. Next, we have proposed an adaptation mechanism for regulations based on a variation of standard case-based reasoning. Specifically, this variation proposes a complete CBR cycle with a novel knowledge representation in the case base that includes a new evaluation component. In the literature, CBR has been widely used in many application domains as it can deal with complex forms of knowledge and generalises fast from few (even noisy) examples. With this approach, OCMAS learns its adaptation policy at run-time, instead of being fixed at design-time. Moreover, it is easily applicable to new OCMAS scenarios.

We test empirically the proposed adaptation mechanism, and the corresponding results show that the proposed CBR version is significantly better than non-adaptive approaches. As we want to apply our approach in open systems, we have also evaluated the robustness of our proposal when some agents do not comply with organisational regulations. Specifically, we have evaluated our approach with different percentages of misbehaving agents that show different norm-deviant behaviors. Results show that our adaptive approach is robust and it significantly outperforms non-adaptive ones. In future studies, we plan to address other structural MAS issues, such as agents joining and leaving the open MAS at any point or considering MAS architectures with subsequent meta-level organisational layers.

ACKNOWLEDGMENTS

In memoriam, we acknowledge Dr. Marc Esteva for his valuable work in this research.

This work has been partially supported by the Spanish Research projects TIN2009-14702-C02-01/TIN2009-14702-C02-02, TIN2009-14404-C02-02, and TIN2012-38876-C02-01/TIN2012-38876-C02-02; the Consolider AT project (CSD2007-0022); EU-FEDER funds; and the Catalan Goverment (Grant 2005-SGR-00093).

REFERENCES

Aamodt, A. and Plaza, E. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications 7,* 1, 39–59.

Alda, S. and Cremers, A. B. 2005. Towards composition management for component-based peer-to-peer architectures. *Electronic Notes in Theoretical Computer Science 114,* 0, 47 – 64. <ce:title>Proceedings of the Software Composition Workshop (SC 2004)</ce:title> <xocs:full-name>Software Composition Workshop 2004</xocs:full-name>.

Artikis, A., Kaponis, D., and Pitt, J. 2009. *Multi-Agent Systems: Semantics and Dynamics of Organisational Models*. V. Dignum, IGI Global, Chapter Dynamic Specifications of Norm-Governed Systems, 460–479.

Axelrod, R. 1986. An Evolutionary Approach to Norms. *The American Political Science Review 80,* 4, 1095–1111.

Becker, G. S. 1968. Crime and punishment: An economic approach. *The Journal of Political Economy 76,* 2, 169–217.

Bittencourt, I. I., Costa, E., Silva, M., and Soares, E. 2009. A computational model for developing semantic web-based educational systems. *Knowledge-Based Systems 22,* 4, 302 – 315.

BitTorrentInc. 2001. BitTorrent protocol specification. http://www.bittorrent.org/beps/bep_0003.html.

Boissier, O. and Gâteau, B. 2007. Normative multi-agent organizations: Modeling, support and control. In *Normative Multi-agent Systems*. 1–17.

Bou, E., López-Sánchez, M., and Rodríguez-Aguilar, J. A. 2009. Autonomic Electronic Institutions' Self-Adaptation in Heterogeneous Agent Societies. In *Organized Adaption in Multi-Agent Systems*. Vol. 5368. Springer, 18–35.

Brooks, L., Iba, W., and Sen, S. 2011. Modeling the emergence and convergence of norms. In *IJCAI*, T. Walsh, Ed. IJCAI/AAAI, 97–102.

Campos, J. 2011. *Adaptive Organisation Centered Multi-Agent Systems (PhD thesis)*. University of Barcelona.

Campos, J., Esteva, M., López-Sánchez, M., and Morales, J. 2009a. An organisational adaptation simulator for p2p networks. In *Engineering Societies in the Agents World X (ESAW'09). Lecture Notes in Artificial Intelligence*. Number 5881. Spinger, 240–242.

Campos, J., Esteva, M., Lopez-Sanchez, M., Morales, J., and Salamó, M. 2011. Organisational adaptation of multi-agent systems in a peer-to-peer scenario. *Computing 91*, 169–215.

Campos, J., López-Sánchez, M., Esteva, M., Novo, A., and Morales, J. to appear 2009b. 2-LAMA Architecture vs. BitTorrent Protocol in a Peer-to-Peer Scenario. In *Catalan Congres of Artificial Intelligence (CCIA09)*.

Campos, J., López-Sánchez, M., Esteva, M., and Piqué, N. 2010. Comparison of Topologies in Peer-to-Peer Data Sharing Networks. In *Artificial Intelligence Research and Development, proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence. Frontiers in Artificial Intelligence and Applications series*. Number 220. IOS Press, 49–58.

Carley, K. 1995. Computational and mathematical organization theory: Perspective and directions. *Computational & Mathematical Organization Theory 1,* 1, 39–56.

Carley, K., Lee, J.-S., and Krackhardt, D. 2002. Destabilizing networks. *Connections 24,* 3, 79–92.

Conte, R., Andrighetto, G., and Campenni, M. 2010. Internalizing norms: A cognitive model of (social) norms' internalization. *IJATS 2,* 1, 63–73.

Corkill, D. D. and Lesser, V. R. 1983. The use of meta-level control for coordination in a distributed problem solving network. In *IJCAI'83: Proceedings of the Eighth international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 748–756.

Costa, A. and Demazeau, Y. 1996. Toward a formal model of multi-agent systems with dynamic organizations. In *Proceedings of the International Conference on Multi-Agent Systems, MIT Press, Kyoto, Japan*. 431.

CUEVAS, R., LAOUTARIS, N., YANG, X., SIGANOS, G., AND RODRIGUEZ, P. 2009. Deep diving into bittorrent locality. *CoRR abs/0907.3874*.

DASTANI, M., MEYER, J.-J., AND TINNEMEIER, N. 2012. Programming norm change. *Journal of Applied Non-Classical Logics 22,* 1-2, 151–180.

DE PINNINCK, A., SIERRA, C., AND SCHORLEMMER, M. 2008. Distributed Norm Enforcement Via Ostracism. *LNCS 4870*, 301.

DEMŠAR, J. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal Machine Learning Research 7*, 1–30.

DIGNUM, V. 2004. *A model for organizational interaction: based on agents, founded in logic*. Universiteit Utrecht PhD Monography.

DIGNUM, V., DIGNUM, F., FURTADO, V., MELO, A., AND SONENBERG, L. 2005. Towards a Simulation Tool for Evaluating Dynamic Reorganization of Agents Societies. In *Proc. of WS. on Socially Inspired Computing, AISB Convention*. Vol. 230. 153–162.

FERBER, J., GUTKNECHT, O., AND MICHEL, F. 2004. From agents to organizations: An organizational view of multi-agent systems. In *Agent Oriented Software Engineering IV*. 214–230.

FRIEDMAN, M. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association 32,* 200, 675–701.

GARLAN, D., ALLEN, R., AND OCKERBLOOM, J. 1994. Exploiting style in architectural design environments. In *Proceedings of the 2nd ACM SIGSOFT symposium on Foundations of software engineering*. SIGSOFT '94. ACM, New York, NY, USA, 175–188.

GASPARETTI, F., MICARELLI, A., AND SCIARRONE, F. 2009. A web-based training system for business letter writing. *Knowledge-Based Systems 22,* 4, 287 – 291.

GLASER, N. AND MORIGNOT, P. 1997. The reorganization of societies of autonomous agents. In *Multi-Agent Rationality*, M. Boman and W. Van de Velde, Eds. Lecture Notes in Computer Science Series, vol. 1237. Springer Berlin / Heidelberg, 98–111.

GOLOBARDES, E., LLORÀ, X., SALAMÓ, M., AND MARTÍ, J. 2002. Computer Aided Diagnosis with Case-Based Reasoning and Genetic Algorithms. *Knowledge-Based Systems 15,* 1-2, 45–52.

GOMEZ-SANZ, J. J. AND PAVON, J. 2005. Implementing Multi-agent Systems Organizations with INGENIAS. In *Proceedings of ProMAS'05*. Vol. 3862. 236–251.

GRIZARD, A., VERCOUTER, L., STRATULAT, T., AND MULLER, G. 2007. A peer-to-peer normative system to achieve social order. In *LNCS - Proc. of the Coordination, organizations, institutions, and norms in agent systems II*. Vol. 4386. Springer, 274.

GUESSOUM, Z., ZIANE, M., AND FACI, N. 2004. Monitoring and organizational-level adaptation of multi-agent systems. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*. IEEE Computer Society, Washington, DC, USA, 514–521.

HORLING, B., BENYO, B., AND LESSER, V. 2001. Using self-diagnosis to adapt organizational structures. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*. ACM, New York, NY, USA, 529–536.

HÜBNER, J. F., SICHMAN, J. S., AND BOISSIER, O. 2004. 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)*. Vol. 3171. Springer, 506–515.

HÜBNER, J. F., SICHMAN, J. S., AND BOISSIER, O. 2005. S-MOISE$^+$: A middleware for developing organised multi-agent systems. In *AAMAS Workshops*. LNCS Series, vol. 3913. Springer, 64–78.

JIM DAVIES, J., GOEL, A. K., AND YANER, P. W. 2008. Proteus: Visuospatial analogy in problem-solving. *Knowl.-Based Syst. 21,* 7, 636–654.

JONES, J. AND GOEL, A. K. 2004. Revisiting the Credit Assignment Problem. In *Challenges of Game AI: Proceedings of the AAAI*. Vol. 4. 4.

KOTA, R., GIBBINS, N., AND JENNINGS, N. 2008. Decentralised structural adaptation in agent organisations. AAMAS Workshop Organised Adaptation in MAS, 54–71.

MARTIN, C. AND BARBER, K. S. 2006. Adaptive decision-making frameworks for dynamic multi-agent organizational change. *Autonomous Agents and Multi-Agent Systems 13,* 3, 391–428.

MUKHERJEE, P., SEN, S., AND AIRIAU, S. 2008. Norm emergence under constrained interactions in diverse societies. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 779–786.

PASQUIER, P., FLORES, R. A., AND CHAIB-DRAA, B. 2005. Modelling flexible social commitments and their enforcement. In *Proceedings of Engineering Societies in the Agents World V, ESAW*. Lecture Notes in Computer Science Series, vol. 3451. Springer, 153–165.

PATTERSON, D., ROONEY, N., GALUSHKA, M., DOBRYNIN, V., AND SMIRNOVA, E. 2008. Sophia-tcbr: A knowledge discovery framework for textual case-based reasoning. *Know.-Based Syst. 21,* 5, 404–414.

PIATEK, M., ISDAL, T., ANDERSON, T., KRISHNAMURTHY, A., AND VENKATARAMANI, A. 2007. Do incentives build robustness in bittorrent. In *In NSDI 2007*.

POWELL, J. H., HAUFF, B., AND HASTINGS, J. D. 2005. Evaluating the effectiveness of exploration and accumulated experience in automatic case elicitation. In *Case-Based Reasoning Research and Development*. Lecture Notes in Computer Science Series, vol. 3620. Springer, 397–407.

RICHTER, M. M. 1995. The knowledge contained in similarity measures: Invited talk at iccbr-95.

RIESBECK, C. K. AND SCHANK, R. C. 1989. *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, Hillsdale, NJ, US.

SALAMÓ, M. AND LÓPEZ-SÁNCHEZ, M. 2011a. Adaptive case-based reasoning using retention and forgetting strategies. *Know.-Based Syst. 24,* 2, 230–247.

SALAMÓ, M. AND LÓPEZ-SÁNCHEZ, M. 2011b. Rough set based approaches to feature selection for case-based reasoning classifiers. *Pattern Recogn. Lett. 32*, 280–292.

SAVARIMUTHU, B., CRANEFIELD, S., PURVIS, M., AND PURVIS, M. Identifying prohibition norms in agent societies. *Artificial Intelligence and Law*, 1–46. 10.1007/s10506-012-9126-7.

SAVARIMUTHU, B. T. R. AND CRANEFIELD, S. 2009. A categorization of simulation works on norms. 09121.

SEN, O. AND SEN, S. 2010. Effects of Social Network Topology and Options on Norm Emergence. *Coordination, Organizations, Institutions and Norms in Agent Systems V*, 211–222.

SEN, S. AND AIRIAU, S. 2007. Emergence of norms through social learning. In *IJCAI* (2007-03-05), M. M. Veloso, Ed. 1507–1512.

SERUGENDO, G. D. M., GLEIZES, M. P., AND KARAGEORGOS, A. 2006. Self-Organisation and Emergence in MAS: An Overview. *Informatica 30*, 45–54.

SHOHAM, Y. AND TENNENHOLTZ, M. 1997. On the emergence of social conventions: Modeling, analysis, and simulations. *Artif. Intell. 94,* 1-2, 139–166.

SIMS, M., CORKILL, D., AND LESSER, V. 2008. Automated Organization Design for Multi-agent Systems. *Autonomous Agents and Multi-Agent Systems 16,* 2, 151–185.

SUTTON, R. S. AND BARTO, A. G. 1998. *Reinforcement learning: An introduction*. The MIT press.

VILLATORO, D., ANDRIGHETTO, G., CONTE, R., AND SABATER-MIR, J. 2011a. Dynamic sanctioning for robust and cost-efficient norm compliance. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, T. Walsh, Ed. IJCAI/AAAI, Barcelona, 414–419.

VILLATORO, D., SABATER-MIR, J., AND SEN, S. 2011b. Social instruments for robust convention emergence. In *IJCAI*, T. Walsh, Ed. IJCAI/AAAI, 420–425.

WATSON, I. 1997. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann Publishers Inc.

WOOLDRIDGE, M. 2009. *An introduction to multiagent systems*. Wiley.

XIE, H., YANG, Y. R., KRISHNAMURTHY, A., LIU, Y., AND SILBERSCHATZ, A. 2008. P4P: provider portal for applications. *ACM SIGCOMM Computer Communication Review 38,* 4, 351–362.

ZHANG, C., ABDALLAH, S., AND LESSER, V. 2009. Integrating Organizational Control into Multi-Agent Learning. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 757–764.

Received April 2012; revised May YYYY; accepted July YYYY