1

The MaBE Middleware

Alois Reitbauer [1], Alessandro Battino[2], Anthony Karageorgos[3], Nikolay Mehandjiev[4], Paul Valckenaers[5], Bart Saint Germain[5]

[1]Profactor Produktionsforschungs GesmbH, Wehrgrabengasse 1-5, A-4400 Steyr
alois.reitbauer@profactor.at
[2]Profactor Produktionsforschungs GesmbH, Wehrgrabengasse 1-5, A-4400 Steyr
alois.reitbauer@profactor.at
[3]Dept. of Communications and Computer Engineering, Univ. of Thessaly, 37 Glavani - 28th October Str, 382 21 Volos - Greece
karageorgos@acm.org
[4]Dept. of Computation, UMIST, Sackville Str. M60 1QD, Manchester, UK
mehandjiev@acm.org
[5]Profactor Produktionsforschungs GesmbH, Wehrgrabengasse 1-5, A-4400 Steyr
alois.reitbauer@profactor.at

The research project MaBE aims at the development of multi-agent middleware supporting cooperation in open business environments, using real-world case studies of dynamic virtual enterprises. These case studies provide us with a library of scenarios of inter- and intra- organisational cooperation, which impose certain requirements to effective middleware platform supporting such cooperation. Some of these requirements can not be directly implemented within the FIPA-compliant agent platform which underpins the middleware we aim to develop. For example, dynamic handling of multiple and evolving ontologies, security and trust issues as well coordination in open environments require further research before building the middleware functionality supporting them. Since these issues are perceived as key enablers for open business collaborations, this paper provides an overview of the research undertaken to address them.

1. INTRODUCTION

The Virtual Enterprise (VE) paradigm (Cam, 2001) describes the development of mechanisms to support collaboration of existing business entities in a distributed environment.
Multi-agent systems (MAS) are considered to be a suitable approach for modelling various cooperation scenarios within a Virtual Enterprise. Existing agent platforms,

however, lack built-in support for some important requirements stemming from these scenarios.
The EU funded research project MaBE is developing a middleware system for supporting the development of collaborative business environments. Within this remit it has focused on several important areas where further research is necessary to provide effective agent-based support to a number of real-world collaboration scenarios within Virtual Enterprises.
This paper provides an overview on the main research issues addressed. Because of the apparent diversity of research issues, the aim is to provide a holistic broad-brush picture of the set of issues rather than to go into a considerable level of depth exploring an individual issue.

2. INDUSTRIAL USE CASE

The solution described below is motivated by requirements from real world industrial use cases. As an example, the scenario of a company performing hardening processes is presented. The company consists of nine SME organized as a Virtual Enterprise.

The main objective for forming a Virtual Enterprise was the optimized usage of information as well as physical resources. However this possibility cannot be fully exploited due to the amount and the complexity of the information stored (e.g. part-treatment programs, quality control data, production plans). In the following the main impediments for collaboration within one company as well across companies are presented.

Within a single company, the main concerns are related with the production coordination. At present, the definition of batches and the scheduling are activities carried out manually and therefore implying big efforts. In order to reach a higher level of flexibility, there is the need for a resource coordination system allowing a production planning, which takes into account the actual workload of the facilities, as well as real-time data about breakdowns and other disturbances.

Interaction between the companies induces mechanisms for communication (flow of information) as well as logistics (flow of material). These mechanisms make use of services currently available within the virtual enterprise but additional services not yet available are needed. Choosing the structure of a Virtual Enterprise it is expected to enable all involved entities to act as if they belonged to just one company, exploiting at the same time the logistic advantage constituted by a distribution of the factories in the territory. The transport of the parts (logistic service) can be carried out with VE means, with means of the customer or through logistic provider. The parts received from one customer can have as a destination the same customer of origin or another customer or even another company of the Virtual Enterprise (in case of distributed processes). The resulting big amount of possible interactions can only be efficiently coordinated with support of a system supporting automatic selection of services. Since such a system would connect different actors of the supply chain, topics like trust and security have to be addressed. It is, in fact, fundamental to provide a sufficient visibility in order to permit a services discovery, but without revealing reserved information or compromising the security of the communication. Moreover, to allow the interoperability between companies working in different business fields, a common language or at least a common frame of reference is required. For instance the data about a transport can be described in different ways inside the Logistic Provider and inside the Virtual Enterprise; and new concepts can be used when a new transport unit is introduced.

3. REFERENCE MODEL FOR COOPERATION

To provide an overall framework for analysing and understanding the case study, for modelling and implementing cooperation scenarios and for handling the requirements implied there, it is necessary to employ a theoretical reference model of collaboration within a Virtual Enterprise. The following model was found to fit the scenario described above, and to resonate well with authors' experience. The model is based on different types of collaboration as follows.

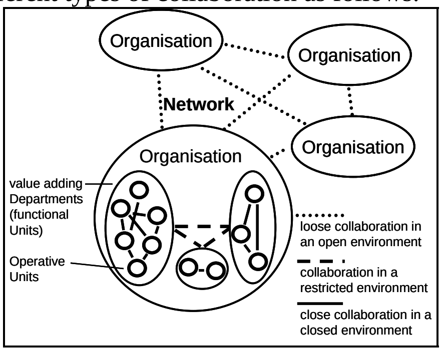

Figure 1 - Levels of Collaboration

Organisations at the lowest level are built on operative units representing atomic resources. These units cooperate among each other forming functional units also called departments or groups. Interactions between these functional units represent the collaboration taking place at enterprise or intra-organisational level. Networks of organisations are the next level of collaboration. Inter-organisational cooperation can further be divided into cooperation of a fixed number of partners or cooperation in an open environment where partners permanently can appear and disappear. Figure 1 presents a graphical representation of the different level of collaboration.

4. SUPPLY NETWORK COORDINATION

As presented in the use case above coordination not only implies coordination of all orders and resources meaning production logistics but also information logistics. A satisfactory solution has to provide monitoring and short term forecasting abilities as well as decision taking mechanisms. While section presented 2 the static structure of various forms of collaboration, a dynamic view on the system is required for resource coordination. The Supply Network paradigm [REF] is seen as fitting well in here.

An entity in a supply network is defined by the services it provides to the environment. Combination of entities can either be established by establishing relations or by aggregation. An important factor concerning the established structure especially in open collaboration environments is that it is by no means fixed. The structure will dynamically change while the environment is changing.

Both in intra- and inter-organizational collaboration, autonomy and specificity of an entity has to be respected. As shown in section 3 there is no single level of autonomy and specificity evolved in the collaboration scenarios. While in intra-organisational collaboration there is only single goal to be achieved among participating entities this is not the case in inter-organisational collaboration. Production entities in the system are subject to be striving towards different and often also confliction goal. Still, decisions made by one entity have to be respected by the other entities in some respect. Consequently a coordination mechanism must be able to define win-win solutions for all involved entities.

For the definition of a supply chain architecture the PROSA [REF] model has been chosen. PROSA allows the modeling of holonic system structures using three main building blocks called holons.

- Product holons provide the process knowledge of a product, meaning all possible production steps combination and their order.
- Order holons manage one order (customer order, production order, a aggregation of orders, etc) throughout all production stages.
- Resource holons represent a material or immaterial resource within a supply network.

In addition to the architectural building blocks a mechanism for coordination of these entities is required. Open environments impose due to their dynamically changing structure great complexity regarding coordination. Centralised approaches are not applicable, as they require the perception of the whole environment. This however is either not or only hardly possible in open environments.

The coordination mechanism presented here is based on the insights from the behaviour of ants searching for food. The main advantage of such an ant-based design is that individuals are not exposed to the complexity and dynamics of the global situation.

There are three different types of ants and respectively three stages of the resource coordination process each one fulfilling one dedicated purpose.

Feasibility ants are responsible for discover the topology of the supply network; finding out all respective services there can be accessed. This information can be used by explorer ants, which are responsible for exploring possible solutions through the supply network for a specific order. After the selection of a possible candidate solution found by one of the explorers, intention ants are used to spread the order agent's current intention along all resources involved in the orders respective objective. Based on these declared intentions, service providers can build a dynamic load forecast, which can be used by explorer ants to make their performance estimate reliable.

All information produced during each of the three phases has only a limited period for which it is valid. After this period it vanished if it is not produced again. This procedure represents an analogy to the evaporation of pheromones – the analogy of information – used in ant communication. As a consequence of this evaporation of information each of the phases above has to be executed cyclically. This approach keeps information in the system always up-to-date and additionally and even more important makes the system robust to changes.

5. TRUST AND SECURITY

Current agent systems lack support for security and other trust building mechanisms (Pos, 2001). A trusted environment is however a vital point in business collaborations. Current standardisation activities within FIPA(FIPA, 2004) deal with the implementation of basic security mechanisms for message based security. Security mechanisms however only factor of trust. Reputation and confidence building mechanisms are additional factors that have to be addressed. As not all of these factors are relevant for any kind of collaboration – a clarification on this topic is needed.

Requirements for trust issues depend on the form of collaboration. We will now relate them to the model presented in section 2, concentrating on where ICT support is needed.

Collaboration within Departments happens within a closed system and in a trusted environment. An established legal framework forms the basis for confidence in the actions of the parties involved in collaborations. This confidence is supported by striving towards a common goal and by the fact that the acting parties are employees of the same company, bound by contracts.

Inter-department collaboration happens on a regular basis but less frequent than within a department. The collaboration scenarios are well defined and all actions take place in a closed, trusted environment as well. Trust however declines due to reduced personal contact in collaborations.

Security at this level addresses the protection of information and resources based on organisational roles as well as on the context of collaboration.

In inter-organisational collaboration the environment is insecure, more dynamic, and in general not trusted. Security for collaboration scenarios within an organisation is completely controlled by the organisation itself. In an open environment a part of this control is handed over to the other involved parties. Mechanisms are needed which allow communication scenarios, in which not all parts of the infrastructure are controlled by the organisation itself.

Contracts between companies are as powerful as contracts regulating inter-organisational activities. However they more difficult to enforce as independent third parties are required for enforcement. As companies might have conflicting or competing goals, confidence in the actions of others is not present per se. The establishment of reputation also takes longer as interactions occur on an infrequent, business need driven basis.

In inter-organisational collaboration different scenarios of doing business exist. Collaboration based on strictly defined processes with known business partners represents the simplest form of inter-organisational collaboration. This can be extended to a fully open dynamic organisational network, where processes are not fixed, and collaboration takes place with unknown partners.

A different view on the roles of the importance of trust related issues are the states collaboration goes through in its lifecycle. Following (Cam, 2001) collaboration has 4 states. The initial state is the creation state, followed by the operation of the virtual organisation. During operation the organisation is suspect to evolve by changing its structure. The dissolution of the virtual organisation is then the last stage. The evolution stage can however be eliminated in closed and fixed collaborations. Confidentiality and data origin identification play a vital role in all stages. Authorization issues as well as reputation and confidence enforcing mechanisms are key requirements for the execution phase. The creation and respectively evolution phases are responsible for setting up and defining reputation and confidence mechanisms.

Due to the specific requirements to authorization mechanisms in collaborative environments (Edw, 1996) that cannot be coped with using standard mechanisms like access control lists (ACLs) or access control matrixes (ACMs). For reputation and confidence building mechanisms no standard approaches are available at all. The challenge here is to satisfy these requirements and the fact that mechanisms have to be independent of the underlying implementation. Agent-oriented systems however offer a very attractive way for implementing these features.

Agent communicate using messages. Having available these messages and knowledge of complete interactions context based trust mechanisms can be implemented. By applying the mediator pattern (Gamma, 94) agent communications can be intercepted and trust related tasks can be performed. This mechanism is for example also used for systems monitoring the access of medical databases (Liu, 2000) or access control in web service environments (Sko, 2003).

6. DYNAMIC ONTOLOGIES

Dynamic ontology handling

To handle semantic interoperation at the tactical level of the Hanomag case study, it is preferable to use open interoperation based on ontologies which provide formal specification of the meaning of a set of concepts and their relationships. Indeed, current agent system use ontologies for high-level semantic interoperation and communication. Explicit modelling of communication and interactions [refs FIPA ontologies].is one of the key concepts of agent-oriented programming. This approach is consistent with the approach used on the Semantic Web initiative, and by some recent approaches to Enterprise Application Integration (EAI).
Having decided to use ontologies for semantic interoperation and agent communication brings to the fore two technical research issues: ontology mapping and dynamic handling of evolving ontologies. This section will address both issues in turn.

Mapping of multiple ontologies

Two basic approaches can be taken to deal with differences in ontologies. Standardisation of IT systems offers the possibility to let all system communicate using the least common denominator of all ontologies. This causes the loss of information and respectively reduces the communication abilities making a deep integration impossible. The second approach is to create a common “world model” integrating all sub-models of all involved systems. The model created using this approach is undoubtedly very complex. Additionally this approach cannot be taken in open environments where not all models will be known at design time.

Taking the model given in Section 3 the ontology issue can be discussed with respect to specific forms of collaboration.

Within functional units all entities use the same ontology as the have the same vocabulary and the same view on the perceived concepts of the world. As their functional scope is also very limited the ontologies are likely to change in a very infrequent manner.

In intra-organisational cooperation one ontology still represents knowledge of the same domain, however different view on the common vocabulary exists and not all concepts are relevant for all functional units. These different view make in necessary to have apply mapping rules translating from one view into another. This requires additional administrative effort. However administration is done in a central point within one company.

In open environments different ontologies from different domains with different views are used by the collaboration participants. In closed environments integration is possible although not easy as all ontologies are known advance. The control of the evolution of ontologies and the rules for mapping between them however is distributed throughout the collaborative environment.

Describe out approach use description of partially overlapping ontologies from one of the deliverables.

Dynamic handling of evolving ontologies

Interoperation at tactical level of the case study should handle previously unknown logistics suppliers, for example. It may be possible to map the ontologies used by them to the common ontology used within the system at the time of joining, but this may require some modifications of that common ontology. The middleware platform should therefore provide facilities for handling evolving ontological concepts at run-time.

The platform underpinning MaBE is Jade. Jade does not support dynamic ontologies at the moment, since every ontology concept to be used by Jade should have corresponding Java classes defined and available at compile-time.

There are three main approaches to introducing dynamic ontologies in this context:

Ontology Server

A promising approach to realise dynamic ontology management is to have a specific software component acting as Ontology Server (Fig 1.) The Ontology Server (OS) acts as a run-time middleware, which administers the knowledge base containing various ontologies. The OS includes powerful reasoning mechanisms (inference engines) which deliver answers to queries to the knowledge base submitted by the client applications. During this processing, rules describing domain specific relationships between concepts and instances of concepts are evaluated. Ontologies are stored in various formats, such as XML, RDF, DAML-OIL and OWL. The OS is able to translate between different representations of ontologies and to map elements of different ontologies that represent the same information. Applications using ontologies and the corresponding knowledge bases are built on top of the ontology management system.

Such ontology-based applications are easily adaptable to new data and information by changing the underlying ontologies and knowledge bases. Adaptation can be done automatically, for example by having machine learning algorithms analyse personal needs and changing demands and then the ontology management system to modify the relevant ontologies accordingly. Alternatively, users can modify the underlying ontologies manually to reflect known changes to the application requirements.

Ontology Agent

In agent-based systems, such as MaBE, the aforementioned ontology management approach needs to be realised using agents. The main reason is that the client applications are agent-based and hence the ontology server will have to engage in complex interactions with numerous client agents providing translation, mapping and interoperation services. Therefore, it is deemed as more efficient to develop the ontology management system as an agent, which is termed the Ontology Agent (OA).

OA will be the link between the agents residing in the MaBE system and any internal and external ontology servers that will be used. In the general case, more

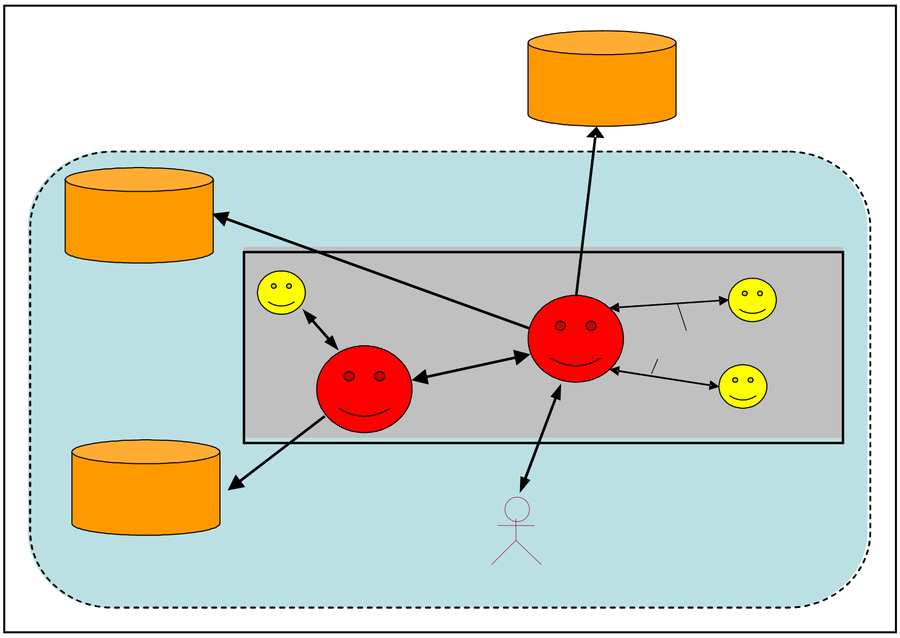

Figure 1 The Ontology Agent structure in MaBE System

than one OA will be operating in MaBE linked together via the Directory Facilitator of the MaBE agent platform (Fig 2). Ontology agents should be able to interact via appropriate protocols based on the ontology service provided. The interactions concerning ontology services in MaBE should be complying to the FIPA ontology Service specifications [2].

The model of agent communication in FIPA is based on the assumption that two agents, who wish to converse, share a common ontology for the domain of discourse. It ensures that the agents ascribe the same meaning to the symbols used in the message. For a given domain, designers may decide to use ontologies that are explicit, declaratively represented (and stored somewhere) or, alternatively, ontologies that are implicitly encoded with the actual software implementation of the agent themselves and thus are not formally published to an ontology service.

This FIPA specification deals with technologies enabling agents to manage explicit, declaratively represented ontologies. An ontology service for a community of agents is specified for this purpose. It is required that the service be provided by a dedicated agent, called an Ontology Agent (OA), whose role in the community is to provide some or all of the following services:

- discovery of public ontologies in order to access them,
- maintain (for example, register with the DF, upload, download, and modify) a set of public ontologies,
- translate expressions between different ontologies and/or different content languages,
- respond to query for relationships between terms or between ontologies, and,
- facilitate the identification of a shared ontology for communication between two agents.

The FIPA specification deals only with the communicative interface to such a service while internal implementation and capabilities are left to developers. It is not mandated that every OA be able to execute all those tasks (for example, translation between ontologies, and identification of a shared ontology are in general very difficult and not always possible to realize), but every OA must be able to participate into a communication about these tasks (possibly responding that it is not able to execute the translation task).

To implement dynamic ontology management using an ontology agent, the agents in MaBE should be supplied with appropriate protocols so that OAs can act as brokers for messages involving unknown ontologies.

Modifications of the JADE Kernel

The last approach requires that the JADE agent system kernel is reengineered so that new ontologies are able to be instantiated on run-time. This could be packaged as a JaDE add-on.

Trade-offs between solutions to managing dynamic ontologies

The final approach is more efficient when long-term, user initiated changes to the ontologies available to MaBE agents take place. This also depends on the size of the ontologies introduced. If the new ontologies are relatively small and frequently used by MaBE agents, such as ontologies reflecting changes to every day business rules and legislation it may be more appropriate to have ontologies explicitly known to MaBE agents.

The second solution is more appropriate when the agents need to use external, relatively large ontologies for short time periods. In such cases it is preferable for OAs to broker agent interactions. The type of ontology service provided by OAs should be configurable by MaBE system users.

9. REFERENCES

(Cam, 2001) Camarinha-Matos, L., Afsarmanesh, H.: Virtual Enterprise Modeling and Support Infrastructures: Applying Multi-agent System Approaches. Lecture Notes in Artificial Intelligence, Vol. 2086. Springer Verlag Berlin Heidelberg (2001) 335 -364.

(Pos, 2001) Poslad S, Charlton P., Calisti M.: Protecting what your agent is doing: AgentlLink Newsletter 7, May/June 2001.

(FIPA, 2004) FIPA: FIPA Agent Message Security Object Proposal: FIPA Security TC Working Document (2004).

(Edw, 1996) Edwards, W.: Policies and Roles in Collaborative Applications. ACM Conference on Computer Supported Cooperative Work (CSCW'96), Cambridge, Mass., (1996) 11-20.

(Gamma, 94) Gamma E., Helm R., Johnson R., Vlissides J.: Design Patterns – Elements of Reusable Object-Oriented Code. Addison-Wesley Professional Computing Series. Addison-Wesley (1996)

(Lakatta 1993) Lakatta EG, Cohen JD, Fleg JL, Frohlich ED, Gradman AH. Hypertension in the elderly: age- and disease-related complications and therapeutic implications. Card Drugs Ther 1993; 7: 643-54.

(Liu, 2000) Liu D., Law K. Wiederhold G.: CHAOS: And Active Security Mediation System. Proceedings of the 12th International Conference on Advanced Information Systems Engineering. Lecture Notes in Computer Science, Springer-Verlag: London, UK (2000) 232 - 246

(Sko, 2003) Skogsrund H. Benatallah B. Casati F.: Model-Driven Trust Negotiation for Web Services, IEEE Internet Computing – Vol (7) No (6) (2003) 45 - 52.

(Smith 1989) Smith, Adam. “An Inquiry into the Nature and Causes of the Wealth of Nations”. In Classics of Economics, Charles W. Needy, ed. Oak Park, IL: Moore Publishing, 1989.

(Ward 1972) Ward, Benjamin. What’s Wrong with Economics. New York: Basic Books, 1972.