INTERACTIVE INFORMATION AGENTS AND INTERFACES

Matthias Klusch
German Research Center for Artificial Intelligence

Elisabeth Andre
University of Augsburg

Thomas Rist
German Research Center for Artificial Intelligence

Toru Ishida
Hideyuki Nakanishi
University of Kyoto

INTRODUCTION

Research and development in the domain of intelligent information agents aims at providing innovative solutions to the challenges of the current and future information landscape, which is largely dominated by Internet-based, pervasive computing infrastructures, and huge volumes of heterogeneous, volatile, redundant, and geographically dispersed data. The main challenges are how to discover, integrate, and provide relevant information to the user in a flexible, knowledge-based, scalable, and resource-efficient way. Prominent but limited solution approaches to this problem include federated database systems to integrate heterogeneous, structured data; Web indices, directories, and search engines; and data warehouses and distributed data mining systems to analyze massive, distributed data sets. *Intelligent information agents* rely on the paradigm of agent-based computing (Luck, McBurney, & Preist, 2003; Weiss, 1999) and advances in artificial intelligence. They have been originally invented to significantly leverage database and information systems by adding one new characteristic feature: intelligent autonomous behavior (Klusch, 2001a, 2001b; Klusch & Sycara, 2001; Papazoglou, Laufmann, & Sellis, 1992). It is fair to say that, most information agents to date have been developed mainly in response to the information overload problem of the common user of the Internet and the Web, as seen from both the technological and human user perspective. In this chapter, we focus on the latter perspective and review the current state of the art in developing *interactive information agents*.

Such agents are supposed to be able to interact with their users by means of appropriate intelligent user interfaces. They promise their users the benefit of more rapid task completion with less work, but more personal and natural interaction, and tailoring the content and form of the interaction to the situated context of both the application and the user. Currently, there are many interactive information agent systems and prototypes available and situated within a variety of domains. For reviews of the field, we refer the interested reader to, for example, Mladenic (1999).

The remainder of this chapter is structured as follows. In the following section, we briefly introduce the notions of interactive information agent and interface agent, and provide selected examples of techniques for visualizing information and virtual characters for interface agents. Main approaches to and prominent examples of conversation modeling, audiovisual embodiment, the modeling of personality and emotion, and behavior control of interface agents are then surveyed in detail in later sections. Then, prominent infrastructure architectures and tools for creating such agents are reviewed. Selected criteria for evaluating interface agents are also discussed.

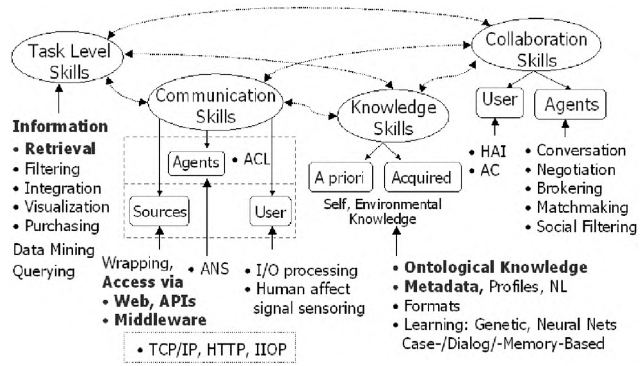


FIGURE 12.1. Required classes of skills of information agents.

INTERACTIVE INFORMATION AGENTS

Information Agents in Brief

An *information agent* is a computational software entity, a special kind of an intelligent agent, that is able to access one or multiple distributed and heterogeneous information sources available, and proactively acquire, mediate, and maintain relevant information on behalf of its user(s) or other agents preferably just in time. In other words, information agents are supposed to cope with the difficulties associated with the information overload of the user. This requires an information agent to semantically broker information by

- Providing a proactive resource discovery
- Resolving the information impedance of information consumers and providers
- Offering value-added information services and products to its users or other agents

Information agents that are currently deployed in the Internet and the Web (Klusch, 2001b; Klusch, Bergamaschi, Edwards, & Petta, 2003; Montaner, Lopez, & De La Rosa, 2003) may be further classified according to one or more of the following features:

- *Noncooperative or cooperative* information agent, depending on the ability to cooperate with other agents for the execution of its tasks. Several protocols and methods are available for achieving cooperation among autonomous information agents in different scenarios, such as hierarchical task delegation, contracting, and decentralized negotiation.
- *Adaptive* information agents are able to adapt themselves to changes in networks and information environments (see Stephanidis et al., chap. 14, this volume). Examples of such agents are learning personal assistants on the Web.
- *Rational* information agents behave utilitarian in an economic sense. They are acting, and may even collaborate together, to increase their own benefits. The main application domains of such agents are automated trading and electronic commerce in the Internet. Examples include the variety of shop bots and systems for agent-mediated auctions on the Web.
- *Mobile* information agents are able to travel autonomously through the Internet. Such agents enable, dynamic load balancing in large-scale networks, reduction of data transfer among information servers, and migration of small business logic within medium-range corporate intranets on demand.

Regarding required skills of information agents we can differentiate between communication, knowledge, collaboration, and task-related skills (Fig. 12.1). Communication skills of an information agent refer to its ability to communicate with information systems and databases, human users, or other agents. Standardized agent communication languages, such as FIPA Agent communication language (ACL) and knowledge query and manipulation language (KQML), have to be considered on top of, for example, common middleware platforms or specific application programming interfaces (APIs). The proper application of techniques for representing and processing ontological knowledge, metadata, profiles and natural language input, translating data formats, and machine learning techniques will enable information agents to acquire and maintain knowledge about itself and its environment. High-level collaboration with other agents can be implemented, for example, by suitable coordination techniques such as service brokering, matchmaking, negotiation, and social filtering. Whereas collaboration with its human users requires the application of techniques borrowed from the domain of human–agent interaction and affective computing. Klusch (2001a) provides a more comprehensive survey of key enabling techniques for developing different kinds of intelligent information agents.

What Are Interactive Information Agents?

From the notion of information agent introduced in the previous section, it is intuitively clear that such agents are supposed to interact with the human user in due course of their acquisition, mediation, and management of information. Consequently, the term *interactive information agent* puts special emphasis on the ability of such an agent to interact with its users by means of some kind of intelligent user interface.

Intelligent user interfaces (Maybury & Wahlster, 1998), are human–machine interfaces that aim to improve the efficiency, effectiveness, and naturalness of human–machine interaction by representing, reasoning, and acting on models of the user, domain, task, discourse, and media, including graphics, natural language, and gesture. Such user interfaces are multifaceted, in purpose and nature, and include capabilities for multimedia input analysis, multimedia presentation generation, and the use of user, discourse, and task models to personalize and enhance interaction. In particular, adaptive user interfaces are designed to put special emphasis on the support of dynamic changes in the system, environment, and user requirements.

Agents that are essential parts of the functionality of intelligent user interfaces are called *interface agents*. Unfortunately, in the literature the term "interface agent" is quite overloaded with different meanings. Depending on its kind of interaction and architecture, one interface agent may be incorporated into different information agent systems. The functionality of the user interface of an interactive information agent may rely on the functionality of one or multiple special interface agents, or may not be agent based at all. Thus, an interface agent may be developed as an integral part of, but in contrast to Sheth and Maes (1993), is not equal to an interactive information agent.

As shown in Fig. 12.2, we distinguish between interactive user interfaces of information agent systems with interface agents or none at all. In the first case, we further differentiate between nonembodied interface agents, such as natural language dialogue agents, and more popular embodied conversational agents, such as avatars or face-to-face dialogue partner agents. In the case of embodied conversational agent-based user interfaces, the interface agent makes use of various communicative behaviors, such as mimics, body posture, speech, and intonation. To summarize, an interactive information agent is an information agent that is able to effectively and efficiently apply appropriate techniques and methods for multimodal interaction and visualization of relevant information. It provides a more natural environment to its user with which to interact in a given application, allowing a broad range of input devices. The kind of user–agent interaction ideally should produce the impression of an observer as if the user is interacting with some reasonably smart, believable human interlocutor. Evaluation of such interaction may rely on quantitative measures, such as time to perform tasks and accuracy of tasks, as well as qualitative measures, such as user indication of utility, ease of use, believability, and naturalness of the agent behavior and interaction.

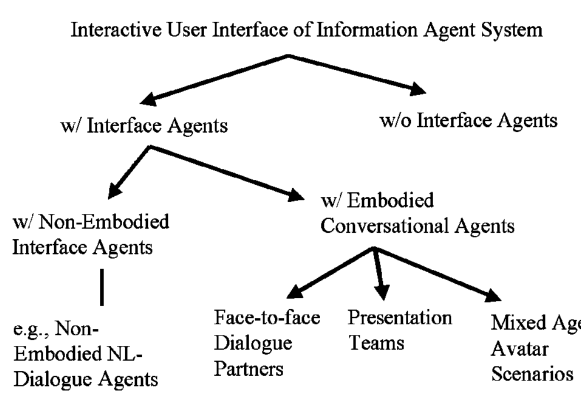


FIGURE 12.2. Classification of information agent user interfaces.

SELECTED USER INTERFACES WITHOUT CONVERSATIONAL AGENTS

Information agent systems are supposed to assist the user in searching, browsing, and modifying his or her rich personal information space. As mentioned in the previous section, this can be achieved by appropriate collaboration between information agents of the system merely focusing on processes related to information retrieval, mediation, and management, and special interface, or user agents that are presenting the results and interacting with the user of the system. That requires both information agents and interface agents to effectively cooperate within the system for proactive information search and provision. Alternatively, many tasks that are related to user interaction such as querying the system, manipulation of the results, administration and workflow of the system, creation and management of user profiles can be pursued by appropriate nonagent-based user interfaces, such as those of advanced database management and information systems.

In the following, we briefly survey prominent techniques for information visualization used in user interfaces, and then present examples of different types of user interfaces of selected agent-based information systems for the Internet and the Web. Some of these interfaces are based on interface agents, some of them are not, and some rely on the combined use of interface agents and administration tools, such as MOMIS, Xyleme, and VisualHarness, respectively. However, these user interfaces do not use special embodied conversational agents that make use of various communicative behaviors, such as mimics, body posture, speech, and intonation.

Information Visualization Techniques

A variety of tools and widgets for visual query formulation and visual display of results have been developed since the mid-1990s; they are to be combined with successful strategies of direct manipulation by the user. For a comprehensive introduction to the field of information visualization, we refer the reader to, for example, Spence (2000) and Card, Mackinlay, and Shneiderman (1999). In general, the main tasks of visualization techniques are to enable the user to

- Gain an overview of the entire information item collection.
- Zoom in on items of interest, and filter out uninteresting items.
- Get details of single or multiple items on demand, and view the relationships among items.

- Keep a history of actions to support undo, replay, and progressive refinement.
- Extract subcollections and query parameters. Different techniques are appropriate for visualizing different types of data such as timelines, histories of data (temporal); linear data, text, and lists (2D); planar data, images, maps, and layouts (2D); volumetric data, 3D images, and solid models (3D); many attributes, relational, and statistical data (M-D); networks (graph); and hierarchical data (tree).

Prominent techniques to visualize structured data include trees, fish-eye views, perspective walls, force directed placement, information spheres, and statistical clustering. Hierarchical and tree-structured data are often visualized by means of hyperbolic trees, cone trees, and space trees; they are space limited, although familiar and appealing to the user.

Force-directed placement-based visualization of networks is treating links between nodes as spring forces; nodes are initially placed either randomly or with some nodes in specified locations. Repulsive spring forces between nodes are then simulated to bring connected nodes into proximity with one another. This may yield a reasonable density of nodes to be displayed, unraveling too dense hyperstructures.

The spectrum of 3D visualizations encompasses chart junk 3D via bar charts, pie charts, and histograms; desktop 3D visualization of medical, architectural, and scientific data, or artificial worlds such as Microsoft's TaskGallery, file cabinets, shopping malls, and digital cities; and immersive virtual environments.

When visualizing artificial worlds to the user, his or her desktop screen could become, for example, a long gallery with shelves, books, or paintings on the walls that represent different tasks. In such a virtual world, the user moves quickly and easily from one to another space and task with a simple series of mouse and keyboard commands. A survey on interactive visual interfaces including those that can be used on personal digital assistants is provided, for example, in Murtagh, Taskaya, Contreras, and Mothe (2003). Workspaces for user–system interaction include, for example, 2D/3D high-resolution desktop or wall-size displays, that present perceptually rich and information-abundant displays, and 3D-immersive virtual environments with head-mounted stereo display, and body tracking. One very useful online library of information visualization environments is OLIVE.

Selected Examples of Information Agent User Interfaces

In this section, we briefly present selected examples of user interfaces of information agents and system. We particularly focus on the type of user interaction, visualization of information, and the use of interface agents.

USC Travel Assistant Agent Interface. Heracles and Theseus, two information-gathering and monitoring tools, have been developed at the Information Sciences Institute of the University of Southern California (USC) (Ambite et al., 2002). ***Heracles*** is a hierarchical constraint planner that aids in interactive itinerary development by showing how a particular choice (e.g., destination airport) affects other choices (e.g., possible modes of transportation, available airlines). Heracles builds on an information agent platform, called ***Theseus***, which provides the technology for efficiently executing agents for information-gathering and monitoring tasks. One example of an information agent based on Theseus and Heracles tools, is the *USC Travel Assistant*, which is capable of interactive planning and monitoring the travel plans of its user. The travel assistant agent organizes the process of trip planning and the associated information hierarchically. The left pane of Fig. 12.3 shows this hierarchical structure, with the particular choices made for the current plan. In this example, the roundtrip consists of three tasks: flying to the meeting, staying at a hotel at the meeting location, and flying back home. Some tasks are further divided into subtasks, for example, how to get to and from the airport when flying. In Heracles, these tasks are represented by related slots and constraints that are encapsulated into template units, which are organized hierarchically.

The travel assistant helps the user evaluate trade-offs that involve many different pieces of information and calculations. The system actively maintains the dependencies among slots such that changes of earlier decisions are automatically propagated throughout the travel planning process. The result is displayed to the user as shown, for example, in Fig. 12.4.

Once the user has completed the planning of the trip, Heracles then generates a set of agents, which are executed in Theseus, for monitoring the trip. If a planned flight has been cancelled, the user is notified via the e-mail operator. For each pending flight, the agent checks if the arrival time of the flight is later than 5 p.m. and, if so, uses the fax operator to notify the hotel (it only does this once). It then compares the current departure time with the previously retrieved departure time. If they differ by a given threshold, the agent does three things: (a) it faxes a message to the car rental agency to notify them of the delay, (b) it updates its local database with the new departure time (to track future changes), and (c) it e-mails the user. These agents interact with the user through e-mail, telephone, fax, or text messaging. These capabilities are provided by a set of specialized agents, such as a fax agent and an e-mail agent. The system does not provide any additional means of visualization and interaction with the user.

Xyleme System User Interface. Xyleme developed by Xyleme SA* is a large-scale distributed extensible markup language (XML) data repository, that provides services such as document loading, versioning and change notification, querying and data integration using views. That is, Xyleme is mostly a "server software," the user interface part is left for the client applications that access the repository. The only exceptions are the QueryShell and the ViewBuilder graphical tools (Java Swing), provided by Xyleme. *QueryShell* enables an "expert" user to query the repository.

*http://www.xyleme.com.

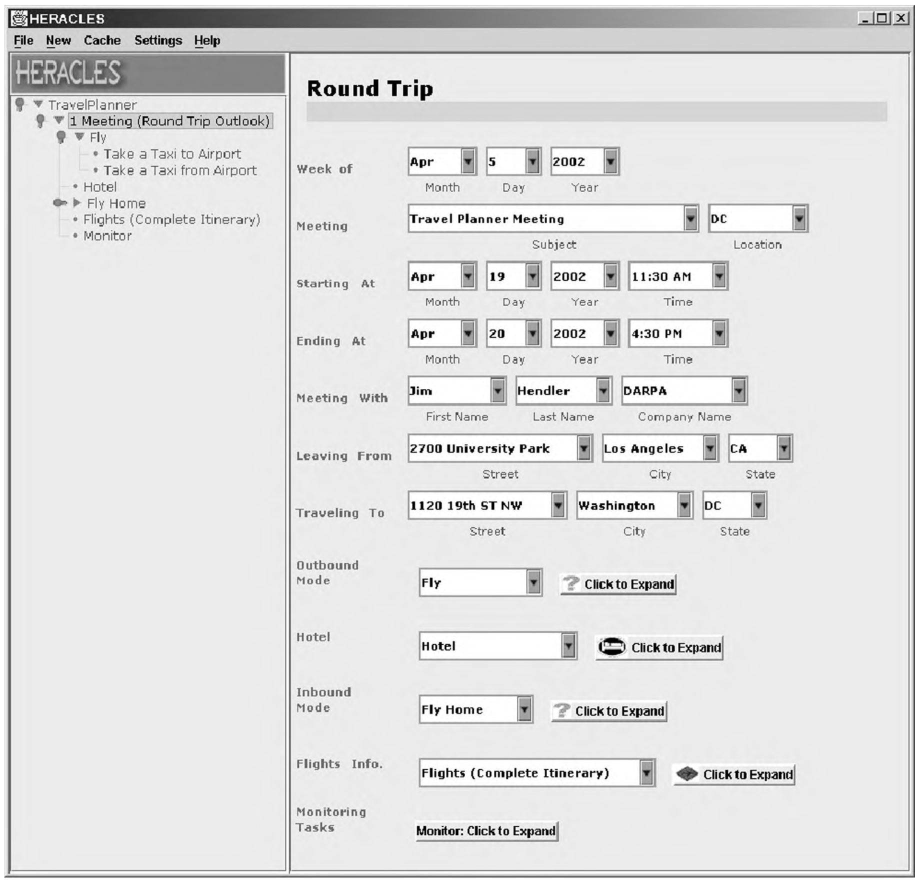


FIGURE 12.3. USC Travel Agent interface for planning a roundtrip.

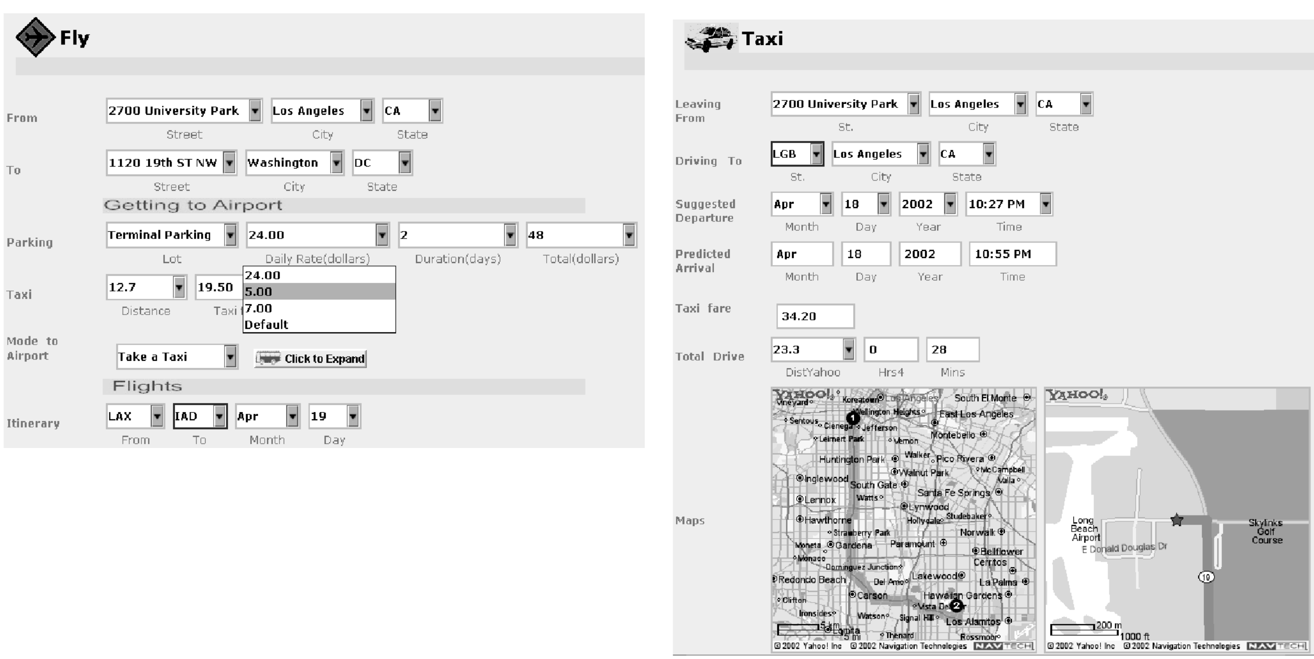


FIGURE 12.4. Propagation of changes and display of results.

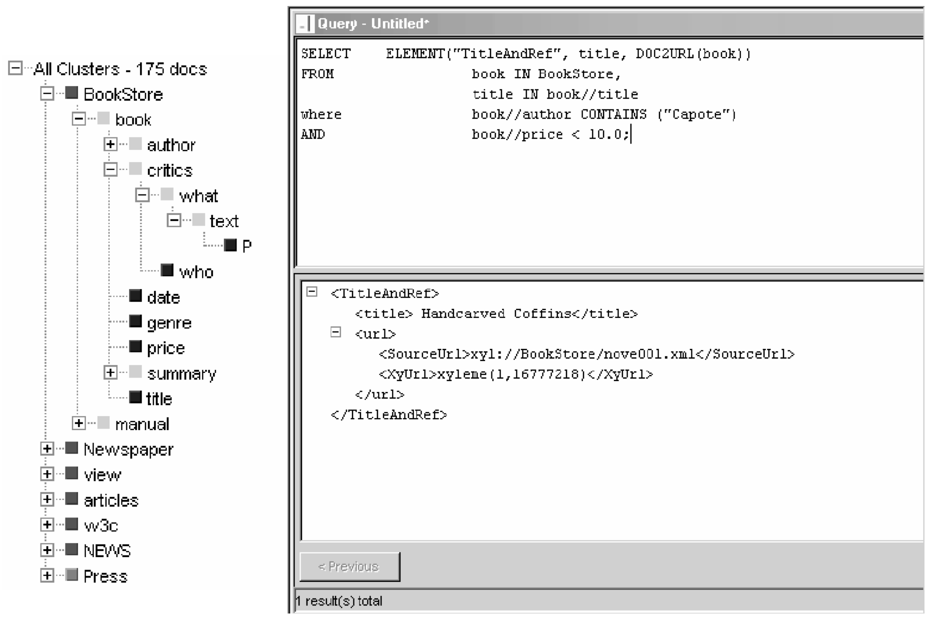


FIGURE 12.5. Xyleme's QueryShell.

The user interface is composed of three panels: one that presents the tree structure of the documents in the repository, one that allows writing queries as text by using XyQL, the Xyleme query language for XML, and one presenting query results.

QueryShell is connected to a Xyleme repository and displays the treelike structure of the XML data in this repository. The user may browse this tree structure through folding and unfolding nodes, and may write XyQL queries that respect the tree structure. The query result is displayed as a tree of XML text (composed elements may be folded/unfolded) within a scrolled window as shown in Fig. 12.5. A given maximum number of results is displayed—Next and Previous push buttons enable the user to ask for the next/previous results.

ViewBuilder allows creating homogeneous views over heterogeneous repository data. The user creates the view by defining a treelike structure (the view schema), and then the mappings between the view schema and the schemas of the real documents stored within the repository. A mapping is simply a correspondence relation between a node of the view schema tree and a node of the document schema tree. The mapping generation process is semiautomatic; the system proposes mappings based on a semantic matching algorithm, and the user may validate, manually add, and delete mappings.

To create a view with ViewBuilder, the user must define the domain (the list of data clusters), the schema (a treelike XML structure), and the definition (the set of node-to-node mappings). The user interface panel for domain definitions simply uses list selection, whereas the panel for schema definition uses a basic tree editor. Mappings creation uses the interface shown in Fig. 12.6. The left panel displays the view schema (virtual document type definition), whereas the right panel shows the current real document schema, both indicating the number of mappings for each node. The mappings are first computed by the program (by using semantic information about tags and structure), but the user may then delete and add mappings.

By clicking on a node (at left or at right), the corresponding node(s) on the other side are highlighted in blue. Nodes in mappings may be displayed either as nodes in the tree or as XML paths starting from the root. The blue "Text" tag in the virtual path (below the virtual DTD panel) shows that Text is the context (considered at mapping generation) of the current "Paragraph" node (i.e., this node means a paragraph of the text of an article, not a paragraph of the abstract). The virtual context is defined by the user at view schema creation. This feedback is useful to understand why eventually some mappings were not automatically generated. Once created, the view may be saved in Xyleme, then later reopened and updated. A view may be queried with QueryShell as any real document structure in Xyleme. Both QueryShell and ViewBuilder are Java Swing applications, communicating in Corba with the Xyleme modules (Corba servers). Of course, they are managing local in-memory data structures for displaying the information. There are no interface agents used in Xyleme's system user interface.

VisualHarness and InfoQuilt System User Interface. The *VisualHarness* system, developed by Amit Sheth and his colleagues (Sheth, Shah, Parsuraman, & Mudumbai, 1999), provides integrated keyword, attributed and content-based search of and access to heterogeneous text and image data based on the assignment and integrated use of relative weights, as shown in Fig. 12.7. The system is a successor of InfoHarness, probably the first web-based systems supporting attributed- or metadata-based search (Shah & Sheth, 1999; Shklar, Sheth, Kashyap, & Shah, 1995).

Au: Pls. Provide Citation for Fig. 12.8 in text.

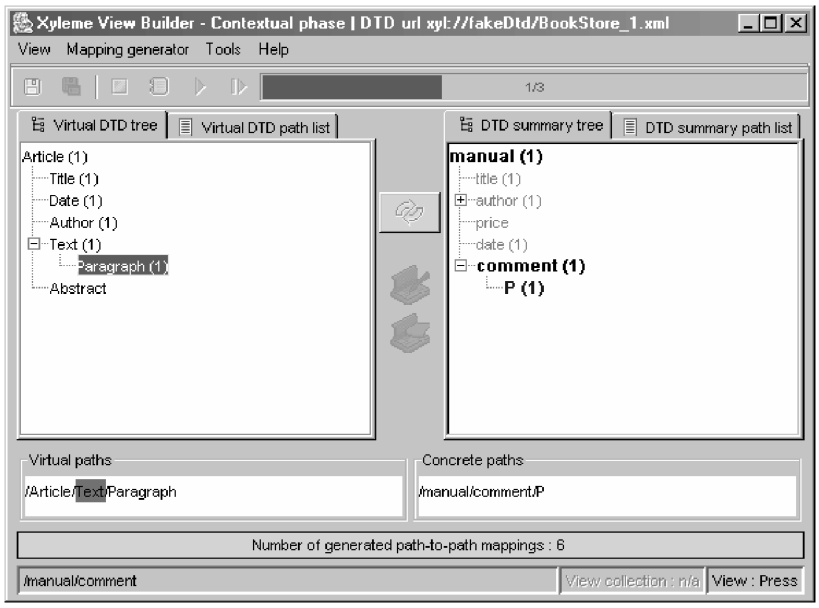


FIGURE 12.6. Xyleme's ViewBuilder.

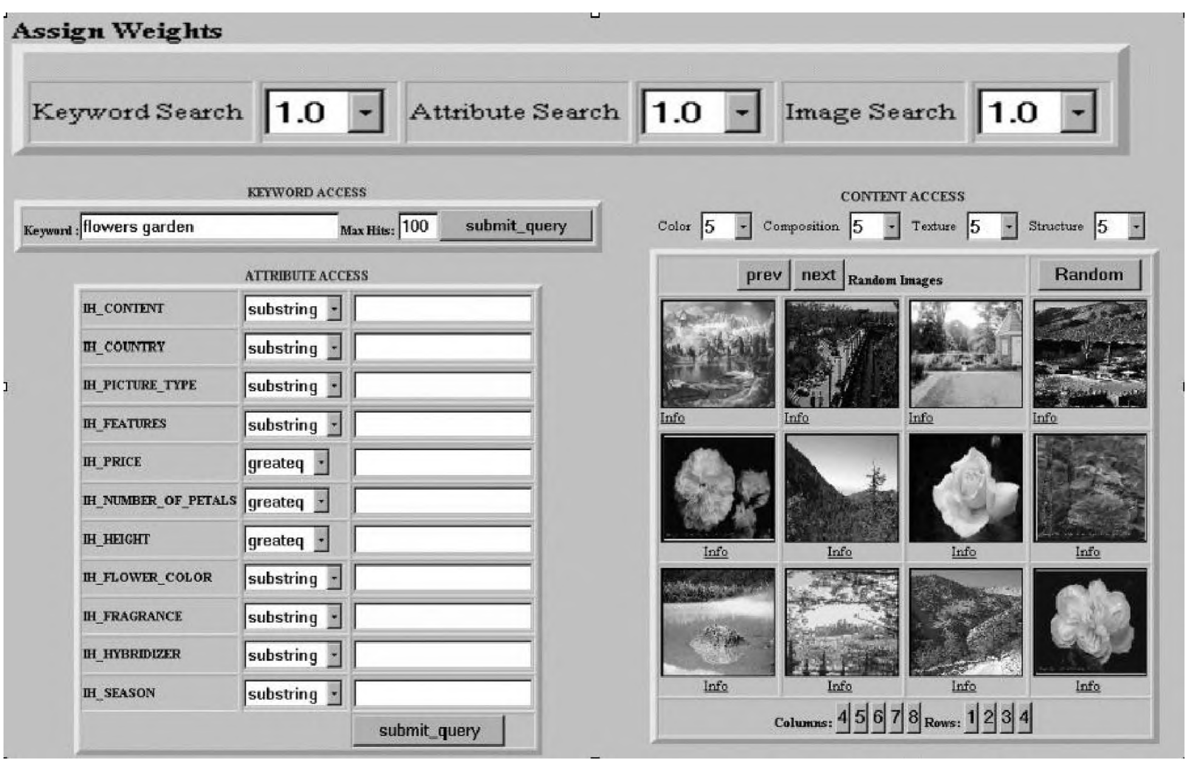


FIGURE 12.7. VisualHarness interface.

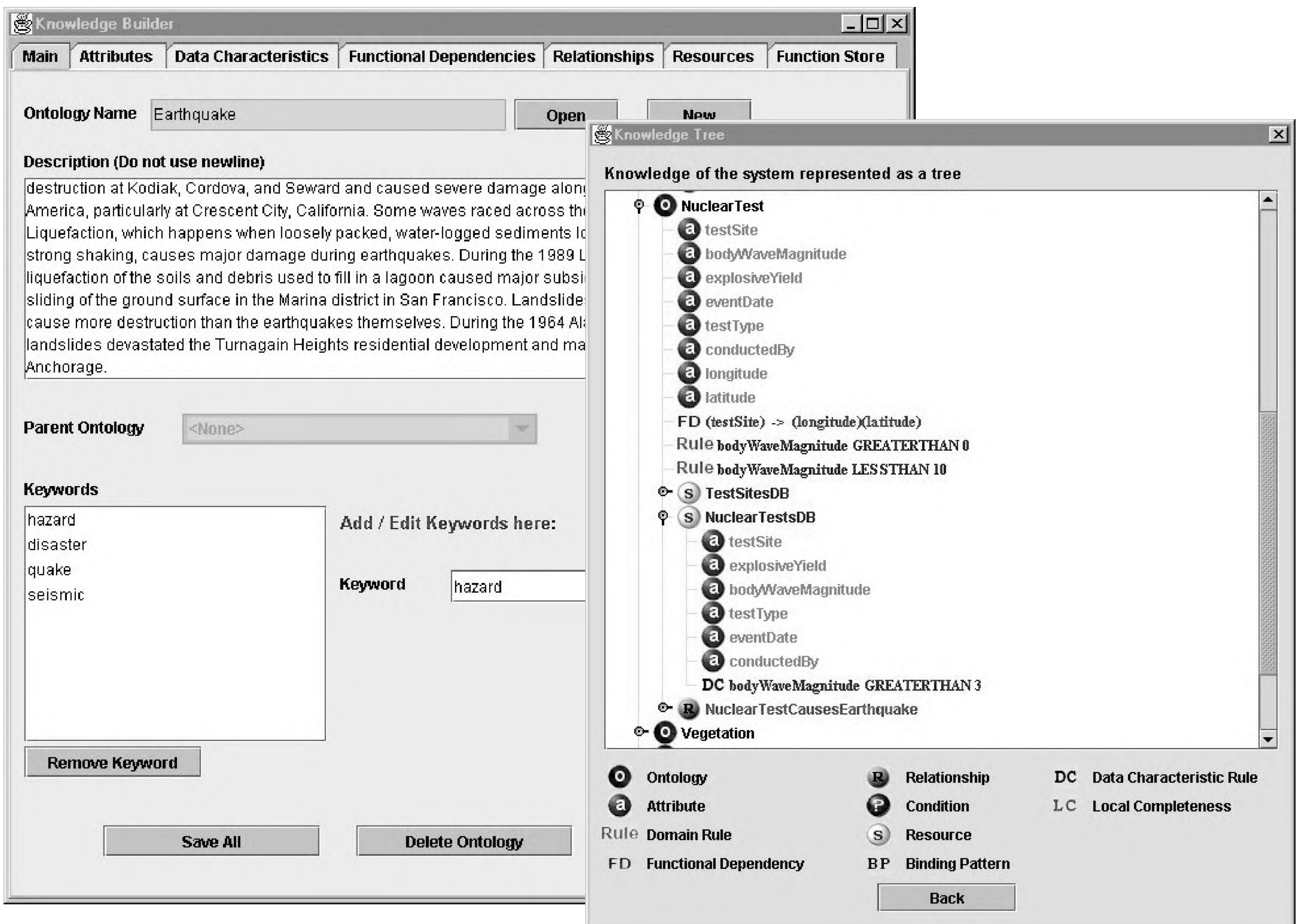


FIGURE 12.8. InfoQuilt system interface: Ontology management.

The *InfoQuilt* system allows specification of complex semantic relationships between data that may involve multiple ontologies and computable functions. Fig. 12.9 shows example screenshots of the InfoQuilt system interface providing the user with comprehensive features for ontology design and specification, and browsing capabilities (Sheth, Thacker, & Patel, 2003; Sheth et al., 2002). The displayed search result consist of metadata and images of content from multiple sources.

MOMIS System User Interface. The MOMIS (Mediator envirOnment for Multiple Information Sources) system (Fig. 12.10), developed by Sonia Bergamaschi and her colleagues at the University of Modena and Reggio Emilia, Italy, is a global-as-view mediator-based information integration system working on structured and semistructured sources (Bergamaschi, Castano, Beneventano, & Vincini, 2001). MOMIS implements a semiautomatic methodology, where each data source provides a schema and a global virtual view (GVV) of all the sources (i.e., a domain ontology). A language ODL_{I3} defined as a subset of the corresponding standard in ODMG augmented by primitives to perform integration is used to describe both the sources and the GVV.

Au: Pls. spell out ODMG and ODl.

The system architecture is composed of functional elements that communicate using the Corba standard. The *SI-Designer tool* supports the MOMIS integration process and provides an easy to use graphical user interface.

To interact with the local sources, MOMIS uses wrapper that automatically translate given metadata descriptions of local information sources into common ODL_{I3} representation. The integration designer has to provide the system with information where the individual sources are located (Fig. 12.11).

For integration purposes MOMIS uses a Common Thesaurus (ontology) to semantically map different words used in the schemas. ODL_{I3} provides terminological relationships for synonym (SYN), more general or more specific terms (BT, NT), and related terms used in the same context (RT). The thesaurus is built through an incremental process during which different kinds of relationships are added in the following order: schema-derived relationships (not modifiable by the designer), lexicon-derived relationships, designer-supplied relationships,

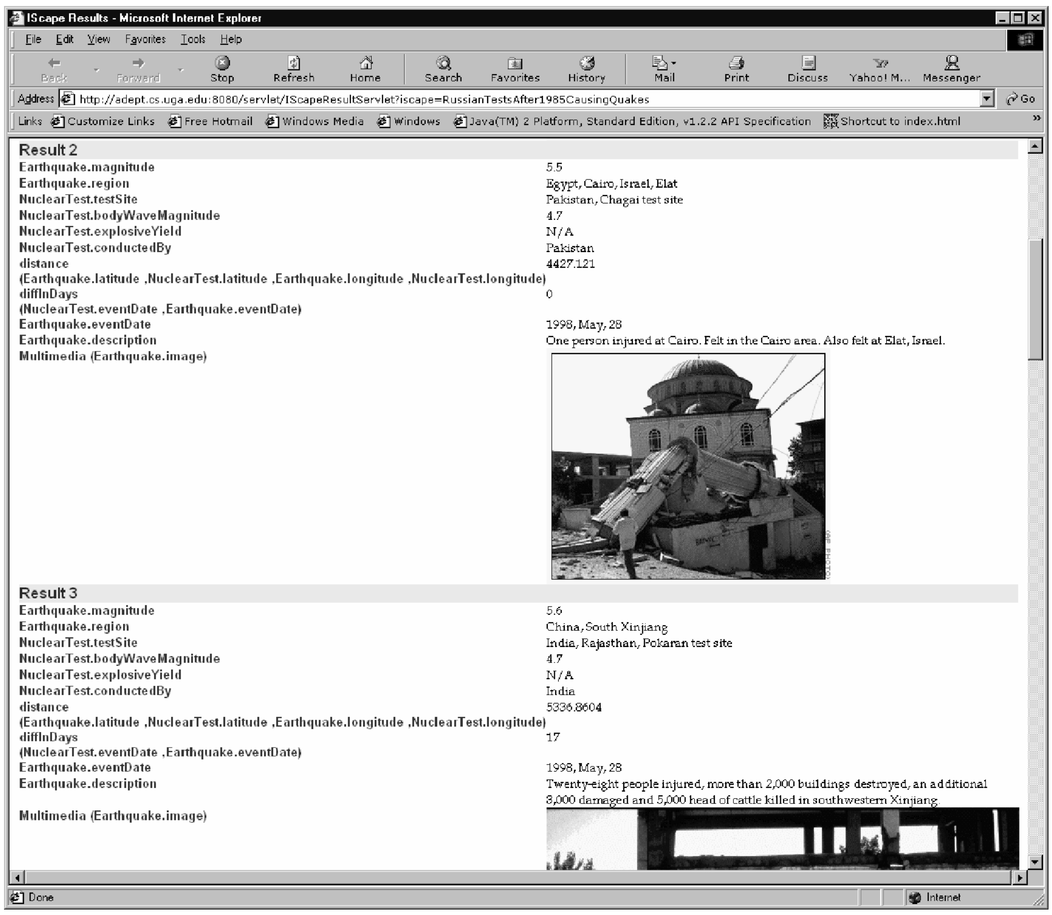


FIGURE 12.9. InfoQuilt system Interface: Display of search results.

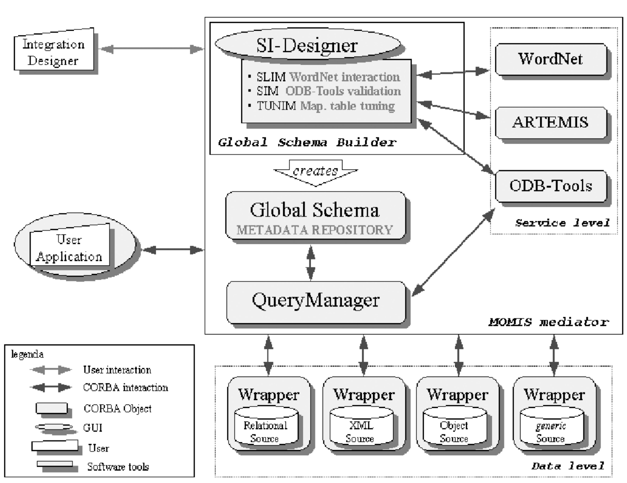


FIGURE 12.10. The MOMIS system architecture.

and inferred relationships. MOMIS extracts the lexical relationships by analyzing different source schemas, according to the WordNet (www.cogsci.princeton.edu/wn) lexical database. The designer has to choose the associated meaning in the WordNet system for each element of the schemas of the involved sources. This choice consists of choosing both a base form (suggested automatically by the system) and a meaning (Fig. 12.9). Exploiting the lexical relationships obtained by WordNet for the selected meanings, lexicon-derived relationships to be inserted in the Common Thesaurus are proposed to the designer; the ones accepted are inserted in the Common Thesaurus. MOMIS checks the consistency of the relationships of the Common Thesaurus and infers new relationships by exploiting ODB-Tools (Beneventano, Bergamaschi, Sartori, & Vincini, 1997), an integrated environment for reasoning on object-oriented database based on description logics. Figure 12.12 shows the graphical interface for managing the Common Thesaurus (MOMIS Ontology Builder).

To integrate the ODL_{I3} classes of given source schema descriptions into global ODL_{I3} classes, MOMIS uses the ARTEMIS tool (Castano, De Antonellis, & De Capitani Di Vimercati, 2001) for hierarchical and affinity-based clustering of ODL_{I3} classes

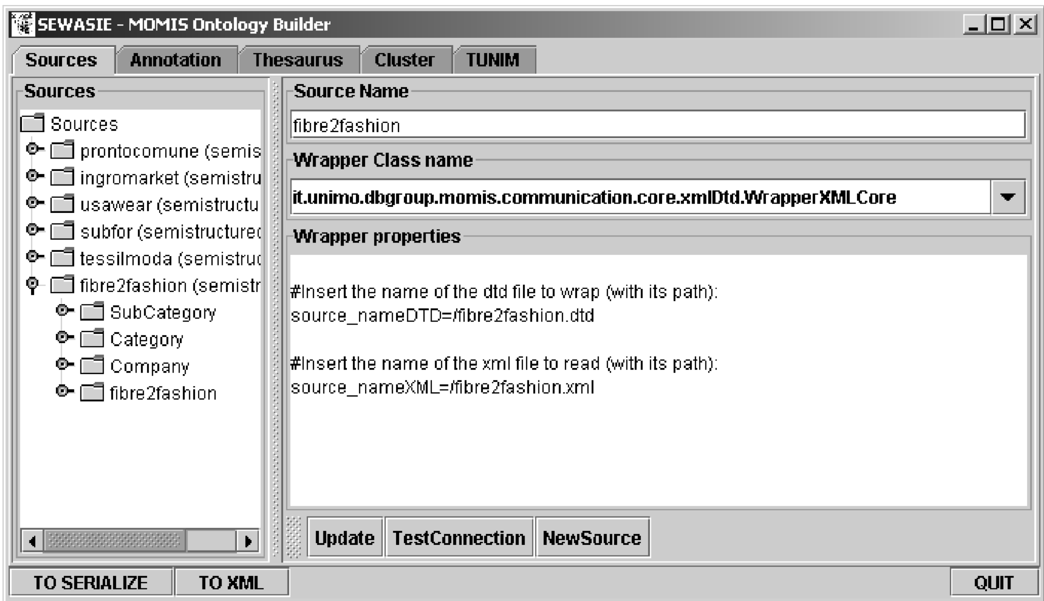


FIGURE 12.11. Specifying the location of information sources to be wrapped and integrated by MOMIS.

based on the relationships of the Common Thesaurus. The clustering procedure outputs an affinity tree, in which ODL_{I3} classes are the leaves and intermediate nodes have associated affinity values (Fig. 12.13).

As mentioned previously, like many other classic information integration systems such as TSIMMIS and Observer, interaction between the human user and the system is coordinated by a special-purpose tool, the SI-Designer of the MOMIS system, which is part of the mediator agent software. The MOMIS integration methodology will be further developed within the EU IST-2001-34825 SEWASIE (Semantic Web Agents in Integrated Economies) European research project (www.sewasie.org, www.dbgroup.unimo.it/Sewasie/). The SEWASIE goal is to design and implement an advanced search engine that enables intelligent access to heterogeneous data sources on the Web via semantic enrichment that provides the basis for structured secure Web-based communication.

Brief Discussion

Regarding the user interfaces of information systems described in the previous section, one may ask how interface agents should behave and present themselves to the user in a given application context? For pragmatic reasons, we do not comment on sporadically upraising general discussions about whether agent-based user interfaces would make sense at all. Instead, we refer to a discussion of direct manipulation versus agent-based interfaces by Negroponte (1997) and Shneiderman (1997). The intriguing characteristic of interactive information agents is their ability not only to assist the user in administering the system, control its workflow of information gathering and processing, and nicely visualize relevant information, but also to individually and proactively assist the user in gaining real benefit out of the displayed wealth of data and information at the same time. We claim that leveraging user interfaces by the embodiment of conversational agents may be one important step in this direction, significantly increasing the quality of user's perception and use of his or her information environment.

In the case of immersive environments, for example, such kind of interface agents may either be uploaded to, or are inherent part of, the virtual workspace of the user. They have to effectively collaborate with information agents of the system, while intelligently guarding the user through his or her 3D information spheres, and proactively anticipating future operations, directions, and opportunities. This is still far from being realized, but remains to be an intriguing vision of user interfaces of future agent-based information systems. Therefore, in the remaining sections of this chapter, we report on state of the art in the domain of intelligent user interfaces with embodied conversational agents.

AGENT-BASED USER INTERFACES

One of the fundamental interaction paradigms in human–computer interaction aims to emulate aspects of human–human interaction (e.g., by providing computers with the ability to engage with the user in a natural language dialogue). Central to this paradigm, from the point of view of the human user, is the illusion to communicate with a virtual entity that exhibits a number of humanlike qualities, such as the ability to engage with the user in a natural language dialogue, to infer communicative intent from user input and observed user behavior, and in some cases, even to react quasiemotional by including emotion triggering signals in generated output. To refer to such a virtual

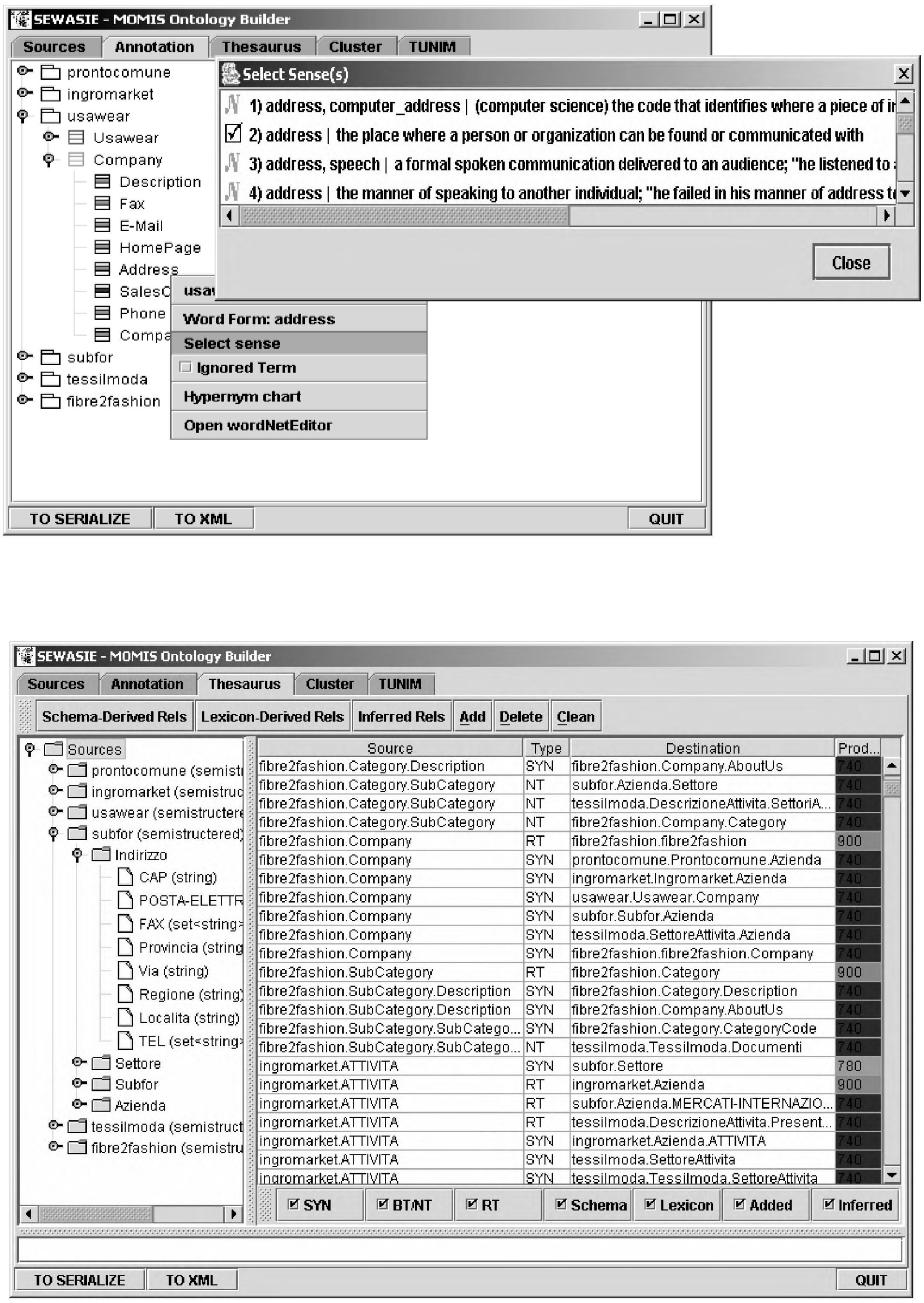


FIGURE 12.12. Interaction with the MOMIS Ontology Builder for determining semantic relationships .

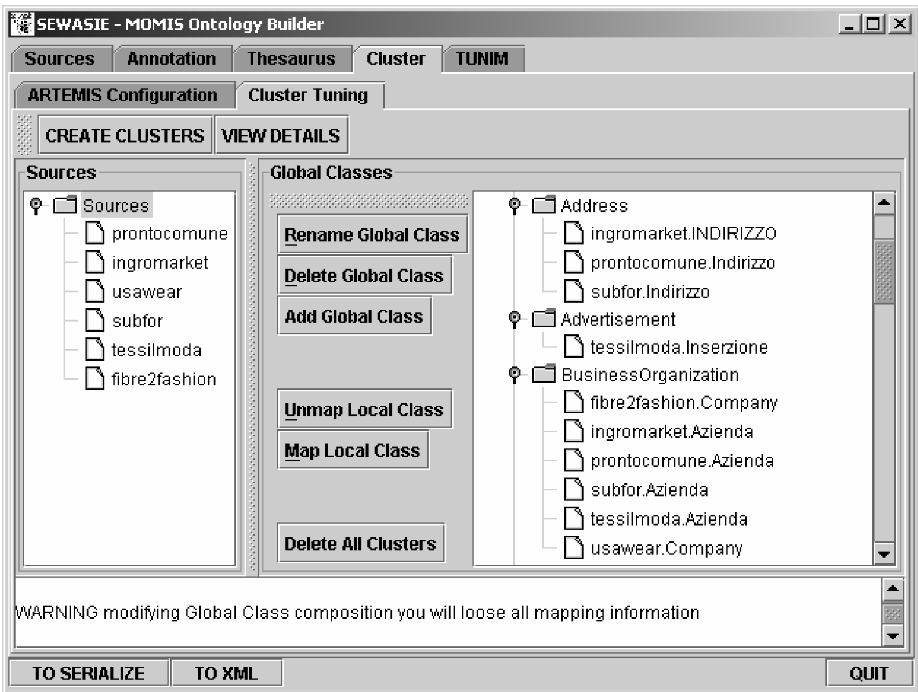


FIGURE 12.13. Clustering the ODL$_{I3}$ classes of given source schema descriptions into global ODL$_{I3}$ classes: The left panel shows the source classes to be integrated, and the right panel shows the obtained global classes.

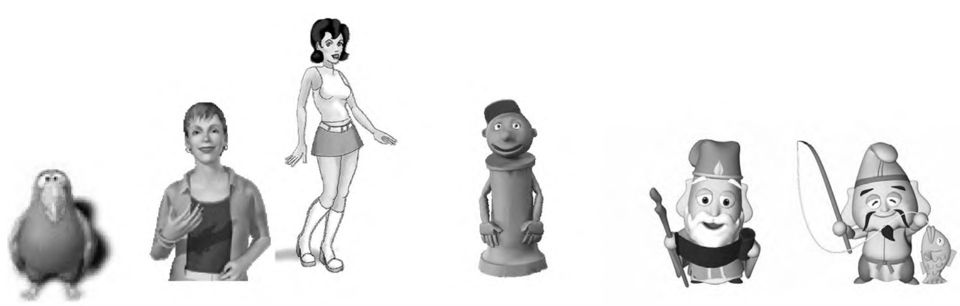


FIGURE 12.14. Examples of virtual characters: Peedy (Microsoft), Expert Coach (Extempo), Cyberella (DFKI), Smartakus (DFKI), and Venus&Mars characters (Lab of Image Information Science and Technology, Japan).

entity researchers and application designers have coined a variety of different terms, including intelligent virtual agents, animated interface agents, conversational characters, talking head interfaces, embodied conversational agents, personas, etc.

User Interfaces With Conversational Agents

It is interesting to note that the use of any embodiment—be it virtual by means of an audiovisual representation or be it even physical in the form of a robot—almost always suggests agency that raise expectations of having human-style conversational skills, too. However, there are also examples of user interfaces that suggest agency without emphasizing it through the any embodiment. A good example of this are the so-called chat-bots that become popular with the advent of the first Internet Relay Chat systems [e.g., Mauldin (1994)]. In these systems, the human user interacts with a virtual conversation partner via the exchange of text messages only. Other examples include early instances of natural language-based tutoring systems in which a human student interacted via text and menu input with an "invisible" virtual tutor that gave advise and instructions through text messages.

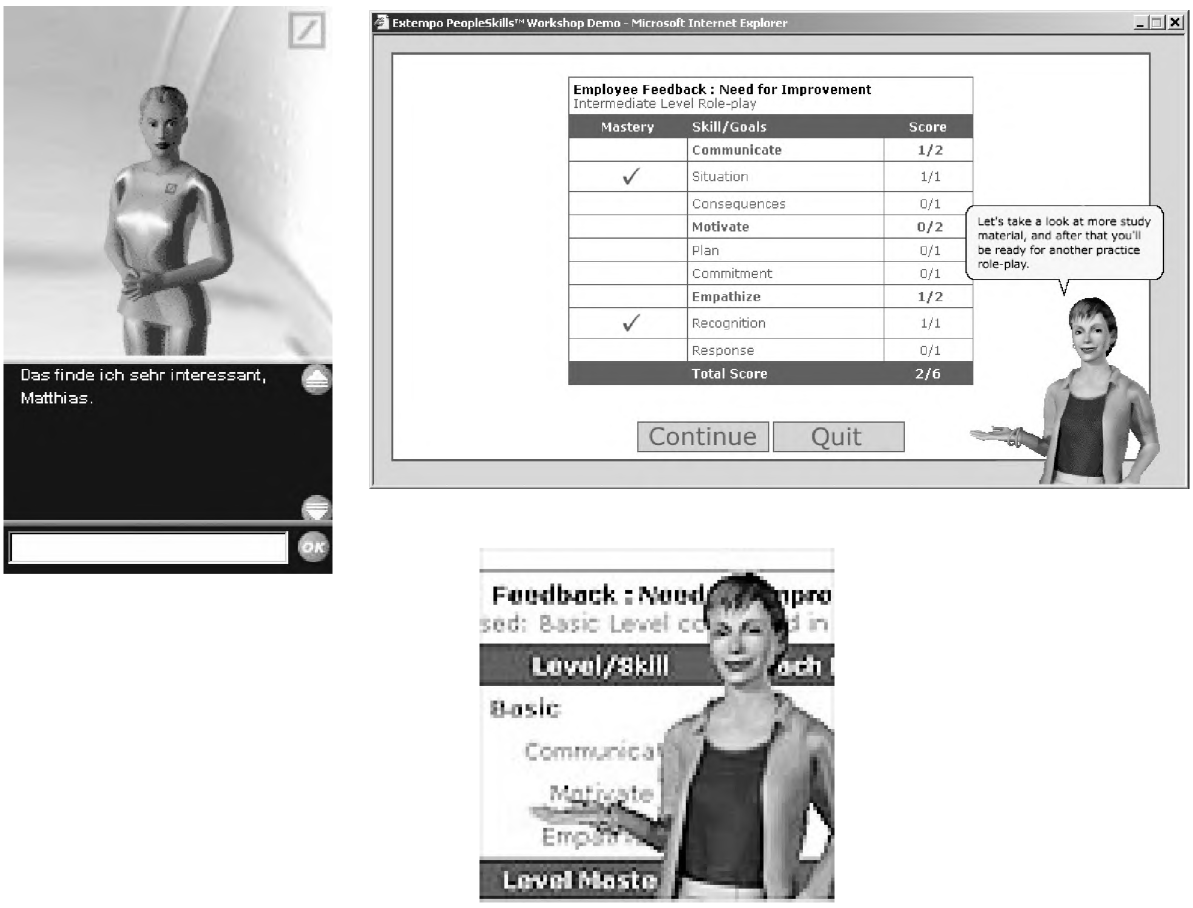


FIGURE 12.15. Examples of applied interface agents: Cor@ of Deutsche Bank Web portal, and Expert Coach of Extempo for eLearning environments.

In these applications, it is only a matter of the conversational skills of the virtual agent regardless of whether the human user is willing to cherish the illusion interacting with an intelligent agent. Due to the increasing availability of multiple input modalities and output media, however, use and exploitation of (audiovisual) embodiment has become dominant in the area. Therefore, the rest of this chapter concentrates on varieties of interfaces with conversational agents that also deploy and exploit embodiment. Some examples of embodied interface agents are shown in Fig. 12.14.

Face-to-Face Interaction With Embodied Agents

A great deal of contemporary systems aims at emulating aspects of a face-to-face interaction in settings where a user faces a virtual conversation partner. Differences among systems concern both available input modalities and output modalities. Interface agents are supposed to use pedagogical strategies and available assessment tools to individually monitor each user's progress against objectives, and make individualized recommendations based on user profile, history of advice, feedback, and other commentary.

For its use in applications, an interface agent has to be integrated with an appropriate information agent, existing system, and network environment, which finally yields an application-dependent interactive information agent. Prominent examples of commercial applications of embodied conversational information agents include Cor@ for information and navigation assistant at the Web site portal of Deutsche Bank; Robert T-Online from Deutsche Telekom; Cyberella from DFKI for guiding the user's navigation through the Web pages on the company's Web site; and Extempo's Expert Coach for one-on-one training of users to master Web-based learning content (cf. Figures 12.15 and 12.16).

Other potential applications of interactive information agents include guidance through virtual marketplaces, Web advertisement, interactive games, personal assistance in visual interfaces to digital libraries (Börner & Chen, 2002), and pervasive and situation-aware information and service provision on mobile devices (Bergenti & Poggi, 2002; Paterno, 2002).

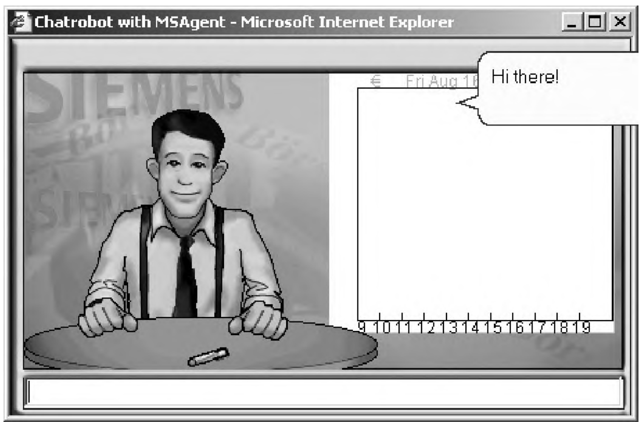


FIGURE 12.16. Stock agent Rudi converses with the user and retrieves online information about the development of stock shares (developed by Augsburg University in cooperation with Siemens AG).

FIGURE 12.17. Shots from the SmartKom system developed at DFKI. SmartKom allows the user to engage in a multimodal conversation with the interface agent Smartakus.

However, most currently deployed virtual characters in commercial applications remain strictly limited, for example, in their way of communicating with the user and response time. To emulate more closely face-to-face human–human dialogue settings, it is desirable to avoid asymmetries in communication channels that are available to the human and to the character. However, especially Web-based interfaces suffer from such asymmetries. In most applications, user input is handled in a rather restricted and controlled fashion. Usually, the user "talks" to these agents by typing natural language expressions into a text input widget. Well-known examples include the chatter-based products offered by companies, such as Extempo or Virtual Personalities.

Multimodal Face-to-Face Interaction

Ambitious attempts to overcome asymmetries in available communication channels are made by the SmartKom project (Wahlster, Reithinger, & Blocher, 2001). First, the interface character Smartakus comprises a more sophisticated spoken dialogue subsystem. Second, Smartakus has also a "visual sense" that enables it to recognize and understand pointing gestures of the user. Third, its visual sense allows the character to read a limited number of emotional expressions from the user's face. Recognized user emotions can be used to disambiguate speech input and to generate affective responses. Figure 12.17 shows the agent Smartakus acting as a TV program advisor and two shots of the SmartKom installation for public spaces. As shown by the close-up, Smartakus interprets multimodal user input, such as verbal utterances that are accompanied by pointing gestures.

Even though projects like SmartKom break new ground in the area of multimodal human–character interaction, the emulation of multimodal human–human face-to-face communication—especially on the Web—remains a long-term challenge.

Character Teams

Frequently, systems that use presentation agents rely on settings in which the agent addresses the user directly as if it were a face-to-face conversation between human. Such a setting seems quite appropriate for a number of applications that draw on a distinguished agent–user relationship. For example, an agent may serve as a personal guide or assistant in information spaces like the World Wide Web. There are situations in which direct agent–user communication is not necessarily the most effective and most convenient way to present information. Empirical evidence suggests that, at least in some situations, indirect interaction can have a positive effect on the user's performance. For example, Craig and colleagues (Craig et al., 1999) found that, in tutoring sessions, users who overheard dialogues between virtual tutors and tutees subsequently asked significantly more questions and also memorized the information significantly better.

The eShowroom (also called "Inhabited Market Place") is an example of a system that employs a new style of presentation and makes use of presentation teams to convey information about products, such as cars, by performing role plays (André, Rist, van Mulken, Klesen, & Baldes, 2000).

Figure 12.18 shows the characters Tina and Ritchie engaging in a simulated car sales dialogue. The overall system's

FIGURE 12.18. The virtual seller agent Ritchie meets the virtual buyer agent Tina in the eShowroom to inform her about the features of cars in stock. The eShowroom is being developed in the European Union-funded project NECA.

presentation goal is to provide the user with facts about a certain car. However, the presentation is neither just a mere enumeration of facts about the car, nor does it have a fixed course of dialogue between the agents. Rather, the eShowroom allows the user to specify prior to a presentation (a) the agents' roles, (b) their attitude toward the product, (c) some personality traits (extravert vs. introvert, agreeable vs. not agreeable), and (d) their interests about certain aspects relevant for cars (e.g., the car's relation to prestige, comfort, sportiness, friendliness to the environment, costs, etc.). Based on these settings, a variety of different sales dialogues can be generated for the same product.

Interacting With Embodied Agents and Avatars

Finally, there is an increasing number of scenarios that combine aspects of the different interaction situations as described in the previous sections. Of particular interest are settings in which the user is represented through an audiovisual embodiment in a virtual world that is cohabited by virtual agents and representations of other users. For this purpose, we distinguish explicitly between the terms "agent," "avatar," and "co-avatar."

An *agent* is an autonomous entity or character that takes decisions on its own based on its perception of the environment, own goals and desires, knowledge and model about the world, its mood, etc. Usually, a user has no direct control on autonomous characters although user interactions may have an impact on the character's behavior. An *avatar*, in contrast, is a perceivable representation of a person (a digital online stand-in for a human; e.g., the user) in a virtual 2D or 3D spatial environment. Usually, an avatar is (almost) fully controlled by the user. Because an environment can be cohabited with avatars of several users, we may additionally introduce the term *co-avatar* to denote from the perspective of a user, another person's avatar. Usually, the user has no control on co-avatars, and if no explicit glues are provided, may even not be able to differentiate co-avatars from agents.

In the following sections, we provide a detailed description of FreeWalk, an interactive user interface environment with embodied agents and avatars used in the digital city of Kyoto on the Web.

FreeWalk: A 3D Interface With Embodied Agents and Avatars. FreeWalk is a collaborative virtual environment (Benford, Greenhalgh, Rodden, & Pycock, 2001) for agent-mediated communities. Increasing graphics performance and spreading broadband network are facilitating 3D geographic contents on the Web. Creating embodied information agents that inhabit the 3D content spaces is a promising way to make them more navigable and accessible (Rickel & Johnson, 2000). Because the agents can search the content, they can guide users in the 3D space very well.

Virtual or digital cities, which are typical 3D contents, are usually plain 3D models where users can only change their viewpoints. Although many guide agents have been developed for Web sites, hardly any of them can be reused for digital cities. The 3D interface FreeWalk partly enables this reuse (Ishida, 2002b).

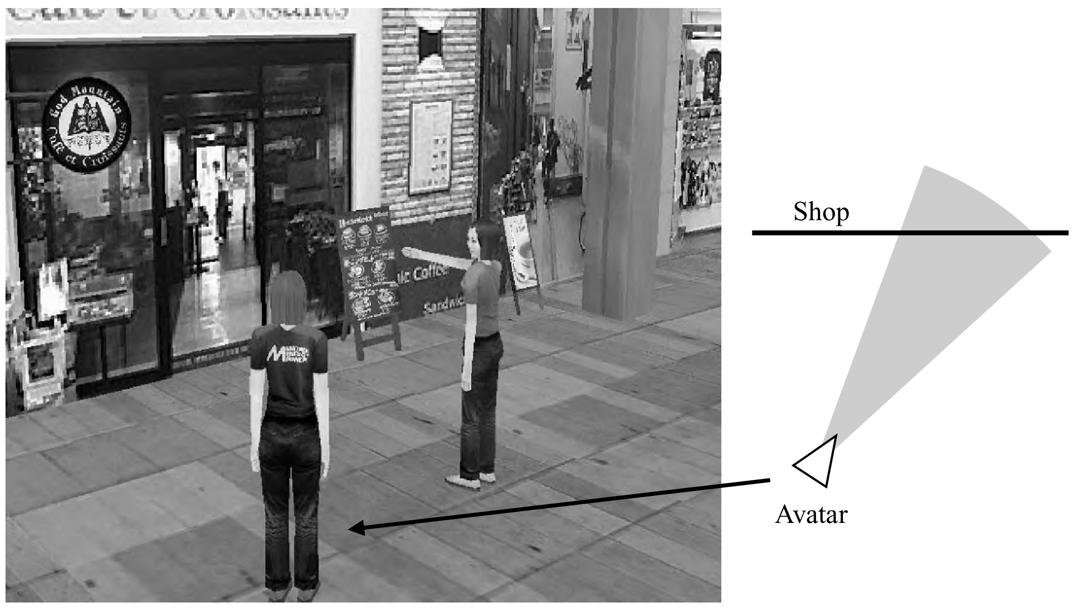
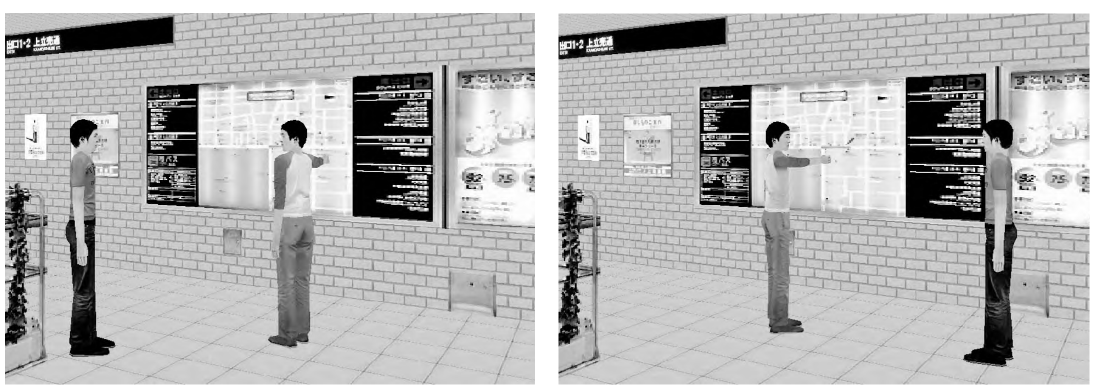


FIGURE 12.19. Conversation invoked by looking behavior.



Avatar from the left side   Avatar from the right side

FIGURE 12.20. Adapative "explain" behavior.

It is a 3D social interaction platform, which has shared memory to be connected with outside controller that calls FreeWalk's interface functions to control agents (cf. Figure 12.19). The *Q* script language and its interpreter for describing 2D agents can be reused as the controller of FreeWalk agents.

Unlike 2D Web interface, guide agents tend to be equivalent conversation peers in a 3D virtual city because they and the users' avatars exist in the same virtual space. Basically, FreeWalk provides communication parity among human users and agents. Both of them can walk, gesture, and speak equivalently in the virtual space. Human users control their avatars by themselves and talk with others in voice. Agents walk, gesture, and talk according to the prepared script. Walking paths of agents are smoothed by the collision avoidance function. Speech synthesis and recognition engines enable agents to talk with humans.

In virtual cities, conversation between agents and humans is inherently multimodal. Looking behavior that means interest may invoke a new topic to talk about. In Fig. 12.20 the guide agent is explaining the shop to which the avatar is paying attention. Such style of conversation has not been considered in the study of traditional mechanism that requires clear task and goal of conversation. FreeWalk support the outside controller to track events including nonverbal cues occurring in a virtual city so the agents can handle event-driven conversations effectively.

It is tough to describe a script that generates natural movement of an agent character equivalently interacting with people in multimodal style. It is necessary to specify many 3D parameters. Furthermore, it is difficult to guess avatars' behavior to which the agent's movement should be tailored. To eliminate this problem, we referred to social manners to implement

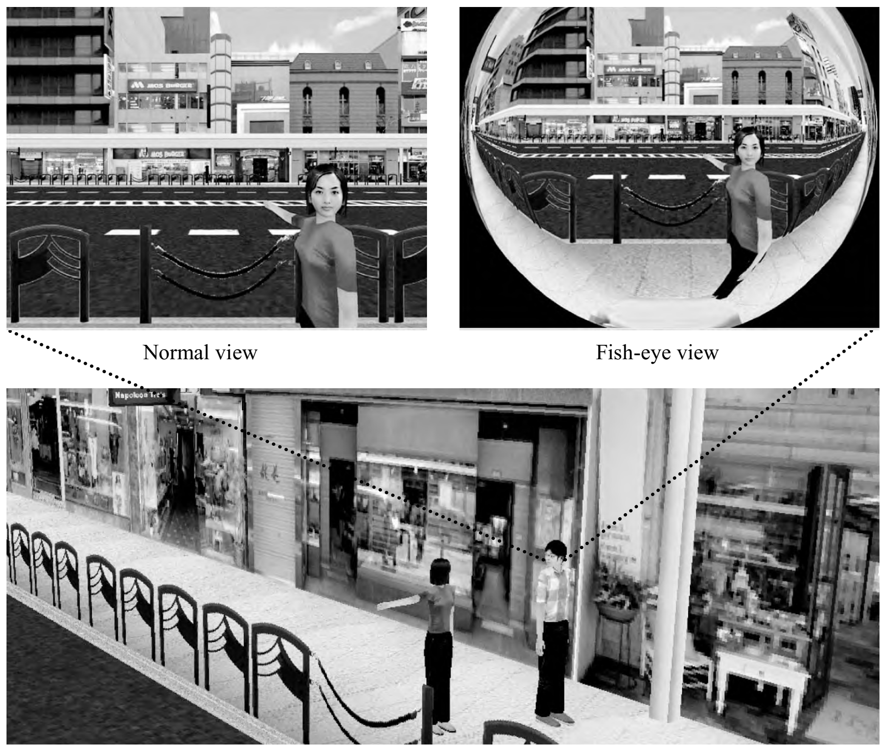


FIGURE 12.21. Guide agent in the 3D digital city Kyoto.

interface functions called by the outside controller. When the function explain is called, the guide agent explains some target to the addressed person. The right-hand side of Fig. 12.20 shows this behavior changed adaptively according to the avatar's position.

The agent explains the map to the avatar. To decide the standing position of the agent, the function applies three rules. The first rule is that the agent should not block the view between the map and the avatar. The second is that the agent should not invade the personal space of the avatar. The third is that the agent should be seen by the avatar. These rules can reduce the number of 3D parameters required to specify each movement and make the movement more natural and adaptive.

Verbal and nonverbal communication channels of FreeWalk are designed to support 3D social interaction between agents and people. The feature of verbal channels is that distance attenuates voice volume in FreeWalk's space. This is a difference from ordinary graphical chat tools and causes socially realistic behavior as approaching others and gathering to have a conversation. Fully controllable head and arm directions are important characteristics of nonverbal channels in FreeWalk. Because facing and pointing gestures are necessary to realize deictic communication (Rickel & Johnson, 2000), human interface and interface functions enable agents and human users to control their heads and arms freely.

Standing position also plays an important role in deictic communication. As an upper layer of the particle model that is dedicated to collision avoidance, a "formation model" is created to simulate social behavior of human crowds. In this model, each agent or avatar tries to adjust his or her location and orientation according to the formation of the group to which he or she belongs. The formation can be a cluster, a line, or a circle. An example of such formation is F-formation (Kendon, 1990), which is a group behavior to form a circle to have a conversation. In this behavior, people approximately recognize the center and the radius of their shared circular space. They position their bodies on the circle to divide it equally and orient their bodies to its center. FreeWalk is equipped with the mechanism of the formation model and automatically detects a group and its formation to adjust its members' locations, orientations, and velocities.

Although a wide field of view helps human users to perceive nonverbal cues of deictic communication, such immersive displays as large screens and HMDs have not been widely spread. **Au: Pls. Spell out HMDs.** Alternatively, FreeWalk uses fish-eye projection to include peripheral view in a normal desktop monitor. Figure 12.21 shows this optional display mode in the 3D Kyoto example. You can perceive the guide agent's position and posture better. Third-person view is a more popular solution, especially in video games. In contrast to that view, fish-eye view can keep consistency between the displayed view in a monitor and the avatar's gazing behavior that is an important nonverbal cue.

Application of FreeWalk to the 3D Digital City Kyoto. The 3D digital city Kyoto is an example of a virtual city on the Web (Ishida, 2002a). We constructed a virtual shopping street that mirrors its real-world counterpart Shijo Street in

Kyoto city. As shown in Fig. 12.21, the guide agent walks along the street together with the user's avatar to provide information of shops in which he or she takes an interest.

A short scenario of *Q* enables the agent to adjust its walking speed to the user's control and detect the avatar's looking behavior that indicates the user's attention. It is easy to include facing and pointing gestures in guiding behavior of the agent. Preliminary experiments show encouraging positive feedback from the users. In particular, the guidance application has been well received and found attractive as a daily Web service.

MODELING OF PERSONALITY AND EMOTIONS

People infuse their verbal and nonverbal behaviors with emotion and personality, and modeling such behavior is essential for building believable virtual humans. Consequently, researchers have developed computational approaches for a wide range of applications. Computational approaches might be roughly classified to communication-driven and simulation-based approaches.

Communication-Driven Approaches

In communication-driven approaches, a system chooses its emotional expression on the basis of its desired impact on the user. Tutoring applications usually follow a communication-driven approach, intentionally expressing emotions with the goal of motivating the students and thus increasing the learning effect. For instance, to increase a student's motivation, an agent may express admiration that is conveyed with applause if a difficult problem has been solved. An example includes the Cosmo system in which the selection and sequencing of emotive behaviors (Lester, Towns, Callaway, Voerman, & FitzGerald, 2000) is driven by the agent's pedagogical goals. For instance, a congratulatory act triggers an admiration emotive intent that is conveyed with applause. To convey appropriate emotive behaviors, agents like Cosmo need to appraise events not only from their own perspective, but also from the perspective of other agents.

Simulation-Based Approaches

The second category of approaches aims at a simulation of "true" emotion (as opposed to conveyed emotion). These approaches build on appraisal theories of emotions, the most prominent being Andrew Ortony, Gerald Clore, and Allan Collin's cognitive appraisal theory—commonly referred to as the OCC model (Ortony, Clore, & Collins, 1988). This theory views emotions as arising from a valenced reaction to events and objects in the light of the agent's goals, standards, and attitudes. For example, an agent watching a game-winning move should respond differently depending on which team is preferred. The earliest implementations of the OCC model include the emotion model developed within the Oz project (Reilly, 1996) and Elliott's Affective Reasoner (Elliott, 1992). More recent work by Marsella and Gratch (2002) integrates the OCC model with coping theories that explain how people cope with strong emotions (Lazarus, 1991). Their implementation includes both problem-focused coping and emotion-focused coping. Problem-focused coping strategies refer to the selection and execution of actions in the world and are applied if there is an action that could improve the agent's emotional state. Emotion-focused coping strategies do not act directly on the world, but rather on the agent's mental state. An example of an emotion-focused coping strategy is shift-blame, which might be applied if an agent's superior is asking about an event that the agent is potentially responsible for and there is another agent that is also potentially responsible. Further simulation approaches are based on the observation that an agent should be able to dynamically adapt its emotions through its own experience. To account for the dynamic nature of emotions, these approaches make use of learning mechanisms (El-Nasr, Ioerger, & Yen, 1999; Yoon, Blumberg, & Schneider, 2000).

Expressing Affective Behaviors

In addition to generating affective states, we must also express them in a manner easily interpretable to the user. Effective means of conveying emotions include conveying a character's emotions through body gestures, acoustic realization, and facial expressions [see Collier (1985) for an overview of studies on emotive expressions]. Several researchers make use of Bayesian networks to model the relationship between emotion and its behavioral expression. Bayesian networks enable us to deal explicitly with uncertainty, which is of great advantage when modeling the connections between emotions and the resulting behaviors. An example of such an approach was presented by Ball and Breese (2000), who constructed a Bayesian network that estimates the likelihood of specific body postures and gestures for individuals with different personality types and emotions. For instance, a negative emotion increases the probability that an agent will say "Oh, you again," as opposed to "Nice to meet you!" More recent work by Pelachaud, Carofiglio, De Carolis, de Rosis, and Poggi (2002) employs Bayesian networks to resolve conflicts that arise when different communicative functions need to be shown on different channels of the face, such as eyebrow, mouth shape, gaze direction, head direction, and head movements. In this case, the Bayesian network serves to estimate the likelihood that a movement of the face overrides another.

Modeling the Relationship Between Emotions, Personality, and Social Roles

Obviously, there is a close relationship between emotion and personality. Moffat (1997) differentiates between personality and emotion by using the two dimensions of duration and focus. Whereas personality remains stable over a long period of time, emotions are short lived. Moreover, whereas emotions are focused on particular events or objects, factors determining personality are more diffuse and indirect. Due to this obvious relationship, a number of projects aim at an integrated model of emotion and personality. An example includes Ball and Breese (2000) who model dependencies between emotions and personality within a Bayesian network. To enhance the believability of animated agents beyond reasoning about emotion and personality, Prendinger and Ishizuka (2001) investigated the relationship between an agent's social role and the associated constraints on emotion expression. Rist and Schmitt (2002) developed a test-bed for the simulation of changing interpersonal relationships during negotiation dialogues. The simulation is based on sociopsychological theories of cognitive consistency dynamics (Osgood & Tannenbaum, 1955) and captures some aspects of group dynamics.

BEHAVIOR AND ANIMATION CONTROL FOR ANIMATED COMMUNICATIVE AGENTS

Most of the current systems with animated characters distinguish between a character's embodiment and a behavior control component. Some relate this distinction to the biological body/brain dichotomy. Others take a more technically oriented view and associate embodiment with an animation engine (often called *character player*), whereas behavior control is related to some sort of automated behavior generation, often based on artificial intelligence techniques, such as task-oriented hierarchical planning, or the simulation of certain aspects of humanlike cognition.

To address the first part of the problem—namely, to animate an agent—many systems make use of MPEG-4, which allows us to define the geometry of a character as well as the animations of the face and the body in a standardized format. Although earlier work concentrated on procedural animation, more recent work focuses on the higher-level control of coordinated animations that reflect a character's emotion and personality as well as its communicative intent. For instance, Poggi and Pelachaud's (2000) Greta system follows a communication-theoretic approach to facial animation, and interprets and generates facial displays in terms of the conversational and affective functions they have to fulfill in a dialogue. A promising approach for body animation control has been developed by Badler and colleagues (Chi, Costa, Zhao, & Badler, 2000), who developed a parameterized system based on Laban Movement Analysis, a method for observing, describing, rotating, and interpreting human movement. Depending on the settings of parameters, such as effort and shape, several variants for the same basic animation data may be generated.

To address the second part of the problem—namely, to specify an agent's behaviors—scripting has been widely used in projects that deal with interface characters (Perlin & Goldberg, 1996). Thereby, a script is a temporally ordered sequence of actions, including body gestures, facial expressions, verbal utterances, locomotion, and (quasi-) physical interactions with other entities of the character's immediate environment. A straightforward approach is to equip the character with a library of manually authored scripts that determine what the character might do in a certain situation. At runtime, the remaining task is to choose from the library a suitable script that meets the constraints of the current situation and, at the same time, helps to accomplish a given task. What is specified in a character script is also a matter of the level of abstraction and the expressiveness of the scripting language. In some cases, the scripting language is built on top of an existing general-purpose script-based programming language. For instance, the Microsoft Agent characters can be easily scripted either in Visual Basic or in Java Script, allowing the script writer to use the standard control structures of these languages like conditional statements or loop constructs. As an alternative to character-specific adjuncts to programming languages, XML-compliant character scripting languages have been be defined, such as VHML (www.vhml.org) or MPML (www.miv.t.u-tokyo.ac.jp/MPML/). In any case, the script may be seen as a kind of an API for the character player that allows the agents behavior to be specified at a certain level of abstraction.

A particular problem with manually authored scripts and script libraries is that the author has to anticipate scripts for all possible situations and tasks, and that the scripts must allow for sufficient variations in order to avoid characters that behave in a monotonic and too predictable way. Furthermore, the manual scripting of characters can become quite complex and error prone because synchronization issues have to be considered. Creating scripts manually is, however, not feasible for many applications because it would require anticipating the needs of all potential users and preparing presentations for them. In the case of most web-based applications, manual scripting is even more impracticable because the information to be presented usually dynamically changes and there is simply not enough time to manually create and update presentations.

Plan-Based Approaches

To automate the script writing process, plan-based approaches have proved promising. An example of such an approach has been presented by André, Rist, and Müller (1999) for the AiA travel agent. The basic idea of the approach was to formalize action sequences for composing multimedia material and designing scripts for presenting this material to the user as operators of a planning system. The effect of a planning operator refers to a complex communicative goal (e.g., to describe a hotel), whereas the expressions of the body of the operator indicate which acts have to be executed to achieve this goal (e.g., to show an illustration of a typical hotel room). In addition, the plan operators allow us to specify spatial and temporal layout constraints for the presentation segments corresponding to the single acts. The input of the presentation planner is a complex presentation goal. To accomplish this goal, the planner looks for operators whose headers subsume it. If such an operator is found, all expressions in the body of the operator will be set up as new subgoals. The planning process terminates if all subgoals have been expanded to elementary production/retrieval or presentation tasks.

Approaches Based on Finite-State Automata

Although planning techniques have proved useful for the specification of scripted behaviors, the generation of immediate reactions and smooth animation sequences requires a method that is computationally less expensive. One solution is to precompile declarative behavior specifications into finite-state machines (Ball et al., 1997), which are also a suitable mechanism for synchronizing character behaviors. For instance, Cassell and colleages (1994) use so-called parallel transition networks (PaT-Nets) to encode facial and gestural coordination rules as simultaneously executing finite-state automata.

A combination of a plan-based approach and a finite-state approach has been applied in the CrossTalk system (Gebhard, Kipp, Klesen, & Rist, 2003). In CrossTalk scene, flow (i.e., the transition between scenes) is modeled by means of a cascaded finite-state machine in which states refer to scenes with characters that have been either preauthored or that are generated automatically on the fly using a plan-based approach.

Machine Learning Approaches

Machine learning approaches have been essentially employed to learn affective behaviors. For instance, El-Nasr and colleagues (1999) made use of a reinforcement algorithm to learn about event sequences and possible rewards. Action selection is modeled as a Markov decision process. When deciding between several actions, the agent considers with which probability a certain follow-up state may be reached if a certain action is performed and what reward it may expect.

According to the OCC, an object may also directly trigger an emotional response. For instance, Yoon et al. (2000) implemented a simple concept learning algorithm based on Bayesian belief update. At the first encounter with an object, the agent tries to come up with a description of minimal length that characterizes the object. As a result of this process, the agent might learn, for example, that dark animals are scary. If the agent encounters objects that break its current belief, the concept description is updated to reflect the agent's new belief.

Approaches Based on Bayesian Networks

Bayesian networks enable us to deal explicitly with uncertainty, which is of great advantage when modeling the connections between a number of psychological factors, such as emotions and personality, and the corresponding behaviors. An example of such an approach was presented by Ball and Breese (2000), who constructed a Bayesian network that estimates the likelihood of specific body postures and gestures for individuals with different personality types and emotions. For instance, a negative emotion increases the probability that an agent will say "Oh, you again," as opposed to "Nice to meet you!" More recent work by Pelachaud et al. (2002) employs Bayesian networks to resolve conflicts that arise when different communicative functions need to be shown on different channels of the face, such as eyebrow, mouth shape, gaze direction, head direction, and head movements. In this case, the Bayesian network serves to estimate the likelihood that a movement of the face overrides another.

SUPPORTING THE DEVELOPMENT OF INTERFACE AGENTS AND APPLICATIONS

The development of reusable software tools, modules, and platforms for interface agents is a prerequisite for both rapid prototyping of new applications of such agents and the promotion of commercial technology uptake. Work in this direction comprises the development of

- Authoring tools that enable nonprogrammers to develop applications with interface agents
- Dedicated platforms atop of which applications can be realized
- Application-independent languages for the specification of characters and behavior scripting

Authoring Suites for ECAs

In many applications, the behavior of the involved characters are either fully or at least partially scripted by a human author. To facilitate the authoring process, a number of editing suits are currently under development. Most of them feature a graphical user interface that allows nonprogrammers to compose complex behaviors, such as a branching user–agent dialogue structures and interactions between agents and objects in their virtual environment. Examples of authoring tools include the SceneMaker tool developed at DFKI (Gebhard et al., 2003), and a graphical editor for the composition of MPML scripts is currently being developed at the University of Tokyo (Prendinger et al., 2003). The Beat tool (Cassell, Vilhjálmsson, & Bickmore, 2001) supports a human author in converting text to multimodal output. It automatically annotates text with hand gestures, eye gaze, eyebrow movement, and intonation in XML-based markup languages by interacting with a domain knowledge base and making use of behavior generation rules. In addition to the automated annotation process, Beat handles the synchronization of verbal and nonverbal signals.

Dedicated Platforms for ECAs

A number of agents are based on Soar (Laird & Rosenbloom, 1996), a general cognitive architecture for building intelligent agents. In this chapter, we concentrate on architectures for interface agents. Most systems that deploy lifelike characters make a concrete commitment to one specific conversational setting and reflect this decision in a particular system architecture. However, if the desire later emerges to support other conversational settings as well, an almost full reimplementation of the application often becomes unavoidable. In contrast, the development of the MIAU platform (Rist, André, & Baldes, 2003) shows that it is indeed possible to develop a single platform that (a) can be used to construct a broad range of character applications, (b) even allows to switch on the fly between director- versus character-centered scripting approaches, and (c) supports a clear separation between the specification of scripting knowledge (being a knowledge-engineering task), and the required computational machinery for behavior generation (being an implementation task). MIAU can be conceived as a multiagent platform that has been tailored to applications with conversational characters.

The key components of the MIAU platform are a director component and one or several character components. Different interaction styles are realized by shifting the control between the director components and the character components. For instance, the MIAU architecture allows both for the realization of scenarios in which a directors scripts all actions for the single characters in detail as well as scenarios in which the characters have greater autonomy and the director only intervenes from time to time (e.g., to increase coherency or to introduce new topics). More examples of flexible switches between different conversational settings (monolog vs. dialog vs. performance vs. interactive performance) can be found in Rist et al. (2003).

In addition to building concrete agents, researchers also focus on the development of application-independent technology for the implementation of interface agents. One of the most prominent frameworks is the Collagen system (Rich, Sidner, & Lesh, 2001), which is based on a model of collaboration between a human and a computer agent. An interesting feature of the framework is that it allows for different degrees of system activity. For example, the default strategy in a help system is to have a task performed by the Collagen agent. This strategy would, however, be less appropriate in a tutoring application, where it is important that the student learns how to perform a task. Unlike the MIAU platform, Collagen focuses on two-party conversation between a human and a computer agent even though an extension to multiparty applications seems obvious and feasible.

Markup Languages for Specifying Agent Behaviors

The increasing number of applications that deploy animated conversational characters motivates work toward standardized scripting and specification languages for characters and applications with characters. Many of the representation languages aim to provide a common representation format for the interchange of data and control among components in character systems.

Examples of this kind are the rich representation language (RRL) developed by Piwek et al. (2002), and the multimodal markup language (M3L) developed in the SmartKom project (Wahlster et al., 2001). In this case, at least simplicity of use becomes primarily a matter of more technical criteria, including compactness, uniformity, and suitability for efficient processing.

Other proposed languages are actually meant to support human authors in the development of applications with lifelike characters. Especially scripting languages for web applications with characters fall into this category. Examples include MPML or the Microsoft Agent scripting language (even though expressions in these languages can and are also automatically generated by a behavior control system).

Another criterion for distinguishing markup languages is their level of description. Languages, such as MURML (Kopp, Jung, Lebmann, & Wachsmuth, 2003), specifies the behavior of an agent at the signal level and directly refer to the gestures the agent needs to perform. Another group of languages controls the agent at the discourse level and specifies the agents' behavior by means of communicative functions, such as suggestions or criticism. An example of such language includes the XML-based markup language used in the BEAT system or APML. In this case, an XML tag corresponds to a higher-level concept that will be translated into a facial expression (e.g., gaze, head movements, eyebrow frowns) and gestures (e.g., hand shape, arm direction).

Modelling Applications With the Scenario Specification Language *Q*

Agent internal mechanisms form the basis for many of the languages proposed for describing agent behavior and interagent protocols. These mechanisms include Soar (Laird & Rosenbloom, 1996), a general cognitive architecture for developing systems that exhibit intelligent behavior, and KQML (Finin, Fritzson, McKay, & McEntire, 1994), a language and protocol for developing large-scale sharable and reusable knowledge bases. For the web, however, we should also consider the needs of application designers such as sales managers, travel agents, and schoolteachers. To this end, Ishida and his colleagues developed *Q*, a language for describing interaction scenarios between agents and users based on agent external roles (Ishida, 2002b). In the following sections, we use this language to provide the reader with an illustrative example of how interaction scenarios can be specified for a given application.

Specifying Scenarios in Q. *Q* is implemented on the top of Scheme, a Lisp programming language dialect. We first introduce sensing and acting functions and guarded commands in Scheme. *Cues* are events that trigger interaction. Scenario writers use cues to request that agents observe their environment. No cue can have any side effect. Cues keep on waiting for the specified event until the observation successfully completes. Comparable to cues, *actions* request that agents change their environment. Unlike functions in programming languages, *Q* does not define the semantics of cues and actions. Because different agents execute cues and actions differently, their semantics fully depend on the agents. *Q* introduces *guarded commands* for use in situations that require observing multiple cues simultaneously. A guarded command combines cues and forms, including actions. After either cue becomes true, the guarded command evaluates the corresponding form. If no cue is satisfied, it evaluates the otherwise clause. A scenario defines state transitions, and each state is defined as a guarded command. Although scenarios are written in the form of simple-state transitions, scenario writers can describe fairly complex tasks because each state can contain concurrent activities of agents and any form can be evaluated in the body of states.

A simple example is given as follows:

```
(defscenario reception (msg)
(scene1
  ((?hear “Hello” :from $x)
```

```
(defscenario card14 ()
 (scene1
  (t
   (!speak "Hm-hum, you are so enthusiastic.")
   (!speak "Then, how about this page?")
   (!display :url "http://kimono.com/index.htm")
   (go scene2)))
 (scene2
  ((?watch_web :url "http://kimono.com/type.htm")
   (!speak "There are many types of obi.")
   (!speak "Can you tell the difference?")
   (go scene2))
  ((?watch_web :url "http://kimono.com/fukuro.htm")
   (!gesture :animation "GestureLeft")
   (!speak "Fukuro obi is for a ceremonial dress.")
   (!speak "Use it at a dress-up party!")
   (go scene2))
```

FIGURE 12.22. Sample *Q* scenario: description and result of execution.

```
   (!speak "Hello":to $x)
   (go scene2))
   ((?hear "Bye")
   (go scene3)))
(scene2
   ((?hear "Hello" :from $x)
   (!speak "Yes, may I help you?" :to $x))
   (otherwise (go scene3)))
(scene3 ...))
```

In this example, each scenario defines states as scene1, scene2, and so on. Cues start with a question mark and actions start with an exclamation point. The same observation yields different actions in different states. In scene1, the agent says "Hello" when it hears someone say "Hello." In scene2, however, the agent responds with "Yes, may I help you?" when it hears "Hello" again.

Applications of Q Scenarios. *Q* can be applied to various legacy agents. Let us start with Microsoft agents. In the example, a user who wants to learn more about the traditional Japanese dress, visits the ***kimono*** Web site, and freely follows the links to related Web pages. Each time the user visits a new page, the agent summarizes its content. If the user does not react for awhile, the agent moves to the next subject. Figure 12.22 shows this *Q* scenario and its outcome.

Q and Microsoft agents are used to develop a multicharacter interface for information retrieval in which domain-specific search agents cooperate to satisfy users' queries (Kitamura et al., 2001). Previous research often used blackboard systems to integrate the results from multiple agents. However, given that computing professionals develop search agents independently, attempts to integrate these results are often unsatisfactory. Instead of integrating the results at the backend, our multicharacter interface increases user satisfaction by displaying the integration process to users as a dialogue involving multiple characters. Because *Q* is a general-purpose scenario description language, it grants too much freedom for describing scenarios for specific domains. We thus introduced ***interaction pattern cards*** (***IPCs***) to capture the interaction patterns in each domain. Figure. 12.23 shows an IPC equivalent of the *Q* scenario shown in Fig. 12.22. Scenario writers can use Excel to fill in the card. IPC provides a pattern language, and so it should be carefully designed by analyzing the interactions in each domain.

Designing and Executing Scenarios. A diverse variety of applications use interactive agents on the Web. To allow application designers to use fairly complex agents, we use *Q* in a scenario design process that provides a clear interface between application designers and computing professionals. The three-step process of creating a scenario focuses on a dialogue that bridges two perspectives (Murakami, Ishida, Kawasoe, & Hishiyama, 2003).

- First, a scenario writer and a computing professional agree on the cues and actions to use as the interface between them. Rather than using cues and actions assigned a priori, the two

<table>
<tr><td>Card ID</td><td>14</td><td>Card Name</td><td>Visiting Kimono Web site</td><td>Card Type</td><td>User Initiative</td></tr>
<tr><td rowspan="2">Opening</td><td colspan="5">Action</td></tr>
<tr><td colspan="5">Hm-hum, you are so enthusiastic.
Then, how about this page?
http://www.kimono.com/index.htm</td></tr>
<tr><td rowspan="6">Reactions to Users' Mouse Click (Repeat)</td><td rowspan="4">Mouse Click</td><td colspan="2">Cue</td><td colspan="2">Action</td></tr>
<tr><td colspan="2">http://kimono.com/type.htm</td><td colspan="2">There are many types of obi.
Can you tell the difference?</td></tr>
<tr><td colspan="2">http://kimono.com/fukuro.htm</td><td colspan="2">(GestureLeft)
Fukuro obi is for a ceremonial dress.
Use it at a dress-up party!</td></tr>
<tr><td colspan="2">http://kimono.com/maru.htm</td><td colspan="2">(Evaluate Card42)</td></tr>
<tr><td rowspan="2">No Reaction</td><td colspan="2">Seconds</td><td colspan="2">Action</td></tr>
<tr><td colspan="2">20</td><td colspan="2">(End of Repeat)</td></tr>
<tr><td rowspan="2">Closing</td><td colspan="5">Action</td></tr>
<tr><td colspan="5">Did you enjoy Japanese Kimono?
OK, let's move on to the next subject.</td></tr>
</table>

FIGURE 12.23. Interaction pattern card for the sample *Q* scenario in Fig. 12.22.

parties define a scenario vocabulary for each application domain through negotiation.

- Second, the scenario writer uses *Q* to describe a scenario, while the computing professional implements cues and actions.
- Third, the design process can introduce another actor, the interaction designer. This third actor observes the patterns of interaction in each domain and proposes IPCs. These cards trigger a dialogue between scenario writers and interaction designers, leading to a better understanding of the interaction patterns in each domain. IPCs also improve a scenario writer's productivity.

For autonomous agents, scenarios can be simple, whereas the scenario writer has to specify all details in case the agents are dependent. The granularity of cues and actions depends on two independent factors: the level of agent autonomy and the degree of precision scenario writers require.

EVALUATION OF INTERFACE AGENTS

Impact on Recall and Learning

One of the first field studies in this area was conducted by Lester and colleagues (2000). The study revealed that the presence of a persona of a persona might have a positive effect on the user's attitudes and experience of the interaction. Although Lester and colleagues only examined different agent variants, van Mulken and colleagues conducted an experiment to compare the effect of presentations with and without persons on the user's understanding, recall, and attitudes. Twenty-eight subjects were shown Web-based presentations with two different types of content. In the experimental condition, a speaking and gesturing PPP persona made the presentations. In the control condition, the (audiovisual) information presented was exactly the same, except that there was no persona and pointing arrows replaced all gesturing. After the presentations, the subjects were asked comprehension and recall questions and were subsequently provided with a questionnaire that measured their attitudes regarding the system and PPP persona. Statistical analyses of the results showed that there was no effect on comprehension or recall. However, analysis of the data on the subjects' attitudes indeed revealed a significant positive effect of persona. Subjects who had seen presentations guided by persona indicated on a questionnaire that they found the presentations themselves and the corresponding tests less difficult than subjects who had seen presentations without persona. In addition, subjects found these presentations significantly more entertaining (van Mulken, André, & Müller, 1998).

Au: Pls. Provide year.

Another study in the context of a tutoring system has been performed by Johnson, Shaw, Marshall, and LaBore (2003), who evaluated Adele, a web-based medical tutor agent. The study revealed that Adele was easy to use and that students found it helpful. The believability of the agent suffered, however, from the poor quality of speech synthesis.

Trustworthiness of Interface Agents

An important prerequisite for the acceptance of characters on the web is that the user trusts the agent. Therefore, it comes as no surprise that a number of empirical studies focus on factors that might have an influence on a character's trustworthiness. Van Mulken and colleagues (1999) conducted an experiment to compare the trustworthiness of different embodiments of agents. In this experiment, subjects had to conduct a navigation task. They were in turn assisted in navigation by one of four agents: one was invisible and merely gave textual recommendations as to how to proceed with the task, the second presented these recommendations acoustically, the third was a speaking cartoonstyle agent, and the fourth was a speaking agent based on video images of a real person. In the text and audio conditions, reference to a recommended path was accompanied by a highlighting of the corresponding parts of the navigation tree. In the conditions with an agent, such a reference was accompanied by pointing gestures. The authors hypothesized that the embodied agents would appear more convincing or believable and that the subjects would therefore follow the agents' recommendations more readily. This hypothesis, however, was not supported by the data. They found numerical differences only in the expected direction: The proportion of recommendations actually followed by the subjects dropped off, going from video based to cartoonstyle, audio, and text agents [for further details, see van Mulken, André, & Müller (1999)].

Another experiment to investigate the credibility of agents has been conducted by Nass, Isbister, and Lee (2000), who showed that computer agents representing the user's ethnic group are perceived as socially more attractive and trustworthy.

Impact of Personality and Social Behaviors of Interface Agents on User Acceptance

Cassell and Bickmore (2003) examined how to model the social relationship between a user and an embodied conversational agent and how to influence this relationship by means of the agent's dialogue behaviors. An empirical evaluation of the system revealed that users with different personalities indeed respond differently to social language.

Isbister, Nakanishi, Ishida, and Nass (2000) empirically evaluated an agent's ability to assist in cross-cultural conversations. The experiment revealed that the presence of the agent may have a positive influence on the perception of each other and of each other's cultural group.

Smith, Farnham, and Drucker (2000) investigated the influence of 3D features on social interaction in chat spaces. For instance, they showed that even experienced users actively make use of proximity and orientation features to enhance their interactions.

CONCLUSION

The detailed, anthropomorphous design of an interface agent is as important to its success in applications, as its believable, and emotional behavior in any conversation and dialogue with the user. This includes multimodal interaction via speech, mimic, gesture, simulation of movement in real time, as well as the tight coordination and synchronization of multiple modalities. Key questions for future research include: How should an information agent interact with its users in what kind of application context? In particular, how to effectively design and set up enabling technological infrastructure for interface agents to behave believably to the user? How to measure and evaluate the success of the interaction between interface agent and human user?

In this chapter, we provided some answers to these and related questions in detail, reviewing the state of the art in this domain at the same time. We do not expect the ultimate class of interactive information agents ever to be developed, but the next generation of interface agents may be available to the common user of the Internet and the Web within this decade.

One prestigious long-term research project toward the design and development of such intelligent interface agents is the ongoing large-scale VirtualHuman project (VirtualHuman Project, 2003) funded by the German ministry of research and education.

ACKNOWLEDGMENTS

Many of the presented systems that have been developed at DFKI comprise contributions from our colleagues Stephan Baldes, Patrick Gebhard, Michael Kipp, Martin Klesen, Markus Schmitt, and Thomas Schleiff and our colleagues from the SmartKom project. We also want to thank our graphics designer Peter Rist for providing us with the virtual actors Cyberella, Tina, Ritchie, and Smartakus.

Work on the scenario language *Q* would not have been possible without the contributions of Yasuhiko Kitamura, Yohei Murakami, Shiro Takata, and Akishige Yamamoto. *Q* has been developed by the Department of Social Informatics, Kyoto University, and the JST CREST Digital City Project. The source codes can be obtained from http://www.digitalcity.jst.go.jp/Q/.

Also, we thank Shinya Shimizu, Toyokazu Itakura, Kaori Sugiyama, and Nobutoshi Yamanaka for their vital contributions. FreeWalk was developed by the Department of Social Informatics, Kyoto University, and the JST CREST Digital City Project.

References

Ambite, J. L., Barish, G., Knoblock, C. A., Minton, S., Muslea, M., & Oh, J. (2002). Getting from here to there: Interactive planning and agent execution for optimizing travel. *Proceedings International Conference on Applied AI (IAAI 2002).*

André, E., Rist, T., & Müller, J. (1999). Employing AI methods to control the behavior of animated interface agents. *Applied Artificial Intelligence, 13*, 415–448.

André, E., Rist, T., van Mulken, S., Klesen, M., & Baldes, S. (2000). The automated design of believable dialogues for animated presentation teams. In J. Cassell, J. Sullivan, S. Prevost, & E. Churchill (Eds.), *Embodied conversational agents* (pp. 220–255). Cambridge, MA: MIT Press.

Ball, G., & Breese, J. (2000). Emotion and personality in a conversational character. In J. Cassell, J. Sullivan, S. Prevost, & E. Churchill (Eds.), *Embodied conversational agents* (pp. 189–219). Cambridge, MA: MIT Press.

Ball, G., Ling, D., Kurlander, D., Miller, J., Pugh, D., Skelly, T., Stankosky, A., Thiel, D., van Dantzich, M., & Wax, T. (1997). Lifelike computer characters: The persona project at Microsoft. In J. M. Bradshaw (Ed.), *Software agents*. Menlo Park, CA: AAAI/MIT Press.

Beneventano, D., Bergamaschi, S., Sartori, C., & Vincini, M. (1997). ODB-Tools: A description logics based tool for schema validation and semantic query optimization in object oriented databases. *Proceedings of International Conference on Data Engineering (ICDE-97)*.

Benford, S., Greenhalgh, C., Rodden, T., & Pycock, J. (2001). Collaborative virtual environments. *Communications of the ACM, 44*(7), 79–85.

Bergamaschi, S., Castano, S., Beneventano, D., & Vincini, M. (2001). *Semantic Integration of Heterogeneous Information Sources* [Special Issue on Intelligent Information Integration, Data & Knowledge Engineering], *36*(1), 215–249.

Bergenti, F., & Poggi, A. (2002). Ubiquitous information agents. *Cooperative Information Systems, 11*(3&4).

Börner, K., & Chen, C. (Eds.). (2002). *Visual interfaces to digital libraries*. LNCS State-of-the-Art-Survey, 2539. Springer. Heidelburg.

Card, S. K., Mackinlay, J. D., & Shneiderman, B. (Eds.). (1999). *Information visualization: Using vision to think*. San Francisco: Kaufmann.

Cassell, J., & Bickmore, T. (2003). Negotiated collusion: Modeling social language and its relationship effects in intelligent agents. *User Modeling and User-Adapted Interaction, 13*(1–2), 89–132.

Cassell, J., Pelachaud, C., Badler, N. I., Steedman, M., Achorn, B., Becket, T., Douville, B., Prevost, S., & Stone, M. (1994). Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. (SIGGRAPH '94 Proceedings). *Computer Graphics, 28*(4), 413–420.

Cassell, J., Vilhjálmsson, H., & Bickmore, T. (2001). BEAT: The behavior expression animation toolkit. In *Proceedings of SIGGRAPH 01* (pp. 477–486). New York: ACM.

Castano, S., De Antonellis, V., & De Capitani Di vimercati, S. (2001). Global viewing of heterogeneous data sources. *IEEE Transactions on Knowledge and Data Engineering TKDE, 13*(2), 277–297.

Chi, D., Costa, M., Zhao, L., & Badler, N. (2000). The EMOTE model for effort and shape. *ACM SIGGRAPH '00* (pp. 173–182). New Orleans, LA.

Collier, G. (1985). *Emotional expression.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Craig, S. D., Gholson, B., Garzon, M.H., Hu, X., Marks, W., Wiemer-Hastings, P., & Lu, Z. (1999). Auto Tutor and Otto Tudor. In *AIED-Workshop on Animated and Personified Pedagogical Agents* (pp. 25–30). Le Mans, France.

Elliott, C. (1992). *The affective reasoner: A process model of emotions in a multi-agent system*. PhD dissertation, Northwestern University. The Institute for the Learning Sciences, Technical Report No. 32.

El-Nasr, M. S., Ioerger, T. R, & Yen, J. (1999). Peteei: A pet with evolving emotional intelligence. In *Proceedings of the 3rd International Conference on Autonomous Agents* (pp. 9–15). New York: ACM.

Finin, T., Fritzson, R., McKay, D., & McEntire, R. (1994). KQML as an agent communication language. *Proceedings of International Conference on Information and Knowledge Management*, 456–463.

Gebhard, P., Kipp, M., Klesen, M., & Rist, T. (2003). Authoring scences for adaptive interactive performances. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 725–732) New York: ACM.

Isbister, K., Nakanishi, H., Ishida, T., & Nass, C. (2000). Helper agent: Designing an assistant for human–human interaction in a virtual meeting space. In *Proceedings of CHI 2000* (pp. 57–64). New York: ACM.

Ishida, T. (2002a). Digital city Kyoto: Social information infrastructure for everyday life. *Communications of the ACM, 45*(7), 76–81.

Ishida, T. (2002b). *Q*: A scenario description language for interactive agents. *IEEE Computer, 35*(11), 42–47.

Johnson, W. L., Shaw, E., Marshall, A., & LaBore C. (2003). Evolution of user interaction: The case of agent Adele. In *Proceedings of the 2003 International Conference on Intelligent User Interfaces* (pp. 93–100). New York: ACM.

Kendon, A. (1990). Spatial organization in social encounters: The F-formation system. In A. Kendon (Eds.), *Conducting interaction: Patterns of behavior in focused encounters* (pp. 209–237). Cambridge University Press.

Kitamura,Y., Yamada, T., Kokubo, T., Mawarimichi, Y., Yamamoto, T., & Ishida, T. (2001). Interactive integration of information agents on the web. In M. Klusch & F. Zambonelli (Eds.), *Cooperative information agents V* (pp. 1–13). New York: Springer-Verlag.

Klusch, M. (2001a). Information agent technology for the Internet: A survey. *Data and Knowledge Engineering, 36*(3).

Klusch, M. (Ed.). (2001b). Special issue on intelligent information agents: Theory and applications. *Cooperative Information Systems, 10*(1&2).

Klusch, M., Bergamaschi, S., Edwards, P., & Petta, P. (Eds.). (2003). *Intelligent information agents: The agentLink perspective*. LNAI, 2586. Springer. Heidelburg.

Klusch, M., & Sycara, K. (2001). Brokering and matchmaking for coordination of agent societies: A survey. In A. Omicini et al. (Eds.), *Coordination of internet agents*. Springer. Heidelburg.

Kopp, S., Jung, B., Lebmann, N., & Wachsmuth, I. (2003). Max—A multimodal assistant in virtual reality construction. In E. André & M. Rehm (Eds.), *Künstliche Intelligenz, Special Issue on Embodied Conversational Agent* (Vol. 4, pp. 11–17). are DTaPPubl., Bremen: Germany.

Laird, J. E., & Rosenbloom, P. (1996). The evolution of the SOAR cognitive architecture. In D. M. Steier & T. M. Mitchell (Eds.), *Mind matters: A tribute to Allen Newell* (pp. 1–50). Hillsdale, NJ: Lawrence Erlbaum Associates.

Lazarus, R. S. (1991). *Emotion and adaptation.* New York: Oxford University Press.

Lester, J. C., Towns, S. G., Callaway, C., Voerman, J. L., & FitzGerald, P. J. (2000). Deictic and emotive communication in animated pedagogical agents. In J. Cassell, J. Sullivan, S. Prevost, & E. Churchill (Eds.), *Embodied conversational agents* (pp. 123–154), Cambridge, MA: MIT Press.

Luck, M., McBurney, P., & Preist, C. (Eds.). (2003). *Agent technology: enabling next generation computing—A roadmap for agent-based computing.* University of Southampton, Southampton: U.K.

Marsella, S., & Gratch, J. (2002). A step towards irrationality: Using emotion to change belief. *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*.

Mauldin, M. (1994). Chatterbots, TinyMUDs, and the Turing test: Entering the Loebner prize competition. *Proceedings of Twelfth National Conference on Artificial Intelligence (AAAI-94)*.

Maybury, M. T., & Wahlster, W. (Eds.). (1998). *Readings in intelligent user interfaces.* San Francisco: Kaufmann.

Microsoft Task Gallery. Available: research.microsoft.com/ui/TaskGallery.

Middleton, S. E. (2001). *Interface agents: A review of the field.* ECS Technical Report ECSTR-IAM01-001. University of Southampton, Southampton, U.K.

Mladenic, D. S. J. (1999). Text learning and related intelligent agents: A survey. *IEEE Intelligent Systems.*

Moffat, D. (1997). Personality parameters and programs. In R. Trappl & P. Petta (Eds.), *Creating personalities for synthetic actors* (pp. 120–165). New York: Springer.

Montaner, M., Lopez, B., & De La Rosa, J. L. (2003). A taxonomy of recommender agents on the Internet. *Artificial Intelligence Review, 19*, 285–330.

Murakami, Y., Ishida, T., Kawasoe, T., & Hishiyama, R. (2003). Scenario description for multi-agent simulation. In *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*. New York: ACM.

Murtagh, F., Taskaya, T., Contreras, P., & Mothe, J. (2003). Interactive visual interfaces: A survey. *Artificial Intelligence Review, 19*(4).

Nass, C., Isbister, K., & Lee, E. J. (2000). Truth is beauty: Researching conversational agents. In J. Cassell, J. Sullivan, S. Prevost, & E. Churchill (Eds.), *Embodied conversational agents* (pp. 374–402). Cambridge, MA: MIT Press.

Negroponte, N. (1997). Agents: From direct manipulation to delegation. In J. M. Bradshaw (ed.), *Software agents* (pp. 57–66). Menlo Park, CA: AAAI/MIT Press.

OLIVE (Online Library of Information Visualization Environments). Available: http://www.otal.umd.edu/Olive/.

Ortony, A., Clore, G. L., & Collins, A. (1988). *The cognitive structure of emotions* Cambridge, England: Cambridge University Press.

Osgood, C., & Tannenbaum, P. (1955). The principle of congruity in the prediction of attitude change. *Psychology Review, 62*, 42–55.

Papazoglou, M. P., Laufmann, S., & Sellis, T. K. (1992). An organizational framework for cooperating intelligent information systems. *International Journal of Cooperative Information Systems, 1*(1), 169–202.

Paterno, F. (Ed.). (2002). *Human computer interaction with mobile devices*. LNCS, 2411. Springer, Heidelburg.

Pelachaud, C., Carofiglio, V., De Carolis, B., de Rosis, F., & Poggi, I. (2002). Embodied agent in information delivering application. *Proceedings of Autonomous Agents and Multiagent Systems.*

Perlin, K., & Goldberg, A. (1996). Improv: A system for scripting interactive actors in VirtualWorlds. *Computer Graphics, 28*(3).

Piwek, P., Krenn, B., Schröder, M., Grice, M., Baumann, S., & Pirker, H. (2002). RRL: A rich representation language for the description of agent behaviour in NECA. *Proceedings of the Workshop "Embodied conversational agents—let's specify and evaluate them!", held in conjunction with AAMAS-02.*

Poggi, I., & Pelachaud, C. (2000). Performative facial expressions in animated faces. In J. Cassell, J. Sullivan, S. Prevost, & E. Churchill (Eds.), *Embodied conversational agents* (pp. 155–188). Cambridge, MA: MIT Press.

Prendinger, H., & Ishizuka, M. (2001). Social role awareness in animated agents. In *Proceedings of the Fifth Conference on Autonomous Agents* (pp. 270–377). New York: ACM.

Reilly, W. S. (1996). *Believable social and emotional agents*. PhD thesis. Technical Report CMU-CS-96-138, School of Computer Science, Carnegie Mellon University, Pittsburgh.

Rich, C., Sidner, C. L., & Lesh, N. (2001). COLLAGEN: Applying collaborative discourse theory to human–computer interaction. *AI Magazine, 22*(4), 15–25.

Rickel, J., & Johnson, W. L. (2000). Task-oriented collaboration with embodied agents in virtual worlds. In J. Cassell, J. Sullivan, S. Prevost, & E. Churchill (Eds.), *Embodied conversational agents* (pp. 95–122). Cambridge, MA: MIT Press.

Rist, T., André, E., & Baldes, S. (2003). A flexible platform for building applications with life-like characters. In *Proceedings of the 2003 International Conference on Intelligent User Interfaces* (pp. 158–165). New York: ACM.

Rist, T., & Schmitt, M. (2002). Applying socio-psychological concepts of cognitive consistency to negotiation dialog scenarios with embodied conversational characters. *Proceedings of AISB'02 Symposium on Animated Expressive Characters for Social Interactions*, 79–84.

Shah, K., & Sheth, A. (1999). InfoHarness: An information integration platform for managing distributed, heterogeneous information. *IEEE Internet Computing*, November/December, 18–28.

Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., & Warke, Y. (2002). Semantic content management for enterprises and the web. *IEEE Internet Computing*, July/August, 80–87.

Sheth, A., & Maes, P. (1993). Evolving agents for personalized information filtering. In *Proceedings of the Ninth Conference on Artificial Intelligence for Applications '93*. Orlando FL: IEEE Computer Society Press.

Sheth, A., Shah, K., Parsuraman, K., & Mudumbai, S. (1999). *Searching distributed and heterogeneous digital media.* 8th IFIP 2.6 Working Conference on Database Semantics (DS-8).

Sheth, A., Thacker, S., & Patel, S. (2003). Complex relationship and knowledge discovery support in the InfoQuilt system. *VLDB Journal, 12*(1).

Shklar, L., Sheth, A., Kashyap, V., & Shah, K. (1995). InfoHarness: Use of automatically generated metadata for search and retrieval of heterogeneous information. *Proceedings of CAiSE '95*.

Shneiderman, B. (1997). Direct manipulation versus agents: Paths to predictable, controllable, and comprehensible interfaces. In J. M. Bradshaw (Ed.), *Software agents* (pp. 97–106). Menlo Park; CA: AAAI/MIT Press.

Smith, M. A., Farnham, S. D., Drucker, S. M. (2000). The social life of small graphical chat spaces. In *Proceedings of CHI 2000* (pp. 462–469). New York: ACM.

Spence, R. (2000). *Information visualization*. Reading, MA: Addison-Wesley.

van Mulken, S., André, E., & Müller, J. (1998). The persona effect: How substantial is it? In H. Johnson, L. Nigay, & C. Roast (Eds.), *People and computers XIII (Proceedings of HCI-98)* (pp. 53–66). Berlin: Springer.

van Mulken, S., André, E., & Müller, J. (1999). An empirical study on the trustworthiness of lifelike interface agents. In H.-J. Bullinger & J. Ziegler (Eds.), *Human–computer interaction (Proceedings of HCI-International 1999)* (pp. 152–156). Mahwah, NJ: Lawrence Erlbaum Associates.

VirtualHuman Project. (2003). Available: http://www.virtual-human.org.

Wahlster, W., Reithinger N., & Blocher, A. (2001). SmartKom: Multimodal communication with a life-like character. *Proceedings of EUROSPEECH'01.*

Weiss, G. (Ed.). (1999). *Multiagent systems*. Cambridge; MA: MIT Press.

Yoon, S., Blumberg, B., & Schneider, G. E. (2000). Motivation driven learning for interactive synthetic characters. In *Proceedings of the Fourth International Conference on Autonomous Agents* (pp. 365–372). New York: ACM.