Software Agents for Electronic Business: Opportunities and Challenges

Jörg P. Müller, Bernhard Bauer, and Michael Berger

Siemens AG, Corporate Technology, CT IC 6, D-81730 Munich, Germany
Joerg.mueller{bernhard.bauer,michael.berger}@mchp.siemens.de

Abstract. Electronic business (eBusiness) is one of the main global drivers of information technology. In this paper, we investigate how eBusiness can benefit from methods, components and solutions based on software agent technology. We give an overview of the current electronic business mainstreams and outline a number of important the challenges eBusiness has to tackle. We then describe how agent technology can be applied to in eBusiness to resolve some of these challenges. We discuss related work and the need for integration of any agent technologies into existing eBusiness landscapes and infrastructures. We describe the opportunities for agent technology in this area. In addition risks and areas of future work are discussed.

1 Introduction

Since the early nineties the Internet usage has been growing exponentially. Stationary and, increasingly, mobile information services are reaching users world-wide at anytime. Based on these facilities for communication and data exchange, many kinds of businesses are moving to the electronic world of the web to connect customers, suppliers and partners to respect the globalization of markets. For *electronic business (eBusiness)*, the Internet is much more than just a mine of information. It is capable of helping to attract a whole new customer base whilst ensuring that existing customers remain. It can enable an increase in efficiency and better cost control by allowing customers, as well as nominated suppliers and distributors, to work together and share applications and knowledge in a secure environment, based on a uniform platform.

eBusiness is the overall term for the complex integration / transformation of existing infrastructures, business processes, enterprise applications, and organizational structures into a high-performance business model using information technology based on electronic media such as the Internet, other computer networks, and wireless transmission to facilitate the business. eBusiness is changing the traditional business landscape and the way business is done in general. The future success of a company depends largely on their ability to transform themselves into eBusiness companies. This means the implementation of new business strategies and the introduction of innovative marketing techniques, as well as the better management of information and the reduction of time-wasting paper-based processes. The aim of this e-transformation is increased speed of service, improved customer satisfaction, integrated solutions, convergence of sales and service chains, leveraging legacy systems, connecting the entire corporate, contract manufacturing, information

security, and protection of intellectual property. A special trend is mobile business (mBusiness). That's the overall term for eBusiness based on mobile electronic media and wireless networks. Beside and on top of the traditional eBusiness applications mBusiness can provide additional value through e.g. location based services.

An important branch of eBusiness are electronic marketplaces (e-marketplaces). An e-marketplace is a central place on the Internet functioning as a meeting place for buyers and sellers of specific products. Buyers and sellers can form interest groups to bundle their capacity. An electronic marketplace can be open or closed. It has intermediary functions like auctioning and negotiation. Additionally, electronic marketplaces usually have a portal functionality as well providing e.g. catalogues, availability information, and supply order. Parts of a marketplace are seller and buyer sites (e.g. e-procurement at the buyer site).

eBusiness is the most growing market in the world. The world-wide business-to-business (B2B) Internet commerce market is on pace to total $8.5 trillion in 2005, according to Gartner Group in March 2001 [25]. In 2000, the value of world-wide B2B Internet commerce sales transactions surpassed $433 billion, a 189 percent increase over 1999 sales transactions. World-wide B2B Internet commerce is projected to reach $919 billion in 2001, followed by $1.9 trillion in 2002. Forrester Research expected in February 2000, that 75 percent of B2B will migrate to e-marketplaces over the next five years and that e-marketplaces will capture 53% of all online business trade by 2004 [22]. The same company predicted in January 2001 that businesses world-wide would increase their spending on B2B e-marketplaces from $2.6 billion in 2000 to $137.2 billion by 2005. Despite the disillusionment brought by the dot-com crisis, this will still be a huge market.

Today, many different software components and systems offer a wide range of standard business services and partially isolated eBusiness solutions. No single vendor can offer software, which covers all eBusiness services. Thus, a distributed global architecture based on open architectural standards, such as Java, HTTP, EJB, and XML, is needed. This architecture will offer component based functionality, wide range of scalability, interoperability between applications, and secure access. Furthermore, a more personalized and intelligent support of the eBusiness processes is becoming more and more important.

In this paper, we investigate how *software agent technologies* can help enterprises tackle the challenges of eBusiness. Software agents are software components characterized by *autonomy* (to act on their own), *re-activity* (to process external events), *pro-activity* (to reach goals), *co-operation* (to efficiently and effectively solve tasks), *adaptation* (to learn by experience) and *mobility* (migration to new places). Agents can be individualized to act on behalf of users, teams, or organizations. They coordinate their activities and collaborate with humans, other agents and external components in order to achieve their goals (see e.g. [31], [39], [43]).

Agent technology has the potential to play a key role in combining the existing heterogeneous eBusiness solutions, adding advanced functionality and automating standard processes. For that, agents provide task delegation, enriched higher level communication, enable more intelligent service provision and process management, provide individualization (also personalized visualizations and avatars), provide service integration to value added services to deal with the enlarging amount of information and functions, and allow self-organization of processes and systems.

To realize this big potential, agents need an infrastructure that allows them to communicate, to discover service-peers, to negotiate and to co-operate in open

environments [48]. This requires standards to ensure the interoperability between agents of different vendors and domains (see e.g. FIPA [19]). Most importantly, agents will need to build on and interface with a variety of existing and upcoming developments and standards, like DAML+OIL [7] and ebXML [11]. Establishing agents as enabling technology will touch on supporting a wide range of devices, integrating telecommunication and Internet, and interfacing with Enterprise Resource Planning (ERP) and supply chain management systems and eBusiness platforms.

Another important precondition for the acceptance of agent technology in eBusiness is the availability of generic added-value services, such as team coordination, process scheduling, recommendation engines, mobility support and location-aware services. Such services are designed for multiple reuses and cover areas requiring higher levels of flexibility, individualization, or intelligence. In addition, successfully bringing agent technology to market, techniques that reduce the perceived risk inherent in any new technology are required. Such a technique is to present the new technology as an incremental extension of known and trusted methods, and to provide explicit engineering tools to support proven methods of technology deployment.

In this paper we give an overview over agent-based approaches to eBusiness and show the main advantages in building agent based eBusiness systems. Section 2 gives a short introduction on eBusiness providing several views: a process view, an architectural view, and a functional view. Based on these views, we will investigate current limitations of eBusiness, and look at fields where we believe agent technology can play an important role over the next few years. Section 3 provides an introduction to agent technology and gives a detailed view on agent enabled eBusiness architectures, in particular having a focus on the benefits of agent technology for eBusiness. Section 4 illustrates three case studies in the domains of Human Resource Matching, Distributed Team Management and Travel Support. Section 5 describes agent standards and outlines relationships between agent technologies and related and integration technologies. Section 6 concludes and outlines areas for future work.

2 Areas and Challenges of eBusiness

In this section, we will outline several views on eBusiness: a process view, an architectural view, and a functional view. Based on these views, we will investigate current limitations of eBusiness, and have look at fields where we believe agent technology can play an important role over the next few years.

2.1 Process View

Electronic business, as far as regarded from a corporate perspective, is neither driven by specific functions nor by technologies. It is mainly driven by processes: the objective of electronic business is to support and optimize business processes. Hence, any discussion of eBusiness will be incomplete without taking a process-oriented stance.

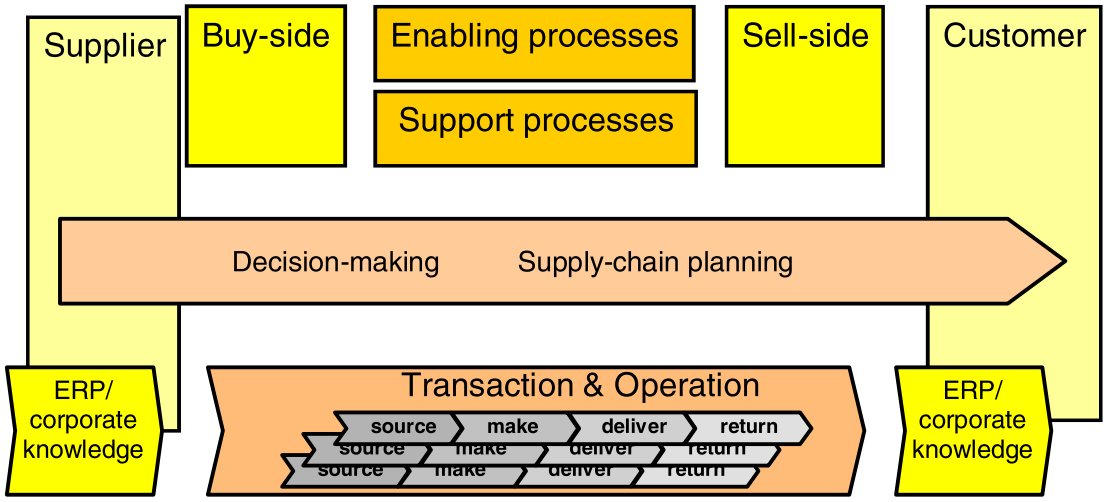

Fig. 1. eBusiness Process Architecture

Fig. 1 outlines the process view of eBusiness. It shows the basic application domains for eBusiness, from the perspective of an enterprise. It depicts the value chain, from suppliers (buy-side, procurement) to customers (sell-side, customer relationship management). The transaction layer that describes the value chain forms the basis of a companies' eBusiness process. It feeds on information from Enterprise Resource Planning (ERP) systems and corporate knowledge bases. On top of this process, there is a decision-making and supply-chain planning flow. Furthermore, a number of *enabling processes* such as content/knowledge management, e-payment, registration and single login management as well as *support processes* including travel management and Human Resources (eHR) round up the corporate eBusiness picture.

2.2 Architecture View

A second important perspective of an eBusiness system is in terms of its underlying software architecture. The software architecture and the technologies used in its different tiers have a great influence on the flexibility and scalability of such a system.

eBusiness systems are typically based on multi-tier architectures. A typical example is illustrated in Fig. 2. In the following, we briefly describe the individual tiers.

2.2.1 Client Tier

The client tier contains the part of the system that runs on client-side hardware. It typically provides a Web / WAP browser interface but may also incorporate a fat-client approach to facilitate rich functionality and personalization on the client side. There typically is a trade-off between rich client-side functionality on the one hand, favoring a fat-client approach, and ease of change, maintenance and administration, favoring a thin-client solution.

Especially in the context of mobile business, more powerful mobile devices and networks, and of new software paradigms such as peer-to-peer technology, fat client solutions are attractive. However, we perceive a strong trend to thin client solutions in

many corporate, intranet-based eBusiness environments, mainly for the above mentioned reasons.

2.2.2 Client Interaction Tier

The role of the client interaction tier is to process client requests and to transform results produced by the business logic tier (see below) into presentations that are tailored with respect to the requesting user and its context (preferences, location, device, etc.). At the time of writing, using XSL transformation engines and Java-based front-end technologies such as Java Server pages appears a state-of-the art solution. Having said this, we must state that up to our knowledge, it is still rather the exception than the rule that this technology is actually used in practice.

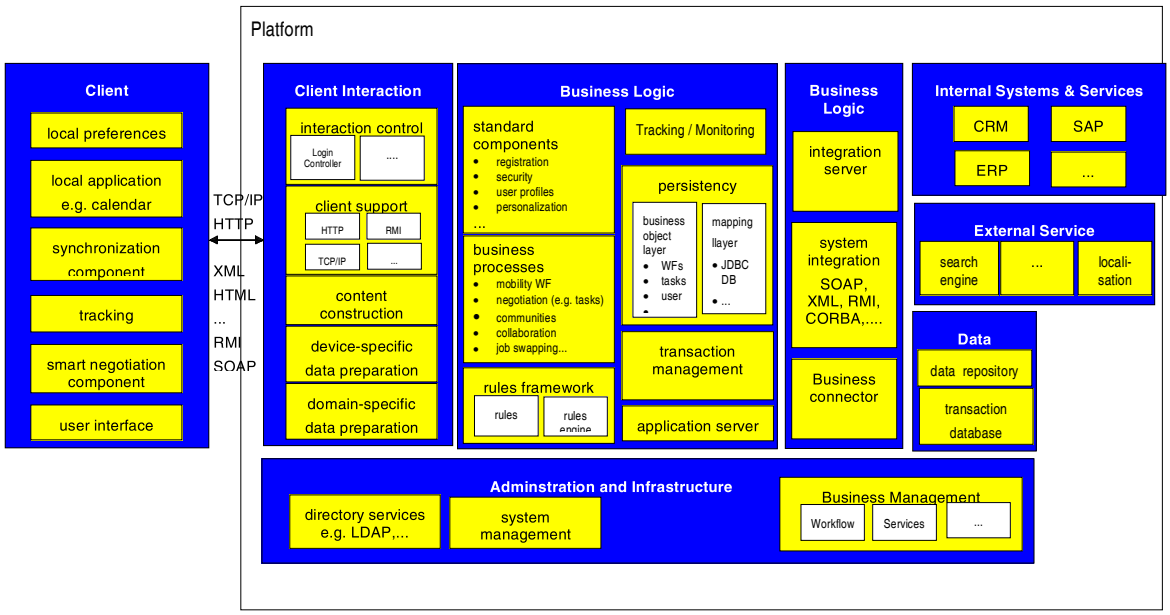

Fig. 2. Software Architecture of an eBusiness System

2.2.3 Business Logic Tier

This tier is the core of an eBusiness architecture, implementing the business objects (e.g., users, customers, orders) and the basic business processes and transactions (e.g., register, order, search). Scalability and flexibility are the most important requirements to be satisfied by the business logic tier. On the one hand, these objects and processes may be requested a large number of times by applications; on the other hand, as the processes in a company change, their implementation needs to be updated. Also, the attempt to standardize interfaces and processes in eBusiness and the need to reuse existing building blocks throughout multiple applications brings about the need and the possibility to reuse pieces of business logic. The dominant solutions today are based on Enterprise Java Beans Application Servers. These Application Servers offer good performance and availability and provide support for transactions and persistence. They often play the role of wrappers to ERP systems or other external or legacy applications.

2.2.4 Integration Tier

More advanced eBusiness architectures provide an extra layer of abstraction for interaction with / integration of legacy systems and external business applications. The integration tier offers uniform interfaces for enterprise application integration to the business tier (often based on XML/SOAP); furthermore, it provides single login support (e.g., based on LDAP) to deal with the authentication regimes of multiple external systems. For asynchronous communication between applications, the integration tier will usually support messaging. Towards the backend, adapters for different legacy systems are provided (e.g., SAP). Today, there are a growing number of products available to implement the integration (e.g., WebMethods, IBM MQSeries).

2.2.5 Systems and Services Tier

This tier provides the physical connectivity between an eBusiness system and the information it accesses, in the form of connectors to data bases and legacy systems, but in addition to external information systems.

State of the art eBusiness architectures define generic invocation interfaces for flexible communication between the tiers, e.g., based on the command pattern.

2.3 Functional View

One way to look at eBusiness systems is by the functions they perform, from the perspective of the parties involved. Figure 3 illustrates these functions. They are briefly discussed in the following:

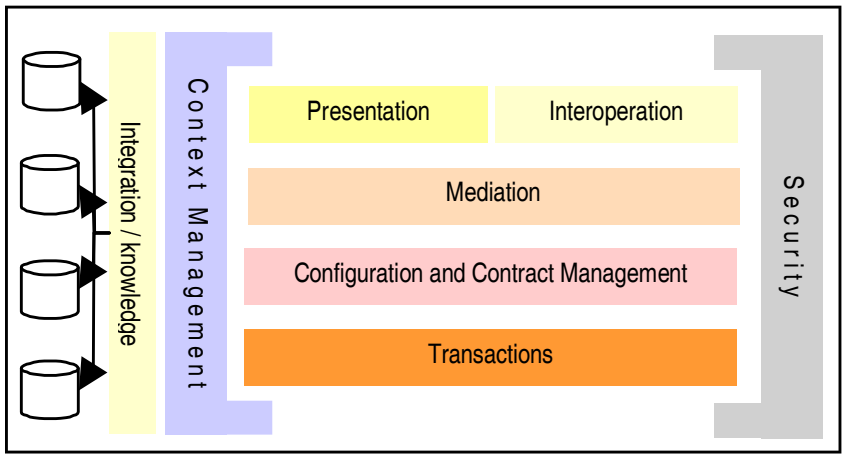

Fig. 3. Functional eBusiness Architecture

2.3.1 Communication: Presentation and Interoperation

The first function of eBusiness is to make communication between enterprises, customers, partners and suppliers across the value chain more efficient. This involves **presentation** (e.g., attractive web shops, efficient marketing strategies, sales cam-

paigns, communities) but also **interoperation**, i.e., the ability of business applications to communicate with each other across the borders of departments and enterprises. Providing this function offers visibility and thus the basis for mutual attraction and interaction.

2.3.2 Mediation

By mediation we mean the solicitation of demand and supply in eBusiness systems and marketplaces. Mediation allows customers to find and contact potential products, partners or merchants[1].

2.3.3 Configuration and Contract Management

This function subsumes the phase of negotiation between buyer(s) and seller(s) with the goal to achieve a business agreement, i.e., a signed contract on some matter. This typically involves the configuration of the product and the negotiation of the contract for the selected product configuration. In practice, however, negotiation may involve modification of the product configuration; hence, the two phases can be inter-linked.

2.3.4 Transactions

Once an agreement has been reached, the actual business transaction is carried out, connected with processes such as charging, payment, order fulfillment, and delivery. Typically, transactions require interactions with ERP systems or with external systems (such as payment servers).

These core functions need to be supported by a number of supporting functions.

2.3.5 Context Management

In order to automate any eBusiness activity, eBusiness systems need access to the context in which the activity is located. In particular, that means the access to and maintenance of user profiles. E.g., to provide personalized services, a Customer Relationship Management system will store information about a customer and derive information about the users' preferences from this history, which is then used for dedicated marketing campaigns, or to deal with complaints and requests by that customer quicker and more efficiently. Another example is the use of location information for mobiles personalized services.

2.3.6 Security

For technical, business and psychological reasons, the provision of a secure infrastructure is (e-)business critical. Security needs to cover all levels, ranging from security of information transfer to application-level security, and requires appropriate measures to be taken at the hardware, software, and process level.

[1] In [28] Guttman et al. split this function into Product Brokering and Merchant Brokering.

2.3.7 Integration and Knowledge Management

For companies, the core of eBusiness is information about their products, customers and processes (normally stored in ERP systems) and the implicit and explicit corporate knowledge available in numerous databases, document repositories, and in addition in the heads of the employees. The ability to access eBusiness relevant information from legacy systems and organize the knowledge within a company in an easily accessible form is crucial.

2.4 Challenges in eBusiness

Today, there are well-established product suites to support these eBusiness functions. E.g., *IBM Websphere* for sell-side, *i2* for Supply Chain Planning and Procurement, *CommerceOne* for procurement and trade exchanges, *Siebel* for Customer Relationship Management, and *SAP* for ERP systems (see also Sect. 5.3).

In this section, we will outline some of the remaining challenges and opportunities for research on eBusiness. We believe that in order for a technology to succeed in eBusiness it will be necessary to take an incremental approach, thus adding value on top of existing platforms and services.

2.4.1 Challenge: Individualization and Privacy

How can eBusiness systems provide individualized content and services while at the same time ensuring adequate (and user-defined) levels of privacy? How can user profiles be created, maintained, and adapted to cope with multiple threads of a user's activity, with disruptive events, and with longer term changes in the user's preference and activity structure? How can teams and organizations be profiled?

The importance of this question has been recognized. P3P is an attempt to provide a uniform representation of privacy preferences for an individual or company. However, it only covers privacy related preferences, and not the individualization of presentation, content, and services.

2.4.2 Challenge: Secure Delegation

How can humans delegate complex tasks to machines? How can they be sure the machine has understood the task correctly? How can they specify the permissions and limitations of authorization of machines during task execution? How can they monitor the progress of task execution? How can a flexible hand-over between humans and machines be implemented to guarantee human and machine abilities are used for their best? How can transactions done by machines be verified and liabilities be enforced? What are legal aspects to be observed?

While many researchers investigate secure protocols and efficient strategies e.g., for automating negotiation, it is our firm belief that the most difficult part of the problem is at the user interface layer in developing appropriate models for efficient delegation and transparent collaboration between humans and machines.

2.4.3 Challenge: Semantic Interoperability

How can machines better understand the content of communication with humans or other machines? How can eBusiness consumers find supply matching their demand? How can suppliers find potential customers? How can applications created by different companies interoperate more smoothly even if there is no explicit standard for the interaction? How can techniques for achieving interoperability at the level of semantics be used to add value to existing platforms and standards in eBusiness?

We believe that semantic interoperability is maybe the hardest of all challenges to be overcome, because it requires the understanding of semantics by machines which is still in its beginnings.

2.4.4 Challenge: Support for Flexible Organization Structures

How can eBusiness infrastructures keep up with ever changing structures of enterprises? When departments change their responsibility or structure, when a partner becomes a competitor or vice versa, how can communication and security policies be maintained smoothly? How can the participants in the value chain communicate and collaborate securely and efficiently over firewalls and enterprise boundaries? How do suppliers know which events in their internal supply chain are of relevance and need to be communicated to their (downstream) partners? How can an existing IT infrastructure migrate towards this functionality?

While various theoretical approaches exist, supporting flexible organization structures in practice is a huge challenge.

2.4.5 Challenge: Intelligent Collaboration and Coordination

How can distributed individuals, teams, departments, and enterprises collaborate and coordinate their activity? How can the information necessary for this be exchanged efficiently? How can tasks be formulated, associated with skills, responsibilities, and permissions, and be allocated? How can humans collaborate with applications acting on their behalf or on behalf of other humans?

In particular in a corporate context, huge efforts are made to set up infrastructures that allow humans to share information, resources, and applications, and to work towards a common goal. Still, the problem remains hard due to its dynamic nature and through increased mobility.

2.4.6 Challenge: Mobility Support

How can mobile users be ensured effective access to corporate processes and knowledge? How can they find and communicate with team members remotely? How can location and situation information be used to make services more useful and intelligent? How can information be routed, transported and presented in a way that accommodates available networks and devices? What infrastructure is necessary to leverage the potential of ubiquitous computing for enterprises and individuals?

In a society where mobility plays a more and more important role, providing answers to these questions is crucial.

2.4.7 Challenge: Pro-active, Adaptive Processes

How can complex, distributed processes be modeled as active, situated entities that can interoperate with other processes that can monitor their environments and change their behavior on demand? How can these processes deal with change? What needs to be done to implement (pro-)active processes based on existing process models, e.g., in distributed, flexible manufacturing and supply chain management.

While many approaches to process modeling exist, automating them still remains a challenge due to the required level of introspection (i.e., a process needs to be able to judge a situation with respect to its competence, e.g., to recognize unforeseen conditions) and due to incomplete sensor-actor loops (i.e., decisions relevant to the process are often made by entities outside the control (and visibility) of that process.

2.4.8 Challenge: Adaptive Decision-Making Assistance

As machines change their role from slaves to assistants, more complex domain models are required to enable them to provide assistance in complex decision situations such as contract negotiations. How can these models be defined? How can machines be effectively instructed for these decision-making assistance tasks? How can decisions (or suggestions for decisions) by machines be made transparent to humans?

This challenge is closely related to Challenges 2.4.2 (Secure Delegation) and 2.4.5 (Intelligent Collaboration and Coordination).

2.4.9 Challenge: Intelligent Selection and Evaluation of Products and Services

How can configuration and purchasing processes be automated? How can machines support users in finding products and services matching a demand profile? How can they evaluate different configuration and product offers? How can they interoperate with existing eBusiness platforms? How can transactions made by machines be tracked down to the users? How can fraud be prevented? How can the risk for users or enterprises be minimized that faulty software makes wrong suggestions or buys the wrong items at too high a price?

Finding answers to these questions will be a precondition to any automation of matchmaking, contracting, or shopping / procurement processes in eBusiness.

In the following section we shall investigate where and how agent technology can provide means for dealing with these challenges.

3 Agents in eBusiness

In this section we define our notion of agent technology and then elaborate opportunities for agent technology to tackle the eBusiness challenges defined in Sect. 2.

3.1 Agents Definitions and Characteristics

In [1], an agent is defined as

> *... a computer system, situated in some environment, that is capable of flexible, autonomous action in order to meet its design objectives.*

We would like to enhance this definition by one aspect. Put more simply:

> *an agent is an intelligent autonomous computer system that does something useful on behalf of a human or an organization.*

With software agents usually special properties are associated, see e.g., [44], [43] for an overview. Fig. 4 illustrates the most important properties and dimensions of agents. These properties are well reflected in the different dimensions of agents, namely mobility, intelligence, cooperation and distribution, showing moreover that agent technologies combines these functions with the existing base technologies in agent-oriented integrated development environments supporting agent platforms and agent-oriented software engineering to develop agent-based frameworks and applications.

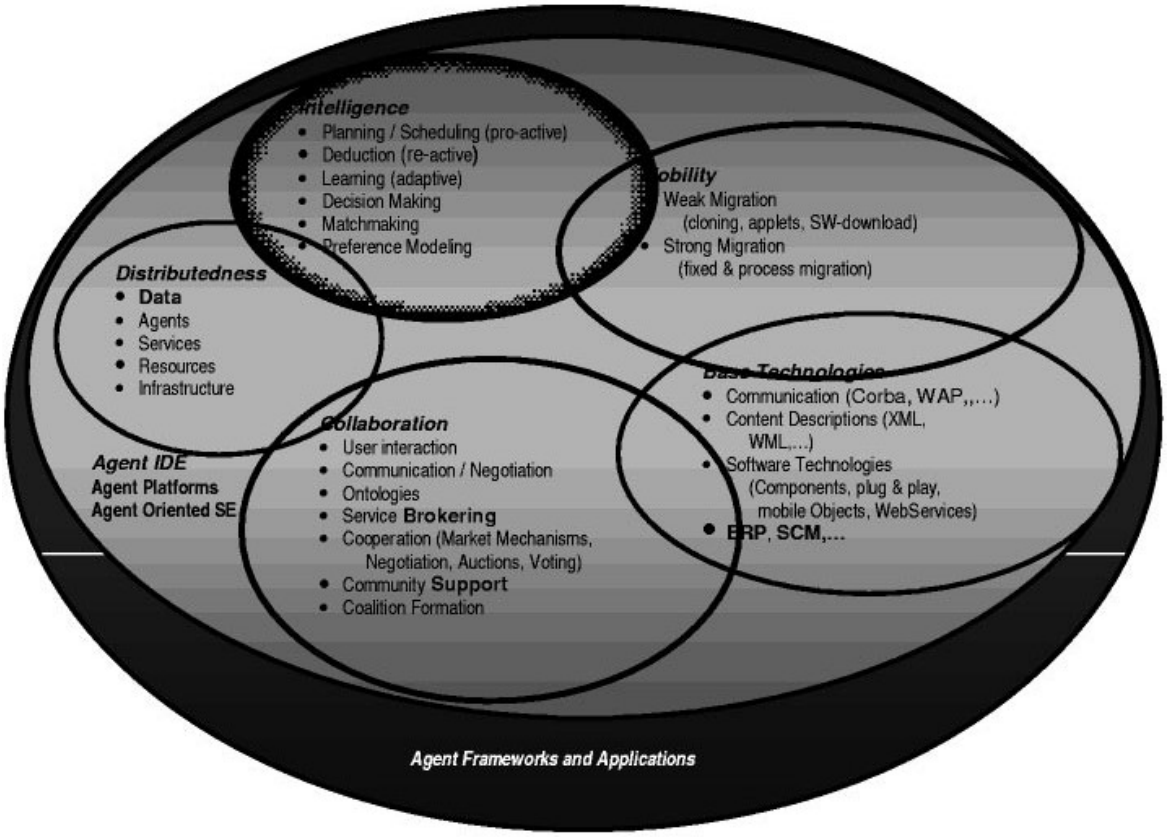

Fig. 4. Dimensions of agents

It becomes clear from Fig. 4 that agent technology is based on and needs to be developed in the context of existing basic technologies, such as software technologies. Content is described using standards like XML or RDF; existing communication mechanisms, like SOAP or WAP, are applied within agent technology. In the case of eBusiness we also view platforms as those described in Sect. 2.1 as base technology.

Software agents add a number of additional features to these technologies:

- Agents enable **collaboration** between humans and machines, thus providing a new metaphor for multimodal *user interaction.* Users instruct agents to act on their behalf instead of having to program them. Two core areas within the discipline of intelligent user interfaces include intelligent dialog systems, and believable virtual personalities. At the same time, agents introduce the notion of *machine-machine collaboration*. *negotiation and communication* on a semantics level allowing agents to coordinate their tasks and resources. In order to support the communication at a semantics level *ontologies* are applied to assign meaning to symbols and expressions within a given domain language. Based on negotiation, communication and ontologies *cooperation* between different agents can take place. *Coalition formation* is useful in eCommerce where virtual purchasing cooperative societies can be built to obtain quantity discount. *Organizational models* can be applied for the mapping of real organizations within a company to software organizations based on agents.
- The **intelligence** of an agent is manifested by its ability to adapt its behavior to changing environmental conditions, and to employ *planning and decision-making*, e.g., for process scheduling and resource planning based on negotiation techniques. In particular planning can be performed by a single agent or in cooperation with several agents. *Learning* can be applied in several dimensions, e.g. learning of user preferences, learning of negotiation strategies of other agents up to adapting processes in the context of eBusiness. *Matchmaking* is the process of mediating demand and supply based on profile information. Matchmaking (see Sect. 3.4.9) plays a crucial role, e.g. in agent based electronic marketplaces: the task to be solved is to find the most appropriate agents, products, or services for a task, negotiation, or market transaction.
- Agent technology supports **mobility** at different levels: mobility of users, devices, and code. In the latter case we speak of *mobile agents*. They are either characterized by *weak mobility* or *strong mobility*. Examples of weak mobility are cloning where an agent is deployed on another machine with the initial state. Applets can be seen as a specific kind of cloning. The download of a new version of software on an agent platform is another kind of weak migration. Strong migration is characterized by the complete movement of the execution state of an agent to another machine; this can be obtained either through real process migration or through specific migration points where an agent is allowed to migrate.
- Multi-agent systems are **distributed** systems; therefore distribution is an inherent feature of agent-based systems. In this context both agents and resources can be distributed. The agent infrastructure enables transparent access to distributed resources and transparent interaction among distributed agents.

These pillars of agent technology result in individualized added value services, supporting a possible nomadic user with intelligent assistance, based on search, integration and presentation of distributed information and knowledge management, advanced process control support, and eBusiness and enterprise applications. To conclude, agent technology provides

- the integration of distributed heterogeneous systems
- the mobility of persons and software

- the coordination of heterogeneous components and services
- the support of users for routine tasks
- the personalization of tasks and the exchange of personalized information and
- the easy integration of additional components and frameworks

Based on these potential benefits, we will now investigate how agent technology can help us tackle the eBusiness challenges identified in Sect. 2.4.

3.2 Agents in eBusiness: Areas of Application

With the broad acceptance of the Internet technology, future business processes will increasingly access electronic market places. The fact that these business processes will involve larger number of different parties from different organizations makes multi-agent systems a very promising approach for the management of these processes. Agents must be able to make autonomous decisions, e.g., in electronic negotiations or auctions. The intelligence of the agents will be essentially determined by the degree of their adaptability, covering the intelligent evaluation of the data, the adaptive reaction to changes of the environment, matchmaking, distributed planning, negotiation protocols and strategies (see Sect. 3.1).

As we have seen in Sect. 2, the today's landscape of eBusiness platforms and architectures is very heterogeneous. eBusiness architectures need to support the integration of the existing products, solutions, processes with new functionality, such as those supported by agent technology to enable end-to-end eBusiness. Within the different views of eBusiness depicted in Sect. 2, we see the following focus areas of application for agents:

- **Sell-Side / Customer Relationship Management:** the electronic networking of business relations with the customers of a company can be essentially supported by agent technology. When applied wisely, avatar technology supports appealing personal assistance functionality based on virtual characters. Negotiations with customers can be supported by assistant agents – participating e.g. in an auction – to help a user through the negotiation process. In this context learning techniques can be applied to optimize the negotiation strategy of the user agent. Furthermore mobile agents can migrate to electronic marketplace to minimize communication costs and time delays and to ensure maximum flexibility and expressiveness of negotiation strategies. Security issues are fundamental in this context; therefore we believe that marketplace agents will only succeed in the market within a timeframe of three to five years. Another topic in sell-side eCommerce is the search and evaluation of existing shops and products with respect to their qualities. Here agents can be applied using matchmaking and act as recommendation engines.
- **Knowledge Management:** the electronic networking of all knowledge within a company and across companies. An important agent-related activity that has the potential to leverage knowledge management to a new quality is the semantic web initiative. Agents can use the metadata included in the semantic web to perform better quality searches and to provide users with better explanations for their decisions.
- **Buy-Side eBusiness (eProcurement):** the electronic networking of business relationships with the suppliers of a company. Software agents are geared to

support a company to automate (pre-)negotiation with existing suppliers based on given framework agreements and contracts. Details of these contracts can be negotiated automatically, like the amount of goods and the delivery date, based on the knowledge of the agents about production capacity and orders on the supplier side and the supply chain management information on the buyer side. Intelligent sourcing techniques and automated negotiation can be used for obtaining required goods or services on spot markets, e.g., if demand increases or an existing supplier has delivery problems. Recommendation engines based on matchmaking can provide sufficient information to select new potential suppliers.

- **Process Optimization:** adapting processes for end-to-end networking. Processes within a company can be optimized by integrating different single solutions into a company-wide network. Agent technology helps enterprises to optimize their business processes by (a) reacting quickly and autonomously to extraordinary events (e.g. breakdown of a machine); (b) ad-hoc planning and scheduling of processes; (c) using the available resources optimal or working to capacity; (d) integrating existing solutions on the interface and process side; (e) applying learning techniques to optimize the planned and scheduled processes; (f) dynamic negotiation and skills-based assignment of tasks; or (g) collective decision making.
- **eManufacturing:** lean manufacturing supporting cost-efficient, customer-driven management of operations. Agent technology can be applied for developing distributed intelligent manufacturing systems, including manufacturing enterprise integration, manufacturing planning, scheduling and control, materials handling, and holonic manufacturing systems. The key issues for developing agent-based manufacturing systems are enterprise integration and supply chain management, agent encapsulation, system architectures, dynamic system reconfiguration, design and manufacturability assessments, distributed dynamic and concurrent scheduling and execution, and factory control structures.
- **Supply Chain Management:** the networking from the purchase of raw goods and materials for manufacturing to the delivery of a finished product to an end user and cash flow. It includes eProcurement/eSourcing and eManufacturing (see above), and logistics. In the area of logistics dynamic advanced market and negotiation based resource management can be applied to achieve vertical (within a company) and horizontal coordination (task distribution across companies). Tracking information of the ordered goods and materials can help optimizing (re-scheduling) the processes dynamically in case of unexpected events, like traffic jams. Organizational techniques can be applied for forming teams and coalitions and to re-organize the supply chain.
- **mBusiness:** the mobile networking of eBusiness including mCommerce with additional value like location based services. Software agents can add advanced functionality for supporting the workflow of mobile employees by being situation-aware (including e.g., device, location, tasks, and time). It can be used to automate, optimize, enable or create mobility-supporting businesses and processes through individualization and process scheduling on the fly.

Today, many different software components and systems offer a wide range of standard business services and partially isolated eBusiness solutions. The objective of using agent technology in this area is to combine the heterogeneous solutions, to add advanced functionality and automating standard processes to connect companies worldwide. Thus, the major contributions of agent technology will be to help *em-*

power eBusiness users, and to *connect enterprises* internally and with their customers and partners world-wide

Agent technology must not reinvent the wheel; therefore it is necessary that these agent architectures are based on open standards, like XML or WebServices (see Sect. 5.3.3), allowing us to deploy their functions as components, and to provide wide scalability from mobile phones up to server farms, interoperability between applications, and secure access.

3.3 Agent Enabled eBusiness Architecture

Recalling the typical eBusiness software architecture presented in Sect. 2.2, we now discuss how agents can support the design of eBusiness systems on the individual tiers:

On the **client tier,** intelligent dialog systems and believable virtual personalities assist users in interacting effectively with the eBusiness systems. User agents can be instructed to provide a wide range of assistance services, to perform searches, negotiations, and transactions on behalf of their users. The instructed agent can reside on the end-user device and cooperate with other agents on end-user devices, servers, or marketplaces. Alternatively, it can migrate to the servers or marketplaces. This flexibility is supported by scalable agent platforms (see Sect. 4.2.1) that run on client devices (ranging from mobile phones up to usual personal computers). Individualized services such as personal negotiation support will be available to users on their **devices**. User agent on the device will increasingly monitor and learn user preferences from observing the user's behavior. In addition personal agents can support the user in planning and scheduling of tasks and managing appointments on behalf of the user. Tracking the user's tasks, appointments, time and positions allows a situation aware reaction of the user agent.

Client interaction can be integrated and extended with agent technology. At the moment this tier mainly deals with the device and domain specific representation and physical communication with the end-user device. The communication can support agent communication languages and in particular interaction protocols. The device specific content preparation can be improved using learning techniques about the user's behavior and interaction with the graphical user interface; in particular in the context of mobile devices with small displays the presentation can be optimized. This tier can perform ontology translations for different domains. The client interaction tier can support code mobility and provides the necessary security infrastructure, allowing mobile agents to migrate between servers and end-user devices.

At the **business logic tier,** agent technology can be applied for preference modeling of users and communities, and for collaboration and integration of business processes. Individualization is supported on different levels, like tasks, processes, services, and negotiations. The business logic tier can provide market places where agents can participate in auctions or other negotiations. The agents' collaboration facilities can be used to integrate with internal and external services and solutions. Distributed, on the fly planning and scheduling can be managed by agents using smart planning and scheduling techniques based on market mechanisms. Ontologies in connection with catalogue systems and ontology mapping enable the integration of heterogeneous data, workflow and processes. Negotiation and communication on a

semantics level allow agents to coordinate their tasks and resources. eBusiness specific user preferences can be learnt, leading to better negotiation strategies and improved processes. Matchmaking allows agents to find the most appropriate agents, products, or services for a task, negotiation, or market transaction and support decision making.

Within the **integration tier** agent wrappers can be applied as a specific technique for the integration of heterogeneous components, systems and services. Moreover on this layer process scheduling, dynamic reaction and on the fly scheduling like in the business logic tier can be done if appropriate. In addition agents can control the usage of appropriate services depending on their quality of service (QoS). In particular the agent can learn about the reliability of services and function as recommendation engines for external services and data providers.

Internal and external **systems and services** can be agentified and thus integrated into the agent framework or can provide agent-based services.

In the following section we will investigate how agent technology can overcome the shortcomings of today's eBusiness solutions.

3.4 Benefits of Agent Technology

The promises of agent technology in eBusiness are:

- To achieve enriched, higher level communication
- To enable more intelligence service provision, and process management e.g. by personalization and integration of different services to value-added
- To deal with the enlarging amount of information and functions, and
- To allow self-organizing of processes.

To realize this potential, agents need to communicate to discover their peers, to negotiate and to co-operate in open environments where everybody can add their contribution when and how it is deemed appropriate. Most importantly, agent systems will need to build on and interface with a variety of existing and upcoming developments and standards, like DAML+OIL, ebXML or WebServices. This includes support for a wide range of devices, but also integration of telecommunication and internet, and the integration with ERP, supply chain management systems, m/eCommerce platforms up to application servers and WebServices.

Thus, agents will only be able to fulfil their potential if they provide a standardized, open and generic infrastructure (see Sect. 5.1). Another important requirement for the success of agent technology is the availability of generic services, like for virtual team coordination, self-organization of processes, recommendation engines, mobility support and location-aware services. Such services are designed for multiple reuses and cover areas where higher level of intelligence is needed and agents seem more relevant than ever. In addition successfully marketing agent technology requires presenting it as an incremental extension of known and trusted technologies (e.g., object-oriented programming), and providing explicit engineering tools to support proven methods of technology deployment.

3.4.1 Challenge: Individualization and Privacy

Agent technology can support different kinds of personalization and individualization. First of all agent technology can support end-to-end processes from the end-user to some service and their scheduling using planning and scheduling mechanisms based on market-based resource management. These processes can be optimized according to the users' preferences like preferred working times.

Tracking the user, i.e. taking the actual time, the actual position and the actual tasks of the user into consideration and comparing these information with the planned activities can result in *situation awareness and reaction*, i.e. depending on the current situation, e.g. time, position, tasks to be performed in the future; specific action options are triggered in order to help to fulfil the schedule of a user if some extraordinary events occur like a traffic jam on the way to the next customer or if tasks take longer than expected. In particular *automating standard procedures*, e.g. appointment scheduling, using knowledge about the working and travel times, the position of a user, the performed work or the consumption of some material, e.g. semi-automatic ordering of new material could be included in the supply chain.

At the service level user preferences, like preferred jobs, hotels or brands, can be taken into consideration and adapted depending on user behavior. User agents can select what information to present and how to present it depending on the user's profile.

An important topic is *privacy*. Using agent technology ensures adequate user privacy, since user-side agents hold the private data and decide (based on instructions by the user) which information are given to others. The same holds for user agents on trusted server which can guarantee that user information is kept private. The user agent is the instance which selects the appropriate information for the user depending on their preferences.

3.4.2 Challenge: Secure Delegation

In order to be able to accomplish tasks (e.g., negotiations) automatically, user agents have to be equipped with knowledge (e.g., negotiation strategies). In addition, models of user preferences and profiles can be developed, to better seize dependencies between individual dimensions by mapping domain-specific attributes (e.g. price, material, extras) to general preference dimensions (security, comfort, thriftiness, achievement). This allows users to express their preferences via these general dimensions.

Distributed planning and optimization as well as coalition formation are other important components in multi-agent systems for delegating tasks. An interesting point in purchasing in electronic market places represents the formation of a buyers' coalition to achieve better prices in negotiations. In automated negotiation, planning and optimization can be applied to deal with global dependencies between different processes or supply chains, e.g. several parts have to be available for the construction of a larger product. In this case a constraint could be to have either all kinds of raw material available or the constructed product out of these different parts. I.e. the negotiation with all raw material suppliers can be seen as one transaction which has to be successful either for all parts or for none. The transaction paradigm is well known from data base applications and is supported by all major data bases. This technology

allows applications to place a transaction e.g. from one account to another account and obtain even in the case of some error a consistent state. In the case of an error a rollback is used to obtain a consistent state otherwise the transaction is committed and fixed. Agent systems are decentralized, distributed applications, thus transaction are important concepts for agents, too, in particular in the context of automated negotiations, where either an auction is completely accomplished or has to be set back under certain conditions.

3.4.3 Challenge: Semantic Interoperability

Usually agents are not closed units, but cooperate with each other to solve their tasks. The important issue concerning the cooperation between agents is that they should be domain independent and cooperate dynamically with unknown partners. Two levels of cooperation can be considered: The first level - some kind of meta-level - supports mechanisms for searching for cooperating agents, to analyze their features and to start cooperating with them depending on their domain specific behavior. At the second level of the cooperation the agents need a language to solve their domain specific tasks.

In such an open cooperation between software modules one problem arises, namely the usage of a common language between the partners with identical syntax and semantics. Agents can have different terms for the same concept and identical terms for different concepts. A common *ontology*, then, is required for representing the knowledge from various domains of discourse.

As a first approximation an ontology is a collection of well-defined terms of a specific domain, i.e. an ontology is a specification of the objects, concepts, and relationships in an area of interest. Moreover within an ontology semantic constraints can be defined. FIPA 98 states [17]:

> *An ontology gives meanings to symbols and expressions within a given domain language. In order for a message from one agent to be properly understood by another, the agents must ascribe the same meaning to the constants used in the message. The ontology performs the function of mapping a given constant to some well-understood meaning. For a given domain, the ontology may be an explicit construct or implicitly encoded with the implementation of the agent.*

The *knowledge model* is a specification of the set of primitives used by a certain class of representation languages. As such, a knowledge model can be considered a meta-ontology, i.e. an ontology can be defined using some knowledge model. The *ontology sharing problem* is the problem of ensuring that two agents who wish to converse do share a common ontology for the domain of discourse. Minimally, agents should be able to discover whether or not they share a mutual understanding of the domain constants.

In the area of ontologies we usually distinguish between *ontology* and *conceptualization*. A conceptualization is not concerned with meaning assignments, but just with the formal *structure* of reality as perceived and organized by an agent, independently of the language used to describe it and the current occurrence of a specific situation. An ontology, on the other hand, is first of all a vocabulary. Besides that, an

ontology must specify the *intended meaning* of such vocabulary, i.e. its underlying conceptualization.

Currently, quite a few web ontology languages are available, including SHOE, OML, XPL, ONTOBROKER, RDF, and RDF Schemas, DAML, and OIL. The most influential ones are the FIPA 98 ontology service based on Ontolingua, and OKBC, DAML, and OIL. The standardization of DAML and OIL is based on existing languages, and the combination of both, namely DAML+OIL, was published [7] and submitted to the world wide web consortium.

In eBusiness semantic interoperability is a crucial point. The standardization efforts shown in the related work can help to manage part of these problems. Semantic Web activities (see Sect. 5.4) and the ontology work in the area of agent-based systems can help to overcome additional interoperability issues.

In addition agent technology supports standardized interaction protocols as pattern of communication which are also in the focus of W3C's Protocol working group and of WSCL. These protocols are a necessity for the interoperability of standardized automated negotiations.

3.4.4 Challenge: Support for Flexible Organization Structures

Organizational models and their representation and support are a core topic in eBusiness research. In [20], the notion of an organization is defined as *all regulations, which provide for a co-ordination of the enterprise and for its adjustment at the company target.* Current research in organizational models focus on techniques to analyze organizations and their processes.

Due to the nature of agents as decentralized, autonomous entities, organizational modeling technique can be transferred to multi-agent systems (for details see e.g. [65]; parts of the section are based on this reference). As already stated not all agents can perform all activities, but some agents are specialized for appropriate tasked. Organizational models are a structuring mechanism to obtain results which are not achievable by an individual agent, but need coordination of the individual agents. One rationale for the existence of organizations is to overcome the limitations of individuals (agents). Transferring this to agents we have four basic limitations [65]:

- **Cognitive Limitations:** agents have cognitive limitations and need to cooperate to achieve higher levels of performance
- **Physical Limitations:** agents have only limited resources and therefore must coordinate their actions, especially accessing non-local resources
- **Temporal Limitations:** agents are temporally limited and therefore must join together to achieve goals which transcend the lifetime of a single agent
- **Institutional Limitations:** agents are legally or politically limited and therefore must attain organizational status to act as a corporate actor rather than as an individual actor

However there is no individual organizational model which fits all problem domains. E.g. in some domains specialization tasks whereas in other domains division of labor can be appropriate. But over-specialization and excessive division of labor can reduce performance and flexibility by de-skilling individuals, decreasing attention due to

boredom and increasing decision making time and increasing coordination costs in situations of uncertainty or failure. These facts are also reflected in eBusiness.

Research in the field of computational organizational theory uses computational and mathematical methods to study both human and automated organizations as computational entities, in particular necessary in the eBusiness domain and can be solved using agent technology:

- Human organizational models are characterized by acquiring, manipulating and producing information through joint, interlocked activities of human beings and automated information processing.
- Automated organizational models are characterized by multiple distributed agents which exhibit collective organizational properties
- For the modeling of an organization usually the following points have to be considered:
  - agents - humans as well as artificial - comprising the organization by modeling their organizational roles and decision making influenced by their capabilities and knowledge
  - the organization's design or structure specifying the aspects like tasks, roles, skills, collaborative teams, agents hierarchies, and resources.
  - tasks the organization carries out, especially taking into consideration temporal constraints or other dependencies on the single tasks, and similarities of tasks
  - any environment of the organization, and the organization is involved in.
  - the organizational material transformation and / or information processing technology
  - any stressors, like time pressure, deadlines, and turnover, on the organization.

Thus, virtual organizations and flexible organizational structures show many of the characteristics that make an agent-based approach seem adequate. However, any agent-based solution must build on established standards, especially in the area of process modeling and enterprise application integration (see Sect. 5.3).

3.4.5 Challenge: Intelligent Collaboration and Coordination

Communication is a key feature in multi agent systems. The communication between different agents is necessary to exchange information, to distribute tasks, plans and goals, to coordinate actions, to negotiate prices and resources, to manage shared resources and to recognize, avoid and manage conflicts.

The easiest way of communication between two agents is simple method or remote method invocation. More complex interaction would include those agents that can react to observable events within the environment, i.e. they receive events regarding the state of the environment. Even more complex interaction is found in systems where agents can be engaged in multiple, parallel interactions with other agents. Here, agents begin to act as a society. Based on these considerations one can distinguish different levels of abstraction as shown in Fig. 5.

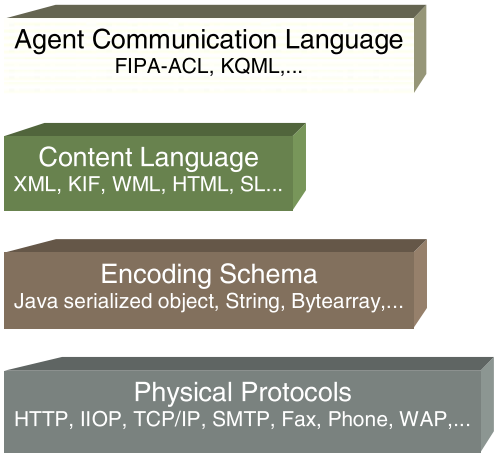

Fig. 5. Levels of abstraction for communication

At the lowest level, communication is done via physical protocols. Messages can be sent synchronously or asynchronously. Transportation mechanisms support unique addressing as well as role based addresses (i.e., ”white page” versus ”yellow page” addressing). Furthermore, they should support unicast, multicast, and broadcast modes and services like broadcast behavior, non-repudiation of messages, and logging. On top of this level different programming languages and tools provide different encoding of information, namely in the case of Java for example as serialized objects, strings or other programming specific representation of information. In the context of agent based systems an abstraction is made from this programming language specific coding by defining *content languages*: content languages can be XML, WML/HTML, or logic based languages like KIF or the FIPA content language SL. To enable negotiation between different agents, e.g. for task delegation, communicative acts are used, grounding on speech act theory. Examples of communicative acts are *cfp* (call for proposals) starting a contract negotiation between agents, or *request* to request an agent to perform some task. These communicative acts and additional information, like sender or receiver of a message, are covered by agent communication languages (ACLs). The main representatives of ACLs are FIPA-ACL [16] from the standardization committee FIPA, and KQML [15]. Within ACL messages the content languages are used to code the real content of the message, e.g. the starting of an auction or the proposed price of a good.

Agents can interact in various patterns called *interaction protocols* also known as *conversation* or *communication* protocols which are built on top of the communicative acts. Agent interaction protocols as defined e.g. by FIPA can be seen as pattern for interaction between different agents that is formally defined and abstracted from any particular sequence of execution steps. Whichever kind of conversation is chosen, the pattern or protocol of that conversation must be understood by the participating agents. The behavior of an agent how to participate in some auction, for example, is fixed by the agent interaction strategies.

In eBusiness, the communicative act, interaction protocol and interaction strategy layer are important for the automation of business processes and automated supply chain management.

3.4.6 Challenge: Mobility Support

Agent technology supports human mobility at different levels. Code mobility is provided by mobile agents, which can migrate from one machine to another. Because of their migration capabilities mobile agents can (1) benefit from the capabilities of a visited computer. This can be data of a special database, or services, such as online banking or the connection to wireless networks which are only accessible on the visited computer; (2) take part in auctions on a user's behalf while the user is offline, for instance because of areas with no reception. Moreover in this scenario the response times are shorter than when using mobile communication; (3) profit from the resources of different servers to calculate solutions in parallel. The performance of the client machine can be low, however the server machines have to be powerful; (4) be supported by the client as well as by the server and the dynamic adaptation to the requirements of the user. This proceeding allows installing the necessary software (dynamic software update); (5) allow independent, off-line operation, e.g., after agents have migrated to some server to perform their tasks a notebook can be switched off. This fact results again in higher availability and robustness, since the agents change their environment system and in contrast to typical client-server approaches no permanent connection to the server has to be established; (6) save bandwidth if the agent as a software program and all its data needs less resources than the data which have to be processed by the agent; (7) communicate fast because of the local communication via e.g. shared memory; (8) support load balancing.

As already stated situation-aware services can be supported by software agents. Moreover the upcoming lightweight agent platforms (see Sect. 4.2.1) deal with mobile communication and mobile collaboration supporting agents on small (mobile) devices.

3.4.7 Challenge: Pro-active, Adaptive Processes

Agent-oriented systems are based on small units with negotiation facilities as shown above which can coordinate the tasks and goals independent of some central process, self-organization is one of the key issues agent-based systems can support. Moreover, the pro-active behavior of agents based on planning and scheduling together with autonomous behavior allow agents to organize complex tasks and processes in agents' inherent distributed environments on their own. In particular with organizational concepts agents can support pro-active processes. Applying learning techniques can improve the process formation.

In particular agents are applied for eManufacturing. According to [29], the scheduling and manufacturing control structure used within the MABES system is a distributed autonomous agent framework. Thereby each agent is responsible for monitoring and acting on a component of the manufacturing process. A component may be a process, such as a press; or a stack of pre-processed components. The agents interact to control the flow of parts through either a traditional push or a lean pull or takt system. Within the adopted approach, the overall desired behavior for a manufacturing line emerges from individual behaviors of, and interactions among, distributed agents. For other research on agent-based manufacturing we refer to [21], [6], [47].

3.4.8 Challenge: Adaptive Decision-Making Assistance

Beyond the benefit of agent technology described in Sects. 3.4.2 and 3.4.5, matchmaking can be applied to deal with the topic of adaptive decision-making assistance. Matchmaking is the process of mediating demand and supply based on profile information. The task to be solved is to find the most appropriate agents, products, or services for a task, negotiation, or market transaction. Most real-world problems require multi-attribute matchmaking, i.e., the ability to combine various dimensions of decision-making to define an overall solution to a matchmaking problem, requiring the interplay of multiple matchmaking algorithms. In addition, in order to be applicable for real-world applications, the matchmaking component must be easily integrated into standard industrial marketplace platforms. The essential requirements to be met by a matchmaking framework are the following:

- Demand and supply profiles in electronic marketplaces are often complex (e.g., matching job profiles against applicant profiles) and require multi-dimensional matchmaking. Often, a combination of existing methods is adequate to deal with different aspects of matchmaking.
- Demand and supply information is distributed and heterogeneous. Thus, distributed search and ontology mapping may be required to achieve comparable profiles.
- Demand and supply profiles differ depending on the application domain; also the underlying business logic to determine the quality of a match (distance functions) is very domain specific in part. A framework must support this variety.
- A framework should restrict the effort of developing new marketplace solutions by enabling reuse of existing profiles and business logic. In particular, developing a matchmaking solution should require only little coding, and should be done mainly through customization.
- The matchmaking framework should assist marketplace developers by supporting a clear process for building matchmaking solutions; appropriate tools should support the enforcement of this process.
- Beyond relying on agent standards such as FIPA, which can be achieved by using FIPA-compliant agent platforms (see Sect. 5.1), it is necessary to serve the integration needs and capabilities of today's eBusiness platforms, which are mostly based on Enterprise Java Beans (EJB) application servers.

In Sect. 4.1 we present an agent-based approach to sourcing and matchmaking in the human resources domain.

3.4.9 Challenge: Intelligent Selection and Evaluation of Products and Services

Helping individuals and organizations to find what they need in distributed electronic marketplaces and vast heterogeneous content repositories requires a number of tasks to be accomplished well: (1) individualization of requests; (2) automation of search; and (3) interpretation and evaluation of candidates. Agents have the potential to contribute to each of these challenges. By maintaining and learning user profiles,

agents can construct appropriate demand profiles. By using brokering services (e.g., yellow pages) and by doing query planning to combine multiple services, agents can find possible matches for a demand profile. In this context, they make use of existing services as described e.g., by [3]. Finally, matchmaking combined with the ability to negotiate provides the means for evaluating possible candidates and helping the user to get the best deal.

The interoperability with existing eBusiness platforms is largely based on the usage of standards and enabling technologies as described in Sect. 5. Lately, we have observed that eBusiness software companies enter the market with products the features of which contain agent support for personalization and marketplace interaction (e.g., [38], [37]). While it is not always easy to see how much of this is marketing and how much is technology, we believe that this is an encouraging sign that agents will play a considerable role to tackle this challenge within the next three to five years, and that the big players in the market might include agent features into their products and solutions once the basic foundations for eBusiness has been established in enterprises.

4 Agents for eBusiness Applications: Case Studies

In this section we describe some case studies for the use of agents in eBusiness applications. The examples include the use of agents for intelligent human resources matchmaking, distributed team coordination, and personal travel assistance.

4.1 Agents for Human Resource Matching

In this section we describe the GRAPPA system for agent-based matchmaking, and an agent-supported tool for e-recruiting based on GRAPPA.

4.1.1 Agent-Based Matchmaking

Matchmaking is not only a key task in multi-agent systems, it is also a crucial function in industrial portals and marketplaces. The provider who will enable the most effective matches between demand and supply will gain a competitive advantage and increase the acceptance and popularity of their marketplaces.

Fig. 6 illustrates a usage scenario for a matchmaking agent in an electronic marketplace. On the supply side, providers of services or products make themselves known to the matchmaker (which can be based on push (e.g., by registering) or pull (e.g., by search initiated by the matchmaker). In Step 2, a requester requests a service (by issuing a demand profile) and the matchmaker returns a list of the k best matching providers. In Step 3, the requester and the selected provider will negotiate a contract and upon agreement the service or product is delivered. Here, we will focus on the matchmaking phase (step 2).

We understand matchmaking as a function which accepts as input a set of offers (supply profiles, candidates) and a request (demand profile) and provides as output a ranked list of the k best offers with respect to the request.

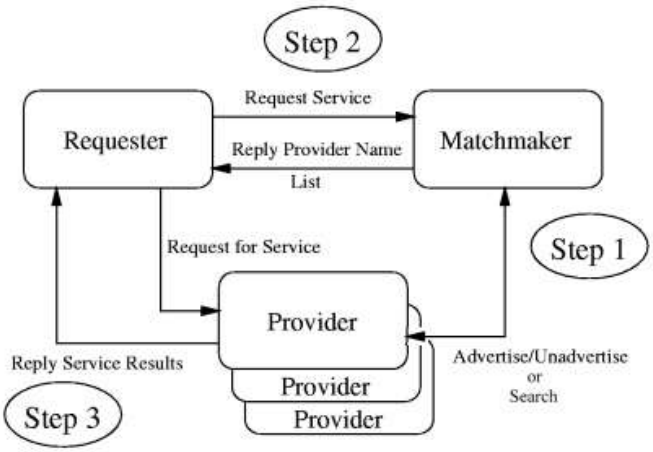

Fig. 6. Matchmaking Procedure

4.1.2 The GRAPPA Matchmaking Framework

Fig. 7 illustrates the structure of the GRAPPA matchmaking framework. It consists of three major parts. Its core is the *matchmaking engine*. It is complemented by the *matchmaking library* and the *matchmaking toolkit*.

4.1.2.1 GRAPPA Matchmaking Engine

The matchmaking engine accepts a set of supply profiles (candidate instances) and a demand profile as input. The supply profiles which have to be provided as instances of the matchmakers candidate class are either stored in the matchmakers service repository (see Fig. 8) or – in case the matchmaker does not keep a service repository – retrieved from different data sources. The request which has to be provided as an instance of the matchmakers demand profile class is matched against each of the candidate instances. The candidate structure as well as the demand profile structure are multidimensional. They consist of complex types constructed from a domain specific set of basic types under application of four complex type constructors: list, array, record and set.

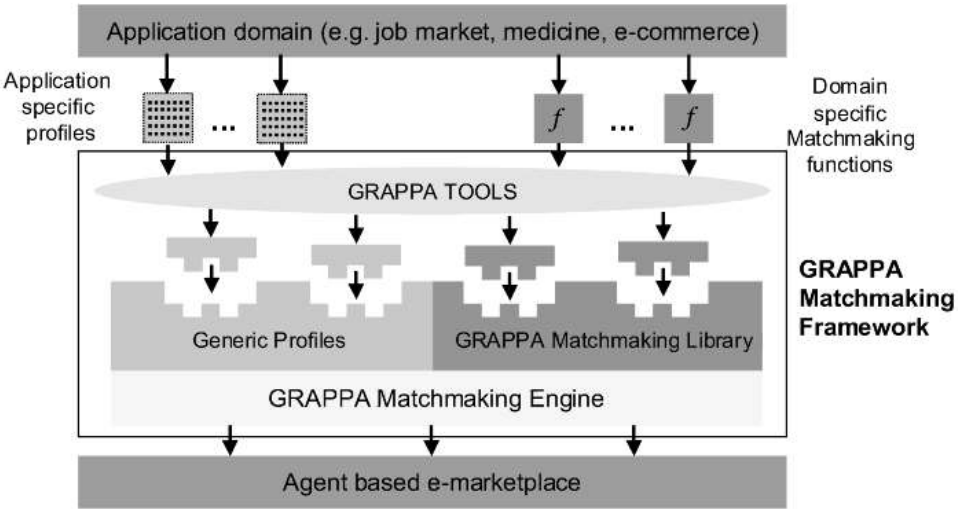

Fig. 7. GRAPPA Matchmaking Framework

The procedure performed by the matchmaking engine is illustrated in Fig. 8.

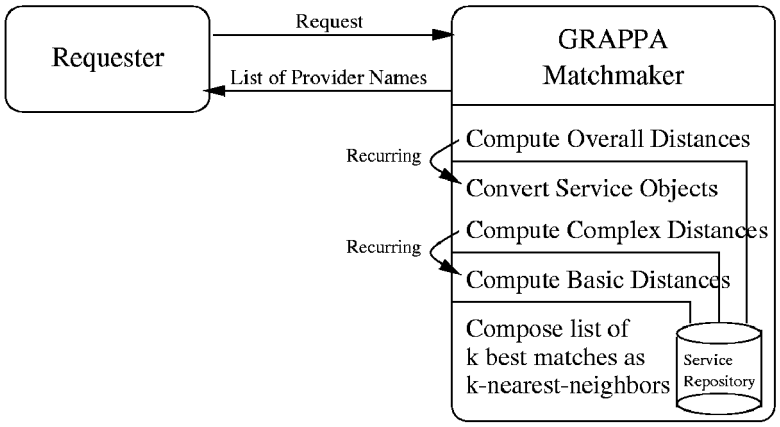

Fig. 8. GRAPPA Matchmaking procedure

The overall distance, a real value between 0 and 1, is computed by recursively computing the distance values for different profile sub-types and propagating them upwards to compute the values for their parents. For the basic types (the atomic attributes of the demand and the supply profile), the specific distance function for the particular type is applied and the result is propagated upwards. Then, at the next higher level, all basic distances between the atomic types in this level are merged to one distance value for this complex type under application of aggregate functions.

The result of the recursive computation of distance values is

- an overall distance (real value between 0 and 1) which reflects the quality of the considered candidate instance for the current demand profile instance.
- a structure (in XML) which consists the individual distance results in each layer.

The best k candidates (with respect to the current demand profile) are returned as the result of the match. This list is ranked using the value the overall distance

The agent (or the agent's principal) can then recur into the XML structure to obtain an explanation how the particular overall result arose (e.g., which aspects of the match contributed to a good or bad overall result).

4.1.2.2 GRAPPA Matchmaking Library

The GRAPPA Matchmaking Library hosts an extensible collection of predefined profile schemas and (general-purpose or domain-specific) distance functions. The profiles schemas can be used as a basis for application-specific profiles; the distance functions provide uniform interfaces that allow us to flexibly combine them to develop specific matchmaking solutions.

It is essential for a matchmaking system to provide powerful distance functions. Currently, we provide distance functions for FreeText, WeightedKeyword, Interval, TimeInterval, DateInterval, Boolean, and Number basic values (i.e. instances of basic types). All distance functions have the property to take two basic values as input and to provide a real number between 0 and 1 as output (distance). Additionally, domain

specific distance functions can be easily integrated to accommodate the requirements of different matchmaking applications.

On top of these basic functions, we can define aggregate distance functions. Currently, WeightedAverage, Average, Minimum, Maximum are supported as predefined aggregate functions. As for basic functions, it is possible to define domain specific aggregate functions and integrate them into a domain specific matchmaker.

4.1.2.3 GRAPPA TOOLKIT

The GRAPPA Toolkit provides a set of tools which enable the development of a multidimensional matchmaker for specific applications mainly through configuration without much coding work. To guide the marketplace designer we have defined a 5-step process to obtain a domain specific matchmaking solution:

- Define the demand and supply profile schemas (basic entries) in XML;
- Define the clusters of attributes in XML (pseudo-orthogonalization); clustering can be recursive;
- Associate the clusters of the demand profile with clusters of the candidate by applying appropriate distance function;
- Combine the results of the distance functions to an overall distance value (e.g., weighted sum);
- Apply feedback regarding the quality of the matches, e.g., by adaptively changing weights or matching functions.

4.1.3 Application: HRNetAgent

Due to the open, flexible architecture of the GRAPPA framework, it can be applied to a wide range of matchmaking problems in all sorts of (agent- or human-operated) electronic marketplaces. In this section, we provide a brief description of two industrial projects in which GRAPPA has been applied successfully: The Siemens Cooperation Market (CoMa) and the Human Resource Network project (HRNetAgent).

4.1.3.1 Siemens Cooperation Market

CoMa is a Siemens corporate service where individuals and teams describe their capabilities, including competence and availability, as service offers and supply this information to some designated provider agent. Departments or customers in need for teams to take over certain projects or tasks forward their project profiles to a requester agent. The CoMa matchmaker keeps a repository of all available provider agents and processes the incoming requests using the GRAPPA matchmaking framework.

4.1.3.2 Human Resource Network (HRNetAgent)

The Human Resource Network (HRNetAgent) is an application of GRAPPA for matching corporate job profiles with profiles of job applicants (i.e., unemployed persons), stored in various data bases. The current version of HRNetAgent is a prototype system that has been developed for the German Labor Exchange Office, and demonstrates the feasibility of a partially automated approach to employment relaying. Based on its success, a full-fledged system is planned for the near future. The potential return on investment is huge: reducing the relaying time of unemployed persons (towards the end of 2001, there are almost 4 million people in Germany

without employment) just by one day on average will save the German government more than a hundred million dollars a year.

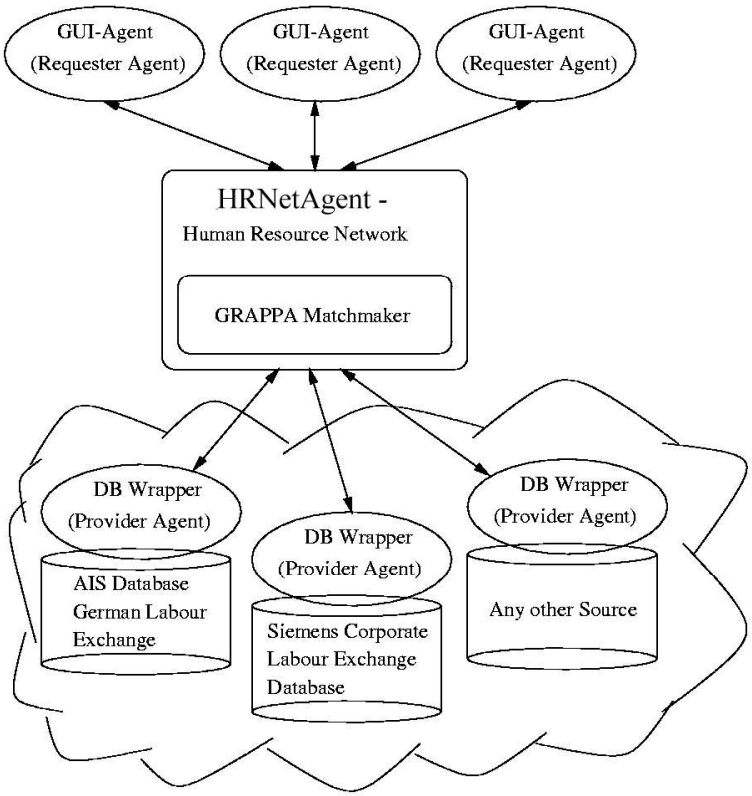

Fig. 9. HRNetAgent System overview

Fig. 9 shows the architecture of the HRNetAgent system. A company specifies its job profiles to a designated GUI-Agent, which takes the role of a requester agent in the system. The GUI-Agent queries the matchmaker by sending to HRNetAgent the description of the open position which should be filled. The scheme for specifying the open positions is the demand profile.

The backend of HRNetAgent consists of a collection of data sources wrapped by information agents, and by a search controller that coordinates a number of search agents. E.g., one data source is the central database of the German Federal Labor Exchange Office, in which all currently unemployed persons in Germany are stored. Others may be corporate skills databases, additional databases can be easily integrated. Note, that the database wrapper agents play the role of virtual provider agents in our architecture.

Wrapper agents perform the task of query translation, connection handling, and result translation. They return a pre-selection of profiles to the matchmaker based on conditions extracted from the demand profile. In HRNetAgent, the demand and candidate schemes are converted to XML-DTDs which are considered as the document classes of these types. Matchmaking thus is done on a pre-selection of candidates. The most successful candidates for a job profile are stored in the local service repository for fast access by the application. In addition, HRNetAgent offers an automated notification service via SMS, Fax, or Email.

4.2 Agents for Corporate Team Management

Within the European project LEAP ([34], [4]) a lightweight extensible agent platform is developed and applied being the precursor of the second generation of FIPA-compliant platforms. It solves a major technical challenge - it is the first integrated agent development environment capable of generating agent applications and executing them on run-time environments implemented over a large family of devices (personal computers, PDAs, mobile phones like the Siemens SL45i etc.) and communication mechanisms (TCP/IP, HTTP, etc.). The feasibility and real-world evaluation of the LEAP platform is performed by field trials dealing with an integrated solution on de-centralized work co-ordination, travel management and knowledge management.

4.2.1 The Platform JADE/LEAP

LEAP emerged as an independent development branch of JADE (see [30]) under the LGPL license and was merged with the JADE mainstream in September 2001, at which point LEAP transformed JADE's kernel. As such, LEAP concentrates on lightweight and extensible aspects, whereas JADE continues independently its evolution towards environmental functions such as monitoring facilities, visualization packages, ontologies and policies. The JADE APIs remain unchanged. Therefore all existing applications continue to run as before. In addition, developers can use JADE/LEAP to migrate existing applications to, or develop a new generation of applications for small wireless devices. The LEAP applications on virtual mobile team management are a good illustration of these new capabilities.

The LEAP activity focused on restructuring the JADE core, compliant to Java 2 Standard Edition (J2SE), in order to match the LEAP requirements and to obtain a single platform that is:

- Lightweight enough to be deployed on small devices, such as mobile phones, supporting only a KVM with Java 2 Micro Edition / Connected, Limited Device Configuration (J2ME/CLDC) and MIDP, instead of a standard JVM.
- Transport layer independent and in particular supporting transport protocols suitable for both the wired and wireless environment, thus providing an homogeneous layer to agent application developers.
- Compliant to the last FIPA specifications.
- Extensible such that, when deployed on a powerful machine, it can provide a number of optional functions such as agent mobility, user-defined ontology's, and platform management GUIs.

The complete external view of the LEAP platform is a distributed system. A JADE agent platform consists of so-called containers, being either an agent container or a main container. All components and agents of a container are loaded into the Java Virtual Machine (JVM). An agent container is a container for agents, which can be either empty or containing one or more agents. The amount of agents running in a container can change during execution time. Each agent container may consist, beyond the agents, of a message dispatcher responsible for the delivering of agents' messages to other containers, and an Agent Communication Channel (ACC) responsible for the message passing between different agent platforms. A main container is a

special case of an agent container. For each JADE platform exists exactly one main container, containing an Agent Management Service (AMS) and Directory Facilitator (DF) agent and a RMI registry, since in JADE the intra platform communication is done using RMI. Therefore, a main container builds up a complete FIPA compliant agent platform. Arbitrary agent containers can register at the main container and thus the agent platform can be distributed over several computers.

Tasks in LEAP are implemented using behaviors. A behavior can be execution and terminated and must be removed from the agent pool of behaviors. Behaviors can be added and removed to/from the agent both during agent initialization and from within other behaviors. All agent behaviors are managed by a scheduler that, at each round, decides which behavior to select for execution.

LEAP is deployable on different types of device and different types of network. The part of a container that handles communication (both intra-platform and inter-platform) is isolated and different implementations dealing with different transport protocol are plug-able without affecting the rest of the container:

- With the outside world, to send and receive FIPA messages to and from other FIPA compliant platforms. This inter-platform exchange of information occurs over one or more Message Transport Protocol (MTP)
- With the other containers in the platform e.g. to dispatch an ACL message to an agent on another container. This intra-platform exchange of information is carried out through platform management commands that the containers send and receive and occurs over one or more Internal Transport Protocol (ITP).
- Inter platform communication (involving a container and a remote platform) where FIPA messages, with a well defined structure (envelop plus payload), are exchanged. This is handled by the `ACC` in Jade which is already able to manage different transport protocols. Therefore, no modification is required as far as this type of communication is concerned.
- Intra platform communication (involving two containers in the same platform) where platform management information are exchanged. In JADE the implementation is done directly performing RMI calls to the container where some information has to be transferred/retrieved. It has to be noticed that ACL message dispatching to an agent residing on a remote container falls in this type of communication. The exchanged information in this case is an ACL message.

The command dispatcher manages different transport protocol objects to which it delegates the operations related to actually sending/receiving data over the network. The transport protocol interface abstracts from the details related to a given transport protocol (HTTP, IIOP, RMI, etc.).

Agent mobility allows agents moving within a LEAP platform exploiting a LEAP-proprietary mechanism. This kind of mobility, that we call *intra-platform*, is fundamental to cope with mobile devices because it might be quite common for agents to move from the fixed network to the mobile device to allow the user disconnecting the device and having the agent running on it. Nevertheless, the implementation of such functionality requires implementing a Java class loader and the J2ME CLDC/MIDP does not allow this. Therefore, we consider intra-platform mobility as part of the optional functionality.

Inter-platform mobility allows agents migrating between different FIPA agent platforms and it requires a precise FIPA specification describing the messages which must be exchanged by platforms in order to implement such a migration. Currently

FIPA does not provide specifications for this type of interoperability. Hence, the LEAP platform design does not currently support inter-platform mobility as part of the optional functionality.

4.2.2 LEAP Field Trials

LEAP is addressing the need for open infrastructures and services which support dynamic enterprises and mobile teams, against a backdrop of information overload, increasing competition and globalization. The project develops key enabling technologies that facilitate virtual team working and managed risk-taking through decentralized policy-based management and workforce empowerment.

The goals of the LEAP agent-based services and applications are:

- prove that agent technology can add value in the management of mobile teams. Three kinds of problems are tackled: knowledge management, work co-ordination, and travel assistance;
- demonstrate the advantage of locally based agents on small devices as part of a distributed application. Agents increase the autonomy of devices and applications;
- prove the viability of both the infrastructure and agent-based services through two independent field trials deployed in Germany (ADAC) and the UK (British Telecommunications), running the same services, but in different operational contexts. Each field trial will involve users in vehicles, roaming over large areas, for a duration of several weeks;
- demonstrate the use of a number of device/operating system combinations, during the field trials.

The agent-based services which address the following key requirements:

- knowledge management - anticipating a user's knowledge requirements by accessing and customizing knowledge (based on the users skill, location, current job and type of display) and providing access to collective knowledge assets in the team (by networking individuals with each other, based on their current needs);
- decentralized work co-ordination - empowering individuals to collectively co-ordinate activities (e.g. by trading jobs, automatically negotiating for work, and expressing personal preferences) within an agreed policy framework;
- travel management - anticipating a user's travel needs, providing guidance and time estimation so as to synchronize the movements of virtual teams working over vast geographic areas.

The field trials are carried out in the open air, using mobile devices such as PDAs and mobile phones. In the case of the BT field trial, the domain is that of telecommunications engineers performing field-based installation and repair task. The domain of the ADAC field trial is roadside assistance for stranded motorists. Each field trial will last approximately several weeks. As the field trials will be carried out in two different countries they will use different existing communications infrastructures, access different knowledge sources, and most important of all, involve users with different cultural backgrounds. The field trials will not only assess the technical quality of the LEAP applications but also the usability and "soft systems" aspects in the context of mobile remote workforces.

4.3 Personal Travel Assistance

The increasing demand for mobility in today's society is leading to drastic increases in traffic volumes, causing undesirable effects such as traffic jams and overcrowded transport system restricting mobility (for details see [3]).

More and more services are becoming available electronically (in particular via the Internet), offering information and support in planning a journey. Unfortunately the typical user is currently overwhelmed with the large number of travel information and booking services, with too many user interfaces and individual offerings. The services which were integrated into a unified system are rarely. Today, in order to plan a trip, the user must initiate an extra session for each transport mode (e.g. automobile, train or plane) and for each layover (e.g. hotel, parking, car rental). These services are mostly offered separately from each other and are seldom individualized for the user. For example, the user needs to enter all of his/her data, such as home address, as well as individual preferences for the travel means (e.g. business class) more than once. When the traveler is on the way, things get even worse: the services are most often not available, especially when the traveler needs those most, such as when the travel plan need to be changed due to traffic jams or delays.

PTA offers a new perspective for individualized and automated handling of the vast amount of information and services, in order to support the traveler effectively in planning and during a journey. PTA is a comprehensive agent-based system comprising the complete chain from basic services up to the end user devices. PTA was part of the MoTiV-initiative of the German industry, for details see [3]. The project has been successfully finished in late 2000 and now the partners are encouraged to develop their own business based on the research results. Project partners included Siemens, BMW, Bosch, DaimlerChrysler, debis, IBM, Opel, VDO car systems, and VW. The agent architecture was an essential part of this prototype. The system architecture of the PTA system is illustrated in Fig. 10.

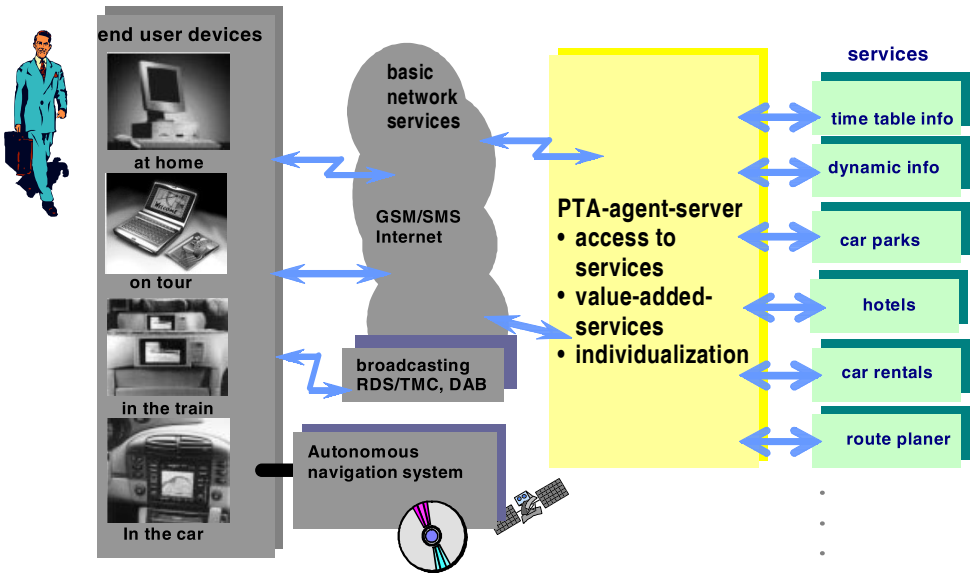

Fig. 10. PTA system architecture

The PTA system offers support for the user at any place and at any time. At home the user access PTA via personal computer and the internet; on tour handheld PCs can be used. Special end user devices support the traveller in the train with information about the delays of the current train, or the surrounding of the destination railway station. Calculated routes based on dynamic traffic information as well as other dynamic information (e.g. positions of car parks with free parking slots) can be loaded into the car navigation system and on-trip usual traffic messages via standard broadcasting mechanism are taken into consideration by the routing system. All the communication is done using standard communication channels.

The heart of the PTA system, namely the PTA agent server accesses the different services, combines the results to value-added information and takes personal preferences and travels into consideration.

The PTA system uses existing travel services, e.g., time table information from the German railway company ”Deutsche Bahn”, Lufthansa, and the public transportation of Hamburg and Munich. Dynamic information about traffic jams and reservations of car parks are accessible for distinguished areas of Germany. One car rental service supporting car rentals all over Germany is integrated too. Two hotel reservation services supporting hotel search and booking nearly all over the world are connected to the PTA server. An individual (car) route planner takes dynamic information, like traffic jams and building sites on roads in consideration. A new developed service, called ”tracking”-service, informs the user on-trip about traffic jams and delays of trains, depending on the current position of the travel, i.e. only information about travel segments in the future are interesting for the traveler. If e.g. a plain cannot be caught any more, a new travel planning can be started. Fig. 11 illustrates the agent architecture underlying PTA.

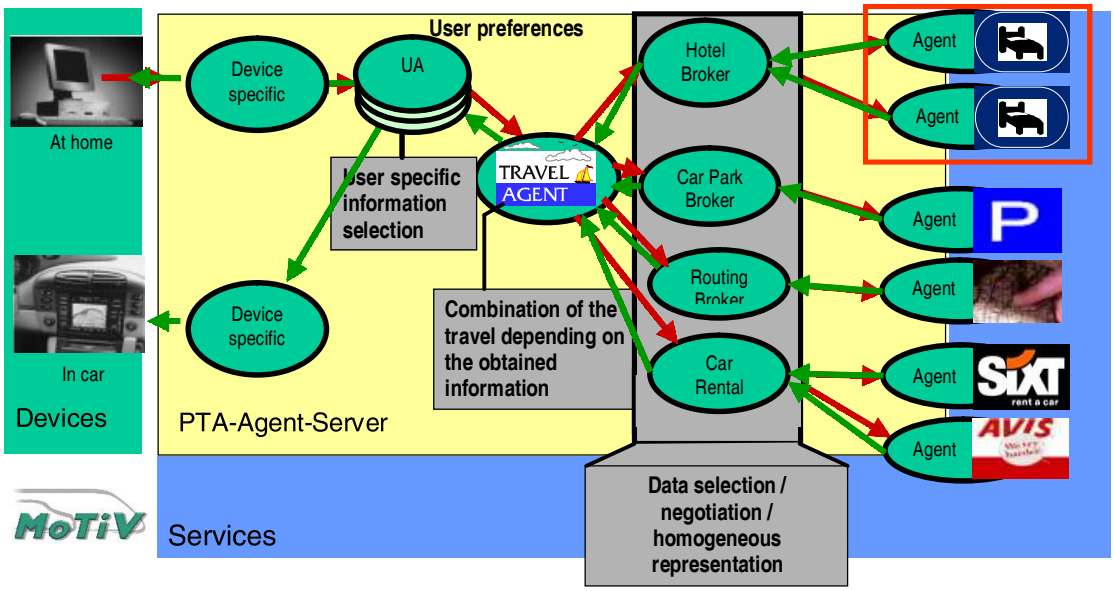

Fig. 11. Agents in PTA

User Device Agents: For each kind of end user device there exists a specialized version of a generic user device agent (UDA). The task of the UDA is to process the information depending on the end user device, like the generation of dynamic HTML

pages for some usual handheld PCs or palm-size devices; or generating WAP compliant pages. Thus the presentation is disconnected from the user agent and it is very easy to add new and different as well as additional end user devices without changing the rest of the system. Adding a new end user device agent can be done during the normal operation of the PTA system.

User Agents: For each human user a user agents exists. The user agent is started on login into the system and is stopped, if all tasks of the user agent are finished and the user is logged out. The user agent (UA) stores and manages all the already planned and booked travels for the user. It knows the preferences of the user and does the selection of calculated travels according to them. Moreover, it analyzes the selections of the user and adapts the user preferences accordingly.

Travel Agent: The travel agent works as a virtual travel agency. It asks the different brokers to prepare information, to book hotel reservations or to make a route planning. It combines the travel results according to special user preferences which are known to the travel agency. The user agent has all the preferences and does the selection or ordering of the travels alternatives for the user.

Broker: The different broker agents access the service agents to get the information or do the booking. The main task of the broker agents is to perform the requests on behalf of the virtual travel agency and combine the results of different data sources e.g. hotel data shops, while the information of this data shops is in a unique representation.

Agent Wrapper: The different kinds of agent wrappers are adapted to already existing travel services (e.g. hotel or airways information services) and access these services directly. Different services support heterogeneous interfaces, like RPC, CORBA, or proprietary interfaces. The task of an agent wrapper is to support the different interfaces and map the information into a unique representation, namely the agent communication language, used internally in the PTA agent system.

PTA reveals all the typical characteristics of an application that can benefit from agent technology:

Distribution and Mobility: Services as well as users are geographically distributed. Moreover, queries to the system can be set up at any place, at any time.

Heterogeneous and Autonomous Components: The PTA system is a heterogeneous system concerning end user devices, supported services and interfaces. The used agents are autonomous and prepare travel plans according to user preferences and other external constraints using coordination.

Value-Added Services: By combining different existing services, new value-added services are created, like the virtual travel agency or an inter-modal route planer, which combines different transport means, like car, train or airlines.

Queries and Booking: PTA allows on the one side the query of information about, e.g. hotels, train schedules, etc. and on the other hand the booking of the different travel services, based on transactions among agents.

User Profiles: The planning of a travel is performed taking the individual preferences of the travelers into account, like preferring the car or a special airline with distinguished rates for companies. The preferences are learned according to the selected travels out of a set of possible travels.

Different Communication Channels and End User Devices: The access to the PTA server can take place using different end user devices and communication

channels, ranging from hand-held PCs, to the integration into car navigation systems or distinguished seats of a train.

Robustness, Extensibility, and Modification: Using multi-agent systems the PTA system is robust wrt. error of components. It can dynamically be extended and modified.

The PTA system supports about 25 different kinds of interfaces to services and user interfaces; about 200 different requests to the services and answers from the services; about 20 parameterized interaction protocols, e.g. for information retrieval, booking, etc.; many non-parameterized protocols, e.g. for the communication with yellow and white pages. It has been shown with the project that an agent-based realization is ideal for connecting services or other existing software with different, heterogeneous interfaces using wrapper technology. The applications of brokers allow an implementation which is easy to grasp, because of the modular implementation. In order to support the dynamic integration of new similar services, like travel monitoring, the implementation is based on specific generic, domain-dependent pattern of interaction, not only taken the interaction protocols into account, but also the content of the communicative acts. Moreover agent technology was an appropriate software engineering paradigm to cope with the complexity of the system; in particular agent technology supports the necessary abstraction mechanisms, like interaction protocols or abstracting from the concrete physical communication between the agents using the underlying FIPA compliant agent infrastructure.

The experience with the application of a FIPA compliant agent platform was twofold. First of all the underlying infrastructure is ideal for the dynamic starting and stopping of services and user agents, also for the connection of different PTA servers supported by different vendors. But the use of the content language FIPA-SL in the framework of a Java-based agent platform makes things more complicated than necessary, although MECCA supports automatic generation of FIPA-SL terms out of arbitrary Java objects.

Within the European project Leonet [36] we added a learning component based on reinforcement learning and evolutionary algorithms to the PTA system to allow the adaptation of the user preferences according to the selections of the user.

5 Related Work

In this section we will have first of all a closer look at agent standardization activities. Afterwards the enabling technologies are presented. We show current state of the art in integration technologies and finish with the semantic web initiative, matchmaking and agent-based marketplaces.

5.1 Agent Standardization

The main standardization efforts relevant for agent technology are the OMG viewing agents as extensions of agents, KQML a de-facto standard for agent communication, FIPA – the standardization organization for agents and the World Wide Web consortium dealing with negotiation protocols and security aspects. Moreover the

Java community process deals with standardizing Java interfaces for agent name services and agent yellow pages.

KQML was the first de-facto standard of ACLs (Agent Communication Languages) supporting a wide variety of interesting agent architectures. Therefore a small number of KQML performatives were introduced used by agents to describe the meta-data specifying the information requirements and capabilities and then to introduce a special class of agents called communication facilitators performing various useful communication services, like maintaining a registry of service names, and providing "matchmaking" between information providers and clients.

FIPA (Foundation for Intelligent Physical Agents [19]) the first standardization effort in agent technology with the remit of producing software standards for heterogeneous and interacting agents and agent-based systems across multiple vendors' platforms. The main focus of FIPA is the interface between the agents and the units of their environment, e.g. human beings, physical surrounding, and existing software. The technical parts address the following issues: agent management; agent communication language; agent software integration; agent management support for mobility; agent security management; ontology service; human/agent interaction and field trial specifications for the verification of the technical specifications. Moreover the FIPA Abstract Architecture specification, identifying architectural abstractions, has been published.

Java™ Agent Services [18] were established within the Java Community Process with the focus on the specification of a set of objects and service interfaces to support the deployment and operation of autonomous communicative agents based upon the FIPA Abstract Architecture defining how agents may register and discover each other, and how agents interact by exchanging messages based on the communicative acts. Only the interface for agent management is standardized via a Java API, but nothing is said about how an agent platform has to look like.

OMG (Object Management Group, [46]) has two activities within the area of agent technology: **MASIF** a mobile agent standard [40], deals with the necessity of interoperation between mobile agent platforms. It defines standardized interfaces for agent transfer, class transfer, and agent management. MASIF does not deal with agent communication. **Agent Platform Special Interest Group** (Agent PSIG) [1] has the identified tasks to extend the OMG Object Management Architecture (OMA) to better support agent technology, to create new OMG specifications or extend existing OMG specifications in the agent area, and to deal with agent modeling techniques like Agent UML to allow a better understanding of how to develop agent based applications. There is a liaison between FIPA and OMG in order to transfer FIPA specifications to the OMG.

Activities of the **World Wide Web Consortium** [60] include the **XML Protocol Working Group** and **P3P.** The goal of the former [59] is to automate negotiations and to standardize application-to-application messaging, especially in the business-to-business e-commerce area, and to satisfy requirements for a lightweight, simple network protocol for distributed applications. **P3P** [58] is a standard for privacy preferences allow servers and clients to automatically check whether the privacy preferences of a user are kept by the information service or the web trader. P3P is a standardized set of multiple-choice questions, covering all the major aspects of a Web site's privacy policies presenting a clear snapshot of how a site handles personal information about its users.

5.2 Enabling Technologies

The Extensible Markup Language (**XML**, [106]) is the universal format for structured documents and data on the Web. XML is a subset of SGML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML. XML is a technology for web applications. Thus XML simplifies business-to-business transactions on the web. It describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them. Markup encodes a description of the document's storage layout and logical structure. XML provides a mechanism to impose constraints on the storage layout and logical structure.

The Resource Description Framework (**RDF** [61]) supported by W3C is a foundation for processing meta-data; it provides interoperability between applications that exchange machine-understandable information on the web. RDF emphasizes facilities to enable automated processing of web resources. The RDF Data Model is described by means of resources, properties and their values. A specific resource together with one or more named properties plus the values of these properties is an RDF description (a collection of RDF statements). RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute-value pairs. RDF properties also represent relationships between resources. In addition to the RDF Data Model, the **RDF Schemas specification** (see [62]) provides a typing system for the resources and properties used in the RDF data. It defines concepts such as classes, subclasses, properties or sub-properties. It also allows expressing constraints. Both the RDF Data Model and RDF Schema propose XML as serialization syntax. In object oriented design terminology, resources correspond to objects and properties correspond to instance variables

At the level of underlying representations, **SMIL** [53], the Synchronized Multimedia Integration Language, is a major standard supported by W3C. SMIL is an XML application for synchronizing television-like audio and video with text and animation. It can be expected that this representation format will also have an important influence on future software solutions for multimedia information presentation within intelligent user interfaces.

SOAP [54] is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: (1) an envelope that defines a framework for describing what is in a message and how to process it; (2) a set of encoding rules for expressing instances of application-defined data types, and (3) a convention for representing remote procedure calls. SOAP messages are fundamentally one-way transmissions from a sender to a receiver, but SOAP messages are often combined to implement patterns such as request/response. A SOAP message does not have to contain a Document Type Declaration, but must contain Processing Instructions. Thus SOAP can be used for encoding messages between agents.

WSDL [67] is becoming a standard for describing Web Services. WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. It attempts to separate services, defined in abstract terms, from the concrete data formats and protocols used for implementation, and define bindings between the abstract description and its specific realization. There are four basic types of operations in

WSDL: a one-way, a (two-way) request-response, a (two-way) solicit-response and a (one-way) notification message.

The Web Services Conversation Language (**WSCL**) [66] provides a standard way to model the public processes of a service, thus enabling network services to participate in dynamic and complex inter-enterprise interactions. WSCL provides an XML schema for defining legal sequences of documents that web-services can exchange, comparable to interaction protocols. I.e., a conversation specification is defined to be a formal description of valid message type-based conversations that a service supports.

The goal of the **DAML** program and the **OIL** project [7] is to create technologies that enable software agents to dynamically identify and understand information sources, and to provide interoperability between agents in a semantic manner. DAML+OIL is a semantic markup language for Web resources (see e.g., [14]). Moreover it has an inference layer for ontologies, which combines the widely used modeling primitives from frame based languages with the formal semantics and reasoning services provided by description logic. Furthermore, OIL is properly grounded in W3C standards such as RDF/RDF-schema, XML/XML-Schema, Extensible Markup Language and Uniform Resource Identifiers and extends these languages with richer modeling primitives. The partners of DAML-OIL have submitted there technology to FIPA and more over to world-wide web consortium (W3C). In particular because of the liaison with W3C it can be assumed that DAML-OIL will also have a major impact in the eBusiness area.

BizTalk [5] is a community of standards users mainly driven by Microsoft, with the goal of a consistent adoption of XML to enable electronic commerce. In particular it defines the BizTalk framework, a set of guidelines for how to publish schemas in XML and how to use XML messages to easily integrate software programs together in order to build rich new solutions. The BizTalk Framework assumes that applications are distinct entities, and application integration takes place using a loosely coupled approach to exchange messages. The message flow between two or more applications integrates applications at the business-process level by defining a loosely coupled communication process that is based on requests.

xCBL (XML Common Business Library, [69]), is a set of XML business components and a document framework that allows the creation of robust, reusable, XML documents to facilitate global trading. xCBL is based on well-known standards like Electronic Data Interchange (EDI), RosettaNet, and Open Buying on the Internet (OBI). They can be obtained from public repositories like XML.org and BizTalk.org. Based on these building blocks companies can build their own documents out of the component library. The idea is that this promotes interoperability between applications and allows corporate parties to easily exchange documents across multiple e-marketplaces, giving global access to buyers, suppliers, and providers of business services. xCBL is made available as a set of SOX schemas (SOX is the schema language for Object-Oriented XML), as a single XML DTD, and in XDR schema forms. xCBL will be able to support all essential documents and transactions for global e-commerce including multi-company supply chain automation, direct and indirect procurement, planning, auctions, and invoicing and payment in an international multi-currency environment.

The Universal Description, Discovery and Integration (**UDDI** [57]) standard (registry) is an industry initiative (IBM, Microsoft, Ariba) creating a platform-inde-

pendent, open framework for describing services, discovering businesses, and integrating business services using the Internet. UDDI is the first cross-industry effort driven by platform and software providers, marketplace operators and eBusiness leaders. UDDI could be regarded as an upper layer in an emerging stack of enabling rich Web Services based on TCP/IP, HTTP, XML, and SOAP to create a uniform service description format and service discovery protocol. The UDDI specifications define a way to publish and discover information about Web Services, to define the interaction with each other over the Internet and to share information in a global registry. This UDDI business registry is simply spoken an XML file storing white pages, yellow pages and green pages describing a business entity and its Web Services. In the focus of UDDI is sharing business information, making it easier to publish preferred means of doing business, find trading partners, and interoperate with these trading partners over the Internet.

The **eCo** (electronic COmmerce, [12]) specification is an architectural framework for interoperability among XML-based application standards and key electronic commerce environments that enables businesses to discover each other on the World Wide Web and determine how they can do business. Following [104] future eCo market places will feature multiple sellers or sources of products and services. Moreover the buyers want to compare these products, their prices and alternatives in order to make the best purchase decision. Sources will include multiple types of product content from multiple seller sources. Agent technology can help to set up such electronic commerce environment or at least can help to add functionality to such platforms, supporting matchmaking facilities or negotiation protocols and strategies for interacting on behalf of the user.

RosettaNet [50] works to create and deploy industry-wide, open eBusiness process standards, in particular the common business processes between trading partners in high tech supply chain. It allows manufacturers, suppliers and end-users to exchange business documents across the entire supply chain based on a comprehensive set of XML based standard business document schemas and data dictionaries. Interoperable protocols for the networked applications are specified that execute the business processes. RosettaNet standards can be divided into three broad groups of data format, business process and protocol specifications. They include the (1) *Business Dictionary* defining the properties used in basic business activities between trading partners; (2) *Technical Dictionaries* defining properties for products, components/devices and services; (3) *Implementation Framework* (RNIF) providing the fundamental prerequisites to execute business processes between the trading partners; (4) *Partner Interface Processes*™ (PIPs™) defining the sequence of steps required to execute a business processes between supply-chain partners, including e.g. purchase order management and distribution of new product information. Structure and content format of the exchanged business documents is specified based on XML DTDs and the time, security, authentication and performance constraints on these interactions.

ebXML [11] has the goal to provide an open XML based infrastructure enabling the use of electronic business information in an interoperable, secure and consistent manner, in particular from a workflow perspective. The main components ebXML deals with (1) Registries and Repositories (in particular an ebXML Registry can be published to UDDI). (2) Business Processes; (3) Collaboration Protocol Profiles;

Business Messages and Business Service Interfaces; (4) Core Library / Core Components, and (5) Collaboration Protocol Agreement.

5.3 Integration Technologies

Crosswolds or WebMethods as any other business integration platform deal with the integration of back-office legacy software, e.g., SAP or SIEBEL, with application interfaces independent of the underlying legacy system. A special type of platforms are e/m Commerce platforms allowing the integration of back-office solutions for electronic commerce.

5.3.1 eBusiness Integration Platforms

eBusiness integration (EBI) platforms define a general category of platform solutions that enable companies to link internal applications and processes with applications and processes of external partners, suppliers and customers. EBI platforms include EAI (Enterprise Application Integration) products and the newer B2B (Business to Business) integration platforms.

Integration platforms bring together the functions of standard application packages, new business logic (components), parts of legacy applications and required data to meet the needs of business processes. Integration platforms often use non-invasive wrapping mechanisms or standard component connectors from various vendors (SAP R/3, Oracle, IBM/CICS, etc).

The inclusion of both EAI and B2B in this category reflects the growing convergence of EAI and B2B integration worlds. Most integration solutions come with a library of pre-built or pre-configured adapters for connecting with packaged applications and other environments. Integration solutions include a set of GUI-based tools for defining and managing the various aspects of the integration process. These tools can be used for defining business process and how the business process integrates with existing applications and external partner applications and processes. There are the following main points which are supported by most of these systems: transactional real-time component integration, a queued messaging model, a publish-and-subscribe messaging or component model and bulk data movement (often as database replication). Usually business process management is support both for internal processes (EAI) and external processes (B2B). Moreover the solution support industry-standard documents, e.g., Open Applications Group (OAG) business object documents (BODs) and process definitions, e.g., RosettaNet (see Sect. 5.2).

An e/mBusiness platform – a specific kind of eBusiness integration platforms - is a software server offering basic selling features on a foundation designed for customizability. Commerce sites must tie into other systems like inventory management, partners' extranets, and eMarketplace software. To do this, a commerce platform, like agent technology used in eBusiness, needs to support integration standards like XML, EDI, .NET, and Java 2 Enterprise Edition (J2EE).

5.3.2 Application Server Platforms

The purpose of application servers is to provide a robust middleware infrastructure to develop and run eBusiness applications. The most important characteristics are

scalability, load balancing, persistency, recovery, and fail-over. Further, the application server provides middleware services (APIs) for communication, transaction control, security and component management. However they are no direct integration technologies, but agents could run on such application server platforms to satisfy the above mentioned requirements.

5.3.3 WebServices

Web services are a new way of building Web application. Web Services can be understood as components in the Internet; they are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. They are mainly based on XML and HTTP with the specific notions of WSDL and UDDI for describing the services and providing the white, yellow and green pages; and SOAP on the protocol level for a Remote Procedure Call function over XML.

Web services perform functions ranging from simple requests to complicated business processes, like in the BizTalk environment. Like a web page a web service can be used by a human user but in the same way by another web service or program, i.e. once a Web service is deployed, it can be discovered - via WSDL and UDDI - and the deployed service can be invoked using SOAP. Despite the name "service", a web service is not restricted to be a server, it can act as well as a client to other web services. In this way widely distributed applications can be built where the services locate each other dynamically.

With the increase of the bandwidth and the decrease of the costs in the last few years more dynamic content, the pervasiveness and diversity of computing devices up to mobile devices like mobile phones or PDAs with multi-access make the need for a glue more important.

Viewed from a software architecture perspective (see Sect. 2.2), the web service is a veneer for programmatic access to a service which is then implemented by other kinds of middleware. Access consists of service-agnostic request handling (a listener) and a facade that exposes the operations supported by the business logic. The logic itself is implemented by a traditional middleware platform.

Web Services can be seen as the next step of development towards agent based services, since WebServices support only the infrastructure and syntactical level of communication, whereas agent technology adds additional functionality like negotiation or communication on a semantic level. Agents can be seen as individual peers which communicate with other peers (agents). It can be assumed that agent technology and peer-to-peer approaches are techniques that can benefit from each other and in the long run both technologies will come together.

5.4 Semantic Web

The Semantic Web will bring structure to the meaningful content of Web pages. The Semantic Web is an extension of the WWW, in which information is given well-defined meaning by providing metadata, for better enabling computers and people to work in cooperation. The first steps in weaving the Semantic Web into the structure of the existing Web are already defined. In the near future, these developments will usher in significant new functionality as machines become much better able to process and "understand" the data that they merely display at present.

For the semantic web to be operable computers must have access to structured collections of information and sets of inference rules that they can use to conduct automated reasoning. Two important technologies for developing the Semantic Web are already in place: eXtensible Markup Language (XML) and the Resource Description Framework (RDF). Another knowledgeable technique is the DAML+OIL approach. Two important question to be resolved is who will define the ontologies required, and, probably more seriously, how the billions of existing pages of web content can be provided with the metadata required for processing by agents in the Semantic Web.

5.5 Matchmaking and Agent-Based Marketplaces

Kuokka and Harada [33] considered matchmaking in the context of emerging information integration technologies, where potential providers and requesters send messages describing their capabilities and needs of information (or goods). They presented two matchmakers: COINS (COmmon INterest Seeker), which is based on free text matchmaking using a distance measure from information retrieval (Salton [51]), and SHADE (SHared DEpendency Engineering), which uses a subset of KIF ([26]) and a structured logic text representation called MAX ([32]). While COINS aimed at e-commerce, SHADE aimed at the engineering domain.

Complementing the theoretical work in ([9],[10]), Sycara and co-workers addressed the matchmaking problem in practice. They developed and implemented the LARKS matchmaker (LAnguage for Advertisement and Request for Knowledge Sharing, see [56]). In LARKS, the matchmaking process runs through three major steps: (1) Context matching, (2) syntactical matching, and (3) semantic matching. Step 2 is divided into a comparison of profiles, a similarity matching, and a signature matching. Compared to previous approaches, LARKS provides higher expressiveness for service descriptions.

In the context of electronic auctions, we refer to Sandholm's theoretical results on the properties of different market protocols and to a number of prototype systems constructed based on these results (see e.g., [52]). Weinstein and Birmingham ([64]) introduce a service classification agent which has meta-knowledge and access to nested ontologies. This agent dynamically generates unique agent and auction descriptions which classify an agent's services and auction subjects, respectively. A requester obtains from it the name of the best auction to its needs. In [49], Reeves et al. describe ContractBot, an interesting approach of automating negotiation based on declarative descriptions of contracts. ContractBot can be viewed as a further development of AuctionBot [68].

In [42], Müller and Pischel present a case study of employing an agent-based approach to a real-world solution in the area of digital libraries and web-publishing.

In IMPACT ([2]), so called Yellow Pages Servers play the role of matchmaker agents. Offers and requests are described in a simple data structure which represents a service by a verb and one or two nouns (e.g., *sell:car*, *create:plan(flight)*). The matchmaking process computes the similarity of descriptions from shortest paths in directed acyclic graphs that are built over the sets of verbs and nouns, respectively, where edges have weights reflecting their distance.

6 Conclusions and Outlook

The contribution of this paper is twofold: Firstly, we outlined what we believe are the key challenges in today's eBusiness. Secondly, we investigated how agent technology can help tackling some of these challenges. We demonstrated possible approaches in using agent technology for eBusiness solutions as well as main advantages in building agent based eBusiness systems. The paper shows that agent technology is well suited to develop highly dynamic, generic and intelligent integration frameworks for eBusiness based on standard vertical and horizontal services on top of existing standard platforms.

We believe that agent technology will play an important role in the development of the next generation eBusiness systems over the next few years. Agent technology first of all plays a key role in combining the existing heterogeneous eBusiness solutions, adding smart functionality and automating standard processes. This becomes also clear if we look at the main abilities which agents provide in contrast to other technologies. The four main abilities are the following:

- Enriched, higher level communication (agent communication languages, based on existing transport encoding and underlying networking protocols, co-ordination of tasks, collaboration based on semantics and ontology's)
- Enabling more intelligence service provision, and process management e.g. by personalization and integration of different services to value-added services (service wrapping, brokering, matchmaking, negotiation, auctioning, preference modeling, adaptive behavior by learning mechanisms)
- Dealing with the enlarging amount of information and functions (agent mobility, intelligent filtering, personalization, presentation)
- Allowing self-organizing of processes (autonomous, flexible and pro-active behavior by planning, scheduling, and learning functionality)

One point this paper makes is to show that agent technology has much to offer to next-generation eBusiness solutions. We believe that the main benefit of agents in eBusiness will be reached in the following application areas:

- Better customer relation by attractive user interfaces and personalized presentations
- Effective and fast assignment of supply and demand in electronic market places by intelligent matchmaking
- Optimization of processes by dynamic negotiations for configuration and contract management
- Integration of heterogeneous software systems by wrapping legacy software using agent standards, e.g. the FIPA standard, for co-ordination and co-operation between software agents

However, we should end with a word of caution. For agents to satisfy the expectations there are some important preconditions. Firstly, the research community needs to establish a focus on pragmatic solutions that build on existing standards. Secondly, existing solutions and technologies need to be used efficiently and enhanced incrementally instead of re-inventing the wheel. Thirdly, while agents are an important enabling technology, they do not liberate software engineers and system

developers from making a careful requirement analysis and design of processes and systems; Fourthly, for some areas, such as advanced supply-chain management, automated negotiations and collaborations, new architectures and methods will be required to achieve the necessary level of scalability and flexibility. Self-organizing manufacturing and supply-chain whose nodes consist of autonomous collaborating agents seem to be appropriate building blocks for these solutions.

References

[1] APSIG: http://www.objs.com/agent/index.html

[2] Arisha, K., T. Eiter, S. Kraus, F.Ozcan, R. Ross, and V.S. Subrahmanian (1999, March/April). IMPACT: A Platform for Collaborating Agents. IEEE Intelligent Systems 14(2), 64-72

[3] Bauer, B.; Berger, M.: Agent-Based Personal Travel Assistance, Proceedings of International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce (MAMA 2000), Wollongong, Australia, 2000

[4] Berger, M, Bauer, B, Watzke, M.: Towards an Agent based Infrastructure for Distributed Virtual Organizations, Proceeding 3rd International WetICE Workshop on Web-Based Infrastructure and Coordination - Architectures for Collaborative Enterprises, MIT, Cambridge, 2001.

[5] BizTalk: http://www.biztalk.org

[6] Bussmann, S.: A Multi-Agent Approach to Dynamic, Adaptive Scheduling of Material Flow. MAAMAW 1994: 191-205.

[7] DAML(-OIL): http://www.daml.org

[8] Daniels, M.: Integrating Simulation Technologies with Swarm paper delivered to the "Agent Simulation: Applications, Models and Tools" conference held at University of Chicago in October of 1999, URL http://www.santafe.edu/~mgd/anl/anlchicago.html

[9] Decker, K., K. Sycara, and M. Williamson (1996, December). Matchmaking and Brokering. In International Conference on Multi-Agent Systems (ICMAS96)

[10] Decker, K., K. Sycara, and M. Williamson (1997, August). Middle-agents for the internet. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), pp.578-583.

[11] ebXML http://www.ebxml.org

[12] eCommerce, http://eco.commerce.net

[13] Etzioni, O. Moving Up the Information Food Chain: Deploying Softbots on the World Wide Web. Proceedings of AAAI-96 (Abstract of Invited Talk), 1996.

[14] Fensel D., I. Horrocks, F. Van Harmelen, S. Decker, M. Erdmann and M. Klein: OIL in a nutshell In: Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference (EKAW-2000), R. Dieng et al. (eds.), Lecture Notes in Artificial Intelligence, LNAI, Springer-Verlag, October 2000

[15] Finin, T. and Fritzson, R.: KQML - a language and protocol for knowledge and information exchange, In Proceedings of the 13th Intl. Distributed Artificial Intelligence Workshop, LNAI 890, Seatle, WA, USA, 1994]

[16] FIPA 97 specification: http://www.fipa.org/repository/fipa97.html

[17] FIPA 98 specification: http://www.fipa.org/repository/fipa98.html

[18] FIPA AGENT: http://www.java-agent.org

[19] FIPA: http://www.fipa.org

[20] Fischer Wirtschaftslexikon.

[21] Fischer, K.: Holonic Multiagent Systems - Theory and Applications. In Proceedings Portuguese Conference on Artificial Intelligence (EPIA) 1999: 34-48.

[22] Forrester Research: http://www.forrester.com

[23] Frey, R.: Agent technology is making the transition from research labs into industry, invited talk, PAAM 2000, 2000.
[24] Gamma, E., Helm, R., Johnson R., Vlissides J. Design Patterns, Addison Wesley, 1997.
[25] GartnerGroup: http://www.gartner.com
[26] Genesereth, M.R. and R.E. Fikes (1992, June). Knowledge Interchange Format, Version 3.0 Reference Manual. Technical Report Logic-92-1, Computer Science Department, Stanford University. http://www-ksl.stanford.edu/knowledge-sharing/papers/kif.ps
[27] Gerber Ch., B. Bauer, D. Steiner: Resource Adaptation for a Scalable Agent Society, (chapter in) Software Agents for Future Communication Systems, ed. Hayzelden, Bigham, Springer, 1999
[28] Guttman R., A. Moukas, and P. Maes. "Agent-mediated Electronic Commerce: A Survey." Knowledge Engineering Review, Vol. 13:3, June 1998.
[29] Ivezic N., Tom Potok, and Line Pouchard. Multiagent Transitional Operations, Autonomous Agents 2000
[30] JADE: http://sharon.cselt.it/projects/jade/
[31] Jennings N. R., K. Sycara, and M.J.Wooldridge. A Roadmap of Agent Research and Development. Journal of Autonomous Agents and Multi-Agent Systems. 1(1), pages 7-36. July 1998.
[32] Kuokka, D. (1990). The Deliberative Integration of Planning, Execution, and Learning. Ph.D. thesis, School of Computer Science, Carnegie Mellon University.
[33] Kuokka, D. and L. Harada (1996). Integration information via matchmaking. Journal of Intelligent Information Systems 6(2/3), 261-279.
[34] LEAP: http://leap.crm-paris.com
[35] Leonet: http://www.nm.informatik.uni-muenchen.de/~leonet
[36] Leonet: http://www.nm.informatik.uni-muenchen.de/~leonet
[37] LivingSystems: www.livingsystems.com
[38] LostWax: www.lostwax.com
[39] Maes, P.: Modeling Adaptive Autonomous Agents. In: Langton, C. (ed.): Artificial Life Journal, Vol. 1, No. 182, MIT Press, pp. 135-162, 1994.
[40] MASIF: http://www.omg.org/cgi-bin/doc?orbos/97-10-05
[41] MoTiV: http://www.motiv.de
[42] Müller, J. P. and M. Pischel. Doing business in the information marketplace: a case study. In Proceedings of the 3rd Intl. Conference on Autonomous Agents (Agents-1999), ACM Press, 1999.
[43] Müller, J. P. The design of intelligent agents. Lecture Notes of Artificial Intelligence, Vol. 1077. Springer-Verlag, 1997.
[44] Odell, J. ed., Agent Technology, OMG, green paper produced by the OMG Agent Working Group, 2000
[45] OIL http://www.ontoknowledge.org/oil
[46] OMG: http://www.omg.org
[47] Parunak, H. Van Dyke: A Practitioners Review of Industrial Agent Applications. Autonomous Agents and Multi-Agent Systems 3(4): 389-407 (2000)
[48] Raymond, E.S. ”The Cathedral and the Bazaar – Musings on Linux and Open Source by an accidental Revolutionary”; O’Reilly & Associates, Inc., Sebastopol, CA, 1999.
[49] Reeves, D.M., Wellman, M.P., Grosof B.N. Automated negotiation from declarative contract descriptions. In Proceedings of the 5th Intl. Conference on Autonomous Agents, pp. 51-58. ACM Press, 2001.
[50] RosettaNet http://www.rosettanet.org
[51] Salton, G. (1989). Automatic Text Processing. Addison-Wesley. (ISBN 0-201-12227-8)
[52] Sandholm, T. eMediator: a next generation electronic commerce server. In Proceedings of the 4th Intl. Conference on Autonomous Agents (Agents-2000), pp.341-348, ACM Press, 2000.
[53] SMIL: http://www.w3.org/TR/REC-smil/
[54] SOAP: http://www.w3.org/TR/SOAP/

[55] SWARM http://www.swarm.org
[56] Sycara, K., J. Lu, M. Klusch, and S. Widoff (1999). Dynamic service matchmaking among agents in open information environments. ACM SIGMOD Record 28 (1), Special Issue on Semantic Interoperability in Global Information Systems, 47-53.
[57] UDDI http://www.uddi.org
[58] W3C P3P: www.w3c.org/P3P/
[59] W3C XML Protocol: http://www.w3.org/2000/xp
[60] W3C: www.w3c.org
[61] W3CRDF: http://www.w3c.org/Metadata/
[62] W3CRDFSchema: http://www.w3.org/TR/2000/CR-rdf-schema-20000327/
[63] Wayflow: http://www.wayflow.de
[64] Weinstein, P. and W. Birmingham (1997). Service classification in a proto-organic society of agents. In Proceedings of the IJCAI-97 Workshop on Artificial Intelligence in Digital Libraries.
[65] Weiss G., Multiagent Systems - A modern approach to Distributed Artificial Intelligence. Cambridge: MIT Press, 1999.
[66] WSCL: http://www.hp.com/go/e-speak
[67] WSDL: http://www.w3.org/TR/wsdl
[68] Wurman P.R., Wellman M.P., Walsh W.E. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In Proc. Of the 2nd Intl. Conference on Autonomous Agents, pp.301-308, ACM Press, 1998.
[69] XCBL: http://www.xcbl.org
[70] XML: http://www.w3.org/XML/