Context dependent management of field diffusion: an experimental framework♣

Flavio De Paoli, Giuseppe Vizzari

Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano Bicocca
Via Bicocca degli Arcimboli 8, 20126, Milano
{depaoli,.vizzari}@disco.unimib.it

Abstract

Agents in the Multilayered Multi Agent Situated System (MMASS) model can perform at-a-distance distance communication by field diffusion over the structure of the environment that hosts the agents. This paper discusses a distributed mechanism to support selective diffusion on the basis of contextual information. The proposed mechanism has been implemented to deliver an experimental framework based on peer-to-peer approach.

1 Introduction

The Multilayered Multi Agent Situated System (MMASS) provides different means of interaction between autonomous entities placed in a structured environment. The MMASS model has already been successfully applied to different contexts and areas such as localization problems [2, 3], and CSCW [4]. Initial exploitation of the model has been carried out to support simulation to study physical phenomena or domain problems [3]. The model is anyway promising also to address the development of distributed applications due to its intrinsic properties. The distributed nature and the communication metaphors defined by the model makes it suitable to represent distributed systems. To make it effective it is necessary to supply a framework that implements the model to avoid discrepancies between specification and implementation, to facilitate the application development, and finally to deliver effective implementations. This paper represents a step toward this goal.

The MMASS environment is described by a network structure, made up of sites that may contain at most one agent. Synchronous *reaction* may only involve adjacent agents and asynchronous, remote communication can be obtained through a field *diffusion-perception-action* mechanism. Both of them are space dependant: the first one due to agent adjacency, the latter because fields are spread throughout the structure of the environment and their value is modulated according to a diffusion function, related to field type. The adoption of this model requires thus suitable mechanisms to manage agent interaction. More precisely they should be able to manage synchronization among autonomous agents, in order to perform reaction, and should provide algorithms and structures able to diffuse information related to fields. This paper will focus on the latter topic, focusing in particular on a distributed mechanism that supports a selective diffusion taking into account contextual information, for instance generated by agents.

In theory, field diffusion could traverse the entire structure of the environment. That is, according to its diffusion function, a field should be perceivable in every site of the structure. At a site, a field is perceived by a hosted agent if, and only if, that agent is sensible to that field. Therefore, the diffusion of fields to sites that do not host sensible agents to that field could be avoided.

The framework presented in this paper has the purposes of providing a mechanism and a diffusion policy that takes into account agents' sensibility to deliver a distributed implementation for the model.

The next section will briefly describe how the model can be exploited in order to define a conceptual architecture for this framework, while Section 3 will describe an experimental implementation, with reference to various mechanisms and phases required to obtain a selective diffusion infrastructure. Related works will be described in Section 4, while conclusions and future developments will end the paper.

2 MMASS and selective diffusion

The MMASS model defines a formal and computational framework for the description of complex systems that are characterized by an explicit and discrete

♣ The work presented in this paper has been partially supported by FIRB project *MAIS: Multichannel Adaptive Information Systems*

representation of spatial structures. The latter are graphs of sites that may be empty or host a single agent. The model supports the representation of different forms of heterogeneity, both with reference to entities and interaction modalities. The former can be obtained defining different agent types, with related perceptive capabilities and behavioral rules. Interaction modalities provide synchronous *reaction* between entities placed in adjacent positions and that agreed to perform a simultaneous change in their internal state. Moreover, it is possible to model asynchronous interaction among at-a-distance agents through *field diffusion* in the environment. Fields are signals that can be emitted by agents and are spread over the environment according to a *diffusion function* related to the specific field type. Field intensity is thus modulated throughout their path from source to destination sites. A perception-deliberation-action mechanism supports the definition of the behavior of various entities that are able to perceive emitted signals. The range of possible actions for the definition of the behavior for agents of a certain type includes four actions:

- *emit*, which allows an agent to add a field of a certain type in its site and, according to the spatial structure and the diffusion function, in a certain part of the environment;
- *trigger*, allowing the definition of a change in the state of an agent that takes place when a specific field is perceived;
- *transport*, that allows the definition of conditions for agent movement from its site to an adjacent one that doesn't contain another entity;
- *react*, which models the synchronous interaction between adjacent agents that agreed to simultaneously change their state.

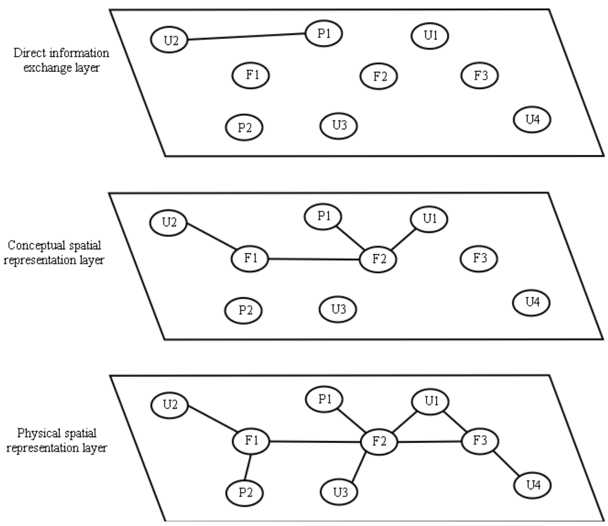

Figure 1: Logical layers and connections.

Interaction between agents is thus strongly space dependant, as it is related to agent adjacency or field diffusion. Through a suitable definition of physical and conceptual spatial structures, it is possible to obtain context-aware behavior for modeled agents. Different superposed spatial representations may be defined, in order to model different aspects of agent context, both related to some kind of representation of the physical space and to conceptual ones. For instance, agent preferences, membership to groups or organizations, assignment to tasks may determine some kind of neighborhood or distance according to specific aspects of their activities or preferences. An MMASS site is a very simple entity that is characterized by the agent and active fields that it is hosting, and by its adjacent sites. This is not enough to obtain a selective diffusion infrastructure, as at least an internal state is required to store information related to paths and agents' concerns. In the following a possible infrastructure for selective diffusion based on the MMASS model will be described.

Figure 1 shows three different layers, representing respectively a physical spatial structure related to network connections among sites and two different representations of conceptual spaces, related to the connection of entities sharing an interest in a specific topic and the adjacency of two entities that agreed to exchange information related to that topic.

Spatial structures are represented with graphs of sites that, in this framework, contain a single static agent. The latter may represent *providers* (Pi) and *users* (Ui), in other words publishers and seekers of information related to a specific topic, and *forwarders* (Fi), that is nodes that are devoted to the diffusion of information related to the publication of new contents and to support users in their searches. These agents do not operate at application level but implement a communication infrastructure, providing selective diffusion.

The lowest layer shows all physical connections among agents in the system, regardless their conceptual adjacencies that are represented in conceptual spatial representation layers. The latter is a partition the set of sites according to the kind of fields that are of interest for the involved agents. The one shown in Figure 1 assumes that agents P1, U1 and U2 are providers and users of a certain kind of information. F1 and F2 are interconnecting sites that diffuse signals among those agents. The involved sites determine conceptual connections that form a graph along whose edges the information flows. In such a way, the information diffusion is optimized since only interested sites and agents are involved. For example, assume that F1 publishes information (it emits a field) that is of interest of U1 and U2 (they are sensible to that field), the information will not reach F3, as no publisher of that kind of content is logically connected to that forwarder.

Conceptual layers may change to follow the system evolution. Therefore, changes in user situations and context might cause modifications that trigger the creation or destruction of certain edges.

The third layer represents the establishment of a direct communication channel between agents that have agreed to perform a reaction. The MMASS model establishes that reactions can only occur between adjacent entities, so an adjacency between the involved agents needs to be established. This layer illustrates this connection though which a direct exchange of information can take place, without involving other entities of the network. For example, assume that U2 receives the information (the field) issued by P1, and deliberate to react with P1. A new connection between the sites hosting the two agents is established to perform the reaction.

Conceptual spatial representation layers are thus sub graphs, parts of the physical spatial structure, representing a constraint in the diffusion of specific fields. They can be constructed and maintained to represent groups of entities sharing an interest and thus defining areas where specific information should be diffused and stored. The primitives of the MMASS model involved in the construction and maintenance of spatial structure are *AddSite(p)* and *DeleteSite(p)* that respectively allow an agent to create and destroy an edge connecting the site it is placed on to another one. Through the definition of suitable *interfaces* between layers, different counterparts of agents at different levels of abstractions may interact to carry out the structural modifications of layers. In this way, configuration is carried out through the diffusion of fields related to agent interests at the network layer, but the effect of the perception of these signals is propagated to the agent interest layers, that are modified accordingly. At last, when agents share the same interest and decide to perform an information exchange, another conceptual level is modified in order to allow a direct communication.

In the following sections, a more detailed and technical explanation of how this kind of process can be carried out in a peer-to-peer environment will be shown.

3 The framework

According to the diffusion mechanism described in the previous section, an experimental framework has been developed. The goal has been to investigate the issues related to the implementation of the model on top of Internet by means of standard tools.

Moreover, an example supporting MP3 file exchange has been developed to test the framework. The sample has been selected for several reasons. Music file exchange is a real and relevant activity in Internet. There are often several sources for the same piece of music. At the same time, the same piece of music is often searched by several users. Moreover, these files are usually quite large and their downloading occupies a large network bandwidth.

Our goal is to develop an application that addresses the problems of current peer-to-peer applications. The idea is to publish concise descriptions of pieces of music to interested users. On the base of such information, they can decide to download the original file, which is usually huge compared to its description. The bet is to reduce network traffic by reducing the number of searches and optimizing the downloading. Moreover, information diffusion prevent from searching for new available music, thus enhancing the quality of service by letting users be aware when a new piece of music becomes available.

Gross music categorization is widely known and does not need further explanations (e.g., anyone can distinguish between classic, jazz or rock music). Music categories are implemented as fields, so that source and perceptive agents deals with fields like “jazz” or to “classic”. The MMASS model is suitable for that application since field diffusion allows for publishing concise information to interested users. Users can react with publishers to download the music.

According to the layered structure of a system illustrated in Figure 1, the framework provides for the definition of a physical layer that represents the real connections among sites, conceptual layers that identify subsets of agents that are interested in specific fields, and finally interaction layers that support direct information exchange between agents.

Java has been selected as implementation language for sites and agents; XML has been exploited to describe the information exchanged between sites; and socket connections have been established to model the connections described in the environment. Agents and sites maintain profiles that include a unique identifier, preferences, and a set of configuration information such as a list of sites to be contacted. Configuration information is used to set up the system. Preferences include description of the fields that are handled. Whenever an agent need to be activated a new site is crated and connected to the existing network, and an agent is installed on it. Connections imply a configuration phase that identifies the conceptual layer and sets up the necessary physical connections. Once the conceptual layer has been established, the field diffusion and the subsequent reaction among agents may occur.

In what follows, a detailed description of the three phases is provided.

3.1 Configuration

We assume that a set of sites are started and connected on different networked computers. Sites and related agents can be created and connected to the network, changing the spatial structure. Some sites host pure forwarders, that is, agents devoted to the support of selective diffusion, while sites hosting agents have the double task of diffusing fields and connecting application level agents. Referring to the physical layer in Figure 1, Fi nodes are of the former type, Pi and Ui nodes are of the latter type. In what follows we will refer to that figure to illustrate the examples. Assume that conceptual layer has been built for agents interested in "pop" music.

When a new agent is created at a new node, it sends out a message – named query – that includes information about the perceivable fields. This context information is initially diffused to let sites set up a conceptual layer that will include the new site, maybe some other site, and the sites that were already forming such a layer. Forwarders and intermediate agents (i.e. agents whose site has more than one adjacent site) will be configured to relay or stop field information according to sensibility of reachable agents. In other terms, a site is a *forwarder* for a certain field, and not for others. Even if potentially, any site can diffuse any field.

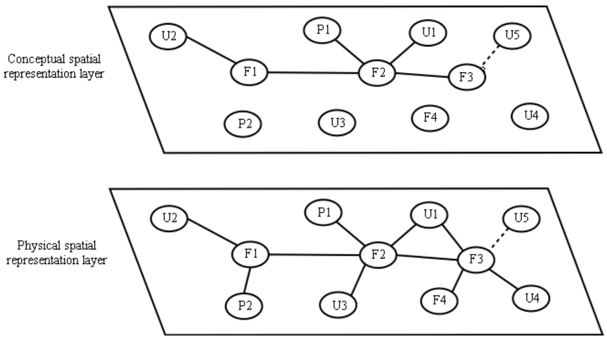

Figure 2: A configuration example: U5 joins the net.

In our example, assume a new agent U5 that sends out a query message about pop music. This context information is initially diffused to the sites that are known to the hosting site. Information about sites is included in the profile of the site. Such sites are named *default* sites to state that they are stable and act as connecting points. Assume that the site hosting U5 contact F3 to diffuse the information that U5 is interested pop music. F3 examines the query and can react in two ways. If it is already a forwarder for pop music, it answers the query and the new site is simply added to the conceptual network that was already present. If F3 does not handle pop music, it forwards the information to its connecting sites. At any site, the query is examined with the same two alternatives: answering or forwarding. Eventually, one or more sites answer the query. The originating site can then select one of the answering sites to establish a connection.

In our example, F3 forwards the query to F2 and F4. F4 does not handle pop music, and it has no connecting sites to forward the query. Therefore, it simply discards the query. F2 is already a forwarder for pop music. Therefore, it sends back an affirmative answer. U5 collects the answers that include information about the sites that have been traversed, and timing information. U5 can now establish a stable connection with F2, or ask F3 to set up a conceptual layer. In the former case, it tries to contact directly F2 that may refuse or accept the connection according to loading considerations. In the latter case, F3 and traversed sites are asked to be part of the forming layer (i.e., to become forwarders for pop music).

Figure 2 illustrates this second option: F3 has been included in the conceptual layer. Note that U5 may collect more answers; a selection process is performed on the base of the content of the answers. Moreover, in the current implementation, a site performs some pings to try to evaluate the most performing forwarders before setting up a connection. Anyway, the selection policy can be defined depending on the application nature. For example, to avoid centralization, the number of connection per forwarder should be limited. Another criterion deals with the connecting computers. In the current implementation, direct connections are limited to computers within LANs to reduce internet connections. A last note deals with loops. To support field diffusion efficiently, connecting loops should be avoided to prevent from the same information to return at the same site. This situation is prevented by construction, since each node can set up one connection at a time for a certain field type.

3.2 Information diffusion

Once a conceptual layer has been established, information can flow from providers to users. Assume that provider P1 wants to publish a new song. It sends out a message – named advertisement – that includes information about the new song and its location. The message is diffused along the conceptual layer to reach every interested site. Figure 3 illustrates P1 that publishes song A. If a user is interested in that song, it can interact with the provider to get the song (see next subsection).

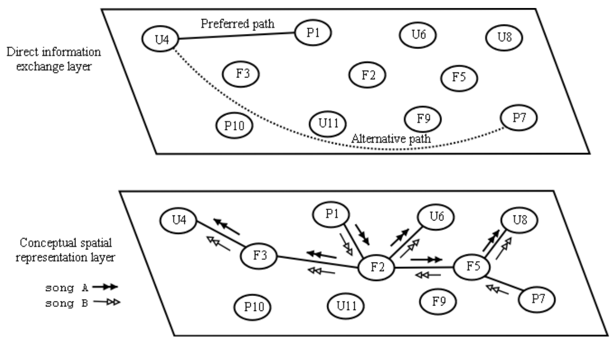

Figure 3: Field diffusion.

Assume now that two agents, located in different locations, publish the same song. The logic of the traversed sites will "compose" field information to deliver one useful piece of information to the receivers. In our case, the system will deliver information about the closest source for the advertised song. For example, in Figure 3 P1 and P7 publish the same song B. F2 receives that information from P1 and F5. In such a case, it composes the information by merging the advertisement messages in one message that report the two providers. P1 will be ranked first since it is closer to the part of the network connected via F3.

To let forwarders compose field information, the advertisement include information on the followed path. Moreover, field diffusion function may define that signals decays after a certain numbers of hops. Therefore, it is possible to define the scope of a field. Composition rules and diffusion functions are programmable to overwrite the default behavior for forwarders.

Over issuing advertising messages, agents may also issue searching messages to look for a certain piece of music. In this case, the field diffusion occurs from users to providers in exactly the same way. Any provider that actually publishes the searched song can answer to let the searcher know the location of the song. To reduce searching costs, the current implementation includes special forwarders that caches field information about available songs and sources. In terms of the MMASS model, they are considered permanent fields that can be set up and cancelled by explicit field emissions.

3.3 Agents' interaction

In the scenario depicted above, a user receives information about a specific kind of music. Therefore, s/he has simply the task of selecting interesting songs and activating the interaction mechanism to download the songs. To react, agents need to be adjacent, therefore if user U4 decides to download song B that has been advertised by provider P1, a direct connection between P1 and U4 need to be established. The information to establish the connection is included in the advertising message. It may happen that P1 refuses the connection – for example, due to high numbers of requests. In such cases, U4 can try to contact P7, which was ranked second in the source list. In this way, availability increases to ensure services even in case of failures.

4 Related work

In recent years, a number of platforms and environments have been proposed to support the development of multi agent systems. Examples of such tools are AgentBuilder [5], JACK [6], JADE [7], JAFMAS [8], and Zeus [9]. A common characteristic is that they generally provide a point-to-point message passing communication model. Generally there is a sort of facilitator agent that is connected to all the others (its name and address are fixed and well known) and is also used as agent name server. MMASS communication model and the implementation proposed in this paper consider information distribution and diffusion as a founding peculiarity to make agents aware of the environment, their context and therefore on at-a-distance communication. To our knowledge, only Repast [10], its ancestor Ascape [11] and Madkit [12] implement some kind of support for space mediated information diffusion, but only in the simulation context.

The framework described in this paper has been inspired by peer-to-peer systems that are emerging as communication tools to distribute information over Internet. Peer-to-peer systems assume a number of disperse actors that connect and disconnect to perform searches and exchange data. The key issue for those systems is the huge amount of exchanged messages, and the capability of actually locate data.

Peer-to-peer systems range from client/server to distributed architecture. For example, Napster [13] provides for a centralized search service. Clients locate files by sending a request to a central server that collects information related to all available files. A point-to-point interaction between the two peers takes place only during file transfer. Therefore, it could be classified as a hybrid system. Napster suffers from typical problems of centralized systems: single point of failure, scalability and availability.

Gnutella [14] is an open, decentralized, search protocol. New peers join to the Gnutella network by contacting a joined peer, and then send broadcasting messages to discover other peers send queries. Each peer receiving a query tries to satisfy the request with its local data or forwards the message to all known peers. In this way, a query is propagated across the system until it meets a server that can answer it. The Gnutella approach does not require any centralization. The drawback is the large amount of generated messages that rapidly overcome the network. Availability is also an issue, since there is no guarantee that a server really answers a query.

P-Grid [15] is a self-organizing P2P system that addresses scalability and robustness by constructing a tree-based distributed indexing structure. In such a way, query messages are directed along certain paths according to indexing. A similar approach is taken by Chord [16] and CAN [17]. The former organizes peers to form a ring according to hash values of stored data; the latter make use of hash function to organize the system. For these systems, network traffic is reduced at the cost of setting up a predefined data organization.

The system organization proposed in this paper follows the trend of structuring the system to facilitate lookup activities by reducing network traffic. Similar to Chord, our protocol involves only a limited set of peers to simplify the setting up phase. In fact, considering the situation depicted in Figure 2, only the nodes that have been contacted directly are involved in the configuration phase. In practice, configuration is a local activity involving part of the nodes in the net. On the contrary, Chord and CAN assume a global organization of the system.

Another important difference with current peer-to-peer systems is the communication mechanism. All the existing systems privilege search of items, while our model takes the opposite approach: users discover the presence of the interesting data by receiving notifications. Sending around notifications may reduce network traffic for some reasons that can be explained using the music-exchange example. Assume a number of users looking up the same piece of music; it is likely that a number of similar query messages are spread over the network almost at the same time. Potentially, one single notification message from the source could reach all the interested users. Note that, since notifications are handled by forwarding sites, they could implement policies to deliver messages according to network condition, e.g., by delaying notifications if no time requirements have to be met. Moreover, caching could be adopted by some forwarders to simulate multiple sources for partitioning the network traffic.

5 Conclusions and future work

This paper has presented a proposal for a framework to support the development of multi agent systems according to the MMASS model. The goal of the framework is to address all the different aspects of the model and support all development phases. The current implementation concentrates on the diffusion mechanism that allows for asynchronous communication among agents.

Currently, the framework is under testing to deliver the first release. Experiments to validate the approach are in progress. The goal is to determine under what condition the proposed solution is effective. That is, to determine the scalability limits. The results of these experiments will drive future development. In particular, we need to determine the default behavior of forwarders with respect to cashing and timing.

Acknowledgments

The authors wish to thank Sara Manzoni and Leonardo Mariani for their contribution. Moreover, they wish to thank Fulvio Moro and Vincenzo Picariello who actually implemented the described framework.

References

[1] Bandini, S., Manzoni, S., Simone, C.: Dealing with Space in Multi-Agent Systems: a model for Situated MAS, Proceedings of the First International Joint Conference on Autonomous Agents and Multi–Agent Sytems (AAMAS 2002), pp. 1183–1190, ACM press, 2002.

[2] S. Bandini, S. Manzoni, G. Pavesi, and C. Simone: Location of extra–urban shopping centres: A multi–agent based approach, In G. Concilio and V. Monno, editors, Proceedings 2nd National Conference on Information Technology and Spatial Planning: Democracy and Technologies, Bari, 2001.

[3] S. Bandini, S. Manzoni and G. Vizzari: MMASS Approach to Localization Problems, Proceedings of The 2003 IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003) (to appear).

[4] Carla Simone, Stefania Bandini: Integrating Awareness in Cooperative Applications through the Reaction-Diffusion Metaphor, Computer Supported Cooperative Work 11(3-4): 495-530 (2002).

[5] Reticular Systems, Inc. AgentBuilder: an Integrated Tookit for Constructing Intelligent Software Agents, February 1999, http://www.agentbuilder.com.

[6] Busetta P., Rnnquist, R., Hodgson A., Lucas A., JACK Intelligent Agents -- Components for Intelligent Agents in Java, http://www.agent-software.com.au/, 1999.

[7] Bellifemine, F., Rimassa, G., Poggi, A., JADE - A FIPA-compliant Agent Framework. Proc. of the 4th International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents, London, 1999.

[8] Chauhan, D. and Baker, A., JAFMAS: A multiagent application development system. In Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98), Minneapolis, ACM Press, May 1998.

[9] Nwana, H. S., Ndumu, D. T., Lee, L. C. and Collis, J. C., "ZEUS: A Toolkit for Building Distributed Multi-Agent Systems", Applied Artificial Intelligence Journal 13 (1/2), 129-185, 1999.

[10] Repast, 2003, http://repast.sourceforge.net.

[11] Ascape, 2000, http://www.brook.edu/dybdocroot/es/dynamics/models/ascape/README.html.

[12] Gutknecht, O. and Ferber, J., Madkit: Organizing heterogeneity with groups in a platform for multiple multi-agent systems. Technical Report 97188, LIRMM, 161, rue Ada - Montpellier - France, dec 1997.

[13] The napster protocol specification, 2001, http://opennap.sourceforge.net/napster.txt.

[14] The gnutella protocol specification v0.4, 2001, http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.

[15] K. Aberer, M. Punceva, M. Hauswirth, and M. Lebofsky, and D. Werthimer. Seti@home systems. IEEE Internet Computing, 6(1):58{ 67, January/February 2002.

[16] J. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrisham. Chord: A scalable peer-to-peer lookup service for internet applications. In ACM SIGCOMM 2001, pages 149-160, San Diego, California, 2001.

[17] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In ACM SIGCOMM 2001, San Diego, California, 2001.