ELSEVIER SCIENCE B.V. [DTD 4.1.0]
JOURNAL DATAK ARTICLE No. 516
PAGES 01-36 DISPATCH 1 November 2000 PROD. TYPE: FROM Disk
DATAK 516 A

ELSEVIER

DATA & KNOWLEDGE ENGINEERING

Data & Knowledge Engineering 000 (2000) 000–000

www.elsevier.com/locate/datak

Information agent technology for the Internet: a survey

Matthias Klusch

German Research Center for Artificial Intelligence, Deduction and Multi-Agent Systems Lab, Stuhlsatzenhausweg 3, 66123 Saarbruecken, Germany

Received 3 August 2000; received in revised form 3 August 2000; accepted 3 August 2000

Abstract

The vast amount of heterogeneous information sources available on the Internet demands advanced solutions for acquiring, mediating, and maintaining relevant information for the common user. Intelligent information agents are autonomous computational software entities that are especially meant to (1) provide pro-active resource discovery, (2) resolve information impedance of information consumers and providers, and (3) offer value-added information services and products. These agents are supposed to cope with the difficulties associated with the information overload of the user, preferably just in time.

Based on a systematic classification of intelligent information agents, this paper presents an overview of the basic key enabling technologies needed to build such agents, and respective examples of information agent systems currently deployed on the Internet. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Intelligent agents; Semantic information brokering; Personal assistants; Cooperative information systems; Agent-mediated electronic business

1. Introduction

In the setting of the hypertext-oriented information service of the Internet, the worldwide Web, a typical user faces mundane, repetitive tasks such as browsing, filtering, and searching for relevant information. Currently, the global information search space on the Internet consists of an estimated 320 million HTML pages on the Web [75]. In the year 2001, probably more than 225 million users will be connected to the Internet, facing around 1000 million indexed Web pages. Large volumes of volatile, redundant, semi- or unstructured heterogeneous data are additionally available in connected legacy databases, file systems, multimedia database systems and software applications. This includes, for example, bibliographic entries, images, speech text, and video data.

The impacts of data, system, and semantic heterogeneity on the information overload of the user are manifold, especially due to potentially significant differences in data modeling, data structures, content representations using ontologies and vocabularies, query languages and op-

E-mail address: klusch@dfki.de (M. Klusch).

PII: S0169-023X(00)00049-5

erations to retrieve, extract, and analyze information in the appropriate context. The impacts of increasing globalization on the information overload encompass the tasks for the user to determine and keep track of relevant information sources, to efficiently deal with different levels of abstractions of information modeling at sources, and to combine partially relevant information from potentially billions of sources.

Other challenges include, for example, how to cope with the problems of the cross-social, cross-cultural and multi-lingual cyberspace. Regarding efficiency in time, the impact of ongoing efforts to increase the transfer rates in Internet-2 (next generation Internet) to 9.6 Gbit/s on the situation of the common user of the public Internet in the next couple of years remains unclear.

Information agent technology (IAT) [67] emerged as a major part of the more general intelligent software agent technology [143,153] around seven years ago as a response to the challenges mentioned above, from both the technological and human user perspective. As such, IAT is an inherently interdisciplinary technology encompassing approaches, methods and tools from different research disciplines and sub-fields such as artificial intelligence, advanced databases and knowledge base systems, distributed information systems, information retrieval, and human–computer interaction.

The driving idea of IAT is the development and effective, efficient utilization of autonomous computational software entities, called intelligent information agents, which have access to multiple, heterogeneous and geographically distributed information sources as in the Internet or corporate Intranets. The main task of such agents is to perform pro-active searches for, to maintain, and to mediate relevant information on behalf of their users or other agents. This includes skills such as retrieving, analyzing, manipulating, and fusing heterogeneous information as well as visualizing and guiding the user through the available, individual information space.

1.1. Web indices and search bots

The most prominent solutions for finding relevant information on the Internet include monolithic Web indices such as Gopher and Harvest [179] as well as search engines and (meta-) search bots [14].

Search bots like AltaVista, Lycos, InfoSeek, Excite or HotBot use basic information retrieval techniques [48,80,140] to automatically gather information from indexed Web pages, maintain and periodically update their own index database, and provide a rating-based one-time query answering mechanism to the user. Each bot has a proprietary method for recursively traversing hyperlinks starting from a given initial list of URLs, and ranking retrieved documents. The information quality of the result relies not only on the ontological organization, size, and methods of access to the internal index but also on the expressiveness of the query language the user is enforced to use to formulate inquiries to the bot.

Among others, the main limitations of search bots are that they do not behave pro-actively due to their one-shot answering mechanism providing a rather simple query language in terms of regular expressions of phrases and keywords. Each search bot has its own idiosyncratic way the user has to deal with, and finally, most of the prominent search bots offer a maximum of coverage of just about 30% of the Web, or less, including up to 5% invalid or broken links [75,188].

Meta-search bots such as MetaCrawler, SavvySearch, Ahoy!, Remora or WebMate execute a given query concurrently over a variety of given search bots, and merge and present the results in a homogeneous, ranking-based view to the user. That allows the user to enlarge the individual search space and may increase the hit rate for some queries.

According to [176], search bots like Excite, HotBot, and Lycos use certain page importance metrics for ranking retrieved Web pages. These include:

- the backlink count measuring the number of links (in-links) to a page *p* that appear over the entire Web. This implies that the more the pages linked to *p*, the greater *p*'s importance, thereby treating all links equally, and pushing out equally important, small fields by the sheer volume of links.
- the page-rank backlink metric recursively measuring the weighted sum of the in-links to a page *p*, thereby exaggerating the above problem in that the more the pages linked to *p* themselves having a high backlink count, the greater *p*'s importance.
- the location metric measuring a page's importance as a function of location, not its content such as preferring URLs with fewer slashes, ending with '.com', or containing the string 'home'.

In any case, portals and sites of (meta-)search bots supported by knowledgeable professionals determine the threshold of relevance at the expense of equally relevant but small, less supported sites. Other reasons for potential page exclusion include (1) the robot exclusion standard limiting each searchbots' access to individual pages at a given site, (2) the common breadth-first search which tends to push search bots to visit more sites but to index only a fraction of each, and (3) the ongoing web page ranking warfare by, e.g., relevancy spamming, purchasing of higher page ranks, tiny keyword text on Web pages, and automatic displaying of banners whenever a certain input is made by the user to the search bot.

The disclosure of algorithms that govern search bots in searching, indexing, and ranking might be one way to avoid ranking warfare. Another option would be to provide common users with easy-to-use software libraries and graphical agent editors to build customized information agents for accomplishing their everyday business tasks thereby treating the Web rather as a public good.

However, neither Web indices nor search bots overcome the hard problems caused by the heterogeneity of systems, data syntax, structure, and semantics in a sufficient way. Methods to solve these types of heterogeneities concern intense data, metadata, and semantic information brokering [180], which goes beyond the capability of any search bot deployed so far on the Web. In the next section, we introduce the notion of an information agent, propose a classification of different types of such agents, and summarize the corresponding basic skills an information agent is supposed to possess.

1.2. Information agents: definition, classification, and basic skills

Information agents are a special kind of so-called intelligent software agents. Software agent technology originating from distributed artificial intelligence is inherently interdisciplinary. Thus, the notion of an agency is quite broadly used in literature; it might rather be seen as a tool for analyzing systems, not an absolute characterization that divides the world into agents and non-agents. However, *intelligent agents* are commonly assumed to exhibit autonomous behavior determined by their:

- pro-activeness, means taking the initiative to satisfy given design objectives and exhibit goal-directed behavior,
- reactive or deliberative actions, means perceiving the environment and timely change management to meet given design objectives, and

• social interaction in groups with other agents and/or human users when needed.

It depends on the concrete application domain and view on the potential solution for a particular problem what an intelligent agent is supposed to do in practice. For a more comprehensive and introductory literature on intelligent agents we refer the reader to [143,152]. Today, agents are deployed in different settings, such as industrial control, Internet searching, personal assistance, network management, games, software distribution, and many others. Agent technology is quite on its way to produce mature standards concerning software agent architectures and applications such as OMG MAF [82] and FIPA specification [38]. Further, the European network of excellence for agent-based computing (AgentLink) [1] set up in 1998, international workshops, and conferences on the subject, like ATAL [92], CIA [71], Autonomous Agents, PAAM [98], and ICMAS [55], strongly push software agent technology since its public breakthrough around five years ago.

Intelligent agents for the Internet are commonly called information agents. But what exactly is an information agent? We define an *information agent* as an autonomous, computational software entity (an intelligent agent) that has access to one or more, heterogeneous and geographically distributed information sources, and which pro-actively acquires, mediates, and maintains relevant information on behalf of users or other agents, preferably just-in-time. Thus, an information agent is supposed to satisfy one or more of the following requirements.

- *Information acquisition and management*. It is capable of providing transparent access to one or many different information sources. Furthermore, it retrieves, extracts, analyzes, and filters data, monitors sources, and updates relevant information on behalf of its users or other agents. In general, the acquisition of information encompasses a broad range of scenarios including advanced information retrieval in databases and also the purchase of relevant information from providers on electronic marketplaces.
- *Information synthesis and presentation*. The agent is able to fuse heterogeneous data and to provide unified, multi-dimensional views on relevant information to the user.
- *Intelligent user assistance*. The agent can dynamically adapt to changes in user preferences, the information, and network environment as well. It provides intelligent, interactive assistance for common users supporting their information-based business on the Internet. In this context, the utilization of intelligent user interfaces like believable, life-like characters can significantly increase not only the awareness of the user on its personal information agent but the way information is interactively dealt with.

Many (systems of) information agents have been developed or are currently under development in academic and commercial research labs, but they still have to wait to make it out to the real world of Internet users broadly. However, the ambitious and pretentious goal to satisfy all of the requirements mentioned above appears to be not very far from being accomplished in the next ten years.

1.2.1. Classification of information agents

Information agents may be categorized into several different classes according to one or more of the following features [67].

1. *Non-cooperative* or *cooperative* information agents, depending on the ability of the agents to cooperate with each other for the execution of their tasks. Several protocols and methods are available for achieving cooperation among autonomous information agents in different scenarios, like hierarchical task delegation, contracting, and decentralized negotiation.

2. *Adaptive* information agents are able to adapt themselves to changes in networks and information environments. Examples of such kind of agents are learning personal assistants on the Web.
3. *Rational* information agents are utilitarian in an economic sense. They act and may even collaborate together to increase their own benefits. One main application domain of such kind of agents is automated trading and electronic commerce on the Internet. Examples include the variety of shop bots, and systems for agent-mediated auctions on the Web.
4. *Mobile* information agents are able to travel autonomously through the Internet. Such agents may enable, e.g., dynamic load balancing in large-scale networks, reduction of data transfer among information servers, applications, and migration of small business logic within medium-range corporate intranets on demand.

According to the definition and classification of information agents we can differentiate between communication, knowledge, collaboration, and rather low-level task skills as depicted in Fig. 1. In this figure, the corresponding key enabling technologies are listed below each of the different types of skills. Communication skills of an information agent comprehend either communication with information systems and databases, human users, or other agents. In the latter case, the use of an agent communication language has to be considered on top of, for example, middleware platforms or specific APIs.

The representation and processing of ontological knowledge and metadata, profiles and natural language input, translation of data formats as well as the application of machine learning techniques enable an information agent to acquire and maintain knowledge on itself and its environment. High-level collaboration of an information agent with other agents can rely, for example, on service brokering, matchmaking, negotiation, and collaborative (social) filtering, whereas collaborating with its human users mainly corresponds to the application of techniques stemming from human–computer interaction and affective computing.

1.2.2. Some approaches for building information agents

The most prevalent approaches for building information agents are the following.

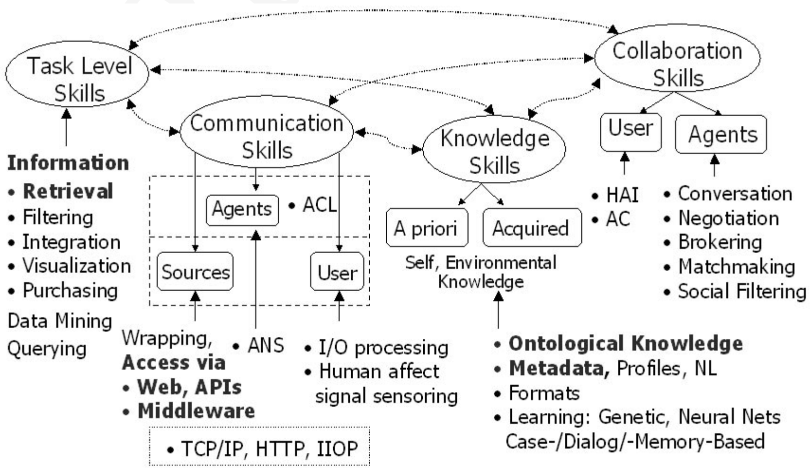

Fig. 1. Basic skills of an information agent.

1. *User programming*. An information agent is programmed explicitly by its user from scratch, for example by using a collection of user-programmed rules for processing information related to a particular task. The main problem with this approach is that it requires too much insight, understanding and effort from the user. For example, a user has to recognize opportunity for employing an agent, take initiative to create it, endow it with explicit knowledge and maintain the underlying rules over time.
2. *Knowledge engineering*. An agent is endowed a-priori with a great deal of domain-specific background knowledge about the application and the user. Related problems with this approach are that it requires substantial efforts from the knowledge engineer, the agent is highly domain-specific and its knowledge is relatively fixed; thus the agent is hardly adaptable to different application domains.
3. *Machine learning*. An agent automatically acquires the knowledge it needs to assist the user by applying appropriate methods from machine learning. Certainly, the condition to be satisfied for this approach to be applicable is that the use of the application involves highly repetitive and different behavior from different users. Adaptation of the agent to individual user preferences and habits is a clear advantage in that it, for example, offers customized results and requires even less work from the user and application developer, but it also raises the issue of trust between the user and his/her learning information agent.

In the remainder of this paper, we provide the reader with an overview of the basic key enabling technologies of IAT needed to build intelligent information agents for the Internet, and point to respective examples of systems of such agents which have been currently developed in the community. The overview is structured in accordance with the classification and basic skills of information agents mentioned earlier. For a more in-depth and technical discussion of individual techniques, methods, and systems we refer to the corresponding, given references. Finally, we will provide an outlook to possible future perspectives of information agent technology for the next decade.

2. Non-cooperative and collaborating information agents

2.1. Basic enabling technologies

According to the basic skills of an information agent the main key supporting technologies concern the following issues independent of whether they collaborate with other agents or not.

- *Access to heterogeneous distributed information systems and resources on the Internet*. This includes standardized middleware platforms as well as efficient techniques for client or server-sided Web-based applications.
- *Retrieving and filtering relevant data from any kind of digital medium worldwide*, such as content-based, multimedia and cross-language information retrieval [47].
- *Metadata management and ontological knowledge processing* facilitate the reconciliation of semantic heterogeneity of retrieved data and information stemming from multiple, heterogeneous sources.
- *Visualization of information*, for example, by utilizing the standardized virtual reality modeling language VRML97 (or its future successor X3D).

2.1.1. Access to heterogeneous distributed information sources

An information agent can be implemented as, invoked by or embedded in any client or server side Web-based application. Relevant techniques for implementing such applications are platform-independent signed Java applets, scripts, platform-dependent ActiveX and CGI, FastCGI, and Java servlets, respectively. Access to relational databases may be realized using generic application programming interfaces such as Java database connectivity (JDBC) with embedded query language SQLJ, and Microsoft's open database connectivity (ODBC). The same goes with the generic open knowledge base connectivity (OKBC) API for an agent to access multiple heterogeneous knowledge bases.

For transparent access in distributed computing environments frameworks such as Microsoft's distributed component object model (DCOM) [27], the Object Management Group's common object request broker architecture and Internet inter-ORB protocol (CORBA/IIOP) [23], and to a lesser extent, Sun's Java remote method invocation (RMI) [111], and JINI have been developed. Each of these frameworks provides an interface description language (IDL) and services that allow distributed objects to be defined, located and (statically or dynamically) invoked. All platforms enhance traditional RPC-based client/server architectures by allowing relatively transparent distribution of service functionality. For example, in JINI, virtual destinations of connections for retrieving services around the network that are available to applications at that time are provided as proxies by a lookup service. These proxies are taken in using the nearest lookup facility and communicating with them as actual destinations, whereas CORBA realizes communication between heterogeneous objects (sources, agents) as client or server side proxies by the transformation of messages to the standard interface in IDL provided by an object request broker. Different ORBs implemented by different providers can communicate via the Internet inter-ORB protocol.

In summary, the primary benefit of using middleware platforms when developing an information agent is to encapsulate the heterogeneity of legacy systems and applications within standard, interoperable wrappers the agent can communicate with. However, any common knowledge representation, behavioral specification of distributed software components, and meaningful interaction among agents goes beyond these frameworks. Besides, asynchronous multicast group communication between a set of agents or resources is difficult to realize using these platforms.

At a lower level of communication between agents and systems the TCP/IP and HTTP protocols have become de-facto standards for the transmission of data packets over networks. Asymmetrical digital subscriber line (ADSL) and high-rate digital subscriber line (HDSL) technology, such as standardized G.lite or HDSL-2, hold tremendous promise for future high-speed Internet access; they current support, for example, up to 6 Mbps downstream and 768 Kbps upstream on the line. Subscribers are connected to ADSL copper lines, assuring bandwidth and making service-level agreements possible. High-speed remote access to the Internet and corporate LANs including transmission of voice over (A/H)DSL may increase the performance of the information agents' operations used by residential or small business subscribers, who typically pull more data from the Internet than they push in.

2.1.2. Retrieving and filtering of information

The process of retrieving relevant information is the main topic of the domain of information retrieval (IR) and is an inherent part of the task-level skill of any information agent. An information retrieval model consists of (1) a set D of documents, (2) a set Q of user queries, (3) a framework F for modeling document representations and queries, and (4) a ranking function $R(q, d)$ on $Q \times D$ which defines an ordering between documents d in D with respect to a given query q in Q. The framework F is composed, for example, of (a) sets of documents and standard operations on sets (the Boolean model), or (b) a t-dimensional vectorial space and standard linear algebra operations (the vector model), or (c) sets of documents and standard probability operations based on Bayes' theorem (the probabilistic model). The typical process of retrieving information then comprises (1) pre-processing of documents, (2) query processing, (3) retrieval of relevant documents, (4) presentation of retrieved documents and evaluation of retrieval performance, and finally, (5) user feedback and query expansion. In the following, we will briefly sketch each of these steps.

One commonly used IR model is the vector model, where documents and queries are represented by vectors of index term weights. Index terms are extracted from the text to capture its semantics and stored in an inverted file (index). Index terms can be obtained by, for example, removing stop words, stemming words, and identifying nouns groups in the text. The similarity between retrieved documents and the given query is based on the correlation between respective term weight vectors. This method is known as term-frequency–inverse-document-frequency (TF–IDF).

Let N be the total number of documents (reference collection), n_i the number of documents in which index term k_i occurs, and $tf_{i,j}$ the term frequency of term k_i in document d_j.

The higher the value of a term document frequency, the more likely it is that the term poorly discriminates between documents. Thus, we compute the inverse document frequency and the normalized term frequency as follows:

$$idf_i = \log_2\left(\frac{N}{n_i}\right), \qquad ntf_{i,j} = \frac{tf_{i,j}}{\max_{l \in ind(d_j)}\{tf_{l,j}\}} \in \Re^+.$$

The latter means, the more times a term occurs in document, the more likely it is that the term is relevant to the document; it favors common words. The document and query term weight are defined as follows:

$$w_{i,j} = ntf_{i,j} \times idf_{i,j}, \qquad w_{i,q} = \left(0.5 \times \frac{0.5 \times tf_{i,q}}{\max_{l \in ind(q)}\{tf_{l,q}\}}\right) \times idf_i.$$

A high value of weight indicates that the term occurs more often in this document than is the average with good discrimination.

The cosine similarity metric measures the correlation between a query and a document vector q and d, respectively:

$$sim(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \times |\vec{q}|} = \frac{\sum_{i=1}^{t}(w_{i,j} \times w_{i,q})}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}.$$

This partial matching strategy allows the retrieval of documents that approximate the query conditions, and produces answer sets which are difficult to improve upon without query q expansion. For this purpose, let R be the set of relevant documents, A the set of retrieved documents for a given query, Rd the set of relevant documents as identified by the user among retrieved documents, and $NRd = A - Rd$ the set of non-relevant documents as identified by the user among retrieved documents; other parameters are for tuning purposes only.

Then a new query can be obtained by expanding the old query q due to relevance feedback

$$\vec{q}_{\mathrm{new}} = \alpha \times \vec{q} + \left(\frac{\beta}{|Rd|} \times \sum_{\vec{d}_j \in Rd} \vec{d}_j \right) - \left(\frac{\gamma}{|NRd|} \times \sum_{\vec{d}_j \in Rd} \vec{d}_j \right).$$

A search on an inverted file (the index) can be performed by:
1. *vocabulary search*: words and patterns present in query are isolated and (binary) searched in the lexicographically ordered vocabulary;
2. *retrieval of occurrences* (postings): occurrences of all the words found are retrieved and listed;
3. *manipulation of occurrences*: occurrences to solve Boolean, proximity (words in close sequence), and phrase (words in sequence) search operations are processed.

Alternatively, or in addition to searching the index, the search bot can perform a sequential (online) search on not pre-processed text using known algorithms such as brute-force, Knuth–Morris–Pratt, or Boyer–Moore.

Some common retrieval performance measures are *recall* and *precision*, which are defined as the ratio of the number of relevant documents retrieved to the number of relevant documents in the document collection, and the ratio of the number of relevant documents retrieved to that of the number of all documents in the collection, respectively. Problems of both measure are that:
- Estimation of maximum recall requires knowledge of all documents available.
- Recall and precision are related but capture different aspects of retrieved documents.
- Both measures are inadequate when a non-linear ordering of retrieved documents (interactive) is considered.
- Both measures assume the relevancy of documents to be independent from the user's point of view.

Alternative, user-oriented measures are, for example, the novelty and coverage ration developed by Korfhage in 1997. For a more detailed, in-depth coverage of the field of information retrieval and text filtering we refer the reader to [177,178,116].

2.1.3. Metadata management and ontological knowledge processing

Any cross-platform exchange of information increases the demand for an uniform view of related sources, which means, in particular, the necessity of an agent to capture data semantics (the information content of each source) using descriptive, domain independent, and semantic metadata. Such descriptions can be achieved by utilizing, for example, the Dublin Core, Web interface definition language (WIDL) [150], resource description framework (RDF) [110] in XML, and (federated) database schemas [158] written in standard common data models like, for example, the EER or ODMG93 ODM models.

The RDF data model and its basic type schema RDF(S) provide a standardized fixed set of modeling primitives for defining ontologies and content descriptions in XML, thereby ensuring consistent representation of data semantics. To reason on content descriptions in RDF the information agent may use a respective parser, compiler, and interpreter, which have been and are currently being developed, such as SiRPAC and SiLRI from the Web consortium and the European On2Broker project. Any kind of metadata descriptions and mappings are used to resolve schematic and structural heterogeneity. For a comprehensive survey of respective techniques we refer the reader to [61,64].

However, metadata are constructed from terms stemming from a given vocabulary, domain, or common-sense ontology, like Cyc or WordNet [154]. Such conceptualizations of real-world notions re-usable across shared domains are the basic key for reconciling semantic heterogeneity. Tools for building such ontologies include standard modeling and markup languages such as UML [42], XML [155], RDF, ontology/conceptual knowledge markup language (OML/CKML), SHOE (simple HTML extension for Web page annotation), XML-based ontology exchange languages OIL and XOL, and description logics [18,31,66]. The latter provide, in particular, the inherent feature of a (concept subsumption) reasoning mechanism that an information agent can use to automatically validate and compare content descriptions which have been written in (or translated into) a concept language. In contrast, XML allows for tagging data in Web pages using common XML-namespaces and to do all forms of publishing from one master XML document. This enables an agent to automatically scan, comprehend and validate the content of imported XML documents but not to reason on them. [1] Examples for using common or domain-specific ontologies to resolve semantic heterogeneity are in [125,157,162].

In summary, for the purpose of semantic information brokering, an information agent has to be capable of dealing with multiple, partially overlapping, standardized, domain-specific ontologies. Inter-ontological relations can be determined via, for example, logic-based reasoning on (parts) of ontological descriptions. The corresponding mapping of concepts across ontologies should be done by minimizing the associated loss of information. A partially integrated ontology can be created by the agent virtually by means of determining inter-ontological mappings of terms within a query and content descriptions on the fly, or statically in the form of a long-term, memorized, machine-readable conceptualization of its local or global domain defined independent from actual data, like in a summary or federated schema.

2.1.4. Design and software reuse to support developing information agents

Design patterns in software engineering have been shown to be a very helpful and effective means of capturing and communicating design experience. They are considered to be a natural way of thinking about software, especially in object-oriented design. Recently started efforts to use such software design patterns for the design of cooperative information agents based on an in-depth requirements analysis and corresponding compositional verification are presented in [181].

A closely related technique is that of component-based software development (CBD) enabling the development of software agents out of prepackaged generic elements which, in fact, is based on object technology. It facilitates the development of components usable with current versions of mainstream software buses such as the CORBA, RMI, and DCOM. This includes, in particular, the design of agent behavioral contracts embedded in programming languages such as Eiffel,

[1] Ongoing efforts in that direction include, among others, e.g., the DARPA agent markup language (DAML) initiative.

iContract for Java, or the object constraint language (OCL) as part of UML, to determine, reliably and in advance, how these components behave in agent-based applications. One may distinguish four different levels of component contracts [13]: (1) syntactic/interface, (2) behavioral, (3) synchronization, and (4) quality-of-service. Reuse of contract-aware components can be an important factor in any agent-based application.

2.2. Cooperative information systems and agents

The rapidly accelerating rate of change in today's information-based business environments, coupled with increased global competition, makes corporations face new challenges in bringing their products and services to the market. This results in an increasing demand for the streamlining of operations and an efficient, unified access to information resources that are distributed throughout a local or worldwide network. Although low-level infrastructure has been developed to support interoperability between heterogeneous databases and application programs, this is not sufficient when dealing with higher-level object organizations such as vertical business object frameworks and workflow. Existing multi-database or federated database systems do not support any kind of pro-active information discovery. This has led to the paradigm of so-called *cooperative information systems* originated by Papazoglou et al. in 1992 [101] (Fig. 2) describing an advanced middleware infrastructure based on the inclusion of intelligent information agents that support higher levels of cooperation and provide the required services.

One key challenge of a CIS is to balance the autonomy of databases and legacy systems, with the potential payoff of leveraging them by the use of information agents to perform collaborative work [100]. On the other hand, information agents should collaborate without losing an individually significant degree of autonomy in planning and task execution.

Mediators for the intelligent integration of information. Many past efforts towards an intelligent-agent-based integration of information rely on the concept of a so-called *mediator agent*, introduced by Wiederhold in 1992 [149]. The main purpose of this special kind of information agent is to enable intelligent interoperability across information systems. A mediator has been defined as

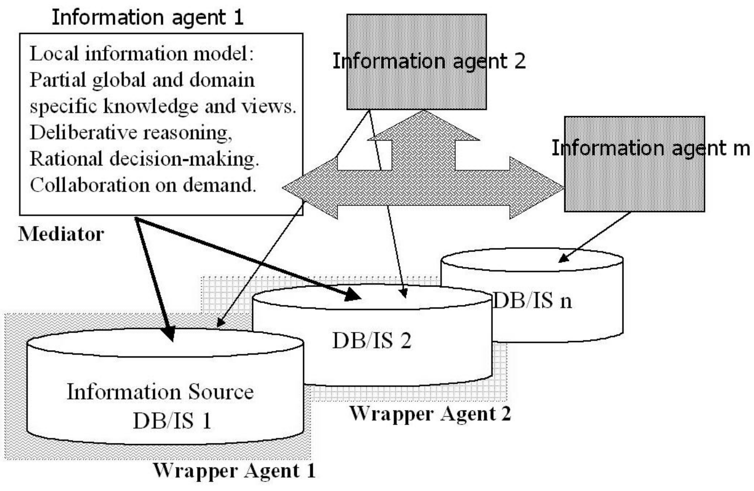

Fig. 2. Macrostructure of a cooperative information system, mediator, wrappers.

"a software module that exploits encoded knowledge about some sets or subsets of data to create information about a higher layer of applications". While the federated or multi-database architecture distinctly focuses on data representation through different types of schemas and appropriate translations, the mediator approach focuses on computational entities that perform value-added activities but keep the information model more or less hidden, typically in the definition of the mediator itself.

A mediator is supported by a set of wrapper agents, each of them providing access to a local information source and extracting content from that source, and performing appropriate data conversion. To provide value-added information services, the mediator may collaborate with other information agents such as broker or matchmaker agents covering different domains and sets of service providers. According to [51] a mediator agent has to

- translate between its own and other domain specific ontologies either by the help of a common ontology maintained by a central ontology agent, or by utilizing its partial global knowledge on ontological inter-relationships between concepts which may occur within requests from other agents and the results of query processing over heterogeneous sources,
- decompose and execute complex queries on available relevant sources with the help of a matchmaker agent, and
- fuse the partial responses obtained from multiple information sources sent by wrapper agents into an uniform, added-value response as displayed to the user.

The perspective of the DARPA intelligent integration of information (I3) research program allows federations among mediators via facilitators on demand. A three-layer reference architecture consists of various types of services such as facilitation, brokering, mediation and integration, wrapping and data access. The architecture is amenable to agents that can support these services offered at each layer. However, most mediator-based information systems implemented to date, such as SIMS/ARIADNE [5], MIX [182], ABS [9], and to some extent TSIMMIS [43], consider scenarios with just one central mediator agent collaborating with multiple wrapper agents. The issue of ontology-based collaboration among multiple mediators or facilitators, like in OBSERVER [124] and InfoSleuth II [94], is absent. The same goes with multi-brokering and distributed matchmaking among different agent communities [59]. A centralized system consisting of just a single mediator and multiple wrappers is not a full-fledged cooperative information system (CIS).

2.2.1. Basic key supporting techniques

The basic key supporting techniques for developing collaborative information agents and systems concern communication and coordination among the agents [52].

2.2.1.1. Inter-agent communication. In Section 2.1, we mentioned middleware platforms for distributed computing to enable transparent access to heterogeneous information and data sources. However, for the purpose of realizing distributed systems with true openness beyond simple communication, "conversations" are necessary. That is the motivation behind the ongoing efforts to design conversations in an agent communication language (ACL) by FIPA [39], or KQML [37] by the KSE initiative at Stanford University.

An ACL defines the syntax and semantics of messages (performative or primitive communicative act) that agents can exchange by describing the desired agent state and complex (propositional) attitudes to reach by each of the language performatives which represents the intention of the conversations prepared. The utilization of such performatives is based on the speech act theory [118]. The receiver sides can understand how to process and proceed with its action by the course of the task-oriented conversations, which are driven by the agents' strategies and behavior but are independent from any content language or ontology (agent's local view). One of the nice things about using an ACL for negotiation is that it permits a model allowing richer communication between the negotiating parties. For example, the FIPA ACL reject-proposal communicative act allows you to give a reason for the rejection.

Many KQML derivatives are on the market designed for different application domains and purposes, but no standard ACL with fixed semantics exists yet.

The requirements for using ACLs in information agent systems include (a) an API for composing, sending, and receiving of ACL messages, (b) a supporting infrastructure, such as an agent naming service and registration, (c) the code implementing action(s) to perform as given by the semantics of message type, particular domain and application. One can distinguish between (1) multi-agent systems using an ACL for inter-agent communication, and (2) APIs facilitating the embedding of ACL-speaking capabilities into an application or multi-agent system.

Another issue of meaningful communication enabling distributed semantic information brokering at the same time is that of understanding the meaning of words, concepts, and notions across multiple application domains which are used within the content of (parts of) the exchanged messages. Related efforts include automated, ontology-based interoperation by using methods, tools, and languages for knowledge representation and sharing, such as non-proprietary languages for knowledge and content interchange, like full-fledged, first-order, predicate logic-based knowledge interchange format (KIF) [63], semantic language (SL) by FIPA, or common ontologies [81,129].

2.2.1.2. Coordinating societies of information agents. Coordination is the process of managing dependencies between activities of one or more actors performed to achieve a goal and to avoid conflicts while having maximum concurrency. It involves task decomposition, resource allocation, synchronization, group decision making, communication, and the preparation and adoption of common objectives.

A variety of approaches for coordination strategies including multi-agent planning and decentralized negotiation protocols for different multi-agent environments exist. Recent works also investigate the benefits of learning to choose an appropriate coordination strategy by a single agent in a multi-agent system [107]. Coordinating collaboration among information agents may follow some sort of social obligations from given or emerging joint intentions, delegation of tasks and responsibilities, or team plans [20,65,130,136].

A comprehensive overview of coordination mechanisms is given, for example, in [33,44,96,193]. Possible types of cooperation in multi-agent systems are discussed, for example, in [32]. Research into modeling cooperative behavior and entailed strategies continues; related works are inspired, in particular, by research in CSCW, cognitive, and social sciences [26].

As an example for coordinating societies of heterogeneous agents we briefly introduce the techniques of *service brokering* and *matchmaking*. For this purpose we differentiate among three general types of agents (see Fig. 3):

1. *Provider agents* provide their capabilities, e.g., information search services, retail electronic commerce for special products, etc., to their users and other agents.

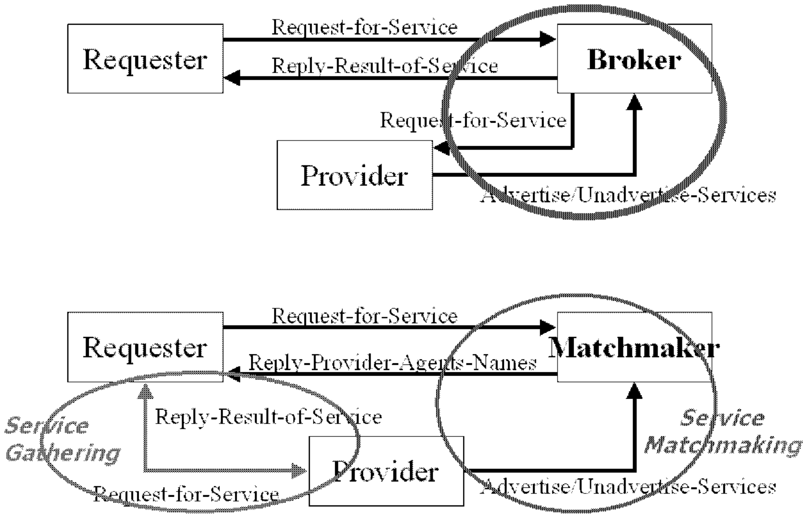

Fig. 3. Service brokering and matchmaking.

2. *Requester agents* consume information and services offered by provider agents in the system. Requests for any provider agent capabilities have to be sent to a middle agent.
3. *Middle agents*, i.e., matchmaker or broker agents [29], mediate among both, requesters and providers, for some mutually beneficial collaboration. Each provider must first register himself/herself with one (or more) middle agent. Provider agents advertise their capabilities (advertisements) by sending some appropriate messages describing the kind of service they offer.

Both brokering and matchmaking require, in particular, a common language enabling the description and automated processing of advertised and requested capabilities of information agents. The first steps have been taken in this direction, such as the development of the agent capability description language LARKS [131] (see Section 2.2.2), or recently started efforts on a general agent markup language DAML. Other capability description languages, like CDL, do not provide any mechanism to enable agents to efficiently reason on respective descriptions.

Every request a matchmaker or broker agent receives will be matched with its actual set of advertisements. If the match is successful, a matchmaker agent returns a ranked set of appropriate service provider agents, together with the relevant advertisements, to the requester. In contrast to a broker agent, it does not deal with the task of contacting the relevant providers itself by means of transmitting the service request to the service provider and communicating the respective results to the requester. This avoids data transmission bottlenecks, but increases the amount of interaction among participating agents. Currently, only a few approaches deal with multiple broker or matchmaker agents [59]. Other works related to matchmaking such as SHADE and COINS [74] are discussed in [135].

2.2.2. Examples

Examples for systems of collaborating information agents include InfoSleuth [58], BIG agent [190], PLEIADES, IMPACT [189], ABS [9], SCOPES [97] and RETSINA. For the sake of conciseness we will just focus on the latter multi-agent system infrastructure.

RETSINA. The reusable task structure-based intelligent network agents (RETSINA) [132,133] multi-agent infrastructure has been developed by the software agents group at the Carnegie Mellon University in Pittsburgh, USA. It consists of a system of three different reusable agent types that can be adapted to address a variety of different domain-specific problems. Interface agents interact with the user, receive user input and display results, task agents help users perform tasks by formulating problem-solving plans and carrying out these plans through querying and exchanging information with other software agents, and resource agents provide intelligent access to a heterogeneous collection of information sources.

A collection of RETSINA agents forms an open society of reusable agents that self-organize and cooperate in response to task requirements. Each agent draws upon a sophisticated reasoning architecture that consists of four different reusable modules. Firstly, the communication and coordination module accepts and interprets messages and requests from other agents. Secondly, the planning module takes as input a set of goals and produces a plan that satisfies the goals. Thirdly, the scheduling module uses the task structure created by the planning module to order the tasks. And fourthly, the execution module monitors this process and ensures that actions are carried out in accordance with computational and other constraints. The RETSINA framework has been implemented in Java and is being used to develop distributed collections of intelligent software agents that cooperate asynchronously to perform goal-directed information retrieval and information integration in support of a variety of decision-making tasks

LARKS: matchmaking among heterogeneous agent systems. In contrast to the rather broker-based InfoSleuth system, RETSINA relies on matchmaking in dynamic agent societies. For this purpose an agent capability description language called LARKS (language for advertisement and request for knowledge sharing) has been developed [135]. When a service-providing agent registers itself with the middle agent together with a LARKS description of its capabilities, it is stored as an advertisement and added to the middle agent's database. Thus, when an agent inputs a request for services, the middle agent searches its database of advertisements for a service-providing agent that can fill such a request. Requests are filled when the provider's advertisement is sufficiently similar to the description of the requested service. Application domain knowledge in agent advertisements and requests can be currently specified as local ontologies written in a specific concept language ITL or by using WordNet.

An advertisement or request in LARKS is a frame comprising the following slots: (1) *Context*: keywords denoting the domain of the description, (2) *Types*: user-defined data types found in the signature definition, (3) *Input* and *Output*: input and output parameter declarations defining the signature of the operation, (4) *InConstraints* and *OutConstraints*: logical constraints on input/output variables (pre-/post-conditions), (5) *ConcDescriptions*: descriptions of the disambiguating words used in the first three slots in concept language, or keyword phrase, and (6) *TextDescription*: a free text description of the agent's capabilities (see Fig. 4).

LARKS is fairly expressive and capable of supporting inferences. The LARKS matchmaking process employs techniques from information retrieval, AI, and software engineering to compute the syntactical and semantic similarity among agent capability descriptions [135]. The matching engine of the matchmaker agent contains five different filters for (1) keyword-based context matching, (2) TF–IDF-based profile comparison, (3) concept-based similarity matching, (4) type-inference rule-based signature matching, and (5) theta-subsumption-based constraint matching of

Ad: AWAC-AirCombatMissions

Context	Combat, Mission*AWAC-AirMission
Types	Date = (mm: **Int**, dd: **Int**, yy: **Int**); DeployedMission = **ListOf**(mType: **String**, mID: **String** | **Int**, mStart: Date, mEnd: Date);
Input	Start: Date, End: Date
Output	missions: DeployedMission;
InConstraints	Start <= End
OutConstraints	deployed(mID), mType = AWAC, launched_after(mID,mStart), mStart >= Start, launched_before(mID,mEnd), mEnd <= End.
ConcDescriptions	AWAC-AirMission = (and AirMission ....)
TextDescription	information on deployed AWAC air combat missions launched in a given time interval

LARKS Advertisement: "Agent is capable of providing information on *deployed air combat missions launched in a given time interval.*"

Fig. 4. Example of an agent capability description in LARKS.

finite Horn clauses. Any user may individually configure these filters to achieve the desired tradeoff between performance and matching quality.

3. Adaptive information agents

Adaptive information agents have to deal with uncertain, incomplete and vague information in an efficient, reliable way such that they are able to make intelligent decisions on the fly [28]. Adaptation of an agent to its environment can be done in an isolated manner or in collaboration with other agents by using methods for single- or multi-agent learning, respectively [109,120,161]. Learning among multiple agents may be collective, that means, the agents adapt themselves in order to improve the benefits of the system. However, system adaptation can even emerge without any collaboration when the individual learning of one agent affects that of the other agents in a beneficial way. An agent may exhibit adaptive behavior relative to a variety of internal reasoning processes concerning communication, coordination, planning, scheduling, and task execution monitoring [131]. All approaches and systems for single or multi-agent adaptation may be evaluated by different criteria; these criteria concern:

- the applied strategy, such as learning by example, analogy, or discovery,
- the kind of feedback and guidance for the agents by means of reinforcement, supervised or unsupervised learning,
- the type of interaction among agents, human users and the multi-agent system in the environment,
- the purpose of learning to improve the skills of a single agent or the whole system, and
- the distribution of data and concurrent computation for adaptation in the multi-agent system.

The most popular application domain of adaptive single- and multi-agent systems is currently electronic commerce and information gathering on the Web. Equally important domains are manufacturing [12,41], digital libraries [34], logistics, and telecommunication networks. Some open questions and challenging research issues are the following:

- Mechanisms for learning successful negotiation and coordination strategies in a multi-agent system [83,107].
- When is adaptation of single agents harmful or beneficial to the system they are involved in [134]?
- How can collaborative behavior among multiple adaptive information agents effectively evolve [6,163,126]?
- Which methods for knowledge discovery, representation and maintenance are most appropriate for an information agent in an open environment?

A variety of machine learning techniques are useful within information agent systems; comprehensive readings in such techniques are, for example, [8,90]. The most popular types of learning methods, ranging from neural-network learning through Q-learning and case-based reasoning (CBR) [76] to genetic learning, for adaptive agents are the following:

- *Supervised learning.* User feedback received by the agent specifies some desired activity; the objective of learning is to match this desired activity as closely as possible.
- *Unsupervised learning.* This refers to adaptation without any feedback from the user or other agents. The objective of learning is to find useful and desired activities or patterns of activities through a self-organizing process.
- *Reinforcement learning.* User feedback specifies the utility of some activity performed by the agent the objective of which is to learn how to maximize this utility.

A comprehensive overview of the research area of adaptation of single agents and multi-agent systems is provided, for example, in [121,144,145].

3.1. The non-cooperative case

A non-cooperative adaptive information agent gradually adapts to changes in the user, information, and network environment by its own without any collaboration with other agents.

3.1.1. Key supporting technologies

Basic key supporting technologies for the development of any single, adaptive information agent include, in particular, human–agent interaction, visualization of information spaces to the user, content-based user profiling, and adaptive knowledge discovery in databases [72]. Other relevant techniques concern, for example, learning of a single agent to select information sources based on the principle of maximum expected utility having limited information on the environment [119].

3.1.2. Human–agent interaction and visualization of information spaces

Any flexible, convenient human–agent interaction (HAI) helps to increase the awareness and thereby the acceptance of the information agent and its work by the user. For this purpose, an adaptive information agent should interact with its users in a most convenient way through an intelligent interface. HAI is largely motivated by the metaphor of indirect management, that is, the user is engaged in a cooperative interaction process in which human and agent both initiate communication, monitor events, and perform tasks [77,84]. Such an interaction encompasses (1) the processing and analysis of the user's input, such as speech, and affective signals, (2) managing the interaction process based on the agent's knowledge of the domain, user, discourse, media, and task model, and (4) the design of the presentation of rendered output using believability-enhancing gestures, natural language, or graphics.

Individually sensitive guidance of the user through the available includes not only anticipating its needs on the fly, but also visualizing the space in real-time. The latter can be realized, for example, by utilizing virtual reality (VR) techniques [24] and life-like synthetic characters [3,35]. This also requires automated speech recognition, body tracking and tracing for affective computing [104], and projecting of potentially terascale data grids resulting from the agent's online data mining activities. Such tele-immersive environments, like CAVE, ImmersaDesk 3, CAVE6D, or TIDE [183], allow the guidance of the user by their personal information agents while s/he is walking through the information landscape. First implementations have been done in that direction through modeling 3D shopping malls in VRML, like culthouse.de or vira.de, though, they still have to be equipped with assisting 3D information agents. [2] The same goes with digital cities [184] as platforms to support community networking while being loosely or tightly coupled with the physical city in terms of shops, offices, and administration.

Besides, any future progress in advanced HAI is due to the rise of multimedia pushed to a new level by, for example, more powerful 3D graphics accelerating and displaying hardware, significant increase of mass storage capacity, ultra-high performance connections among sites in the Internet(-2) and standards for multimedia integration on the Web, like SMIL [128].

In summary, HAI and its application to (systems of) information agents still appear to be an uncharted territory [77], despite the recent research efforts in intelligent interfaces and human agent factors carried out, for example, by projects in the European I3Net initiative started in 1997 [53] as well as in the Special Interest Group on intelligent information agents as part of AgentLink [1].

3.1.3. Content-based filtering and user profiling

Content-based filtering and user-interest profiling is a common approach to tackle the information filtering problem (see Fig. 5). Items are recommended to a user according to correlations found between the items' content, for example, presence of certain keywords, features, and the given user preferences in a profile. The latter is usually generated and updated by the agent automatically by observing the user's online activities such as visiting Web pages, dealing with downloaded documents, adding or deleting bookmarks, and printing, as well as affective signals such as eye movement or gestures, and credit assignments to the agent.

The agent extracts features from the documents, uses them to form training examples, and induces a corresponding user interest profile via application of suitable machine learning techniques such as reinforcement learning. Other related methods include learning models of text categorization, assignment of documents to one or more categories, and the creation of possibly overlapping categories due to levels of user interest. The profile can be used by the agent to predict further actions of the user thereby learning to recommend and pro-actively select relevant documents.

[2] Notably, the impact of HAI in such virtual environments to social interaction and psychological well-being in real life still remains to be investigated.

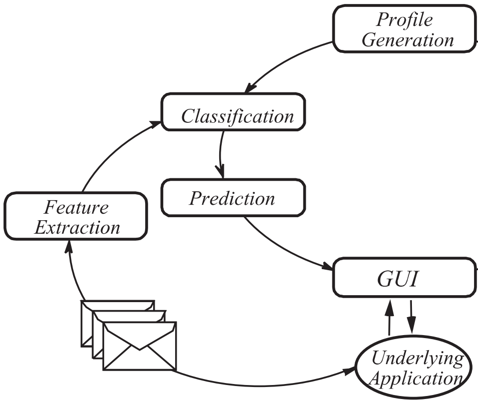

Fig. 5. Content-based filtering and user profiling.

Content-based filtering has a few drawbacks. Firstly, information must be in some machine-parsable form like text or attributes that have to be assigned by hand, but it appears to be difficult to assign attributes to media such as sound, images and video. There is also no inherent mechanism for generating serendipitous finds, meaning the agent may not recommend more of what the user has seen before and liked. Finally, content-based methods cannot filter, based on assessments of style, quality, etc., that is, the agent cannot distinguish between a well-written and a badly written paper, if both use the same terms.

3.1.4. Examples: personal assistants and synthetic characters

Developing personal assistants not only for the Web has become trendy, attracting increasing interest from the common user over the past five years. Though one has to be aware of the fact that an intelligent information agent acting as a personal assistant can present itself to the user in the form of a believable, synthetic character as kind of alter ego, it is never meant to be identical with it.

A lot of synthetic characters and personal assistants have been developed at, for example, NetSage [93], Extempo Imp [36], Microsoft, DFKI, and MIT Media Lab [78]. Fig. 6 shows some examples of synthetic characters which are programmable to behave in accordance with different personality traits specified by the developer in the context of a given application. It is possible, for example, to set up personalized sales dialogues at the portal site of a car dealer between different synthetic characters representing vendors and consumers. Each of these characters adopts a different role, trying to convince the potential customer to purchase an advertised item through role-based argumentations in a simulated conversation.

Some prominent personal assistants are Letizia, Remembrance, ExpertFinder, Butterfly, Let's Browse, and TrIAs [11], AiA (adaptive communication assistant for effective infobahn access), PAN (planning assistant for the Net) Travel Agent [99], WebPersona (presentation agents for the worldwide Web) [4], from MIT Media Lab and DFKI, respectively.

Letizia and Remembrance observe user preferences while s/he is browsing through the Web, use a variety of heuristics for identifying possibly interesting pages for the user and, in contrast to common users, browse the Web breadth-first. Let's Browse allows for collaborative browsing of the Web, which appears to be highly suitable for WebTV application. ExpertFinder and Butterfly

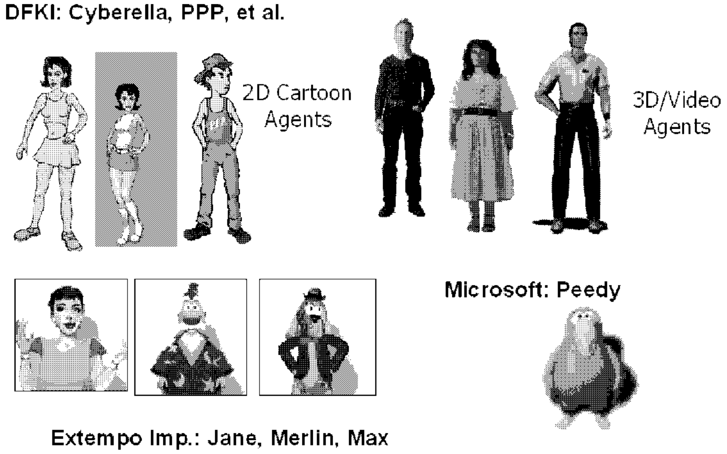

Fig. 6. Examples of life-like synthetic characters.

determine the user's experience and expertise by the observation of dialogues in mailing lists and chat rooms, and gradually learn to find most appropriate experts for a given query of their user; Butterfly recommends a chat channel on the Internet to the user. WebWatcher from Carnegie Mellon University interactively accompanies a user while s/he browses the Web in a tour and adapts to user preferences and information server contents by reinforcement learning from experience, that is, previous tours.

Another but rather simple type of personal assistant is the so-called chatter bot, which basically uses low-complexity case-based reasoning techniques to guide users through product orders, Web-pages or entertain, for example, by ELIZA-like chatting with users on Internet chat channels. In the domain of electronic business it is typically designed to answer the 20% of questions that generate 80% of call volume to customer-service centers.

3.2. Collaborating adaptive information agents

Currently, only a few systems of collaborating information agents show adaptive behavior. In part, this is due to the fact that there is still not much known about the exact relation between single- and multi-agent adaptation, and vice versa. The first steps have been taken to investigate how collaborative information agents may learn to coordinate their actions and task execution in different domains. However, the development of adaptive collaborating information agents remains one of the main challenges in the area of IAT. Regarding this, the future question will be how an information agent ingeniously behaves, rather than what it looks like.

3.2.1. Key supporting techniques and technologies

Some of the most widely used and effective techniques for collaborating adaptive agents are collaborative (social) filtering [30,164] and genetic algorithms [8].

3.2.1.1. Collaborative filtering. This collaborative recommendation technique is a powerful method for leveraging the information contained in user profiles. In contrast to content-based filtering, the agent rates the items chosen by its user and compares the corresponding user preference vector to that of other users projected to the same set of items. It then recommends other items which have been recommended by users who share similar likes and dislikes. For this purpose it has to collaborate with other agents to gain the respective knowledge. Thus, this technique essentially automates the process of "word of mouth" in a given user community. In addition, trust among users and agents is even easier to gain, since it is very difficult to manipulate the recommendations an agent makes to its user via social filtering.

A common technique to find similar users and predict the weighted average of user ratings is to determine the correlation between the users' preference vectors using minimum square error, or the Pearson algorithm:

$$\sum_{i\in\{1,\ldots,x-1,x+1,\ldots,n\}} [\vec{r}_+(u_x) - \vec{r}_-(u_x)] \cdot [\vec{r}_+(u_i) - \vec{r}_-(u_i)], \tag{1}$$

where $\vec{r}_{+/-}(u_k)$ is the positive/negative rating of user k.

However, traditional collaborative recommendation still has a few shortcomings. Initial users can bias the ratings of future users, no different (context-based) points of view of users are taken into account, and there is no learning from negative cases involved.

3.2.1.2. Evolutionary computing: genetic algorithms. Learning across generations of individual, autonomous entities may follow the biologically inspired genetic algorithm model introduced by Holland in the 1960s. Such genetic algorithms model the alteration of genes (phenotypes) during reproduction in order to create new architectural forms [8]. They use a coding of the problem to be solved in a DNA-like string, search for a solution in a population of state points rather than a single point, control the population by a fitness function that rates each individual and reproduces in a non-deterministic manner via cross-over or mutation. A population of strings represents a species, and the most fit species represents the best solution evolved so far. Genetic algorithms are inherently slow but have proved to be useful in many application domains of information agents as in the following examples.

3.2.2. Examples: evolving information agent ecosystems

Some prominent examples of systems of collaborating adaptive information agents are Amalthaea, InfoSpider, LikeMinds and Firefly [40].

Amalthaea [91] is an evolving, market-like multi-agent ecosystem for personalized filtering, discovery and monitoring of sites in the Web. Like in distributed, adaptive knowledge bases [88], the approach uses a genetic algorithm. Two different categories of agents are introduced in the system: filtering agents that model and monitor the interests of the user and discovery agents that model the information sources. The latter type of agent uses the standard IR technique of weighted keyword vector extraction and determines the similarity of documents. The user may give relevance feedback on documents presented by filtering agents in a digest. The phenotypes of filtering agents are represented by the most rated document vector. Filtering agents that are useful to the user may reproduce using methods originated from genetic cross-over and the mutation of phenotypes, while low-performing agents will simply die out and be destroyed by the system. In [91], results from various experiments with different system configurations and varying ratios of user interest are shown and how to achieve an equilibrium in the information agents ecosystem.

Transitive recommendation systems generate recommendations of products and documents on the Web from the connected paths of ratings of human users themselves. This is in contrast to collaborative filtering, where the recommendations are produced from the direct ratings of preferences. An example for such a recommendation system is Histos [91] for highly connected online communities.

InfoSpider agents [87] search online for relevant information by traversing links in the Web considered as a directed acyclic graph. As with Amalthaea, the idea of InfoSpiders is to complement the existing search bots on the Web with respect to two of their main difficulties: scaling and personalization.

The static character of an index database as a basis for any search engine cannot keep up with the rapid dynamics of the growing and changing Web. In addition, the general character of the index building process cannot exploit the different profiles and needs of different users. Search bots on the Web provide global starting points, and based on statistical features of the search space InfoSpiders use topological features to guide their subsequent searches. Each live spider follows a random link from a given document with best-first heuristics. If a new document is not in its cache it pays energy costs for incurred server access bandwidth and receives an energy payoff equivalent to the relevance of the retrieved document. The spider will be destroyed as soon as its energy decreases below a given threshold, otherwise it is allowed to reproduce by splitting and sharing its energy with a cloned offspring.

4. Rational information agents for electronic business

Electronic commerce may be defined as the set of activities of trading goods and services on line. It is a part of electronic business covering a broader range of issues including business processes and transactions on the Internet devoted to customer relationship and supply chain management.

4.1. E-commerce: some facts and figures

E-commerce is steadily growing since around the past five years. According to recent market research reports of Meta Group and Forrester Research between US$ 200 and 350 billion e-commerce sales are expected by the year 2000 in Europe and US, respectively. Remarkably, the business-to-business (B2B) market segment of e-commerce is predicted to outweigh that of business-to-consumer (B2C) worldwide by values of US$ 1.3 and 0.1 trillion, respectively. On the other hand, consumer online spending rose from US$ 7 billion in the whole holiday season of 1999 up to US$ 2.8 billion in the month of January, 2000.

4.1.1. Basic enabling technologies for developing e-commerce and business solutions

The development of any e-business solution relies on basic enabling technologies such as for:

- *standard data representation, retrieval and exchange*, like XML, UML, EDIFACT/WebEDI/EDIINT, domain ontologies, RDF, data retrieval, and data mining methods,
- *secure user profiling and data*, like open profiling standard (OPS), W3C's platform for privacy preferences (P3P), (a)symmetric coding schemes, digital signatures, and digital watermarks,
- *secure electronic payment*, like VISA/MC's SET for payment with credit card, digital cash, like DigiCash's eCash, DEC's MilliCent, and smart cards [127], or deduction from a given customer account like at central virtual markets, and
- *standard protocols covering most issues of electronic trading*, like IETF's internet open trading protocol (IOTP), open trading protocol (OTP), and open buying on the Internet (OBI).

Different trading models and schemes may be compared along (1) the design on economic principles such as dominant, competitive and adaptive strategies and equilibria, (2) privacy of interests of the participants and anonymity of identities, (3) complexity of trading mechanism in terms of computation and communication.

Besides, any emerging consensus on an *accounting and pricing* structure, such as flat-rate, capacity-based or usage-sensitive pricing, is as important as effective trust and security mechanisms to facilitate e-commerce transactions in a digital economy. A remaining challenge is how to model, measure, and reason on *trust*. The situation becomes even more complex since customers as well as vendors, their products, services and quality may change rapidly over time. There is still no satisfactory method known which appears to be suitable for agents to react to such changes in an appropriate way.

4.2. Agent-based e-trading

Despite the enormous potential of electronic commerce a more sophisticated, agent-based trading [192] still remains a key challenge for economists, computer scientists and business managers as well. It might reshape the way we think about economic systems and business processes in an increasingly networked world. In the open and increasingly commercialized cyberspace, personalized information agents not only may pro-actively discover and manage information relevant to their customers but are paid and have to pay for any services they provide. One vision is that agents facilitate e-commerce and business functions such as advertising, negotiating, matchmaking and brokering. Reasonably, trading information agents have to be equipped with effective and efficient methods for making economically rational decisions. This includes scenarios where agents, for example, make purchases up to a preauthorized limit, filter information and solicitation from vendors, dynamically trade commodities such as even bandwidth and components within B2B or B2C digital market, or decide on bids from service providers to take for their customer in some reverse auction on-line in a consumer-to-business (C2B) e-commerce setting, and increase the level of trust in their actions gradually over time, involving only manageable risks for both customers and vendors [114,117].

Though electronic business and commerce is not the original, classical application domain of information agents it certainly is the most steadily growing one. However, e-commerce on the Web might happen without any intelligent agents if agent technology in general fails to be injected into currently emerging Internet-mediated transaction standards and systems.

In any case, due to a Harvard study published by Moon in 1998, e-commerce applications with personalized, digital assistants for information gathering and guidance in on-line shopping are expected to be 80% more convincing, 30% more attractive and 40% more qualitative. This is where rational information agents for agent-mediated trading come into play.

4.2.1. Basic key supporting techniques for agent-mediated trading

Many negotiation and trading mechanisms for intelligent trading agents rely on multi-attribute utility theory [191], price comparison, content-based recommendation and user profiling, blueprint learning of unknown Web pages as well as collaborative recommendation, coalition formation among autonomous agents, auction-based protocols [147,148], dynamic supply chain management [156], agent-based marketplaces [56,146], variations of the well-known contract net protocol and arbitration schemes [112].

The first four techniques are typically used in non-cooperative cases such as shop bots, whereas the latter are devoted to the class of collaborating trading agents. Scenarios for single-agent or multi-agent systems for e-commerce and business can be set up, including virtual marketplaces, auctions (B/C2C), and reverse auctions (C2B), as well as shop bots, on line shops, and web portals (B2C/B).

4.2.1.1. Coalition formation between rational information agents. Self-interested autonomous agents may negotiate rationally to gain and share benefits in stable (temporary) coalitions [115,123]. This is to save costs by coordinating activities with other agents. For this purpose, each agent determines the utility of its actions and productions in a given environment by an individual utility function. The value of a coalition among agents is computed by a commonly known characteristic function which determines the guaranteed utility the coalition is able to obtain in any case. In a characteristic function game, the agents may use imposed individual strategies implied by desired type of economically rational behavior such as altruistic, bounded rational, or group rational. In any case, the distribution of the coalition's profit to its members is decoupled from its obtainment but is supposed to ensure individual rational payoffs to provide a minimum of incentive to the agents to collaborate.

Methods for the formation and maintenance of stable coalitions mainly derive from cooperative game theory, economics, and operations research. They cover the formation of coalition structures, and the distribution of gained benefit among coalition participants. Individual strategies of agents are implied by different types of economically rational behavior such as being altruistic, bounded rational, or group rational.

Most interestingly, non-trivial cases of coalition formation concern non-superadditive environments where at least one pair of potential coalitions is not better off by merging into one which could be caused by, for example, communication and coordination overhead costs, decrease of coalition value as a result of restricting utility constraints posed by agents joining a coalition, or anti-trust penalties for specific coalitions [123]. The meaning of stability of formed coalitions relies on the chosen game-theoretic concept of payoff division within coalitions according to, for example, the Shapley value, the core, the bargaining set, or the kernel [195].

In environments where published interests and utilities used for negotiation to form coalitions cannot be verified, most current protocols allow for fraud by different types of lies [175]. In addition, in scenarios where agents may leave or enter the negotiation process at any time and perform a continuous stream of incoming tasks, an efficient dynamic formation of multiple, overlapping coalitions remains to be solved. Dynamic coalition formation can be applied to multiple online auctions to form temporary customer coalitions on the fly, though the underlying methods and techniques have to be invented yet.

Although well-grounded techniques for automated decision making and coalition formation among self-interested agents are known [62,68,69] none of them has been used so far on the public Web. Other applications for methods of utilitarian coalition formation include, for example, the decentralized power transmission planning [22]. A publicly available simulation environment for coalition formation based on selected coalition theories is provided in [21].

4.3. The non-cooperative case: shop bots

Quite popular but basically very simple examples of non-cooperative rational information agents are shop bots on the Web such as mySimon.com, Junglee/Yahoo!, Jango/Excite, shopfido.com, compare.net, or evenbetter.com. The first and probably best known shop bot, BargainFinder [10] from Andersen Consulting, is not available anymore on the Web. This is mainly due to insufficient profit-making for associated retailers and vendors.

Shop bots do not sell any product but guide the customer to recommended online stores offering these items. The recommendation is either based on the comparison of prices, or multi-attribute utility theory, such as that used by the shop bot frictionless.com. The latter takes additional attributes like quality of product, timeliness in delivery, warranty, customer support, and reputation of the vendor into account. The final recommendation is based on the result of a distributed attribute constraint satisfaction approach. Underlying assumptions for both types of shop bots are that (1) vendors reveal the relevant information on items to the agent, and (2) the content of vendor Web pages can be automatically scanned and understood by an agent.

However, it remains to be seen if the currently deployed shop bots can successfully compete with large portal sites such as amazon.com or barnesandnoble.com, and online retail auctions [7] such as eBay.com in the long run.

4.4. The collaborative case: markets and auctions

Virtual, agent-based marketplaces provide locations where multiple agents from different consumers and vendors may meet each other to negotiate and exchange relevant data and information. Negotiation may concern, e.g., the amount of charges for provided services as well as the kind of services or goods itself. Free markets and auctions [7] are the most common virtual institutions for e-commerce which can be mediated by collaborating rational information agents. They are means for customer-to-customer and B2C e-commerce, respectively. Marketplaces provide locations where multiple information agents from different users and providers may meet each other to negotiate and exchange relevant data and information. Negotiation concerns, for example, the amount of charge to pay for services as well as the kind of services or goods itself.

Auctions theory [85,151] analyzes protocols and agents' strategies in auctions. An auction is a price-fixing mechanism or institution in which negotiation is subject to a very strict coordination process. It consists of an auctioneer who wants to mediate the exchange of given items between buyers and vendors for sale at the highest possible price, and potential bidders who want to buy them at the lowest possible price. Asynchronous bidding mechanisms are mostly based on open-outcry with price changes or sealed-bid with periodic partial revelation. Any auction may be classified along three dimensions of (1) bidding rules including, for example, bid format and many:1 or many:many participation, (2) clearing policy such as pricing, clear schedule and closing, and (3) information revelation policy including, for example, price quotes, quote schedule, etc. Prominent auction protocols include:

- the first-price, open-cry, so-called English auction. The dominant strategy for consumers here is to bid up to their true, maximum value;
- the decreasing price, open-cry, so-called Dutch auction that guarantees the auctioneer the purchase of items at the highest possible price;

- the first-price, sealed-bid auction having the potential to force buyers and seller into price wars since the sealed bid of any bidder depends on what s/he believes of all other opponents bids;
- the second-price, sealed-bid so-called Vickrey auction where the winning bidder pays the price of only the second-highest bid [113].

Almost all current Internet auction types are single-resource, one-sided, and not executed in real-time in a strong sense. Future trends may include combinatorial auctions with lower and upper prices for product bundles and reverse auctions where service providers bid to satisfy some customer's request for a kind of service.

4.4.1. Examples

Prominent examples for agent-based marketplaces and auctions are Kasbah/MarketMaker, AuctionBot [147], UMDL, and FishMarket.

UMDL (University of Michigan digital library) [34] is an agent-based digital library offering electronically available information content and services in a distributed environment. It relies on a multi-agent infrastructure (the service market society – SMS) with agents who buy and sell services from each other using a given set of commerce and communication protocols. Within the SMS, self-interested agents are able to find, work with, and even try to outsmart each other, as each agent attempts to accomplish the tasks for which it was created. Learning in the context of SMS provides a way for agents in the SMS to develop expectations and strategically reason about others, and exploit these expectations to their mutual benefit.

Kasbah/MarketMaker [7,160] is a simple, agent-based marketplace which has been developed at MIT Media Lab. Trading of goods is performed among buyer and seller agents on the central marketplace; each agent has knowledge about the (type of) goods it has to buy or sell by pro-actively seeking out potential best deals and negotiates them on their user's behalf. These deals are subject to user-specified constraints in terms of desired price, lowest (or highest) acceptable price, a date to complete the deal, and one of three simple types of price decay functions. These functions correspond to greedy, moderate, and anxious behavior of buyers or sellers. Upon completion of a deal (and respective transaction) both parties are able to rate the other parties' part of the deal in terms of, for example, product quality, and timely completion of transaction. Agents may use these ratings to determine their willingness to follow up a negotiation with agents whose users do not match a given reputation threshold.

5. Mobile information agents

A mobile agent is programmed to be able to travel autonomously in the Internet from one site to another for the execution of its tasks or queries on different servers. It can be seen as a steadily executing program only interrupted during transport between several servers. The new paradigm of mobile agents or so-called remote programming is in contrast to the traditional client/server computing via remote procedure calls (RPC) conceived in the 1970s. Any two computers or software agents that communicate via RPC agree in advance upon the effect of remotely accessible procedures, the kind of arguments, and the type of results. This appears to be insufficient in dynamically changing information environments. Besides, any request for procedure performance, acknowledgement as well as the data as a result of remote processing must be sent via the network that interconnects the respective computers. This may generate a high level of network traffic and, depending on the network design, can be susceptible to congestion delay. In addition, mobile devices, intelligent broadband [159] and wireless data networks are becoming more powerful and affordable, leading to the growing importance of mobile personal data access and processing [167]. Until today, a large number of mobile agent systems has been developed, and several approaches deal with the integration of these systems and RPC-based middleware such as CORBA.

The most prominent efforts for the standardization of an intelligent mobile agent system are the addition of a mobile agent facility (MAF) into CORBA by OMG, and the proposal by the Foundation of Intelligent and Physical Agents (FIPA). These efforts even try to deal with the problem of misuse involving mobile agents.

What are the main benefits of mobile information agents? Firstly, such agents may execute their services, for example, intensive data processing and information extraction, locally at remote database servers. They can react dynamically on latencies and congestions which may reduce network load significantly. Especially in wireless networks, it is advantageous to do work remotely, in particular when the connection is temporarily lost. Mobile agents can exhibit intelligent strategies for actively searching and integrating information at multiple servers. Resource and service discovery is the fundamental premise of mobile information agents.

Finally, mobile information agents can enhance distributed applications by enabling users to access information ubiquitously, that is, anywhere and at any time [103].

5.1. Non-cooperative mobile information agents

Mobile information agents may enable the migration of small, application-based business logic in corporate intranets on demand and dynamic maintenance of connected data warehouses in the Internet. Any activity of a single mobile information agent basically relies on the existence of appropriate run-time environments allowing it to work on different servers. Some main related issues concern the assignment of server resources to visiting agents, code persistence, recovery from failures, and a platform-independent development of mobile agents.

5.1.1. Key supporting technologies and examples

Mobile agents are almost always written in an interpreted machine-independent language such as Java, so that they can run in heterogeneous environments. It is assumed that an appropriate computation environment is accessible on any server the agent might visit. Actually there are several systems available [103]. They consist of either

- Java class libraries such as IBM's Aglets [168], ObjectSpace's Voyager [171], Mitsubishi's Concordia [173], and MOLE [171], or
- scripting language systems with interpreter and runtime support, like D'Agents/AgentTcl [174], and ARA [169], or
- operating system services accessible via a scripting language like TACOMA [170].

A comprehensive overview of mobile agent systems and their application for distributed information retrieval is given, for example, in [17]. Future research work may provide in-depth investigation of the benefits of using mobile information agents for efficient distributed database query processing including (semi-)join operations in large-scale heterogeneous, distributed (or mobile) databases; the first steps have been taken in this direction [105]. However, the main application area of mobile agent technology is currently the area of telecommunications [106,142] where it is being used as a part of the decentralized service architecture of next-generation networks such as TINA-C [73]. In only a few years, some systems of mobile information agents might be able to operate on different kinds of wireless connected hand-held devices and wearable intelligent computers [102]. The development of mobile agents may benefit in particular from progress in wireless, satellite-based communication, and the mass production of wearable computers.

5.1.2. Transport of code and state, server resource management

Mobility mechanisms include remote method invocation such as Java RMI, agent cloning, and code-on-demand. Most systems use application protocols on top of TCP for transporting agent code and state. Interoperability among heterogeneous mobile agent systems is crucial for any unlimited crawling of mobile information agents through the Internet and Web; this implies the need of facilities to port and persistently store the data and code of mobile agents. Portability among different mobile agent systems may be achieved either by adding appropriate features to a platform-neutral programming language such as Java to support migration via RMI, or middleware platforms. The issue of data persistence still has to be covered in most current mobile agent systems, except for, e.g., Voyager and Concordia. Mobile agents also require access to site resources such as CPU cycles, disk capacities, graphics, memory, persistence service, and threads. Resource management is hardly supported by current mobile agent systems, like in IBM Aglets, and Voyager, or not specified at all.

5.1.3. Issues of security

The question of security goes in both directions [46,139,141]: how can database servers be protected from malicious actions of mobile agents, and, in turn, how can an information agent, packed with private data and information, be protected from hostile servers and other agents while traveling through cyberspace? Remarkably, in many approaches and implementations of mobile agent systems, the server and the computational environment are still assumed to be trustworthy. The same goes with the agents. Traditional security mechanisms rely on cryptographic methods for the implementation of authentication and authorization. More satisfactory solutions to prevent an attack by server include listening to inter-agent communication, refusing to execute selected agents, or access to private data of an agent. Such misuse is hard to prevent since the server has to have access to the agent code to execute it. Other solutions include trusted execution environments and mechanisms for detection or prevention of tampering, such as trusted hardware and secure cryptographic execution of agents using encrypted functions or code obfuscation in a time-limited blackbox, proof-carrying code, and various schemes for access and execution control such as execution tracing. The most mundane form of protection is to disallow agents to move to untrusted hosts; this provides a high level of security, but in many cases will not be appropriate for the application.

5.2. Collaborating mobile information agents

Coordinating a system of mobile information agents is quite a challenge regarding their location and communication independence.

5.2.1. Some basic techniques and examples

Agent spawning is a means of resolving agent overload problems. Agents in a multi-agent system may face situations where tasks overload their computational capacities or do not fit their capabilities. Usually, this problem is solved by passing tasks to other agents or agent migration to remote hosts. Agent spawning is a more comprehensive approach to balancing local agent overloads. Agents may spawn themselves, pass tasks to others, die or merge. In this sense agent spawning is also an appropriate means for creating and maintaining systems of collaborative mobile agents. [122] reports in detail on mechanisms required for deciding upon when, how and where to spawn agents. Simulation results show the advantage of using the implemented agent spawning mechanism under some constraints.

The basic idea of *coordination patterns* is to re-use software patterns of coordination among mobile agents suitable to a given application [138]; mobility may be seen as a way to manage accessibility dependencies between activities of mobile agents accessing information sources. Such coordination patterns can be written, for example, in (a variant of) the coordination language LINDA.

Other approaches of coordination rely on *ant-based swarm intelligence*, which could help mobile information agents to perform, for example, rerouting of their traffic in busy networks automatically in a manner that is similar to how ants raid different food sources and mark their respective paths via evaporating pheromones [185]. Examples of collaborative mobile agents are Concordia [165], and Nomad [186]. Other related work is reported in [187].

6. Conclusions and outlook

The open Internet and worldwide Web allow us to access multimedia data and knowledge located throughout the world. Clearly, these new technologies present enormous opportunities for posting, finding, organizing and sharing vast amounts of information. These are the premises of intelligent information agents, who can play a dominant role in our evolving information infrastructure, if they are proven to be useful to people, organizations and enterprises for intelligent information search and management. Thus, information agent technology attracts the attention of both the community in industry and academia, and professional and private Internet users.

To support the widespread use of intelligent information agents for the Internet, the challenges are, among others, to build libraries of reusable software patterns for different types of such agents, and provide corresponding easy-to-use plug-in information agent components to the common user.

A large number of shop bots has been deployed on the Web so far, but still no fielded system of rational information agents capable of sophisticated, trusted decision-making and providing an advanced, comfortable human–agent interaction exists.

Personal assistants may help to reduce the user's reluctance to start doing everyday business on the Internet and Web. Need-driven, but not necessarily technology-lead, products for agents in Web-based user interfaces should allow for shared context and convenient inspection of agent by user to make its activity and impact of feedback more transparent. This might smooth the raised expectations about anthropomorphous agents and avoid any single agent deployed on the Web as a life-like character being just an individual curiosity to the user. Adaptive resource discovery, selection [119] and change management are some of the key topics for future research in this domain.

The use of mobile information agents may be beneficial in terms of performance, network load balancing and customization of distributed applications. However, costs and efforts to ensure data security [25] in open networks might outweigh these benefits. Besides, the discussion of whether mobility is an essential feature for intelligent information agents has not been decided yet. the main application domain of mobile agents is that of future communication systems including high-performance networks and the management of complex telecommunications services [2,50].

And, of course, the future of the Internet and Web itself as it is actually governed by the Internet Engineering Task Force (IETF) group and the Web Consortium as a kind of moral authority, respectively, strongly affects the development of information agents for a broad range of applications on the Internet.

Possible future application scenarios of intelligent information agents for the next decade include the following.

- Year 2002: Use of information agents on mobile appliances, for example, to assist in UMTS cell phone videoconferencing, and dynamic content provision for eBooks, WAP devices, and embedded databases.
- Year 2003: Information agents enable distributed data mining in wide-area networks and automated customer coalition formation at online auctions.
- Year 2010: Agent-based car and traffic management, navigation of flying cars.
- Year 2012: Real-time affective information agents are co-inhabitants of 3D digital cities.
- Year 2015: Information agents control micro-machines based on nano-technology in the healthcare management domain.
- Year 2016: Intelligent information agents contribute to the coordination of ground/space activities on next Mars mission [194].

7. Uncited references

[16,16,19,45,49,54,57,60,70,79,86,89,95,108,137,166,172].

References

[1] AgentLink, ESPRIT network of excellence for agent-based computing, http://www.agentlink.org; see also: AgentLink's special interest group on intelligent information agents, http://www.dfki.de/~klusch/i2a-SIG.html.
[2] S. Albayrak (Ed.), Intelligent Agents for Telecommunications, IOS Press, 1998.
[3] E. André (Ed.), Animated Interface Agents (special double issue), Applied Artificial Intelligence Journal 13(4/5) (1999).
[4] E. André, T. Rist, J. Müller, WebPersona: a life-like presentation agent for the World-Wide Web, Knowledge-based Systems 11 (1) (1998) 25–36.
[5] Y. Arens, C.A. Knoblock, C. Hsu, Query processing in the SIMS information mediator, in: A. Tate (Ed.), Advanced Planning Technology, AAAI Press, 1996.
[6] A. Armstrong, E. Durfee, Mixing and memory: emergent cooperation in an information marketplace, in: Proceedings of ICMAS-98, Paris, IEEE Press, New York, 1998.
[7] Auctions and marketplaces: Onsale – http://www.onsale.com/, FairMarket – http://auctions.fairmarket.com/, eBay – http://www.ebay.com, Kasbah – http://kasbah.media.mit.edu/, AuctionBot – http://auction.eecs.umich.edu/.
[8] D. Ballard, An Introduction to Natural Computing, MIT Press, Cambridge, MA, 1999.

[9] M. Barbuceanu, M.S. Fox, The architecture of an agent building shell, in: M. Wooldridge, et al. (Eds.), Intelligent Agents II, Lecture Notes in Artificial Intelligence, vol. 1037, Springer, Berlin, 1994.
[10] BargainFinder, http://bf.cstar.ac.com/bf.
[11] M. Bauer, D. Dengler, TrIAS – an architecture for trainable information assistants, in: Working Notes of the Autonomous Agents'98 Workshop on Agents in Interaction – Acquiring Competence through Imitation, 1998; also in Working Notes of the AAAI'98 Workshop on AI and Information Integration.
[12] W. Brauer, G. Weiß, Multi-machine scheduling – a multiagent learning approach, in: Proceedings of ICMAS-98, Paris, France, IEEE Press, New York, 1998.
[13] A. Beugnard, et al., Making components contract aware, IEEE Computer 7 (1999).
[14] BotSpot, a collection of Bots on the Web, http://www.botspot.com/.
[16] C. Boutilier, Y. Shoham, M. Wellman, Economic principles of multi-agent systems, AI Journal 94 (1997).
[16] J. Bradshaw, Software Agents, MIT Press, Cambridge, MA, 1997.
[17] B. Brewington, et al., Mobile agents for distributed information retrieval, in: M. Klusch (Ed.), Intelligent Information Agents, Springer, Berlin, 1999 (Chapter 15).
[18] M. Buchheit, F.M. Donini, W. Nutt, A. Schaerf, A refined architecture for terminological systems: terminology = schemas + views, AI Journal 99 (2) (1998).
[19] K. Cavanaugh, Bandwidth's new bargaineers, Technology Review 101 (6) (1998).
[20] L. Cavedon, L. Sonenberg, On social commitment, roles and preferred goals, in: Proceedings of ICMAS-98, Paris, IEEE Press, New York, 1998.
[21] COALA simulation environment for agent coalition formation, http://www.dfki.de/∼klusch/COALA.
[22] J. Contreras, M. Klusch, T. Vielhak, J. Yen, F. Wu, Multi-agent coalition formation in power transmission planning: bilateral Shapley value and kernel approaches, in: Proceedings of the 13th International Power Systems Computation Conference (PSCC-99), Trondheim, Norway, 1999.
[23] CORBA, Object management group (OMG) – common object request broker architecture, references, http://www.cs.wustl.edu/∼schmidt/corba-papers.html.
[24] F. Dai, Virtual Reality for Industrial Applications, Springer, Berlin, 1998.
[25] I. Damgard (Ed.), Lectures on Data Security, in: Lecture Notes in Computer Science, vol. 1561, Springer, Berlin, 1999.
[26] K. Dautenhahn, C. Numaoka (Eds.), Socially Intelligent Agents (special issue), International Journal on Applied Artificial Intelligence 12 (7/8) (1998).
[27] DCOM, Microsoft distributed component object model, http://www.microsoft.com/com/tech/DCOM.asp.
[28] K. Decker, K. Sycara, M. Williamson, Intelligent adaptive information agents, Journal on Intelligent Information Systems 9 (1997) 239–260.
[29] K. Decker, K. Sycara, M. Williamson, Middle-agents for the Internet, in: Proceedings of IJCAI-97, 1997.
[30] J. Delgado, T. Ura, N. Ishii, Content-based collaborative information filtering: actively learning to classify and recommend documents, in: M. Klusch, G. Weiß (Eds.), Proceedings of CIA-98, in: Lecture Notes in Artificial Intelligence, vol. 1435, Springer, Berlin, 1998.
[31] F.M. Donini, M. Lenzerini, D. Nardi, W. Nutt, The complexity of concept languages, Information and Computation 134 (1) (1997).
[32] J.E. Doran, S. Franklin, N.R. Jennings, T.J. Norman, On cooperation in multi-agent systems, The Knowledge Engineering Review 12 (3) (1997).
[33] E. Durfee, Distributed problem solving and planning, in: G. Weiß (Ed.), Multiagent Systems, MIT Press, Cambridge, MA, 1999 (Chapter 3).
[34] E. Durfee, et al., Strategic reasoning and adaptation in an information economy, in: M. Klusch (Ed.), Intelligence Information Agents, Springer, Berlin, 1999 (Chapter 8).
[35] C. Elliott, J. Brzezinski, Autonomous agents as synthetic characters, AI Magazine 19 (2) (1998).
[36] Extempo Inc., http://www.extempo.com/.
[37] T. Finin, R. Fritzson, D. McKay, R. McEntire, KQML as an agent communication language, in: Proceedings of the Third International Conference on Information and Knowledge Management (CIKM-94), ACM Press, New York, 1994.
[38] FIPA, Foundation for intelligent physical agents, http://drogo.cselt.it/fipa/; see also: L. Chiariglione, FIPA – agent technologies achieve maturity, AgentLink Newsletter 1 (1999).
[39] FIPA agent communication language, http://www.fipa.org/spec/fipa99spec.htm.
[40] Firefly network, http://www.firefly.com/.
[41] K. Fischer, Agent-based design of holonic manufacturing systems, Journal of Robotics and Autonomous Systems 27 (1999) 3–13.
[42] M. Fowler, UML Distilled: Applying the Standard Object Modeling Language, Addison-Wesley, Reading, MA, 1997.
[43] H. Garcia-Molina, et al., The TSIMMIS approach to mediation: data models and languages, in: Proceedings of Workshop NGITS, 1995, ftp://db.stanford.edu/pub/garcia/1995/tsimmis-models-languages.ps.

[44] D. Garlan, D. Le Metayer (Eds.), Coordination Languages and Models, Proceedings of the Second International Conference on Coordination-97, in: Lecture Notes in Computer Science, vol. 1282, Springer, Berlin, 1997.
[45] M.R. Genesereth, A.M. Keller, O. Duschka, Infomaster: an information integration system, in: Proceedings of the ACM SIGMOD Conference, 1997.
[46] M.S. Greenberg, et al., Mobile agents and security, IEEE Communications 36 (7) (1998).
[47] G. Grefenstette, Cross-language Information Retrieval, Kluwer, Dordrecht, 1998.
[48] V.N. Gudivada, Information retrieval on the world wide web, IEEE Internet Computing 1 (5) (1997).
[49] R. Guttman, A. Moukas, P. Maes, Agents as mediators in electronic commerce, in: M. Klusch (Ed.), Intelligent Information Agents, Springer, Berlin, 1999 (Chapter 6).
[50] A.L.G. Hayzelden, J. Bingham, Software Agents for Future Communication Systems, Springer, Berlin, 1998.
[51] M. Huhns, M.P. Singh, Social abstractions for information agents, in: M. Klusch (Ed.), Intelligent Information Agents, Springer, Berlin, 1999.
[52] M. Huhns, L.M. Stephens, Multiagent systems and societies of agents, in: G. Weiß (Ed.), Multiagent Systems, MIT Press, Cambridge, MA, 1999 (Chapter 2).
[53] I3Net, ESPRIT network of excellence for intelligent information interfaces, http://www.i3net.org/.
[54] M. Jeusfeld, M.P. Papazoglou, Information brokering, in: M.P. Papazoglou (Ed.), Information Systems Interoperability, Research Studies, Somerset, England, Wiley, Chichester, 1998.
[55] ICMAS: Proceedings of the Third International Conference on Multi-agent Systems, Paris, France, IEES CS Press, 1998.
[56] K. Indermaur, At your service? A state-of-the-art report on electronic agents in e-commerce, Magazine Database Management Systems DBMS 11 (10) (1998).
[57] T. Ishida, et al., Digital city Kyoto: towards a social information infrastructure, in: Proceedings of CIA-99, in: Lecture Notes in Artificial Intelligence, vol. 1652, Springer, Berlin, 1999.
[58] N. Jacobs, R. Shea, Carnot and InfoSleuth – database technology and the WWW, in: ACM SIGMOD International Conference on Management of Data, May 1995; see also: R.J. Bayardo, et al., InfoSleuth: agent-based semantic integration of information in open and dynamic environments, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, 1997.
[59] S. Jha, P. Chalasani, O. Shehory, K. Sycara, A formal treatment of distributed matchmaking, in: Proceedings of the Second International Conference on Autonomous Agents (Agents-98), Minneapolis, MN, USA, 1998.
[60] P. Kandzia, M. Klusch (Eds.), Cooperative Information Agents, Proceedings of the First International Workshop CIA-97, Kiel, Germany, in: Lecture Notes in Artificial Intelligence, vol. 1202, Springer, Berlin, 1997.
[61] V. Kashyap, A. Sheth, Semantic heterogeneity in global information systems: the role of metadata context and ontology, in: M.P. Papazoglou, G. Schlageter (Eds.), Cooperative Information Systems: Trends and Directions, Academic Press, New York, 1998.
[62] S. Ketchpel, Forming coalitions in face of uncertain reward, in: Proceedings of the AAAI-94 Conference, 1994.
[63] KIF: knowledge interchange format, http://logic.stanford.edu/kif/.
[64] W. Kim, et al., On resolving schematic heterogeneity in multidatabase systems, International Journal on Distributed and Parallel Databases 1 (1993) 251–279.
[65] D. Kinny, et al., Planned team activity, in: Lecture Notes in Artificial Intelligence, vol. 830, Springer, Berlin, 1994.
[66] M. Klusch, Cooperative recognition of inter-database dependencies, in: Proceedings of the International Workshop on Description Logics (DL-95), University of Rome, Rome, 1995.
[67] M. Klusch, Intelligent Information Agents, Springer, Berlin, 1999.
[68] M. Klusch, O. Shehory, Coalition formation among rational information agents, in: Proceedings of MAAMAW-96, in: Lecture Notes in Artificial Intelligence, vol. 1038, Springer, Berlin, 1996.
[69] M. Klusch, O. Shehory, A polynomial kernel-oriented coalition algorithm for rational information agents, in: Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96), Kyoto, Japan, AAAI Press, 1996.
[70] M. Klusch, G. Weiß (Eds.), Cooperative Information Agents II, Proceedings of the Second International Workshop CIA-98, Paris, France, in: Lecture Notes in Artificial Intelligence, vol. 1435, Springer, Berlin, 1998.
[71] M. Klusch, O. Shehory, G. Weiß (Eds.), Cooperative Information Agents III, Proceedings of the Third International Workshop CIA-99, Uppsala, Sweden, in: Lecture Notes in Artificial Intelligence, vol. 1652, Springer, Berlin, 1999.
[72] J. Komorowski, J. Zytkow (Eds.), Principles of Data Mining and Knowledge Discovery, Proceedings of the First European Symposium PKDD-97, Trondheim, Norway, 1997.
[73] S. Krause, T. Magedanz, Mobile service agents enabling intelligence on demand in telecommunications, in: Proceedings of IEEE Global Communications Conference (GLOBCOM-96), 1996, http://www.tinac.com.
[74] D. Kuokka, L. Harrada, On using KQML for matchmaking, in: Proceedings of the Third International Conference on Information and Knowledge Management (CIKM-95), AAAI/MIT Press, 1995, pp. 239–245.
[75] S. Lawrence, C.L. Giles, Searching the world wide web, Science 280 (1998).
[76] M. Lenz, et al., Case-based Reasoning Technology, in: Lecture Notes in Artificial Intelligence, vol. 1400, Springer, Berlin, 1998.

[77] M. Lewis, Designing for human–agent interaction, AI Magazine 19 (2) (1998).
[78] H. Lieberman, Personal assistants for the Web: an MIT perspective, in: M. Klusch (Ed.), Intelligent Information Agents, Springer, Berlin, 1999 (Chapter 12).
[79] LikeMinds, http://www.likeminds.com/technology/.
[80] R.M. Losee, Text Retrieval and Filtering, Kluwer, Dordrecht, 1998.
[81] S. Luke, L. Spector, D. Rager, J. Hendler. Ontology-based Web agents, in: Proceedings of the First International Conference on Autonomous Agents (Agents-97), Marina del Rey, USA, 1997.
[82] MAF, OMG's mobile agent facility (MAF), http:// www.omg.org/library/schedule/CF-RFP3.htm.
[83] N. Matos, C. Sierra, N. Jennings, Determining successful negotiation strategies: an evolutionary approach, in: Proceedings of ICMAS-98, Paris, IEEE Press, New York, 1998.
[84] M.T. Maybury, W. Wahlster, Readings in Intelligent User Interfaces, Morgan Kaufmann, Los Altos, CA, 1998.
[85] R.P. McAfee, J. McMillan, Auctions and bidding, Journal of Economic Literature 25 (1987).
[86] L.W. McKnight, J.P. Bailey, Internet economics: when constituencies collide in cyberspace, Internet Computing November/ December (1997).
[87] F. Menczer, A.E. Monge, Scalable Web search by adaptive online agents: an InfoSpiders case study, in: M. Klusch (Ed.), Intelligent Information Agents, Springer, Berlin, 1999 (Chapter 14).
[88] J.S. Mertoguno, W. Lin, Distributed knowledge-base: adaptive multi-agents approach, in: Proceedings of the International Joint Symposia on Intelligence and Systems, Rockville, MD, USA, November 1996.
[89] Metadata: IEEE Metadata Conferences; ISO TC 46/SC 9 (Standards for presentation, classification and description of documents); W3C metadata and resource description, http://www.w3.org/Metadata/; Dublin core meta-data system, http://orc.rsch.oclc.org:9016/dcms/.
[90] T. Mitchell, Machine Learning, McGraw-Hill, New York, 1997.
[91] A. Moukas, G. Zacharia, P. Maes, Amalthaea and Histos: multiagent systems for WWW sites and reputation recommendation, in: M. Klusch (Ed.), Intelligent Information Agents, Springer, Berlin, 1999 (Chapter 13).
[92] J.P. Müller, M. Singh, A.S. Rao (Eds.), Intelligent Agents V, Proceedings of the Fifth International Workshop on Agent Theories, Architectures and Languages (ATAL), in: Lecture Notes in Artificial Intelligence, vol. 1555, Springer, Berlin, 1999.
[93] NetSage sage, http://www.netsage.com.
[94] M. Nodine, J. Fowler, An overview of active information gathering in Infosleuth, in: Proceedings of the International Conference on Autonomous Agents, USA, 1999.
[95] P. Noriega, C. Sierra (Eds.), Agent-mediated Electronic Commerce, Proceedings of the First International Conference on Agent-mediated Electronic Trading (AMET-98), in: Lecture Notes in Artificial Intelligence, vol. 1571, Springer, Berlin, 1998.
[96] S. Ossowski, Co-ordination in Artificial Agent Societies, in: Lecture Notes in Artificial Intelligence, vol. 1535, Springer, Berlin, 1999.
[97] A. Ouksel, A framework for a scalable agent architecture of cooperating heterogeneous knowledge sources, in: M. Klusch (Ed.), Intelligent Information Agents, Springer, Berlin, 1999 (Chapter 5).
[98] PAAM, Proceedings of the Fourth International Conference on Practical Applications of Intelligent Agents and Multi-agent Technology, London, UK, PA Company Ltd., 1999.
[99] PAN, Planning assistant for the Net, http://www.dfki.de/~bauer/PAN/.
[100] M.P. Papazoglou, G. Schlageter, Cooperative Information Systems: Trends and Directions, Academic Press, New York, 1998.
[101] M.P. Papazoglou, S. Laufmann, T.K. Sellis, An organizational framework for cooperating intelligent information systems, International Journal of Cooperative Information Systems 1 (1) (1992) 169–202.
[102] A.P. Pentland, Wearable intelligence, Scientific American 9 (4) (1998).
[103] V.A. Pham, A. Karmouch, Mobile software agents: an overview, IEEE Communications July (1998).
[104] R.W. Picard, Affective Computing, MIT Press, Cambridge, 1997; see also: http://www-white.media.mit.edu/vismod/demos/affect/.
[105] E. Pitoura, B. Bhargava, A framework for providing consistent and recoverable agent-based access to heterogeneous mobile databases, Journal ACM SIGMOD Record 24 (3) (1995).
[106] M. Plu, Software technologies for building agent based systems in telecommunication networks, in: N.R. Jennings, M. Wooldridge (Eds.), Agent Technology, Springer, Berlin, 1998.
[107] M.V.N. Prasad, V. Lesser, Learning situation-specific coordinating in cooperative multi-agent systems, International Journal on Autonomous Agents and Multi-agent Systems (1999).
[108] Proceedings of the Workshop on Deception, Fraud and Trust in Agent Societies, in: International Conference on Automous Agents (Agents-98), Minneapolis/St.Paul, USA, 1998; see also: Proceedings of the Workshop "Deception, Fraud and Trust", in: International Conference on Autonomous Agents (Agents-99), 1999.
[109] Proceedings of the Workshop on Agents Learning About, From and With other Agents, in: 16th International Joint Conference on Artificial Intelligence (IJCAI-99), 1999.
[110] Resource description framework (RDF) schema specification, http://www.w3.org/TR/WD-rdf-schema/.

[111] RMI and Java distributed computing, SunSoft white paper, 1997, http://www.javasoft.com/features/1997/nov/rmi.html.
[112] T. Sandholm, TRACONET – an implementation of the contract net protocol based on marginal cost calculations, Journal on Group Decision and Negotiation (1996).
[113] T. Sandholm, Limitations of the Vickrey auction in computational multiagent systems, in: Proceedings of ICMAS-96.
[114] T. Sandholm, Unenforced e-commerce transactions, Internet Computing November/December (1997).
[115] T. Sandholm, V.R. Lesser, Coalitions among computationally bounded agents, AI Journal 94 (1997).
[116] G. Salton, Automatic Text Processing: the Transformation, Analysis and Retrieval of Information by Computer, Addison Wesley, Reading, MA, 1989.
[117] M. Schillo, P. Funk, M. Rovatsos, Who can you trust: dealing with deception, in: R. Falcone (Ed.), Proceedings of the Workshop "Deception, Fraud and Trust" of the Autonomous Agents Conference, 1999.
[118] J.R. Searle, Speech Acts, Cambridge University Press, Cambridge, 1969; see also: Collective intentions and actions, in: Intentions in Communication, MIT Press, Cambridge, MA, 1990 (Chapter 19).
[119] S. Sen, A. Biswas, S. Ghosh, Adaptive choice of information sources, in: M. Klusch (Ed.), Intelligent Information Agents, Springer, Berlin, 1999 (Chapter 11).
[120] S. Sen, M. Sekaran, Individual learning of coordination knowledge, Learning in DAI Systems (special issue), Journal of Experimental and Theoretical AI (1998).
[121] S. Sen, G. Weiß, Learning in multiagent systems, in: G. Weiß (Ed.), Multiagent Systems, MIT Press, Cambridge, MA, 1999 (Chapter 6).
[122] O. Shehory, Spawning information agents on the Web, in: M. Klusch (Ed.), Intelligent Information Agents, Springer, Berlin, 1999 (Chapter 17).
[123] O. Shehory, S. Kraus, Feasible formation of coalitions among autonomous agents in non-super-additive environments, Computational Intelligence 15 (3) (1998).
[124] A. Sheth, E. Mena, A. Illaramendi, V. Kashyap, OBSERVER: an approach for query processing in global information systems based on interoperation across pre-existing ontologies, in: Proceedings of the International Conference on Cooperative Information Systems (CoopIS-96), IEEE Computer Society Press, Silver Spring, MD, 1996.
[125] A. Sheth, A. Illaramendi, V. Kashyap, E. Mensa, Managing multiple information sources through ontologies: relationship between vocabulary heterogeneity and loss of information, in: Proceedings of ECAI-96, 1996.
[126] Y. Shoham, M. Tennenholtz, On the emergence of social conventions: modeling, analysis, and simulations, AI Journal 94 (1997).
[127] Smart cards, http://www.visa.com/nt/chip/main.html.
[128] SMIL (synchronized multimedia integration language), W3C rec. status, http://www.w3.org/TR/REC-smil.
[129] L. Steels, The origins of ontologies and communication conventions in multi-agent systems, Journal on Autonomous Agents and Multi-agent Systems 1 (2) (1998).
[130] D.G. Sullivan, et al., Intention reconciliation in the context of teamwork: an initial empirical investigation, in: M. Klusch, O. Shehory, G. Weiß (Eds.), Cooperative Information Agents III, Proceedings of the Third International Workshop CIA-99, Uppsala, Sweden, in: Lecture Notes in Artificial Intelligence, vol. 1652, Springer, Berlin, 1999.
[131] K. Sycara, Levels of adaptivity in systems of coordinating information agents, in: M. Klusch, G. Weiß (Eds.), Cooperative Information Agents II, Proceedings of the Second International Workshop CIA-98, Paris, France, in: Lecture Notes in Artificial Intelligence, vol. 1435, Springer, Berlin, 1998.
[132] K. Sycara, In-context information management through adaptive collaboration of intelligent agents, in: M. Klusch (Ed.), Intelligent Information Agents, Springer, Berlin, 1999 (Chapter 4).
[133] K. Sycara, D. Zeng, Coordination of multiple intelligent software agents, International Journal of Cooperative Information Systems 5 (2/3) (1996) 181–211.
[134] K. Sycara, D. Zeng, Benefits of learning in negotiation, in: Proceedings of AAAI-97, Providence, USA, 1997.
[135] K. Sycara, M. Klusch, J. Lu, S. Widoff, Dynamic service matchmaking among agents in open information environments, Semantic Interoperability in Global Information Systems (special issue), Journal ACM SIGMOD Record (1999); see also: K. Sycara, J. Lu, M. Klusch, Interoperability among heterogeneous software agents on the Internet, technical report CMU-RI-TR-98-22, CMU, Pittsburgh, USA, 1998.
[136] M. Tambe, Implementing agent teams in dynamic multiagent environments, Applied AI Journal 12 (2/3) (1998).
[137] J.D. Thomas, K. Sycara, Heterogeneity, stability, and efficiency in distributed systems, in: Proceedings of ICMAS-98, Paris, IEEE Press, New York, 1998.
[138] R. Tolksdorf (Ed.), Coordination Mechanisms and Patterns for Web Agents (special issue), Journal on Autonomous Agents and Multi-agent Systems (1999); see also in: M. Klusch (Ed.), Intelligent Information Agents, Springer, Berlin, 1999 (Chapter 16).
[139] C. Tschudin, Mobile agent security, in: M. Klusch (Ed.), Intelligent Information Agents, Springer, Berlin, 1999 (Chapter 18).
[140] K. Webster, K. Paul, Beyond surfing: tools and techniques for searching the Web, Information Technology January (1996).
[141] G. Vigna (Ed.), Mobile Agents and Security, in: Lecture Notes in Computer Science, vol. 1419, Springer, Berlin, 1998.
[142] R. Weihmayer, H. Velthuijsen, Intelligent agents in telecommunications, in: N.R. Jennings, M. Wooldridge (Eds.), Agent Technology, Springer, Berlin, 1998.

[143] G. Weiß, Multiagent Systems, MIT Press, Cambridge, MA, 1999.
[144] G. Weiß (Ed.), Distributed Artificial Intelligence Meets Machine Learning, selected papers from ECAI-96 Workshop LDAIS and ICMAS-96 Workshop LIOME, in: Lecture Notes in Computer Science, vol. 1221, Springer, Berlin, 1997.
[145] G. Weiß, S. Sen (Eds.), Adaptation and Learning in Multi-agent Systems, Proceedings of IJCAI-95 Workshop, CA, in: Lecture Notes in Computer Science, vol. 1042, Springer, Berlin, 1995.
[146] M. Wellman, Market-oriented programming: some early lessons, in: S.H. Clearwater (Ed.), Market-based Control: A Paradigm for Distributed Resource Allocation, World Scientific, Singapore, 1995.
[147] M. Wellman, P.R. Wurman, W.E. Walsh, The Michigan Internet AuctionBot: a configurable auction server for human and software agents, in: Proceedings of the Second Conference on Autonomous Agents, 1998.
[148] M. Wellman, P.R. Wurman, W.E. Walsh, Flexible double auctions for electronic commerce: theory and implementation, in: Decision Support Systems, 1998.
[149] G. Wiederhold, Mediators in the architecture of future information systems, IEEE Computer 25 (1992).
[150] WIDL, the W3C Web interface definition language, http://www.w3.org/TR/NOTE-widl.
[151] E. Wolfstetter, Auctions: an introduction, Journal of Economic Surveys 10 (4) (1996) 367–420.
[152] M. Wooldridge, N.R. Jennings, Intelligent agents: theory and practice, Knowledge Engineering Review 10(2) (1995); see also: M. Wooldridge, Intelligent agents, in: G. Weiß (Ed.), Multiagent Systems, MIT Press, Cambridge, MA, 1999 (Chapter 1).
[153] M. Wooldridge, N. Jennings, Agent Technology, Springer/Unicom, Berlin, 1998.
[154] WordNet – a lexical database for English, http://www.cogsci.princeton.edu/∼wn/.
[155] XML, Extensible markup language, World Wide Web Consortium (W3C) working draft, November 1997, http://www.w3.org/TR/WD-xml-link.
[156] D. Zeng, K. Sycara, Dynamic supply chain structuring for electronic commerce among agents, in: M. Klusch (Ed.), Intelligent Information Agents, Springer, Berlin, 1999 (Chapter 10).
[157] P.C. Weinstein, W.P. Birmingham, Creating ontological metadata for digital library content and services, International Journal on Digital Libraries 2 (1) (1998).
[158] A. Sheth, J.A. Larson, Federated database systems, ACM Computing Surveys 22 (3) (1990).
[159] I. Venieris, H. Hussmann, Intelligent Broadband Networks, Wiley, New York, 1998.
[160] A. Chavez, D. Dreilinger, R. Guttman, P. Maes, A real-life experiment in creating an agent marketplace, in: Proceedings of PAAM-97, London, UK, April 1997.
[161] T.R. Gordin, N. Puppala, S. Sen, Evolving cooperative groups: preliminary results, AAAI technical report WS-97-03, in: Proceedings of the AAAI Workshop on Multiagent Learning, 1997.
[162] R. Gruber, Ontolingua: a mechanism to support portable ontologies, Knowledge Systems Laboratory, Stanford University, technical report KSL-91-66, March 1992; see also: A translation approach to portable ontology specifications, Knowledge Acquisition 5(2) (1993) 199–220.
[163] R. Axelrod, The Evolution of Cooperation, Harper Collins, UK, 1984.
[164] M. Balabonovic, Y. Shoham, Combining content-based and collaborative recommendation, Communications of the ACM March (1997).
[165] D. Wong, N. Paciorek, T. Walsh, J. DiCelie, M. Young, B. Peet, Concordia: an infrastructure for collaborating mobile agents, in: Lecture Notes in Computer Science, vol. 1219, Springer, Berlin, 1997.
[166] T. Magedanz, M. Breugst, I. Busse, S. Covaci, Integrating mobile agent technology and CORBA middleware, AgentLink Newsletter 1 (November) (1998), http://www.agentlink.org.
[167] T.E. Truman, T. Pering, R. Doering, R.W. Brodersen, The InfoPad multimedia terminal: a portable device for wireless information access, IEEE Transactions on Computers 47 (10) (1998).
[168] D.B. Lange, M. Oshima, Programming and Deploying Java Mobile Agents with Aglets, Addison Wesley, Reading, MA, 1998.
[169] H. Peine, T. Stolpmann, The architecture of the ARA platform for mobile agents, in: K. Rothermel, R. Populescu-Zeletin (Eds.), Mobile Agents, Proceedings of the First International Workshop on Mobile Agents (MA-97), in: Lecture Notes in Computer Science, vol. 1219, Springer, Berlin, 1997.
[170] Cornell University, USA, http://www.cs.uit.no/DOS/Tacoma/index.html.
[171] Institute for Parallel and Distributed Computer Systems, University of Stuttgart, http://www.informatik.uni-stuttgart.de/ipvr/vs/projekte/mole.html.
[172] ObjectSpace Voyager Inc., USA, http://www.objectspace.com/voyager/.
[173] Mitsubishi Concordia, http://www.meitca.com/HSL/Projects/Concordia/.
[174] D. Rus, R. Gray, D. Kotz, Transportable information agents, in: M. Huhns, M.P. Singh (Eds.), Readings in Agents, Morgan Kaufmann, Los Altos, CA, 1998; see also: Journal of Intelligent Information Systems 9 (1997) 215–238.
[175] T. Tesch, P. Fankhauser, Arbitration and matchmaking for agents with conflicting interests, in: M. Klusch, O. Shehory, G. Weiß (Eds.), Cooperative Information Agents III, Proceedings of the Third International Workshop CIA-99, Uppsala, Sweden, in: Lecture Notes in Artificial Intelligence, vol. 1652, Springer, Berlin, 1999.

[176] J. Cho, H. Garcia-Molina, L. Page, Efficient crawling through URL ordering, in: Proceedings of the Seventh WWW Conference, 1998, http://www7.scu.edu.au/programme/fullpapers/1919/com1919.htm.
[177] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison Wesley, Reading, MA, 1999.
[178] W.B. Frakes, R. Baeza-Yates, Information Retrieval, Prentice Hall, Englewood Cliffs, NJ, 1992.
[179] C.M. Bowman, et al., Harvest: a scalable, customizable discovery and access system, technical report CU-CS 732-94, University of Colorado, Boulder, USA, 1994.
[180] V. Kashyap, A. Sheth, Semantic information brokering, Proceedings of the Conference on Information and Knowledge Management (CIKM), 1994.
[181] C.M. Jonker, M. Klusch, J. Treur, Design of collaborating information agents, in: M. Klusch, L. Kerschberg (Eds.), Proceedings of CIA-2000 Workshop, in: Lecture Notes in Artificial Intelligence, vol. 1860, Springer, Berlin, 2000.
[182] MIX XML mediator agent, http://www.db.ucsd.edu/projects/MIX/.
[183] J. Leigh, et al., Visualization in teleimmersive environments, IEEE Computer 12 (1999).
[184] T. Ishida, K. Isbister (Eds.), Digital Cities – Technologies, Experiences, and Future Perspectives, in: Lecture Notes in Computer Science, vol. 1765, Springer, Berlin, 2000.
[185] E. Bonabeau, G. Theraulaz, Swarm smarts, Scientific American March (2000).
[186] Q. Huai, T. Sandholm, Nomad: mobile agent system for an Internet-based auction house, Internet Computing March/April (2000).
[187] T. Papaioannou, Mobile information agents in cyberspace – state of the art and challenges, in: M. Klusch, L. Kerschberg (Eds.), Proceedings of CIA-2000 workshop, in: Lecture Notes in Artificial Intelligence, vol. 1860, Springer, Berlin, 2000.
[188] S. Lawrence, C.L. Giles, Accessibility on the Web, Nature 400 (6) (1999) 740.
[189] V.S. Subrahmanian, et al., Heterogeneous Agent Systems: Theory and Implementation, MIT Press, Cambridge, USA, 2000.
[190] V. Lesser, et al., Resource-bounded searches in an information marketplace, IEEE Internet Computing March/April (2000).
[191] R. Keeney, H. Raiffa, Decisions with multiple objectives: preferences and value tradeoffs, Wiley, Chichester, 1976.
[192] M. Klusch, Agent-mediated trading: intelligent agents and e-business, in: A.L.G. Hayzelden, R. Bourne (Eds.), Agent Technology Applied to Networked Systems, Wiley, Chichester, 2000.
[193] A. Omicini, F. Zambonelli, M. Klusch, R. Tolksdorf (Eds.), Coordination of Internet Agents: Models, Technologies and Applications, Springer, Berlin, 2000.
[194] W. Truszkowski, H. Hallock, J. Kurien, Agent technology from a NASA perspective, in: M. Klusch, O. Shehory, G. Weiss (Eds.), Proceedings of the Third International Workshop on Cooperative Information Agnets (CIA'99), in: Lecture Notes in Artificial Intelligence, vol. 1652, Springer, Berlin, 1999.
[195] J.P. Kahan, A. Rapoport, Theories of Coalition Formation, Lawrence Erlbaum, Hillsdale, NJ, 1984.

Matthias Klusch is a senior researcher in the Multi-Agent Systems Group at the German Research Center for Artificial Intelligence and an Assistant Professor in the Artificial Intelligence Department at Free University of Amsterdam, Netherlands. He coordinates the Special Interest Group on Intelligent Information Agents as part of the European Network of Excellence for Agent-Based Computing (AgentLink), and is the founder and general chair of the international workshop series on cooperative information agents (CIA). He holds a Ph.D. in computer science from the University of Kiel, Germany, and spent 1998 as a visiting research scientist at Carnegie Mellon University in Pittsburgh, USA, and organized the second European Agent Systems Summer School (EASSS 2000) as local co-chair. Dr. Klusch is a co/editor of seven books, co/author of numerous papers, presenter of several tutorials, program committee member of major conferences in the field, and is on the editorial board of the Semantic Web journal. His past and current research focuses on the application of artificial intelligence and agent technology to advanced databases and intelligent information systems in the Internet and Web.