An organisational adaptation simulator for P2P networks

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

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

Abstract. Organisational centred multi-agent systems (MAS) have proved to be effective to regulate agents' activities. Nevertheless, population and/or environmental changes may lead to a pour fulfilment of the system's purposes, and therefore, adapting the whole organisation becomes key. This paper presents a MAS simulator devoted to test organisations with self-adaptation capabilities in P2P scenarios. More specifically, our simulator implements different sharing P2P methods –some of them with self-adaptation– and so, it can be used as a testbed for comparing them.

1 Introduction

This paper presents a simulator for testing organisation adaptation mechanisms in P2P scenarios. In order to endow an organisation-based MAS with self-adaptation capabilities, we propose to incorporate a meta-level in charge of adapting system's organisation. Hence, we call our approach Two Level Assisted MAS Architecture (2-LAMA) [1]. In this context a P2P system is modelled as an organisation, with a social structure among peers, and a set of protocols and norms that regulate the sharing process. On top of the P2P system, that we call domain-level, we add a distributed meta-level which perceives information about its status and uses this information to adapt peers' social structure and norm values. Meta-level adaptation is based on system performance, which is measured by the time peers spent to share data and the network consumption in such a process.

The simulator, that we have implemented using Repast Symphony, can model the whole process, implements different sharing methods, and help in the analysis of system's behaviour. In the rest of the paper we present the system architecture in section 2, the implemented sharing methods in section 3 and the graphic user interface in section 4.

2 Simulator Architecture

The simulator architecture allows to model both agents (*agent-level*) and the transport of messages among them (*network-level*). On the one hand, the `p2p` module represents the conceptual model defined by the 2-LAMA targeted to drive the simulation at *agent-level.* Among others, it provides facilities to create state-based agents, and to define a problem (number of peers, who has initially the datum, etc). This module is divided into two layers, the *domain-level* composed by peers, and the *meta-level* containing the assistants that implement adaptation services.

On the other hand, the `netsim` module drives the simulation at *network-level.* It provides facilities to transport messages among agents, to define different network topologies, and to collect statistical information about network status. The network level simulates message transport as a packet switching network. We assume that peers are connected to different ISPs. Hence, we define an individual link that connects each peer to its ISP, and some aggregated links among ISPs. The latency of a message between agents depends on the number of links, their *bandwidth* and the current traffic through them.

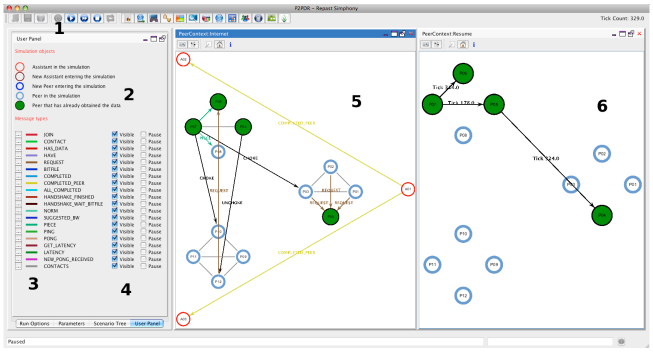
During simulations the tool generates log files containing all occurred events. The simulator includes a module for facilitating the analysis of simulation results. For this purpose, this module processes the generated logs extracting relevant information, which is later on displayed in different types of graphics. Hence, this can be used to compare the time spent to share the data in different configurations, or using different sharing methods.

3 Sharing methods

The simulator offers alternate sharing methods. This way, they can be executed over the same initial configurations and their results can be compared. Currently the tool includes two sharing methods without meta-level, and two different methods with meta-level. The two methods without meta-level are a brute-force algorithm, where peers contact all other peers, and a single-piece version of the BitTorrent protocol (BT). Regarding methods using a meta-level, the tool includes the 2-LAMA approach with just *social structure* adaptation, and the 2-LAMA approach with *social structure* and *norm* adaptation Hence, it allows to compare the performance of our adaptation approach with respect to methods without meta-level.

We used the BitTorrent protocol as a base to design the protocol in our 2-LAMA P2P approach, so both protocols are similar. The main difference is that BT does not have a distributed *meta-level* but a single agent (*Tracker*) that informs just about connected *peers.* Consequently, *peers* do not receive any further assistance to share the datum, and are restricted to use the algorithm in [2]. In contrast, the 2-LAMA approach without *norm* adaptation includes a *meta-level* with *assistant* agents. In this method, assistants just adapt the social structure among peers —i.e. their overlay network. Hence, norm values remain unchanged during the whole simulation. Finally, in the 2-LAMA approach with *social structure* and *norm* adaptation, assistants update both the social structure and norm values during the simulation.

4 GUI



Simulation objects

- ○ Assistant in the simulation
- ○ New Assistant entering the simulation
- ○ New Peer entering the simulation
- ○ Peer in the simulation
- ● Peer that has already obtained the data

Message types

...	JOIN	☑ Visible	□ Pause
...	CONTACT	☑ Visible	□ Pause
...	HAS_DATA	☑ Visible	□ Pause
...	HAVE	☑ Visible	□ Pause
...	REQUEST	☑ Visible	□ Pause
...	BITFILE	☑ Visible	□ Pause
...	COMPLETED	☑ Visible	□ Pause
...	COMPLETED_PEER	☑ Visible	□ Pause
...	ALL_COMPLETED	☑ Visible	□ Pause
...	HANDSHAKE_FINISHED	☑ Visible	□ Pause
...	HANDSHAKE_WAIT_BITFILE	☑ Visible	□ Pause
...	NORM	☑ Visible	□ Pause
...	SUGGESTED_BW	☑ Visible	□ Pause
...	PIECE	☑ Visible	□ Pause
...	PING	☑ Visible	□ Pause
...	PONG	☑ Visible	□ Pause
...	GET_LATENCY	☑ Visible	□ Pause
...	LATENCY	☑ Visible	□ Pause
...	NEW_PONG_RECEIVED	☑ Visible	□ Pause
...	CONTACTS	☑ Visible	□ Pause

Fig. 1. 2-LAMA P2P Simulator Graphic User Interface

In order to display up-to-date visual runtime information of the evolution of our P2P simulation, an advanced GUI has been created as an extension of the Repast GUI. Figure 1 depicts a GUI screenshot that illustrates its general appearance. The **Control toolbar** (1) pertains to the original Repast GUI and allows to play the simulation, pause it or execute it step by step. On the left area, the **Legend panel** shows information about what represents each object of the layout (2), the colours of the different messages exchanged among agents (3), whether they are visible or not, and if execution will pause upon sending this kind of messages (4). All these options can be modified by users. Thus, the legend allows an easy identification of each object and message to interpret what is happening in the simulation at every moment The **Runtime P2P Network layout** (5) shows the elements of the simulation and the communications among them. Peers and assistants are drawn according to the network topology, while messages are displayed as arrows among them with the corresponding colour defined in the legend panel. Finally, the **Resume layout** (6) displays how the data has been distributed among the different peers. It highlights completed peers and displays arrows connecting source and receiver agents. These arrows are labelled with the time step at which the datum was received.

Acknowledgements. Work funded by IEA (TIN2006-15662-C02-01), AT (CONSOLIDER CSD2007-0022), and Marc Esteva's Ramon y Cajal contract.

References

1. Campos, J., López-Sánchez, M., Esteva, M.: Multi-Agent System adaptation in a Peer-to-Peer scenario. In: ACM SAC09 - Agreement Technologies. (2009) 735–739
2. Cohen, B.: The BitTorrent Protocol Specification. http://www.bittorrent.org/beps/bep_0003.html