Building Agents for Service Provisioning out of Components*

Ralf Sesseler
DAI-Lab, TU Berlin
Sekr. FR 6-7, Franklinstr. 28/29
D-10587 Berlin, Germany
+49 (0)30 314-21763

sesseler@cs.tu-berlin.de

ABSTRACT
CASA is an open, scalable agent architecture for service provisioning. Its interaction scheme between agents is based on a formalized concept of services to allow dynamic and flexible selection and combination of services and their providers.

Keywords
agent architectures, agent-based service provisioning, multi-agent collaboration, communication protocols

1. INTRODUCTION
Computer networks like the Internet promise many advantages as electronic platforms for service provisioning. Service supply is very flexible and effective, while service access is independent from time and location of usage. The open, distributed, and heterogeneous character of the networks offers many new possibilities, but also raises new technological demands. Current platforms are often proprietary stand-alone systems that miss to utilise the whole potential of the networks and their dynamics, because they lack interoperability.

An alternative is given by agent technology, which is concerned with flexible interactions in open, distributed systems. In the following, we describe the interaction scheme of CASA[1] (Component Architecture for Service Agents), a multi-agent architecture that facilitates the design of interoperable service platforms. The aim of CASA is automated and flexible provisioning and usage of services in domains like electronic commerce and telematics by dynamic supply, selection, and combination of services. This is done by combining the multi-agent approach and a service concept that maps service provisioning and usage to the agent level.

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

[1] The JIAC IV agent system [2] is implemented based on CASA. Thanks to all at DAI-Lab involved in the implementation.

2. ARCHITECTURE
The architecture of CASA comprises three levels: At the structural level, an agent consists of a set of exchangeable components. The agent behaviour is determined by a control architecture of interrelated components. Finally, an agent infrastructure interconnects agents to a society.

For scalability, an agent is built by selecting an arbitrary set of reusable components as needed to fulfil its tasks. The component set can be changed even at run-time by adding, removing, or replacing components. A core agent manages the component set and supports decoupled interactions between components by directed message passing. The components are interrelated by roles describing their interactive capabilities to abstract from different implementations of the same functionality.

This component framework is open for different kinds of control architectures to be defined by sets of interdependent component roles. A default architecture of components to realise reactive, deliberative, and interactive behaviour exists. This architecture is divided into a knowledge-based kernel consisting of components to store and to process knowledge and a periphery with components that provide interfaces to the environment or additional functionality. The use of explicit knowledge representations allows thereby a more flexible behaviour control and to handle the formal descriptions of services.

Agents form open societies, where agents join and leave and the supply of services changes dynamically. Dedicated agents provide infrastructure services that support to find services and to interact with other agents. According to the FIPA Agent Management Specification [1], there are services to administer the agents of a society (Agent Management Service, AMS) and their service supply (Directory Facilitator, DF).

3. INTERACTIONS BY SERVICES
A society of agents is characterised by the interactions taking place at a communicative level. To allow interactions between different agents, there is a need for interoperability, which is ensured in technical systems by standardisation.

Most agent systems employ standards for interactions at different levels. Communication languages provide a common 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. Thus, communicating agents are enabled to interpret received messages as intended by the sender. In addition, proto-

cols regulate sequences of communicative acts for specific purposes to reduce the space of possible conversations.

On the top of this, CASA adds the concept of services. A service provides a generic scheme to describe interactions between two agents, the provider and the customer. A predefined protocol frames the service usage, allowing embedded service-specific protocols. Furthermore, the default architecture supports the service conception by operators for services and protocols and by a component role that performs ongoing communications.

3.1 Services

The main idea of a service in CASA is that of an act one agent performs for another one. Therefore, it is described by an operator stating conditions and effects of this act from the point of view of the customer. Thus, 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. This service scheme can thereby support competitive as well as cooperative or mixed societies.

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.

3.2 Protocols

While service operators describe the services in abstract, the process of service usage and its communications is described by protocols. Thereby, each service usage constitutes a single conversation, while each of its speech acts belongs to a protocol. This protocol is either the generic meta-protocol of CASA 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.

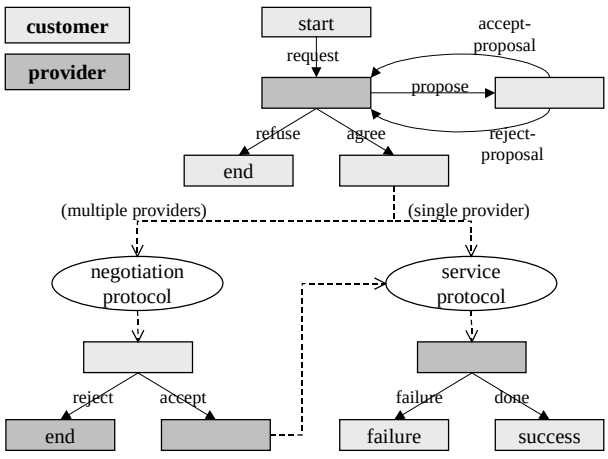

Figure 1. Meta-Protocol for Service Interactions

The meta-protocol (Figure 1) frames every service usage as a generic ordering and negotiation scheme for services. It is always initiated by the customer when executing a service operator. The initiating speech act is a request containing a 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 conversation, because it establishes a new conversation. An optional negotiation phase for the generic parameters follows. Finally, the provider has to agree or refuse the request. 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 other. 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 ends the meta-protocol with a success or failure.

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.

4. CONCLUSION

The CASA architecture provides a pragmatic approach for flexible, automated interactions in open, dynamic environments. Its component framework facilitates the design and creation of a variety of different agents and agent types. Also, it makes agents scalable and allows dynamic reconfigurations at run-time. The default architecture supports a flexible control of agent behaviour by combining reactive, deliberative, and interactive capabilities. The service concept guides interactions by an open, but reliable scheme. Together with standardisations at other levels of interaction and with infrastructure services, it ensures a high level of interoperability within a multi-agent system including dynamic usage and combination of services.

A suitable application domain of the CASA architecture is the realisation of platforms for service provisioning in electronic networks. Services are thereby provided to the human user by particular agents that access the capabilities of the agent society to fulfil their tasks. The flexibility and interoperability of the multi-agent-system utilises the openness and dynamics of the networks in favour of a likewise open and dynamic service environment.

5. REFERENCES

[1] FIPA 98 Specification, Part 1: Agent Management. ww.fipa.org, 1998.

[2] 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.