Adaptation in a P2P scenario with 2-LAMA

Jordi Campos[1] and Maite López-Sánchez[1] and Marc Esteva[2]

[1]MAiA Dept., Universitat de Barcelona {jcampos, maite}@maia.ub.es
[2]Artificial Intelligence Research Institute (IIIA) CSIC marc@iiia.csic.es

Abstract. In this paper, we propose a MAS architecture (2-LAMA) to adapt social conventions from a system's perspective. The proposed architecture consist of two levels: the conventional MAS system (*domain-level*) and an additional *meta-level* in charge of adaptation. We illustrate our approach in a Peer-to-Peer scenario. The resulting model changes organisation depending on environment and agent changes.

1 Introduction

We approach MAS adaptation through the modification of *social conventions*. MAS are distributed by nature, and so it should also be its adaptation mechanism. Accordingly, we propose adaptation to be done by means of an additional distributed level (*meta-level*) on top of a regular MAS [1].

Most work on MAS adaptation assumes it is feasible to identify which tasks are necessary to achieve system goals. In [2][3] once they have identified required tasks, they can assign them to available agents and establish their organisation depending on task dependencies. However, we are interested in contexts where it is not possible to identify which tasks achieve system goals. For example, in a traffic scenario we want to decrease the number of accidents and save control resources [4], but we cannot identify which tasks are necessary to achieve it. In fact, there are norms, and their relationship with global goals is more complex. Other works, such as [5] or [6], share our interests. In [5], agents update social norms by agreement without dealing with goals. On the other hand, agents in [6] change their local *conventions* in a P2P scenario but keep global norms static. [6] also has an additional layer, but with supervision purposes only. Similarly, *organisational agents* of [7] have an extra layer, although they assume the mapping between tasks and goals previously mentioned.

As a motivating scenario, we work on a Peer-to-Peer (P2P) network where *peers* (agents) share some datum –we assume it has a single piece. The relationships they establish change over time depending on network status. Our vision is that these relationships define the system's organisation (i.e. how computers organise themselves to interact), whereas changes in network status constitute its dynamic environment. The performance of a system will be computed in terms of time and network consumptions.We use a simplified version of BitTorrent protocol shown in Figure 1. It has an initial handshake phase in which *peers* indicate if they have the datum ("`bitfile[1/0]`"≡peer does/doesn't have data), and a second phase in which they request the datum to those *peers* having it.

2 General Model: 2-LAMA

We propose a Two Level Assisted MAS Architecture (2-LAMA) by adding a *meta-level* (ML) on top of the previous system called *domain-level* (DL) plus a communication interface (Int) among both levels. Thus, our model can be expressed as: $M = \langle ML, DL, Int\rangle$. Each level has a set of agents (Ag_{xL}, xL is a generalisation of ML and DL) and its social conventions defined by a *social structure* (Org_{xL}) and a set of *norms* (Nor_{xL}). Hence, each level can be defined as: $xL = \langle Ag_{xL}, Org_{xL}, Nor_{xL}\rangle$ (see Figure 2). We see the social structure as a set of roles (Rol_{xL}) and the relationships (Rel_{xL}) among agents playing them: $Org_{xL} = \langle Rol_{xL}, Rel_{xL}\rangle$. Norms limit agent's behaviour and are expressed as first-order deontic logic formulae to define agents permissions, prohibitions and obligations. Furthermore, communication among levels covers bottom-up (Up) and top-down (Dn) information exchanges: $Int = \langle Up, Dn\rangle$. The *meta-level* perceives *domain-level* observable properties, evaluates them, and adapts *domain-level* social conventions. Perceived properties are those that can be observed in the environment ($EnvP$, e.g. date, temperature...) and those that can be observed in agents (AgP, e.g. colour, position...) —i.e. $Up = \langle EnvP, AgP\rangle$. While adapted social conventions correspond to new organisation (Org'_{DL}) and norms (Nor'_{DL}) of the *domain-level* —i.e. $Dn = \langle Org'_{DL}, Nor'_{DL}\rangle$. We assume each *meta-level* agent ($a_{ML} \in Ag_{ML}$) has partial information about such properties, so it only perceives a subset of $EnvP$ and AgP (in many scenarios global information is not available). An a_{ML} has aggregated information about a subset of *domain-level* agents that can share with other *meta-level* agents.

3 Peer-to-Peer Model

In Participant agents in the *domain-level* correspond to peers sharing data that play a single role $Rol_{DL} = \{peer\}$. Their network connections are represented as arcs connecting nodes in a weighted complete graph (costs correspond to latencies). Figure 3 depicts it, although some arcs are omitted for the sake of simplicity (their weights are 30). As peers usually contact a subset of neighbours, we define it as the relationships among agents (Rel_{DL}) that form a sub-graph of the network graph. These relationships, which belong to the agents' organisation, will be updated by the *meta-level* taking into account the system status. However, we have a norm to Nor_{DL} that limits the bandwidth *peers* are allowed to use —it limits the number of message units a *peer* can send at each time step. Thus, *peers* cannot use the network as an infinite resource. We assume agents follow social conventions. Regarding our *meta-level*, it also has a single role $Rol_{ML} = \{assistant\}$. Each agent in Ag_{ML} collects information about a disjoint subset of *peers* ($cluster \subset Ag_{DL}$) and adapts their local organisation. Its decisions are based in local information about its associated *cluster* —such as latencies ($EnvP$) or *peers* having the data (AgP)— and information about other *clusters* they get from their neighbours in the *meta-level* organisation (Org_{ML}). We assume *assistants* are located at Internet Service Providers (ISP) and thus related communications are fast.

3.1 Extended protocol

We extend the P2P protocol to include *meta-level* communications. A *peer* starts handshaking its *assistant* with a "join <hasDatum>" message ("hasDatum" = *peer* does/doesn't have the datum). Then, the *assistant* asks the *peer* to measure its latencies with all other *peers* in its *cluster* by sending "get_latency <peers>" messages. The *peer* measures latencies by means of *ping* messages, and informs back the *assistant* with a "latency <amount>" message. Once an *assistant* has all latencies among their *peers* ($EnvP$) and knows which ones have the datum (AgD), it estimates which would be the best re-organisation. Then it adapts the agent *relationships* ($Rel'_{DL} \in Org'_{DL}$) by sending "contact <peers>" messages to all the *peer*s in its *cluster*. Then, the previously introduced P2P protocol is followed. Additionally, when a *peer* receives the datum it informs its *assistant* with a "completed" message. Then, at *meta-level* this *assistant* informs its neighbour *assistants* with a "completed_peer <peer>" message. Next, contacted *assistants* spread this information towards their *peers* with a "has_datum <peer>" message. In that moment, agents measure their latencies to the new *peer* and request the datum if it is better than any previous source. Finally, an *assistant* sends an "all_completed" message to its neighbour *assistants* when its *peers* are completed.

3.2 Assistant decisions

Mainly, an *assistant* faces two different situations: (a) some *peers* in its *cluster* already have the datum, or (b) no *peer* in its *cluster* has it yet. In the first case (a), the *assistant* computes the shortest paths —using Dijkstra's algorithm over arc latencies— from each *peer* having data to the rest of *peers* in the cluster (in case there are several source nodes, the minimum shortest path is considered instead). Then, it re-organises its *cluster* by telling each *peer* to contact with its predecessor in its shortest path to a data source. This way, the graph of new *relationships* (Rel'_{DL}) may have different arcs than the old relationships (Rel_{DL}). In the second case (b), the *assistant* organises its *cluster* to be prepared for data entering through any *peer*. Accordingly, it assumes any *peer* can become a data source and computes all possible shortest paths. Next, it provides to each *peer* its predecessors in all its corresponding shortest paths. This way, all *peers* are in contact with the neighbours that could provide rapidly the data when it enters through any node in the *cluster*. The resulting *relationship* graph (Rel_{DL}) is larger than in previous case (a) but considering the information available, it still smaller than all possible *relationships* ($EnvP$).

4 Experiments

We have tested the 2-Lama approach on the P2P scenario depicted in Figure 3. We evaluate the system performance in time and network usage. On the one hand, we define the *time cost* (c_t) as the number of time steps from the start of simulation up to when all nodes have the datum. On the other hand, we define the *network cost* (c_n) as the network usage of each message (c_{m_i}) sent among agents: $c_n = \sum_{i=0}^{\#msgs} c_{m_i}$. This usage depends on the message's length (m_{length}) and the latency (Lat) among its origin (m_{org}) and destination (m_{dst}) agents, expressed as: $c_{m_i} = m_{length} \cdot Lat(m_{org}, m_{dst})$. We assign m_{length} their values depending on message types and levels: at DL, ping $=$ 1, data $=$ 10 and control (`bitfile`, `request` and `have`) $=$ 2; at ML, messages among *assistants* $=$ 2 ; at Int, messages among *peers* and *assistants* $=$ 2. Since we propose to add a distributed *meta-level*, we name our implementation *Distributed*. Besides, in order to have reference performance values, we also present two alternative implementations: *All4All* and *Centralised*. Firstly, in *All4All*, all *peers* contact each other at the beginning, and then request data from sources along all possible paths. *All4All*'s parallelism guarantees minimum execution time (c_t), but its lack of *meta-level* does not prevent maximum network cost (c_n), since all *peers* exploit all their communication alternatives simultaneously. Secondly, *Centralised* implements a *meta-level* composed by a single *assistant* agent. This agent has global information so that it can make fully informed decisions when computing shortest paths (see subsection 3.2). As a consequence, it recommends the optimal neighbour to each *peer*, and thus, guarantees the minimum network cost (c_n). *Centralised*'s execution time is slightly longer than *All4All*'s, though. This is because all *peers* in *All4All* send the handshaking (`bitfile`) simultaneously, whereas, in *Centralised*, handshaking is a dialogue: answers are sent once `bitfiles` are received (see Fig. 1). Finally, it is worth mentioning that current simulations start once all *peers* have contacted their *assistants*.

5 Results

The results correspond to the execution of the three alternatives with the *peers* and network latencies depicted in Figure 3. Alternatives are simulated with different bandwidth limits (Nor_{DL} ="max BW message units per time step") as explained in section 3. In *Centralised* and *Distributed*, there are also different simulations for various network latencies between *domain-level* and *meta-level*, and among *assistants* –we assume both have the same value (L_{x2a}). Each combination has been executed once with the datum in each *peer*. Results in Figure 4 show the round average of these executions in time (c_t) and network (c_n) costs. Results confirm our minimum and maximum costs assumptions. Generally, *All4All* requires the minimum time but uses the maximum network, whereas *Centralised* consumes the minimum network. Indeed, they show our proposal of adding a *meta-level* is worth, since the cost derived from adding it is less than its benefit. Specifically, the results of the *Distributed* approach show that adding the *meta-level* provides more savings in network usage than expenses in time. For instance, being $BW = \infty$ and $L_{x2a} = 1$ our *Distributed* approach requires 31% more time than *All4All* but saves 76% network costs. In fact, the *Distributed* is an intermediate point in network consumption among *All4All* —its *peers* need to discover its shortest path to data sources— and the *Centralised* —its *as-*

sistant already has all the information. In *Distributed*, *assistants* already have knowledge about its cluster, but *peers* are required to discover the shortest path with data sources outside its cluster. Currently, *assistants* tell all their *peers* to discover these shortest paths. Even if we increase communication latencies with the new level (L_{x2a}) up to the maximum of *domain-level* —30 among *peers*— it still uses less network than having no *meta-level* at all (*All4All*). For example, being $BW = \infty$ and $L_{x2a} = 30$, *Distributed* consumes 64,2% less network than *All4All*. We test up to this latency to prove that the advantages of our solution were not related to having faster communication channels, but they derive from doing a better use of these channels. This means, that having some agents playing both roles —*peer* and *assistant*— without having fast *assistant* agents located at ISPs, our approach still saves network resources. In *Centralised*, network usage increases with L_{x2a}because *peers* also inform the *assistant* when they are completed. However, as they send their neighbours the "`have`" message before contacting the *assistant*, the execution continues regardless of L_{x2a}, so the required time is constant. In addition to prove the benefits of organisational adaptations, we wanted to check the effects that could have adapting our proposed norm. Results show that limiting bandwidth influences over network consumption. However, with current configuration it has more impact on execution time than on network usage. In future work, we plan to simulate network traffic jams to increase the influence when changing the norm. This way, we could study its adaptation.

6 Conclusion

In this paper we presented the Two Level Assisted MAS Architecture (2-LAMA) that adds a *meta-level* in charge of a system adaptation to dynamic changes. The proposed adaptation is distributed requiring no global information. As a case study we introduced a P2P scenario to which we apply our model obtaining an adaptive P2P MAS. We provided means to evaluate it and we designed some alternatives and experiments to contrast its benefits. The experiments showed interesting results, notably the fact that the cost of adding the *meta-level* is lower than the obtained benefit. We conclude it is feasible and worth to add our proposed *meta-level*. In future works, we plan to experiment different configurations with our current norm, and work on its adaptation by the *meta-level*. Even more, we plan to experiment with a norm at *meta-level* level to bound its weight over the rest of the system (i.e. limit the number of *peers* and *assistant* can tell that another *peer* has data). Besides, we want to update latencies depending on network traffic and study how our approach adapts to these environmental changes. In the medium term, we would like to deal with open MAS, where agents can join and leave and transgress social conventions. Currently, *meta-level* provides adaptation directions that agents follow, but we think about providing advices to agents. We envision an open MAS with an *assistant layer* that improves the *coordination support* the infrastructure provides to its agents.

Acknowledgements. This work is partially funded by IEA (TIN2006-15662-C02-01), AT (CONSOLIDER CSD2007-0022) projects, EU-FEDER funds, the Catalan Gov. (Grant 2005-SGR-00093) and M. Esteva's Ramon y Cajal contract.

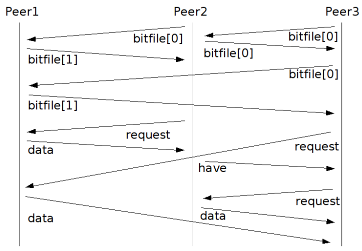

Fig. 1. P2P simplified protocol.

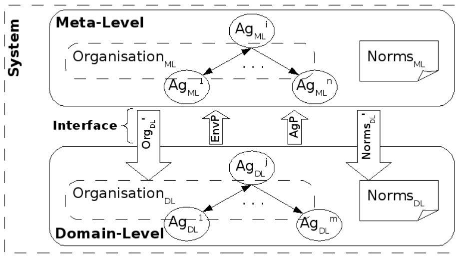

Fig. 2. General Model.

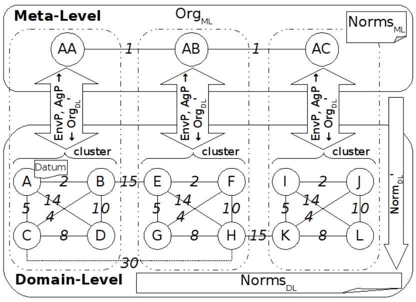

Fig. 3. P2P model example.

		All4All		*Centralised*		*Distributed*	
BW	L_{x2a}	c_t	c_n	c_t	c_n	c_t	c_n
1	1	525	25480	512	2896	648	5926
1	5	-	-	"	2984	688	6348
4	1	476	25053	488	2896	618	5971
4	5	-	-	"	2984	653	6394
∞	1	464	24976	481	2896	610	5979
∞	5	-	-	"	2984	645	6402
∞	30	-	-	"	3534	861	8939

Fig. 4. Resulting costs in simulations.

References

1. Campos, J., López-Sánchez, M., , Esteva, M.: Multi-Agent System adaptation in a Peer-to-Peer scenario. In: Symposium on Applied Computing SAC, Special track on Agreement Technologies. (2009)
2. Kota, R., Gibbins, N., Jennings, N.: Decentralised structural adaptation in agent organisations. In: AAMAS Workshop on Organised Adaptation in MAS. (2008)
3. Sims, M., Corkill, D., Lesser, V.: Automated Organization Design for Multi-agent Systems. Autonomous Agents and Multi-Agent Systems **16**(2) (2008) 151–185
4. Campos, J., López-Sánchez, M., Rodríguez-Aguilar, J.A., Esteva, M.: Formalising situatedness and adaptation in electronic institutions. In: COIN Workshops. (2008)
5. Artikis, A., Kaponis, D., Pitt, J.: Dynamic Specifications of Norm-Governed Systems. In: Multi-Agent Systems: Semantics and Dynamics of Organisational Models. (2009)
6. Grizard, A., Vercouter, L., Stratulat, T., Muller, G.: A peer-to-peer normative system to achieve social order. LNCS **4386** (2007) 274
7. Deloach, S.A., Oyenan, W.H., Matson, E.T.: A capabilities-based model for adaptive organizations. Autonomous Agents and Multi-Agent Systems **16**(1) (2008) 13–56