Agent-based Marketplaces for Electronic Commerce [*]

Ralf Sesseler
DAI-Lab, TU Berlin
Sekr. FR 6-7, Franklinstr. 28/29
D-10587 Berlin, Germany

Sahin Albayrak
DAI-Lab, TU Berlin
Sekr. FR 6-7, Franklinstr. 28/29
D-10587 Berlin, Germany

Abstract *We transfer the common notion of marketplaces as locations for trading to infrastructures for electronic commerce. Then, we argue that multi-agent systems are a promising technology to realise such electronic marketplaces because of the autonomous, task-oriented, and interactive capabilities of agents in dynamic and distributed environments. We present the agent toolkit JIAC IV and describe its special features that make it a suitable candidate for the realisation of electronic marketplaces. In more detail, we describe how interactions between agents are made more flexible and reliable by a service scheme, which covers generic aspects of interactions.*

Keywords: multi-agent systems, electronic commerce, electronic marketplaces

1 Introduction

The popularity of the Internet and its global presence make it an interesting place for trading and other business transactions. Business to business as well as business to consumer relations can profit from the new forms of commerce that electronic networks promise. The independence of time and location and the potential for automation and interconnectivity will enhance convenience and reduce costs.

Reliable platforms are needed that support the openness and dynamics of the networks in a flexible manner. On the other hand, increasing competition urges for more efficient development and introduction of new services to reduce time to market. Also, electronic networks as the base of trading yield additional requirements like security, privacy, personalisation, and device-independent access.

We developed JIAC IV as a generic toolkit for agent systems facing the requirements of real world applications. It covers the whole range from development to deployment. Regarding electronic commerce, the infrastructure functionalities and the service scheme of JIAC IV are of special interest. The infrastructure supports flexible and dynamic interactions between agents in open environments. In addition, it provides security and administration facilities. The service scheme forms the link between the infrastructure and the control architecture of single agents. It covers generic aspects of interactions and integrates them into the behaviour control process.

In the following, we present marketplaces as a metaphor to organise the infrastructure for electronic commerce and business applications. This is done by mapping the concepts of conventional marketplaces to computational functionalities and networking technologies. We argue for multi-agent systems as a basic technology for distributed electronic commerce systems. Then, we give a short overview of the JIAC IV agent toolkit before concentrating on its aspects suitable to realise agent-based marketplaces and the service scheme. Finally, we discuss some related work and give a conclusion and outlook.

2 Marketplaces

Traditional marketplaces are meeting points for traders and customers. They serve the supply and demand of goods and services. The traders offer their products, from which the customers can choose to purchase. In some cases, there are more complex relationships between both parties

[*] This work was funded by T-Nova Deutsche Telekom Innovationsgesellschaft mbH.

like auctions and brokering. In any case, a marketplace is more than a pure meeting point, because it also supports trading by the infrastructure it provides. This comprises common services, the supply of resources, and administration.

An electronic marketplace is one, which's infrastructure is based on electronic networks. Traders and customers meet only via the network. Nonetheless, an electronic market has to provide similar infrastructure functionalities like conventional marketplaces.

2.1 Infrastructure of Electronic Marketplaces

First of all, an electronic marketplace (Figure 1) also serves as a meeting point for trading: The traders will have to present and to promote their products and the customers need the ability to choose and to buy products. From this, several requirements of traders and customers derive, which the infrastructure has to take into account by different kinds of generic services.

Supply services support the traders in supplying their products. They need an appropriate product presentation e.g. by graphical and multimedia facilities and easy supply and maintenance of their products and services. This involves among other access to external databases, data exchange with external systems, and persistent storage of data. *Access services* assist the customers to access the supply on the marketplace according to their individual demands. This comprises personalisation and support of different access devices including mobile ones.

Traders and customers are brought together by *broker services*. The traders register themselves and their products at white and yellow pages, which the customers can access to find products and their vendors. The business transactions have to be facilitated by *trading services* that cover accounting, billing, payment, and delivery. In addition, auctions and other market mechanisms can be supported.

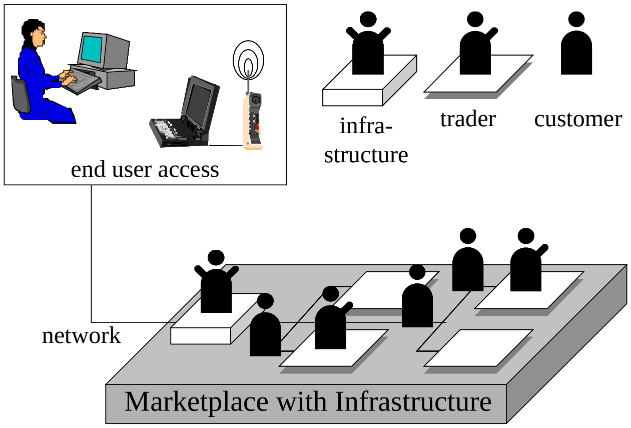

Figure 1. An Electronic Marketplace

The infrastructure has to provide a reliable and trustworthy platform. *Security services* establish trust and privacy and protect against manipulation and misuse. The administration of the infrastructure is done by *management services*, e.g. for configuration, performance analysis, and fault management.

2.2 Advantages of Electronic Marketplaces

To realise marketplaces on electronic networks has many advantages for the traders as well as for the customers. The supply of goods and services is more effective and cheaper because of the potential for automation and the direct integration into other business systems. Also, the supply can be more dynamic and flexible because it is easier to change the marketing strategy, to introduce new products, and to create new services.

A main advantage is the independence from time and location. In principle, the traders can reach every person that has access to the network and the customers can choose conveniently out of an enormous range of products at any time without leaving home. Because of this overwhelming amount of suppliers and products, customers need assistance in finding, comparing, and selecting products according to their demands.

Agent technology is a promising ground to realise electronic marketplaces because agents are capable of solving complex tasks autonomously in dynamic, distributed environments. The infrastructure of multi-agent systems already supports dynamic interactions and negotiations between independent entities that may be distributed over networks.

In addition, agent systems have several advantages compared to traditional approaches like the client/server paradigm dominating the Internet nowadays. The autonomy of the agents and their communicative capabilities allow for more flexible and complex business transactions. Value added services can be generated easily by the

JIAC IV Agent Tool-Kit					
Architecture		Development		Run-Time	
component architecture	*control architecture*	*methodology*	*tools*	*infrastructure*	*administration*
- components - roles - messages	- knowledge languages - reactivity - deliberation - interactions	- analysis - design - implementation - evaluation	- language compilers - configuration - debugger	- marketplaces - migration - white pages - yellow pages - security	- configuration - fault management - end user access

Figure 2. The JIAC IV Agent Tool-Kit

combination of more basic services. Broker services support the agents in finding appropriate partners for interactions. Especially the customers can be relieved from many tasks because of personalised assistance and agents acting autonomously in behalf of them.

3 Overview of JIAC IV

JIAC IV is intended as a complete toolkit for developing and deploying agent systems covering design methodology and tools, agent languages and architecture, a FIPA compliant infrastructure, management and security functionality, and a generic scheme for user access. For a more comprehensive overview, we refer to [1].

The development process is guided by an agent-oriented software engineering model, which is tailored to the specifics of JIAC IV. It comprises tools for agent specification including compilers for the different agent languages and tools to analyse and debug the running system.

On the single-agent level, agents are build out of reusable components, which can be reconfigured and exchanged even at run-time. A control structure for reactive, deliberative, and interactive behaviour is realised by a set of components.

The agent infrastructure has to facilitate dynamic interactions between agents. Based on the specifications of FIPA [2], it comprises the communication infrastructure as well as services to administer the agents of a society (Agent Management Service, AMS) and the services they supply (Directory Facilitator, DF).

Commercial applications have special requirements on the reliability and trustiness of the system. Therefore, the JIAC IV toolkit provides several management and security functionalities, which can be easily integrated and adapted as needed due to the overall scalability. Management services include configuration, fault management, and logging of system processes, which may be used for analysis as well as for accounting. Security issues are addressed by authorisation, authentification, and privacy mechanisms.

To provide a convenient way for the user to interact with the agent system, JIAC IV contains a generic scheme to translate agent functionality into human accessible services. Thereby, dedicated agents provide graphical user interfaces, which can be used by a single access point.

4 Marketplaces for Agents

We will now describe how many aspects of the marketplace metaphor can be mapped on concepts of the generic agent toolkit JIAC IV and how JIAC IV addresses many of the requirements we assessed for an infrastructure for electronic marketplaces. Then, we present the service scheme of JIAC IV that links single-agent behaviour and multi-agent interactions.

4.1 Marketplace Infrastructure

In JIAC IV, a marketplace[1] (Figure 3) is both, a run-time environment and a meeting point for agents. Marketplaces reflect the distribution of the system and can be used for organisational structuring. Mobile agents are able to change the marketplace they reside on at run-time. The agents on a marketplace can be distinguished according to their role as infrastructure agents, provider agents, and customer agents.

[1] Figure 3 is a snapshot of our tool for monitoring marketplaces. It shows the agents on a marketplace, their life-cycle state, and their on-going communications.

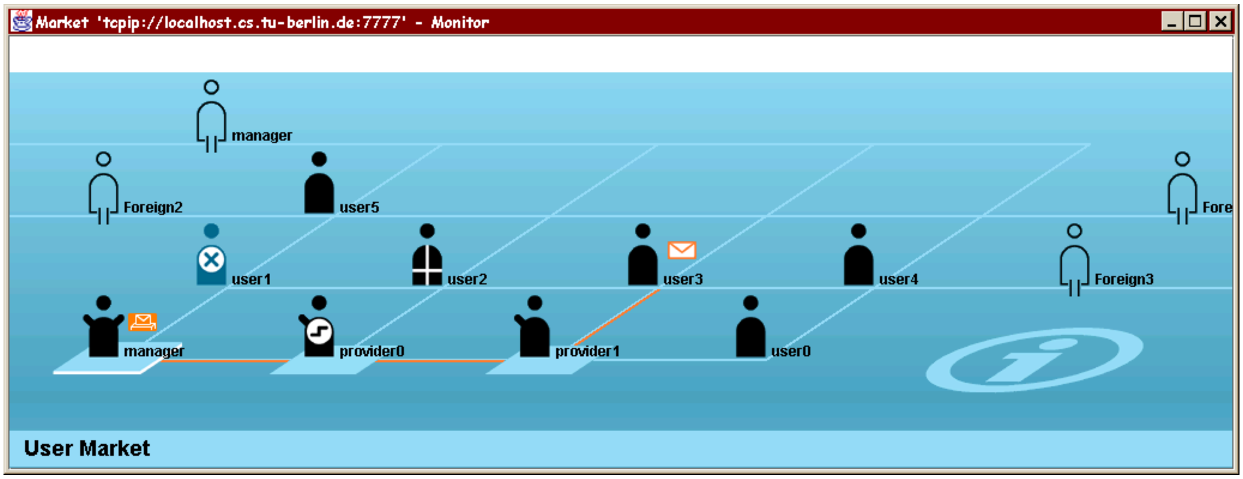

Figure 3. Agents on a Virtual Marketplace

The infrastructure agents realise the infrastructure of the electronic marketplace. The basic agent infrastructure of JIAC IV is compliant to the FIPA specification [2] for agent management consisting of Agent Management Service (AMS), Directory Facilitator (DF), and Agent Communication Channel (ACC). The manager agent provides the AMS functionality and thereby constitutes the marketplace. An agents resides on a marketplace, if it is registered at the AMS. The AMS serves as white pages informing about the location and the communication addresses of agents. The manager agent is also responsible for the transport of agents between marketplaces on migration.

The yellow pages are realised by the DF. This agent administers the services available at a group of marketplaces. Thereby, several marketplaces are interconnected by sharing a single service directory. The local DF at each marketplace provides access to this global directory. Agents register the services they provide at the local DF and can search the DF for services they need. The communication between agents on different marketplaces is handled by the ACC. In addition, an agent can have components for immediate communication with other agents.

This basic agent infrastructure of AMS, DF, and ACC supports dynamic and flexible interactions between agents, which can build the ground for business transactions. In addition, the JIAC IV toolkit contains agents and components for auxiliary services needed for electronic commerce. Security is established by authentification via certificates, encryption for privacy, authorisation to control access to services, and trust relationships between marketplaces. Trading is supported by accounting facilities, which can be integrated into conversation protocols for service interactions.

Accounting consists of three parts: metering, charging, and billing. Metering means to permanently collect data about the effort taken to fulfil a service. Charging converts this data into the costs for the customer. If the costs are not fixed, the customer can retrieve the current amount on request. When the service is finished, billing sums up the total costs into a bill.

Provider agents represent traders and offer commercial services. A commercial service is a service that makes use of accounting to protocol the costs. Customer agents are agents that act autonomously in behalf of the customer. Since provider and customer agents are application-specific, they are not part of the JIAC IV toolkit. Only their generic functionalities like the control architecture and the access to the infrastructure are given.

User access (for customers, but also for traders and infrastructure administration) is supported by JIAC IV in a uniform way by a multi access point. Mediator agents translate services from the agent level that uses formal languages for specification and communication to a human level with natural language and graphical user interfaces. This concept also supports different kinds of end user devices by exploiting a XML representation that may be translated into different interface formats.

4.2 Interactions between Agents by Services

A key feature of multi-agent systems is communication as a way of interaction that respects the autonomy of agents. Thereby, communication standards ensure interoperability by mutual understanding. Communication languages like KQML [3] and FIPA-ACL [2] provide a format to exchange messages between agents. The representational content of these messages is formulated by own languages with uniform syntax and semantics using terminologies expressed in shared ontologies [4]. Thus, communicating agents are enabled to interpret received messages as intended by the sender. Protocols regulate sequences of communicative acts for specific purposes to reduce the space of possible conversations. In addition, JIAC IV employs the concept of services to describe interactions in a standardised way.

4.2.1 Services

The main idea of a service in JIAC IV is that of an act one agent performs for another. Therefore, it is described by a plan operator stating conditions and effects of this act from the point of view of the customer. Thus, interactions are seamlessly integrated into the deliberative control mechanisms of JIAC IV that are based on plan operators. The customer can handle a service as any other act to control its behaviour and only the execution is delegated to the provider via communication.

On the other hand, formalising all interactions by services provides an explicit framework for generic requirements and parameters of interactions like accounting, billing, payment, and security. Services can thereby support competitive as well as cooperative or mixed societies.

The execution part of a service operator contains the service description. It consists of a unique identifier and generic parameters common to all services. These parameters include the content language and ontologies used to express the conditions and effects of the operator. It also comprises a list of protocols that can be used for service usage or negotiation.

To specify interactions by service operators has many advantages. Conditions and effects of using a service are described explicitly in a familiar manner. A dynamic selection of services can be done in the same way as for other actions, and the combination of services means nothing else than generating a plan with several service operators. Thus, service operators allow for flexible and automated interactions between agents.

Some particular aspects have to be considered when using service operators. For each service, several providers and protocols may be available, which have to be selected at planning or execution time. Especially for service operators inside of plans or scripts, commitments from the providers to supply it to the customer should be gathered in advance to ensure all steps of the plan or script to be executable. Also, the evaluation of a service intention may depend on the selected provider and protocol as well as on other service parameters, which might even be negotiable. Thus, components that select, evaluate, or execute service operators have to cope with these different possibilities of performing them.

4.2.2 Protocols

The communication language of JIAC IV is compliant to the specification for FIPA-ACL [2]. In JIAC IV, each speech act (except error handling) belongs to a conversation and is part of a protocol. Each service usage is a single conversation with a global unique identifier. The protocol is either the generic meta-protocol common to all services or a service-specific protocol, be it for negotiation or the service supply itself. Since every interaction is part of a service usage, all protocols have exactly two roles, the customer and the provider. The customer is always just one agent, while the role of the provider may be taken by several agents for a dynamic selection of the provider by negotiation. A service allows multiple providers, if it declares at least one protocol for negotiation.

4.2.3 The Meta-Protocol

The meta-protocol (Figure 4) frames every service usage as a generic ordering scheme for services. It is always initiated by the customer when executing a service operator. The initiating speech act is a request containing the unique identifier for the service and values for generic and specific parameters including the embedded protocols to be used. This is the only kind of speech act an agent has to process outside of an ongoing

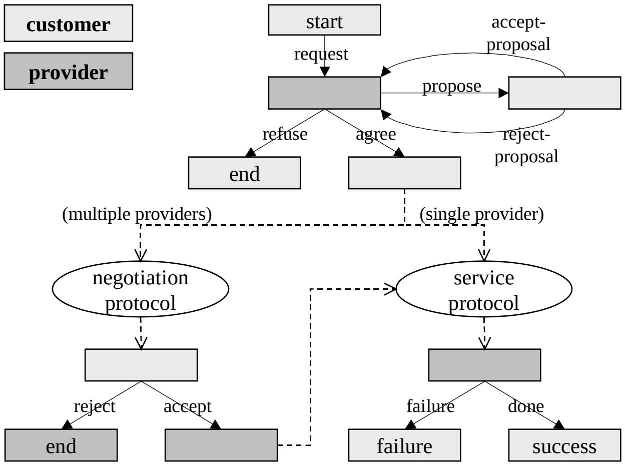

Figure 4: The Service Meta-Protocol

conversation, because it establishes a new conversation. An optional negotiation phase follows, where a provider can propose other values for the generic parameters, which the customer can accept or reject. If a proposal is rejected, the provider can make another proposal for the same parameter, otherwise only for other parameters. Finally, the provider has to agree or refuse the request. An agreement is only possible, if for every negotiated parameter a proposed value was accepted. A refused request finishes the conversation.

On agreement, the service-specific part begins. If multiple providers agreed, the negotiation protocol is performed to select the best provider by negotiating service-specific parameters. When it ends, the customer has to accept one provider and reject all others. Then, for the accepted provider as well as for a single provider, the service protocol starts. When the service protocol is finished, the provider sends the result of the service usage, which may be a “done” or a “failure” speech act. In the first case, the execution of the service operator ends with a success, otherwise with a failure.

In addition to the scheme of Figure 4, the meta-protocol allows communicative act types for error handling which may be used at any time by both roles. The first type is “cancel”, which cancels the conversation. The second type is “not-understood”, which refers to a speech act that can not be processed by the receiver, be it part of a conversation or not. Also, there might be a security protocol framing the meta-protocol handling the security requirements for a conversation.

The meta-protocol is handled by the component implementing the communication role. It provides a uniform scheme for interactions by ordering and negotiating services. Specific communications for a service are covered by embedded protocols.

4.2.4 Embedded Protocols

Inside of the generic meta-protocol, there are two types of embedded protocols, one for negotiation and one for service supply. For each service usage, these protocols may be chosen by negotiation from the ones prescribed by the service operator. Since the protocols are generic parameters, this takes place at the negotiation phase of the meta-protocol. Thereby, agents should only propose or accept protocols for which they have a corresponding protocol operator for their role.

To start an embedded protocol, each agent sets up a new conversation goal to take its role in the protocol, which leads to the execution of an appropriate protocol operator. If the meta-protocol already covers all communications needed in a conversation, the embedded protocol specifies no communications. This is still a kind of protocol in JIAC IV, because it is also realised by an protocol operator determining the decision or result. In this case, only the role sending the terminating speech act of the protocol, which is already part of the meta-protocol, needs to execute a protocol operator.

A protocol is a scheme for conversations determining which role may perform which communicative acts depending on the previous communications. A protocol operator has to reflect the structure of a protocol for one role by executing appropriate communicative operators to send or await speech acts accordingly. In addition, it takes the service-parameters as input and has a result as output.

The description of a protocol consists of a unique name and the type declaring it as a negotiation or service protocol and as containing communicative acts or not. Also, it specifies a content language and the ontologies used in communicative acts. Finally, it contains the service-specific parameters to be processed by a protocol operator. A protocol operator description adds to a normal operator the protocol description and the role.

5 Related Work

Guttman et al. [5] compare agent systems by a Consumer Buying Behaviour model adapted to agent-mediated electronic commerce. This model consists of six steps: need identification, product brokering, merchant brokering, negotiation, purchase and delivery, and finally product service and evaluation. They argue that agents are of benefit mainly for product and merchant brokering and for negotiation because of their autonomy. They discuss two kinds of systems. On the one hand, there are assistances that help the customer finding and comparing offers. On the other hand, agents negotiate autonomously in automated auctions.

Collins & Lee [6] describe the realisation of an electronic marketplace based on the generic agent toolkit ZEUS. They use only one agent type that is capable of buying as well as selling. These agents participate in auctions and negotiate according to exchangeable pricing strategies.

Tete a Tete [7] is an agent-based auction system, in which negotiations consider other parameters like warranties or service contracts besides of the price. In addition, it supports product and merchant brokering.

6 Conclusion and Outlook

Unlike most other agent-based approaches to electronic commerce, JIAC IV is not an application for specific business aspects but a generic agent toolkit with many features that make it a suitable platform for the development and deployment of electronic commerce applications. Many aspects of electronic marketplaces like white and yellow pages can be mapped onto the agent infrastructure of JIAC IV. Additional functionalities like security, accounting, and administration are included that are prerequisites for real-world commercial applications. Openness and scalability of single agents and agent societies are realised by modular approaches on all system levels and ensure the flexibility to cope with the dynamics of business under permanent competition. The service scheme makes business transactions an integral part of the system. Hence, JIAC IV fulfils most of the requirements we stated for infrastructures of electronic marketplaces.

Until now, we have only build a relative small application in the domain of traffic telematics that utilises the marketplace concept of JIAC IV. We intend to gain more experience of the benefits and drawbacks of our system by realising further applications in different domains. Also, we want to extend JIAC IV by additional functionalities specific for electronic commerce, e.g. support for auctions, personalisation, or on-line payment.

References

[1] Stefan Fricke, Karsten Bsufka, Jan Keiser, Torge Schmidt, Ralf Sesseler, Sahin Albayrak. A Toolkit for the Realization of Agent-based Telematic Services and Telecommunication Applications. in: Communications of the ACM, April 2001.

[2] FIPA 2000 Specification. www.fipa.org.

[3] Tim Finin, Yannis Labrou, James Mayfield. KQML as an Agent Communication Language. in: Jeff Bradshaw (Ed.), Software Agents, MIT-Press, Cambridge, 1995.

[4] Thomas R. Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University, 1993.

[5] Robert H. Guttman, Alexandos G. Moukas, Pattie Maes, Agents as mediators in electronic commerce, EM - Electronic Markets, Vol. 8, No. 1, 1998

[6] Jaron C. Collis, Lyndon C. Lee. Building Electronic Marketplaces with the ZEUS Agent Toolkit. Proceedings of the Agent Mediated Electronic Trading (AMET) Workshop, May 1998, p17-32.

[7] Robert H. Guttman, Pattie Maes; Agent-mediated Integrative Negotiation for Retail Electronic Commerce. Proceedings of the Workshop on Agent Mediated Electronic Trading 1998 (AMET'98)