Chapter 8

Applications of Information Agent Systems

M. Klusch, H.-J. Bürckert, P. Funk, A. Gerber, C. Russ
DFKI German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3, D-66123 Saarbrücken
Germany
`{klusch,hjb,funk,agerber,russ}@dfki.de`

Intelligent information agents are autonomous computational software entities that are especially meant (1) to provide a pro-active resource discovery, (2) to resolve information impedance of information consumers and providers, and (3) to offer value-added information services and products. These agents are supposed to cope with the difficulties associated with the information overload of the user preferably just in time. In this chapter we describe selected examples of systems of information agents for applications in telematics, e-business, supply chain and repository management. These systems have been developed and implemented by the research group on multi-agent systems at the German Research Center for Artificial Intelligence.

1 Introduction

An *information agent* is an autonomous, computational software entity (an intelligent agent) that has access to one or multiple, heterogeneous and geographically distributed information sources, and which pro-actively acquires, mediates, and maintains relevant information on behalf of users or other agents preferably just-in-time (Klusch, 1999; Klusch, 2001). Thus, any information agent is supposed to satisfy one or multiple of the following requirements.

1. *Information acquisition and management.* An information agent is capable of providing transparent access to one or many different information sources. It may retrieve, extract, analyze, and filter data,

monitor sources, and update relevant information on behalf of its users or other agents. The acquisition of information covers activities such as advanced information retrieval in databases and purchasing of relevant information on appropriate electronic marketplaces.

2. *Information synthesis and presentation*. The agent is able to fuse heterogeneous data and to provide unified, multi-dimensional views on the relevant information to the user.

3. *Intelligent user assistance*. The agent dynamically adapts to changes in user preferences, the information, and network environment. It provides intelligent, interactive assistance for common users supporting their information-based business on the Internet.

In general, we differentiate between *communication*, *knowledge*, *collaboration*, and rather low-level *task skills* of an information agent (cf. Figure 1). In this figure, the corresponding key enabling technologies are listed for each of the different types of skills.

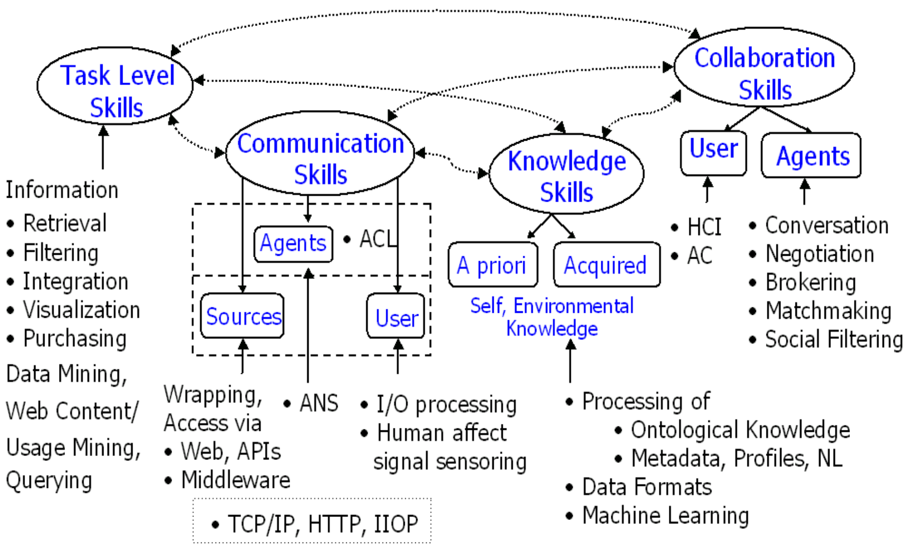

Figure 1. Basic skills of an information agent

• *Communication.* Information agents communicate either with information systems and databases, human users, or other agents. In the latter case, the use of an agent communication language such as FIPA ACL has to be considered on top of, for example, middleware platforms or specific APIs such as CORBA/IIOP, JDBC (Java Database Connectivity), and Microsoft's ODBC (Open Database Connectivity) or OKBC (Open Knowledge Base Connectivity), respectively.

- *Knowledge*. An information agent copes with pre-given or dynamically acquired data, information, and knowledge on its environment and itself. This may include representation and processing of ontological knowledge and metadata, profiles and natural language input, translation of data formats as well as the application of machine learning techniques.

- *Collaboration*. The higher level interaction and collaboration with other agents may rely on techniques and protocols such as for service brokering, matchmaking, negotiation, and social filtering. Interaction with human users mainly corresponds to the application of techniques stemming from human-computer interaction (HCI) and affective computing (AC).

- *Task*. The core tasks of any information agent include retrieval, filtering, integrating, and visualizing relevant information. This implies in particular effective utilization of methods for Web content and/or usage mining as well as optimal query planning and processing.

In fact, the extent of utilizing one or multiple of these categories of basic skills varies depending on the application domain and tasks the information agent has to accomplish. For a comprehensive survey on information agent technology for the Internet we refer the reader to (Klusch, 2001).

The remainder of this chapter is structured as follows. In the sections 2, 3, and 4 we present different applications of holonic-structured information agent systems in the domain of telematics and e-business. Section 2 briefly describe systems for dispatch support (TeleTruck), and remote control and diagnostics of technical systms (TeleService). In section 3 we survey a system of collaborating information for mobile integrated services for e-trading in selected application domains of forestry and agriculture. Section 4 then proposes a general agent-based coordination infrastructure for retail supply webs in the e-business domain. Finally, section 5 presents an approach of information agents to support software repositories. Each of these applications have been developed and implemented by the research group on multi-agent systems at the German Research Center for Artificial Intelligence.

2 Holonic Agents for Telematics

2.1 TELETRUCK – A Dispatch Support System

The TELETRUCK application prototype aims at supporting dispatch officers of forwarding companies. It allows for interaction with on-board systems installed in the trucks of the company using satellite or mobile phone communication. Different copies of TELETRUCK can be connected to distributed system for collaborative transport scheduling – used by dispatch officers of collaborating companies or different sites or departments of one company.

We have modeled the basic physical objects (drivers, trucks, trailers, containers, dispatch officer) of the transportation domain explicitly by agents. The agents can join together forming *holonic agents* or *holons* that act in a corporate way: driver, truck, trailer, and container agents compose to a transport holon representing a physical transportation unit (a road train or an articulated vehicle together with its driver), which is able to execute a transport order. These holons are coordinated by a dispatch agent. Together the transport holons and their dispatch agent form again a holon, which acts as planning and scheduling assistant for a human dispatch officer.

Technically the dispatch agent distributes incoming transport orders to transport holons using contract net protocols. The transport holons compute a bid for an announced order on the basis of their current transport plan (for earlier distributed orders). Dispatch agents select the best bid and award the order to that holon. Not every transport holon has to be composed at this stage. Some may still be incomplete, eg., missing a trailer with further loading space, which could be integrated.

Others still have to be composed of idle component agents from the scratch. Composition and extension of transport holons is organised by a cascading sequence of transport protocols (cf. Bürckert *et al.* 2000 for a detailed description.)

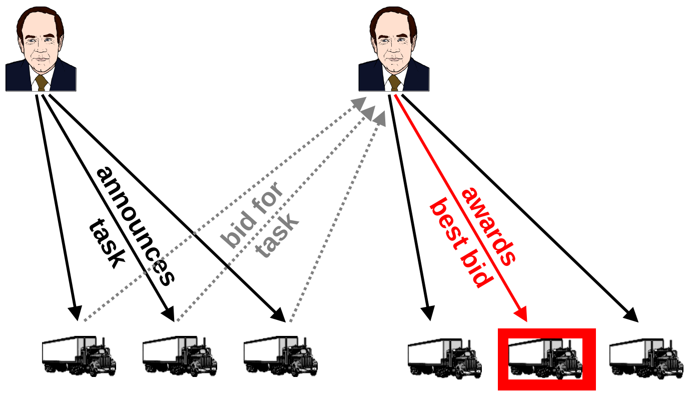

Figure 2. Contract Net Protocol for Order Distribution.

Since such transport plans are computed for a sequential stream of incoming orders only optimizing the local plans of the transport holons, they are suboptimal by nature. Hence the exchange of orders between different transport holons is organized by a simulated trading protocol. Again coordinated by the dispatch agent, transport holons can sell some of their orders and buy others, in order to improve their plans.

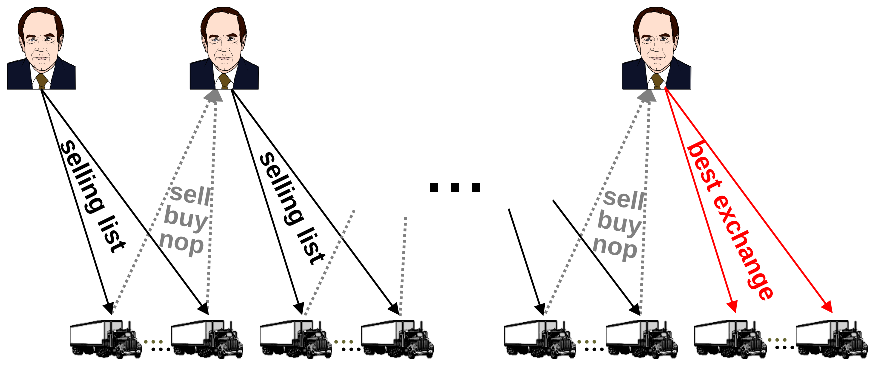

Figure 3. Simulated Trading Protocol for transport plan optimization.

Because of the randomized selection of the orders to be sold or bought, this method results in a simulated annealing like procedure that optimizes the global transport plan of one dispatch officer incrementally. For several connected TELETRUCK systems we use a similar organization of the whole network. One system acts as coordinator – it makes sense to have here a system without transport holons. It announces – again within a contract net protocol – the transport orders to the other systems' dispatch agents, which then in turn announce them to their transport holons as described above. A simulated trading protocol can also be used for optimizing the transport plans globally for the whole network of collaborating systems. However, if we support collaborating companies, we use competitive versions of the contract net and the simulated trading protocol (cf. Vierke 2000).

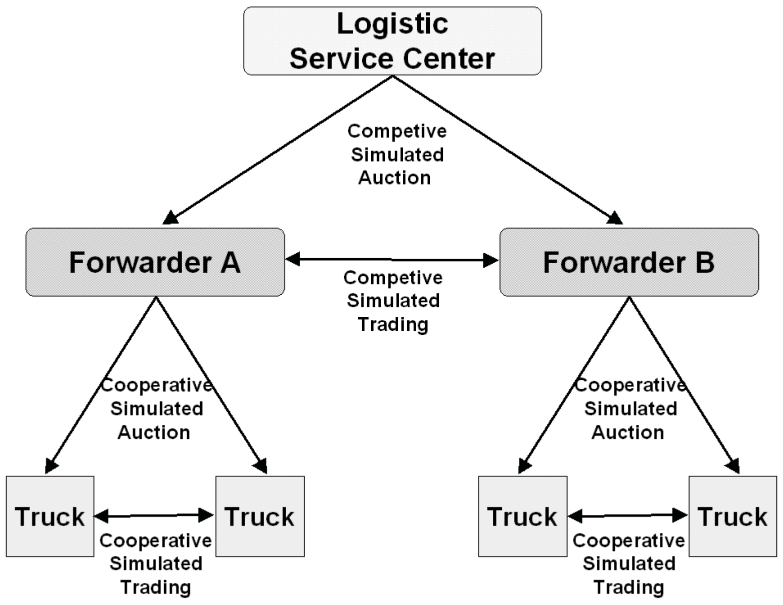
Logistic Service Center
Competive Simulated Auction
Forwarder A
Forwarder B
Competive Simulated Trading
Cooperative Simulated Auction
Cooperative Simulated Auction
Truck
Truck
Truck
Truck
Cooperative Simulated Trading
Cooperative Simulated Trading

Figure 4. Contract Net and Simulated Trading for collaborating companies.

2.2 TeleService – Mobile Agents for Remote Applications

Remote diagnostics, maintenance, control and other tele-services for stationary or mobile technical systems may become a typical field for mobile agent technology. The basic idea is to support service providers of technical systems, which are distributed – in some cases all over the world, in their daily job. Usually a service center sends out service engineers to its customers – for routine checks or on demand. The engineers check the system or solve the problem and move to their next job. Mobile agents could overtake such service tasks – at least for simple or routine cases – provided the technical systems are connected with the server of the service provider.

In our TeleService approach we have software agents representing each technical system. Control devices make the current state (such as the current location in the case of mobile systems) fully available to these agents and also provide the agents with interfaces for communication and interaction.

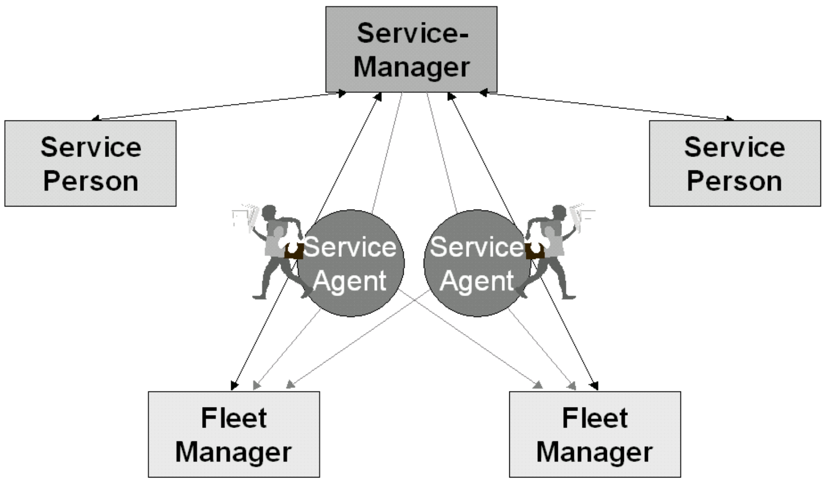

Figure 5. Service agents are sent out by service management agents.

Mobile service agents are used for diagnostics, maintenance and control of the systems. Starting from the server – at a service center or at the customer – they visit local agents and execute their tasks. Where the underlying computing platform is physically mobile, the hosting of agents on the onboard computer is supported.

Currently we are instantiating this approach for forklift trucks which are equipped with control processors and onboard computers. At the service center, an agent coordinates service management for several customers and their forklift fleets. At each customer's server, other agents are responsible for the fleet management. Images of the forklifts are realized as agents which know all about the current status of the forklifts.

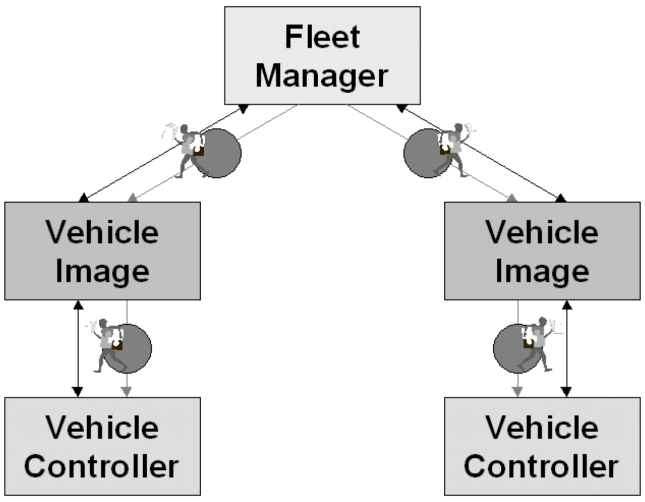

Figure 6. The fleet management agent moves the service agents to vehicle agents which in turn upload the service agents to the board computer.

Service agents for different services can be sent by the service management agent via the fleet management agents to the image agents, which can upload the service agents onto the onboard computers of the forklifts (cf. Figure 5 and 6).

The agent based approach allows for high flexibility and frees the agent from dependence on a permanent data connection between the servers and onboard systems, which would be needed in a conventional architecture for remote applications: as service agents can execute their tasks autonomously a permanent connection is not necessary, even for control tasks. Only uploading of the service agent onto the onboard computer requires a brief data connection. In the reverse direction, only the results of the activity of the service agent are communicated.

Since the onboard computers and control processors are usually of very low computational power, the service agents cannot be realized as big intelligent expert systems but as small agents with very specific expertise. In order to solve, for instance, a complicated diagnostics problem several of these specific agents may be sent one after the other to the onboard system for analyzing different aspects of the failure. Their results are collected by the service management agent which then decides for further activities to repair the failure. Such activities could range from tasks to be done again by service agents over tasks to be solved by technical staff at the customer (e.g. a battery exchange) to even complex tasks that require a service engineer which has to be sent by the service provider.

The intermediate layers (i.e., the fleet manager and the images of the target systems) of the approach allow for secure transfer of the mobile agents provided suitable secure interaction protocols are used. Only the image agents of the target systems are allowed to upload service agents onto the onboard computers.

3 CASA: Agents for Mobile Integrated Commerce in Forestry and Agriculture[1]

3.1 Motivation

The project CASA (Cooperative Agents and Integrated Services for Logistic and Electronic Trading in Forestry and Agriculture) focuses on the development of an agent-based information and trading network (CASA ITN) for mobile, integrated services in selected application scenarios within the domains of forestry and agriculture in the Saarland county. The CASA ITN can be installed at users connected to the Internet; registered users may then access the complete CASA services via a secure Extranet and mobile telecommunication network.

Agent-mediated services of the CASA ITN aim to support the main operative business processes users are performing in each of the following application scenarios: (1) customer-oriented, dynamic timber production, (2) mobile trading of timber via different types of auctions, fixed or negotiable price, and (3) electronic trading of cereals.

[1] This research and development is sponsored by the Saarland Ministry of Economics under grant 032000

The approach taken for providing information and trading services in the CASA ITN focuses on the effective integration of production, logistics and trading processes of respective parts of the particular supply chain for each of the application scenarios.

This approach is motivated by the paradigm of *integrated commerce* (i-commerce) which can be seen as an operational extension of traditional e-commerce. The basic ideas of i-commerce are (a) to get customers more involved in the activities related to his/her orders or tasks to be jointly accomplished by one or multiple contractors, auxiliary/finishing industry, and shipping companies, and (b) to get related processes in the considered supply chain more integrated in practice.

The agent-based CASA service suite for i-commerce can be accessed by users at any time, anywhere; selected information and trading services are available via mobile WAP-enabled computing and telecommunication devices such as smart phones, PDAs and communicators. Efficient coordination of services within the CASA ITN is done via appropriate types of collaborating software agents.

3.2 CASA Agents and Services

3.2.1 Holonic Agent System of the CASA ITN

We differentiate between following groups of participants in the CASA ITN: *producers* offer their goods; *buyers* want to purchase several goods; *retailers* act in agency of a company or of their own profit; *logistics companies* are responsible for transportation tasks, storage and resource management. Each member of these groups is represented in the ITN by an appropriate kind of software agent (cf. Figure 7).

For reasons of effectively accomplishing complex, mostly hierarchically decomposed tasks and resource allocations in the selected application scenarios (cf. section 3.3) we use the concept of *holonic agents* (Gerber, Siekmann, and Vierke 1999) and *holonic multi-agent systems (H-MAS)* (Bürckert, Fischer, and Vierke 1998). A holonic agent (or holon) co-ordinates and controls the activities and information flow of its subagents. In a H-MAS, autonomous agents may join others to form, reconfigure, or leave a holon.

A human user in the ITN is represented by a special holonic agent called *personal assistant*. It pro-actively acts on behalf of its user even if (s)he is off-line; the personal assistant is the coordinating head of a set of other specialized agents for individual negotiation, participation in auctions, finding other relevant partners for trades and information, and elaboration of optimal trading strategies over time.

Each corporation is represented by a special holonic agent system according to its task-oriented subdivision into departments for information management, logistics, and production planing.

In this context we presume that:

- *information management* services provide information either on certain products and related production processes, or on current market situation and potential competitors.

- *logistics* services support the co-ordination of machines for production and transport, human resources, and storage capacities.

- *production planning* services support short-, middle-, and long-term product planning cycles.

A corporation holon is constituted by other holonic agents each of them representing a special department and corresponding complex tasks and services. Since in the CASA ITN the roles of buyer/retailer or and seller/producer may be interchangeably used both are modelled by similar holonic agent structures. In addition, logistics companies are usually contracted by other corporations for the purpose of time- and cost-saving delivery of goods on demand. Finally, to enable different kinds of trading between the participants of the CASA ITN such as multiple online auctions and sales via fixed or negotiable prices in bilateral negotiations at the same time we developed agent-based services for a distributed virtual market place. This enables each participant to, for example, initiate and perform one or multiple auctions or negotiations for sales of its own goods and products.

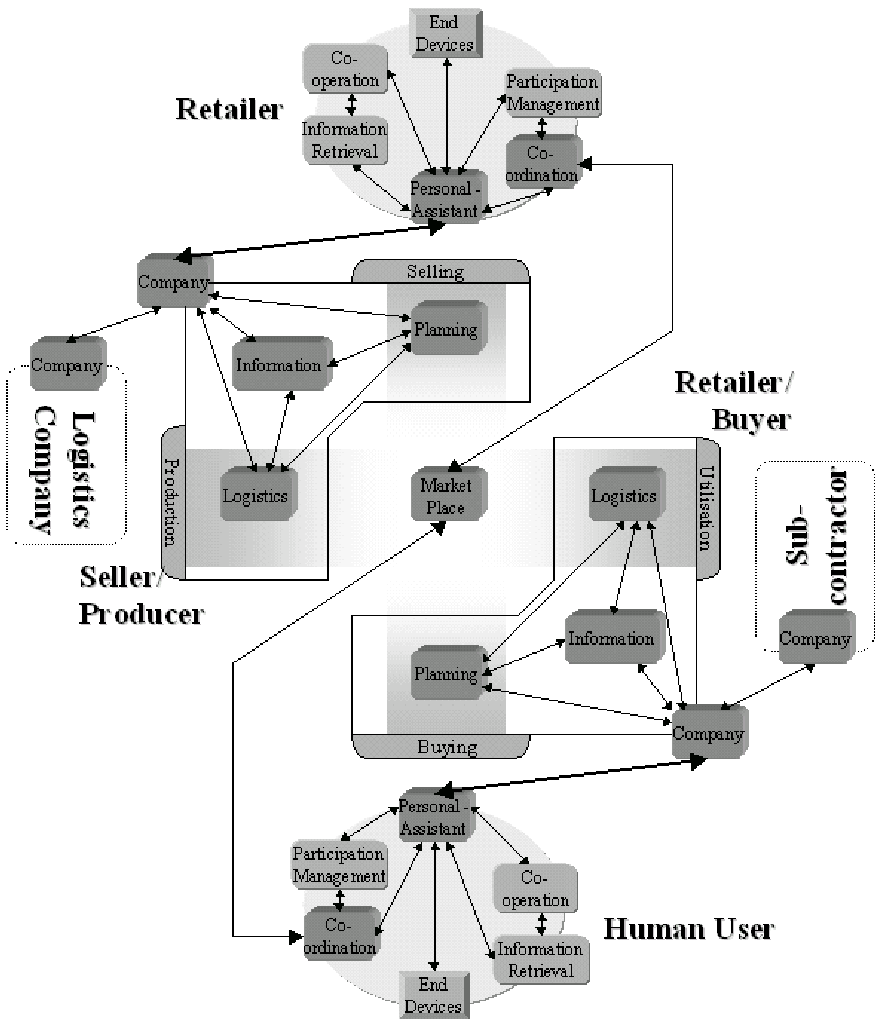

Figure 7. Overview of the holonic CASA agent system

3.2.2 Agent-Based Services of the CASA ITN

In general, the CASA agent society provides the following classes of services to its users and is co-ordinating the effective execution of each of the services.

- *Auction mechanisms* including Dutch, English, Vickrey, and First-Price-Sealed-Bid auctions.

- *Integrated services* for

    - *dynamic pricing*. Agents collect additional information on transportation costs and other constraints to meet as a decision-support service for its users, for example, during the bidding processes of some auctions.

    - *logistics*. These services provide dynamic, approximately optimal (re-)scheduling and (re-)planning of transportation.

    - *information management.* Agents gather relevant information on behalf of its users in different trading and production settings.

- *Mobile services* to let the users access most services of the CASA ITN also on WAP-enabled mobile devices.

3.3 Application Scenarios

In brief, the application scenarios of the CASA ITN are as follows.

- *Customer-oriented dynamic timber production* (KDPS): Foresters and timber harvester appropriately cooperate with pro-active support of service providing agents of the CASA ITN to accomplish the goal of an individual customer's order to deliver a certain quantity of timber with specified quality at a given time of delivery. The approximately optimal, dynamic (re-) planning and coordination of integrated services for harvesting, processing, and transportation is performed by the CASA multiagent system and partly accessible via mobile WAP-enabled devices.

- *Mobile timber sales* (MHS): Registered users of the CASA ITN may set up and participate in one or multiple timber auctions via mobile WAP-enabled mobile devices.

- E-trading of cereals: Similar to the MHS scenario registered users of the CASA ITN may trade grains via (a) auctions (with the farmer being the auctioneer) or (b) multi-lateral negotiations with interested parties following a version of the extended contract net protocol. Many of the required services in this scenario can be appropriately re-used from those developed in the MHS and KDPS scenario

The first two of these application scenarios have been implemented in the CASA ITN.

3.3.1 Mobile Timber Sales: Services, Interactions, and Agents

In this special scenario each ranger may sell timber via different kinds of auctions, fixed or negotiable sales offers to other registered users of the CASA ITN. Main benefits of the agent-based service support are the concurrent monitoring and computation of optimal transport costs per individual bid or buying offer, and the full mobile service support via WAP-enabled mobile devices.

In general, the mobile timber sales services of the CASA ITN enable registered users (a) to initiate and/or participate in one or multiple timber auctions, and (b) to sell or buy timber at fixed or negotiable prices. In the first case, any user may set up an auction thereby acting as an auctioneer of (parts of) his/her own goods for sale. The CASA ITN currently provides the following types of auctions: Dutch, English, Vickrey, and First-Price-Sealed-Bid. There is no central market place. An auctioneer may either be identical with the seller of some goods, or different. In the latter case, an auction could take place at a different but trusted site on behalf of the seller which in turn may increase the trust of potential buyers of the auction goods. The auction server has been built upon a general holonic coordination server (cf. section 4.5.2; Gerber and Russ, 2000). Any user may invoke *integrated services* for decision-support during the participation in one or multiple different auctions. For example, a personal CASA agent may concurrently determine the optimal transportation costs and delivery dates of some auction good for each individual bid of its user. As a result, the agent may notify its user in real-time if estimated optimal transport costs exceed the allowed limit due to given buying preferences or if some deadlines are at risk to be exceeded.

The integrated trading services are provided by CASA agents to support interaction between the participants in the scenario, that is seller and/or auctioneer, buyers, and shipping companies. Figure 8 summarizes such interaction in case of an auction. First a seller initiates an auction on a trusted auction site and informs potential buyers about offered goods, the auction type, and related bidding policy. During an auction bidders may evaluate their bids, monitor the current auction process, and place bids according to their individual bidding strategy. Latter may evolve depending on the result of the integrated services and respective benerfits for the individual bidder.

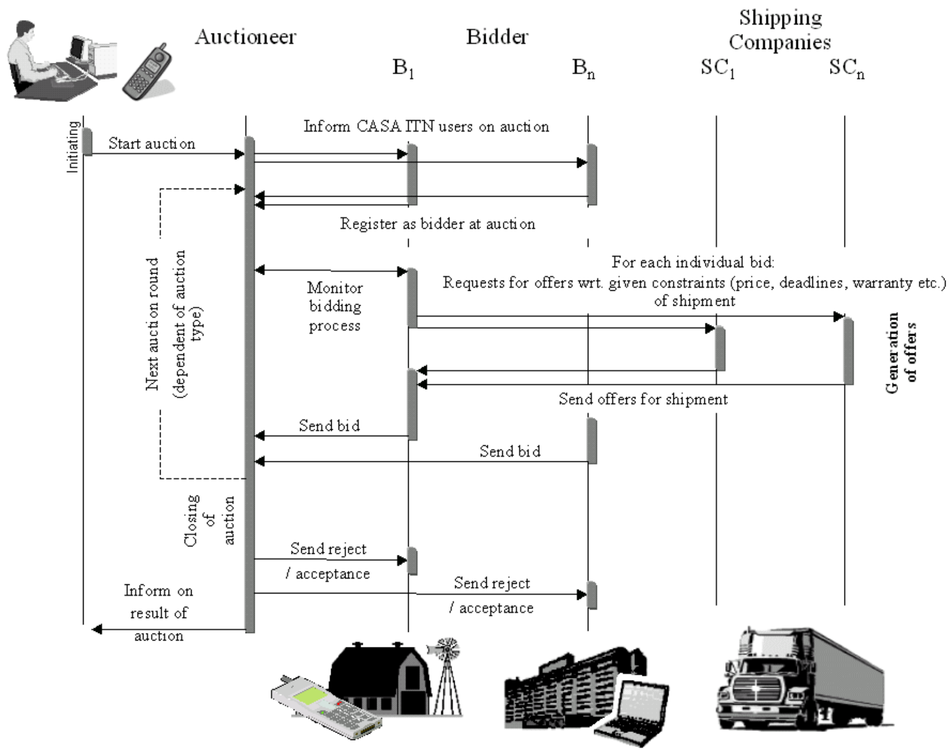
Auctioneer
Bidder
Shipping Companies
B_1
B_n
SC_1
SC_n
Initiating
Start auction
Inform CASA ITN users on auction
Register as bidder at auction
Next auction round (dependent of auction type)
Monitor bidding process
For each individual bid: Requests for offers wrt. given constraints (price, deadlines, warranty etc.) of shipment
Generation of offers
Send offers for shipment
Send bid
Send bid
Closing of auction
Send reject / acceptance
Send reject / acceptance
Inform on result of auction

Figure 8. Interaction between participants in the MHS scenario

In addition, each of the information and trading services is available for registered users of the CASA ITN on mobile WAP enabled devices such as smart phones or PDAs connected to the internet via modem. A synchronisation between the mobile and PC-based home computing environment in the CASA ITN is co-ordinated by appropriate CASA agents. Participation in any trading process can be delegated by the user to its personal agent which then is in charge of negotiating or bidding at an auction, and notifying its user when needed, e.g., via SMS or email.

3.3.2 Relevant Holonic Agents in the MHS Scenario

Figure 9 shows the different kinds of holonic agents which play an active role in the MHS scenario. These are holons for users as buyers or sellers/auctioneers, and shipping companies. We distinguish between buyers with or without logistics capabilities. In the latter case, carriers have to be appropriately contracted by the buyer. A company holon is structured into three different department holons as mentioned above.

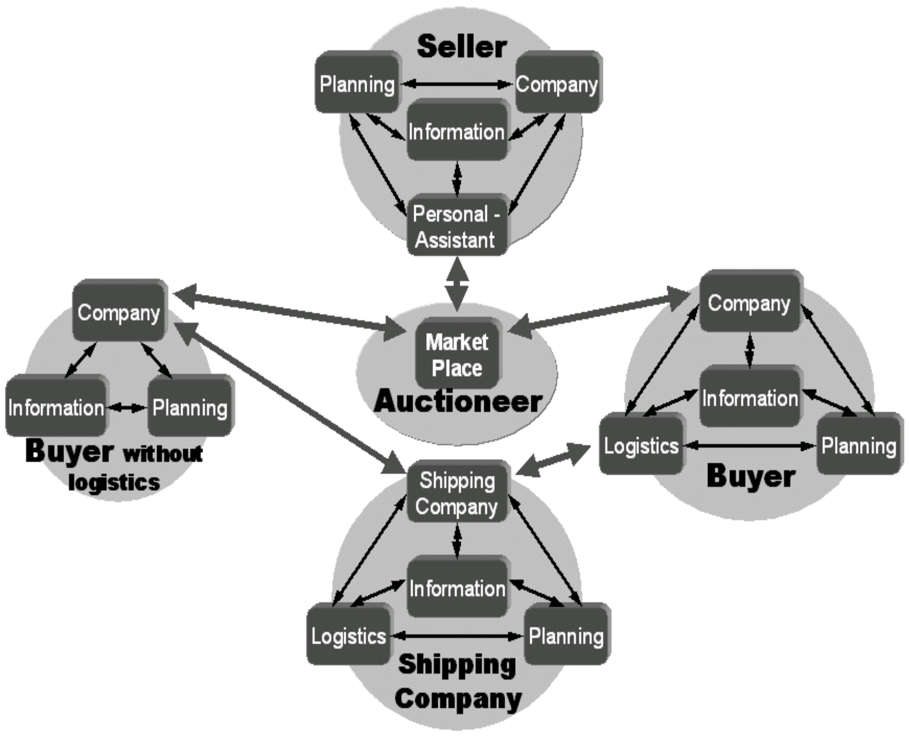

Figure 9: Relevant holons in the mobile timber sales scenario

3.3.3 Implementation

The services of the CASA ITN have been implemented in Java (JDK 1.3); the prototype of the system has been extensively tested on

Windows NT/ME/2000 machines. CASA holonic agents are modeled in accordance with the InteRRaP agent architecture and implemented using the standard open source FIPA-compliant agent system development environment FIPA-OS 2.0. Implementation of the mobile application services of the CASA ITN is compliant with the WAP 1.1 standard. These services are accessible by users of the current version of the ITN via the secure T-D1 WAP-gateway of the Deutsche Telekom AG. Data security is provided by the virtual private network suite of Checkpoint AG.

4 MAS-R/3: A Multi-Agent Co-ordination Infrastructure for Retail Supply Webs[2]

4.1 Motivation

According to Booz-Allen & Hamilton, in retail industry as well as in many other industries supply structures are evolving which may be characterized as "*complex sets of relationships that appear more web-like than chain-like*". Such structures are called *supply webs* (Laseter 1998). Partnerships between autonomous business entities in these supply webs can be flexibly contracted or withdrawn and are predominantly short-dated. This may cause complex coordination problems since the resulting many-to-many interactions and instantiated supply paths are not stable but may dynamically change. The coordination of the manifold interactions and internal planning tasks overstrains present-day inventory control systems.

We tackle these coordination problems by applying the concept of intelligent agents to the design of retail supply webs. Each operative unit of a retail supply web is modelled as an agent which represents either a supplier, producer, broker, wholesaler, retailer, warehouse, distribution center, branch, or logistics service provider. The coordination infrastructure for these *supply web agents* relies on *coordination agents, services, and mechanisms* such as auction server agents. Latter may provide other services such as matchmaking

[2] This work has been sponsored by the SAP AG subsidiary SAP Retail Solutions.

services, and auction types, coalition forming mechanisms for (re-) allocation of resources, respectively

In subsequent sections we briefly describe the considered application domain, our proposed design for a wartehouse agent as an example of a supply web agent, and a short discussion of the implemented coordination server.

4.2 The Supply Web Application Domain

The supply web application domain can be structured into three different layers each of them representing a different view on the domain (cf. Figure 10). The first layer represents the *supply web structure*. As in any supply chain available resources are (a) transformed to goods or products along some supply paths, (b) allocated to brokers, wholesalers and/or retail concerns, and finally (c) distributed by them (potentially via additional intermediators) to the consumers. However, in modern competitve settings of electronic markets supply paths are becoming more and more established quite short-dated, that is not until a real supply flow between appropriate supply web entities is impending. This implies the need to flexibly exchange parts of supply paths. The situation gets even more complex in cases where the outcome of the competition of supply web entities for the assignment of goods in the supply chain is not known in advance (see red circle in Figure 10). Depending on who wins this competition all of the subsequent entities in the “winning” supply path are enforced to adapt their plans correspondingly. The second layer models what we can consider as the *agentification view*, i.e., how a set of supply web agents is mapped to corresponding "real" entities within the supply topology.

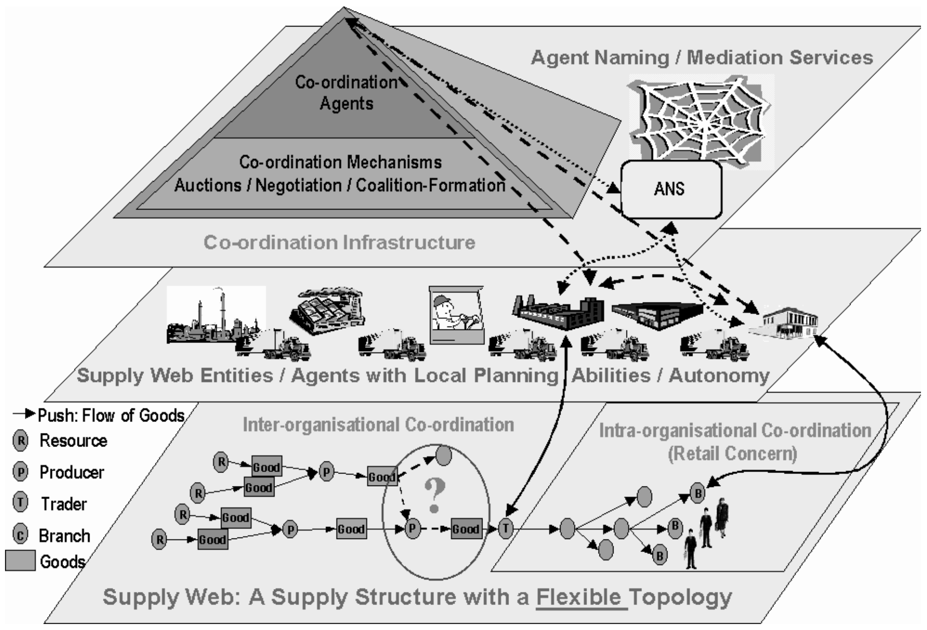
Agent Naming / Mediation Services
Co-ordination Agents
Co-ordination Mechanisms
Auctions / Negotiation / Coalition-Formation
ANS
Co-ordination Infrastructure
Supply Web Entities / Agents with Local Planning Abilities / Autonomy
Push: Flow of Goods
Resource
Producer
Trader
Branch
Goods
Inter-organisational Co-ordination
Intra-organisational Co-ordination (Retail Concern)
Good
?
Supply Web: A Supply Structure with a Flexible Topology

Figure 10: Three views on the supply web application domain

Finally, on top of these layers the *coordination layer* consists of two main components: a mediation infrastructure providing agent naming or yellow page services, and a coordination infrastructure which consists of special coordination agents endowed with appropriate coordination mechanisms. Agents may interact after they registered themselves at an agent name server (ANS).

4.3 Agentification of Supply Web Entities

How can supply web entities be agentified, how can the respective supply web agents be modelled in our scenario? We have implemented the operative supply chain units in Java by using the proprietary agent development tool *gACE* (generic Agent Control Engine) (gACE). *gACE* agents are equipped with a high level control architecture, called *Procedural Reasoning System (PRS)* (Ingrand et al. 1996), which allows the agents to behave goal-directed while staying responsive to their environment. Agents may execute predefined basic actions and given goals represented by plans on stack (out of the agent's plan library) as well as to replan their actions. gACE's basic modules are the *agent control engine (ACE), the agent control unit (ACU), a knowledge base* and a *communication unit* (cf. Figure 11).

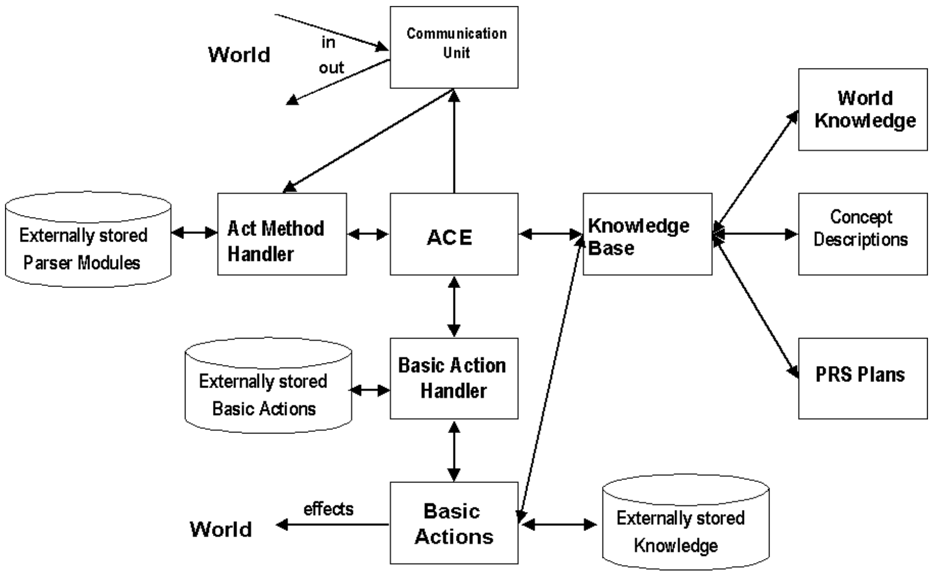

Figure 11: Architecture of the gACE framework

The *Agent Control Engine* (*ACE)* connects all parts of an agent and controls the ACUs spawned at runtime for controlling the processing of activated plans. All incoming and outgoing messages are handled by the *communication unit* using the UDP protocol for data transmission and the KQML language as message format. Incoming messages are parsed by the *Act Method Handler*, the knowledge base is updated, and plans as sequences of selected basic actions are instantiated and activated (*Basic Actions Handler*).

4.4 Warehouse Agents

The holonic warehouse agent is an example of a special supply web agent (cf. Figure 12) and has been implemented with *gACE* .

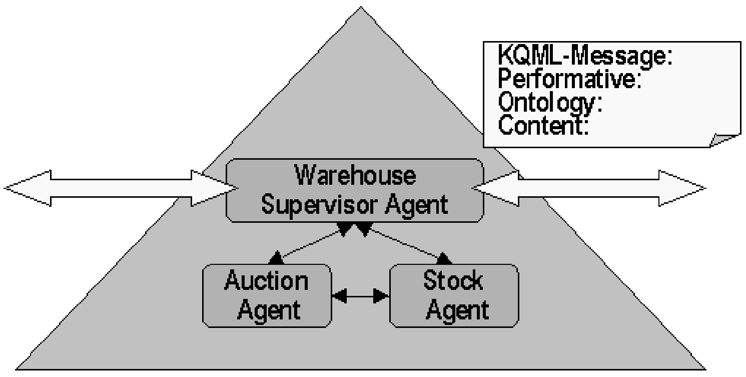

Figure 12: The Warehouse Agent Holon

This holon consists of different types of interacting agents: The *supervisor agent* represents the interface to the supply web environment and coordinates the operation of its subagents; the *auction agent* has to handle all activities related to auctions; the *stock agent* manages the flow of goods as well as policy based reservations.

4.5 Supply Web Coordination

4.5.1 Coordination Policies

The warehouse agent behaves according to given rules (policies) of which the main ones are the following.

- *Supervisor Policy.* The supervisor agent decides on which good has to be traded at what time and for what price. For example, the *keep-the-level policy* describes the re-ordering of goods in case of the entity falls short of storage capacities.

- *Bidding Policy.* The auction agent determines the value of the bids. For example, the *always-one-higher policy* prescribes a continous increase in bid value by one until a certain limit is reached.

- *Decommitment Policy.* Any commitment or reservation of resources may be cancelled with penalty. Various policies could manage the decision if and when to decommit.

- *Reservation Policy.* An agent has to decide on if some proposed reservation of goods for another agent is acceptable, or not.

Other behavior-oriented policies include policies for supply paths, and calculation of delivery dates and delays. Latter kind of policies can be categorized into *just-in-time policies, safety lead-time policies, bid refusal policies*, and *promise date negotiation policies.* A comprehensive overview of these types of policies can be found in (Kjenstad, 1998).

4.5.2 The Supply Web Coordination Server

The coordination server has been structured into a 3-layered holonic agent architecture as follows (see Figure 13).

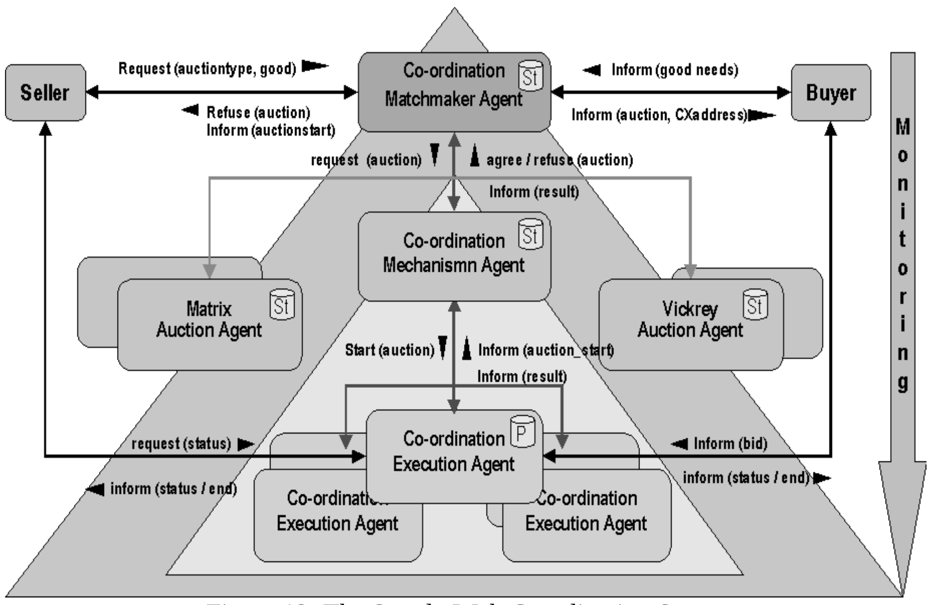

Figure 13: The Supply Web Coordination Server

The Coordination Matchmaker Agent (CMM Agent) serves as an interface to supply web agents which are requesting for coordination of their sales activities in terms of an auction. For each request the CMM agent selects the most appropriate coordination mechanism (CMech) agents for further processing of the request. In case there are enough resources available for initiating an auction of the desired type by the subordinated coordination execution (CX) agents the respective supply web agent(s) will be notified on the details of this special auction matching its request. If the request matches with some currently running auction(s) the CMM agent forwards the contact details of the corresponding CX agent(s) it receives by the respective CMech agent(s) to the supply web agent.

Latter can then directly contact the selected CX agent(s) for bidding or status monitoring depending on its role as buyer or seller, respectively. The CMM agent is in charge of efficiently coordinating and monitoring operative processes of running auctions to ensure appropriate load balancing at the server. For reasons of statistics the result of each auction is also propagated internally in a bottom-up fashion to each of the responsible agents, i.e., CX, CMech, and CMM agent.

4.5.3 Market-based Supply Web Coordination Mechanisms

We have endowed our supply web coordination server with several market-based allocation mechanisms for the coordination of supply web activities such as the simulated trading algorithm (Bachem et al., 1992) and matrix auction (Gomber et al., 1998).

The *simulated trading (ST) algorithm* is a randomized algorithm that realizes a market mechanism where contractors attempt to optimize a task allocation by successively selling and buying tasks in several rounds. *Matrix auctions (MA)* are truth-revealing and - in contrast to ST and VA - applicable for the *simultaneous* assignment of multiple items or tasks to bidders. In a *matrix-k-auction (MA-k)*, *k* items are auctioned-off simultaneously to some bidders. From their transmitted bids an auctioneer identifies the optimal allocation of all *k* items. Prices are set according to the *Vickrey auction (VA)* (Vickrey, 1961), i.e. bidders receiving items pay the second-highest bid made for these items.

Empirical results on the suitability of these coordination mechanisms for the allocation of transportation tasks in a network of shipping companies are reported in (Gerber et al., 1999); the mechanims ST, MA-2, and MA-3 showed promising performance for supply web coordination tasks.

4.6 Future Work on MAS-R/3

We have implemented the coordination infrastructure and supply web agents in Java. Future efforts will mainly be devoted to the integration of additional coordination mechanisms into the infrastructure. This aims to decentralize the configuration and coordination of distributed business processes as well as the (re-)allocation of resources and tasks within complex supply webs.

5 Agent-Based Support of Software Repositories[3]

5.1 Motivation

The availability of knowledge and information about internal business processes and software systems is strategically crucial and relevant for companies. One means to make such knowledge available in a smooth and easily usable way are *repositories*. They are computer-assisted information systems about the information processing activities of an enterprise. Other commonly used names are *data dictionary, development database, information resource system, catalogue,* or *meta information system* (Ortner 99).

The most important features of such a repository are the accuracy and recentness, as well as correctness of the contents made available. In order to keep the information about software systems in use, up to date and in concurrence with the upgrading to new systems, releases and thus in concurrence with software configuration management procedures, we have implemented *autonomous agents as pro-active information sources.*

In the corporate setting of our project partner, the Dresdner Bank AG, software development follows strict rules. One of these includes the use of *archiving servers* for change management and data preservation. Software agents which are monitoring the activities on these servers

[3] This research has been sponsored by the Dresdner Bank AG, Germany.

may provide a highly suitable means to communicate the most recent changes, releases or patches to the corporate development repository in an autonomous and automatic manner.

Currently, Dresdner Bank uses the commercial configuration management tool, *Continuus CM*™ (Continuus 98)[4] for company-wide software configuration management and archiving server. Both, software development center and repository are located within the corporate intranet. Information agents are extracting the required information. We have implemented a prototype, which has mainly been used to identify the basic goals, abstractions, and plans involved for agents in this problem domain. Furthermore, it provided insights into the applicability, robustness and reliability of agents in this context. Prototypical implementation has been done using the Java-based PRS-implementation JamAgent (Huber 98a, Huber 98b); the prototype served as a basis for the final product which will be described in the following.

The modular development and implementation of used agents enables easy future extensions regarding, for example, the interaction with other types of archives or the extraction of different pieces of information. In the current state of the implementation information gathered by the agents is about *released* software systems. Agents communicate via secure channels using XML-encoded messages. The features of tracing and monitoring agents' activities rely on special action log files, which include among others statements in XML on the interaction of agents with some CCM system.

5.2 Domain Characteristics and System Requirements

Software development inside the Dresdner Bank AG is performed in many development centers all over the world. Each of these centers produces and archives software according to well-defined procedures

[4] *Continuus CM*™ is abbreviated by CCM throughout this section. It has been a corporate decision by Dresdner Bank to use CCM for change management, thus most of their software development within c/s-environments is performed with CCM.

and processes. The bank uses a vast number of legacy software, which has been developed for host platforms, mainly MVS-based mainframe systems, in client/server (c/s) computing environments. During the last few years, at Dresdner Bank the problem of change management has been tackled in such c/s-environments by the commercial CCM suite; archiving CCM servers keep track of changes and releases of software.

The main goal of software agents developed in the REPTIL project is to find the most recent releases in the c/s-environment by means of appropriately interacting with given CCM servers. A general overview of the tasks and the flow of information for the agents in their working environment is shown in Figure 14.

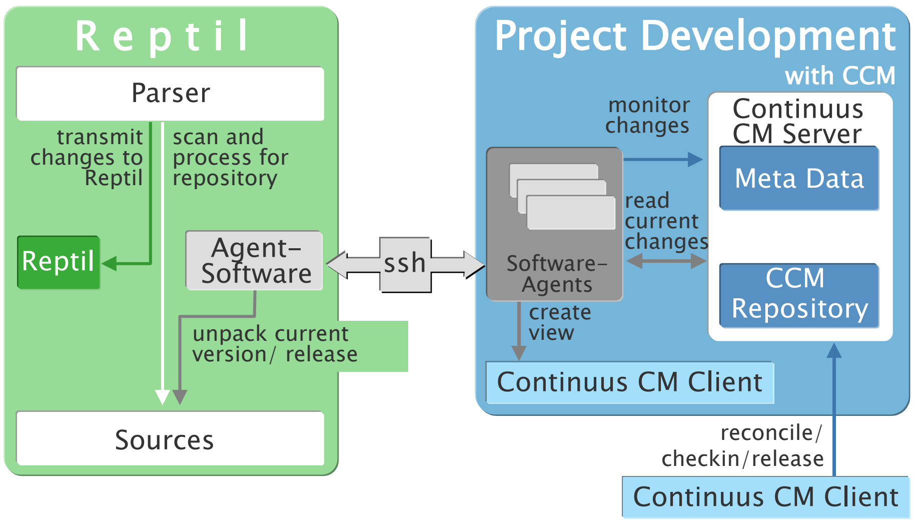

Figure 14: Flow of information between development center (right) and repository REPTIL (left)

This figure also depicts the initial requirements for the agents in general. The project development environment hosts a monitoring agent which interacts with the archiving CCM server, whereas in the so-called REPTIL environment, a software agent receives new incoming information. For obvious reasons, communication and transfer of data including corporate data and values via the network has to be secured.

5.2.1 The Repository REPTIL

The information system, that constitutes the in-house repository solution for the Dresdner Bank AG is called REPTIL and will be commercially available in near-future. REPTIL

- processes, analyses and archives data about software projects and software systems currently in use, and provides respective (meta-) information to the users.

- provides selected operations on archived data for knowledge transfer, information update and refinement, thereby providing a *single point of information* within the software development of the bank.

The *meta information* which REPTIL offers to its users, comprises in particular contextual knowledge about the considered development environment, such as embedded software tools, standards for procedures, lists of users and software components along with an analysis of their data and control flow, as well as organizational entities which are using the software and thus affected by changes to it. In addition, REPTIL provides users with a unified interface in a standard browser (e.g. Netscape or MS Internet Explorer) and is currently applied to the intranet of the Dresdner Bank.

5.2.2 Archive-Based Agent in the Development Environment

We call the special kind of agent in a project development center *archive-based*, since each of the c/s-based CCM server represents one specific kind of *reference location* to be monitored by one or multiple agents. In a more general view of the problem of transmitting the most recent release information as input data to the REPTIL system, such a reference location can be any kind of server or change management and tracking system, like CVS (Fogl 99), Rational ClearCase™ (Rational), or other sorts of interactive archives[5]. The corporate choice of the bank was the CCM, therefore our reference implementation currently interacts with it.

[5] This does not necessarily have to be a configuration management tool.

The agent software located in the development environment has to monitor the contents of the CCM-repository in certain time intervals. It looks for specific tags on the CCM-repository's contents. The CCM-repository is organized in terms of (software) projects. Such a project contains all relevant information concerning the continuos evolution of a software system under development. CCM provides a broad amount of operations to keep track of changes and allows for concurrent development including features such as conflict detection. We omit to go into details but refer interested readers to the extensive system documentation (CCM 98). We limit our description of CCM features to the requirements for the detection of new *project releases*, which is the main task of the agent software.

A new release in the CCM project life cycle is characterized by a *status label*, which has the value *released* (each object under CCM control has a type-specific life cycle; the status label indicates its phase in the life cycle, whereas for projects, the values of this label start with working, and may include *test, integrate* and finally *released* for making the software available). The agent's task is to find those projects, that have been released recently and are not yet updated or known in the REPTIL repository. On detection of a new release, the agent creates its own CCM-specific project view and extracts the respective software and meta data available. The agent's general activity and interaction with CCM comprises the following steps:

- Monitoring of changes at the CCM servers;
- Upon detection of a new, unknown release, the creation of a CCM-specific project view, and
- the collection of the relevant sources code, as well as
- the compilation of available meta information, and finally
- assembling and sending the data packages through the network to the REPTIL system.

These interactions have to be transparent to the software developers using the respective CCM server, to ensure that the agent software does not change substantial system properties.

5.2.3 The Agent in the REPTIL Environment

While the agent software in the CCM environment actively seeks new project information and upon succeeding in this task, gets and sends it, the agent in the REPTIL environment has more simple tasks to achieve. It waits for new incoming data and upon receipt, unpacks the data and notifies the REPTIL system about it. REPTIL processes the data supplied by the agents, which make up the agent-based information source.

5.3 The Agent-Based Approach

5.3.1 Agents at the REPTIL Site

Since the activity on this side of the agent-based information source comprises only the reception and unpacking of data, the agent-based solution is straight forward.

5.3.2 Agents at the Archiving Server

At the side of archiving CCM servers, we developed an agent society, which consists of a collection of specialized agents (cf. Figure 15). Information extraction from a CCM server is performed by an appropriate task-specific CCM agent. Since there may exist multiple CCM servers to be monitored for relevant data we distinguish between the processes of monitoring the archives, retrieving data from those archives and transmitting data to REPTIL.

- *ADS-Agent:* The *Agent Directory Service Agent* provides several structural services, such as initialization and control of action execution, to the agent society. It observes the activities at the CCM servers. Upon detection of new releases, it invokes a task specific CCM-Agent for information extraction.

- *CCM-Agent:* The *Continuus CM Agent* is activated by the ADS-Agent upon detection of a new project release. It interacts with the respective CCM server for retrieving the source code of and meta-information on a given software development project, compresses the data for transmission to the REPTIL system, and then notifies the RDM-Agent.

- *RDM-Agent:* The *Resource Data Manger Agent* handles the interaction with the agents at the REPTIL site through secure data channels. When the CCM-Agent notifies the RDM-Agent, that data packages are ready for transmission, it informs the agent at the REPTIL site and communicates the data packages through the network.

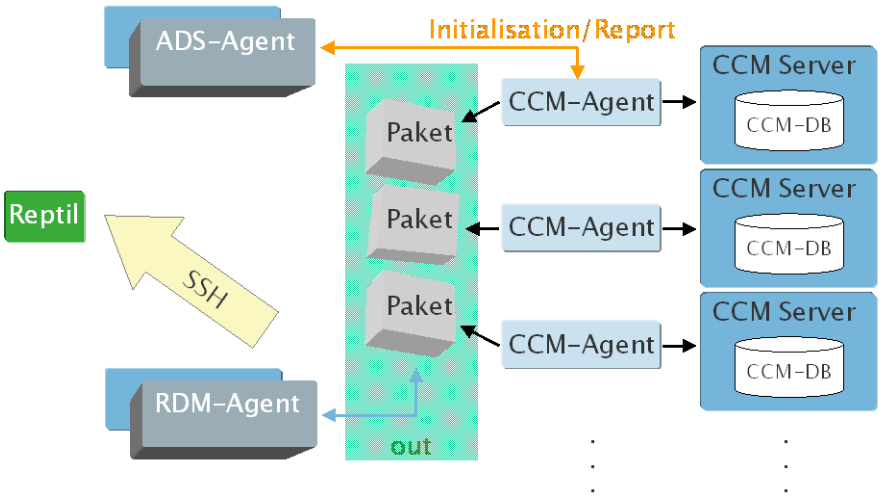

Figure 15: Task-oriented agent society at the archiving server

Within the REPTIL project we developed a society of task-specific agents, which can easily be adapted to the needs of other archiving servers in the considered software development environment.

5.4 Implementation Details

The agents for REPTIL have been implemented in Java on Sun Sparc machines under Solaris 2.6 using JDK 1.2 and the current JamAgent distribution. We use the Continuus CM™ suite CCM 4.5, script-based interactions are realized in Perl 5. The agents exchange data and message by TCP/IP through secure ssh-channels.

5.5 Future Work on REPTIL

The agent software is currently integrated into the REPTIL system within productive real-life environment of the bank. Potential future enhancements are as follows:

- Interaction with other archiving servers or reference locations, e.g. CVS (Fogl 99) or RationalClearCase™ (Rational).

- Currently the agents and REPTIL provide just information on *released* software. However, CCM also allows for easy switching to different status views (integration test, work, etc.) of projects. This additional feature enables the repository to present dynamic information about individual software development progress.

Acknowledgments

We’d like to thank Jürgen Feilhauer and Alexander Natter for proof reading and comments on draft versions of section 5 of this chapter. In addition, we are indebted to Nils Kammenhuber, Patrick Frotscher, Carsten Tichy, and Christoph Stahl for their invaluable efforts in implementing the CASA ITN prototype.

References

Bachem, A. and W. Hochstättler, and M. Malich. Dezember (1992) "Simulated Trading: A New Approach For Solving Vehicle Routing Problems", *Technical Report* 92.125, Mathematisches Institut der Universität zu Köln.

Bürckert, H.-J., Fischer, K., and Vierke, G. (1998), ”Transportation Scheduling with Holonic MAS — The TeleTruck Approach,” *Proceedings of the Third International Conference on Practical Applications of Intelligent Agents and Multiagents* (PAAM´98).

Bürckert, H.-J., Fischer, K., and Vierke, G. (2000), "Holonic Transport Scheduling With TELETRUCK," *Applied Artificial Intelligence*, vol. 14, pp. 697-725.

Continuus Software Corporation, *Introduction to Continuus/CM™*. 1998, see also http://www.continuus.com

(FIPA) www.fipa.org

(FIPA OS) www.nortelnetworks.com/fipa-os, www.emorphia.com

Fogl, K. (1999), *Open Source Development with CVS.* CoriolisOpen(tm) Press.

(gACE) http://www.dfki.de/schuhmac/AgentTools

Gerber, A., Klusch, M., Ruß, C., and Zinnikus, I. (2001), "Holonic Agents for the Coordination of Supply Webs," *Proceedings of the International Conference on Autonomous Agents (Agents'2001), ACM Press, New York, USA*

Gerber, C., Siekmann, J., and Vierke, G. (1999), "Flexible Autonomy in Holonic Agent Systems," *Proceedings of the 1999 AAAI Spring Symposium on Agents with Adjustable Autonomy*.

Gerber, C., Siekmann, J., and Vierke, G. (1999), "Holonic Multi-Agent Systems," *DFKI Research Report RR-99-03, ISSN 0946-008X*.

Huber, M. J. (1999), *JAM Agents in a Nutshell, Version 0.61+0.79i*, available at http://members.home.net/marcush/IRS

Huber, M. J. (1999), "JAM: A BDI-theoretic Mobile Agent Architecture," *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, Seattle, WA, pp. 236-243.

Ingrand, F.F., Chatila, R., Alami, Robert, F. (1996), "PRS: A High Level Supervision and Control Language for Autonomous Mobile Robots", *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, USA.

Gomber, P., Schmidt, C., and Weinhardt, C. (1998), “Efficiency incentives and computational tractability in the coordination of multi-agent systems”, *Proceedings of the Workshop Kooperationsnetze und Elektronische Koordination.*

Kjenstad, D. (1998), “Coordinated Supply Chain Scheduling,” *Trondheim, Norway, NTNU-rapport.*

Klusch, M. (1999), *Intelligent Information Agents*, Springer.

Klusch, M. (2001), “Information Agent Technology for the Internet: A Survey,“ *Data and Knowledge Engineering*, vol. 36, pp. 337-372

Laseter, T.M. (1998), “Balanced Sourcing: Cooperation and Competition in Supplier Relationships,” *Jossey-Bass, ISBN: 0787944432.*

Ortner, E. (1999), “Repository Systems,” *Informatik Spektrum* 22 (4), pp. 235-251 and 22 (5), pp. 351-363, In German.

Rational.*ClearCase*™ ww.rational.com/products/clearcase/index.jtmpl

Vickrey, W. (1961), “Counterspeculation, Auctions, and Competitive Sealed Tenders,” *Journal of Finance* 16, pp 8-37.

Vierke, G., (2000), *TeleTruck - A Holonic Multi-Agent System for Telematics*, PhD Thesis, Saarland University, Saarbrücken, Germany.