COMMERCIAL SCENARIOS OF DIGITAL AGENT DEPLOYMENT: A FUNCTIONAL CLASSIFICATION

Arno Scharl
Vienna University of Economics & Business Administration, Austria
scharl@wu-wien.ac.at

Christian Bauer
Vienna University of Economics & Business Administration, Austria
bauerc@cbs.curtin.edu.au

Marion Kaukal
Vienna University of Economics & Business Administration, Austria
mkaukal@wu-wien.ac.at

ABSTRACT

The evolution and commercialization of Web Information Systems (WIS) since 1991 has gradually shifted the focus from basic design issues to feedback analysis, visualization techniques, and agent-based negotiation support. Even during the early stages, WIS design efforts turned out to be quite substantial in scale and required a planned, managed, and structured approach. Adaptive solutions avoid redundant repetition, facilitate navigation, and increase the perceived value of provided information or services. Digital agents as proactive, goal-oriented systems promise to increase flexibility and will radically change inherent characteristics of electronic commerce. An evolution-based framework for WIS is provided and software agents are classified according to their functionality and architectural attributes into information agents, co-operation agents, and transaction agents. In this last stage of the framework, agents independently search virtual spaces and identify suitable business partners for exchanging information, collaborating in virtual organizations, negotiating contractual terms, and performing commercial transactions.

Keywords: Evolution of Web Information Systems, Digital Agents, Commercial Transactions, Agent-mediated Electronic Commerce, Interactivity, Communication Languages

1. Introduction

Delivering customized contents with adaptive WIS aims at identifying potential users and presenting them information specifically tailored to their individual needs and preferences. Commercial organizations will seek the competitive advantage of deploying adaptive WIS in distinguishing their on-line services from competitors and providing added value to customers. While being motivated by a user-centered design perspective, the problem domain goes beyond the scope of WIS interfaces or document presentation and includes the development of flexible software architectures and corresponding business models to take advantage of adaptive system behavior. Due to the heterogeneous character of customer profiles and market allocation mechanisms, it remains difficult to adequately support them using traditional system architectures.

Changes in information technology, organizational structure, and the corporate value chain strongly influence electronic commerce as a new business paradigm. Since the emergence of the World Wide Web in 1991, interactivity and adaptability have become apparent features of WIS and as such provide the vertical dimension of Figure 1. Changing communication models have raised the need for corresponding modifications, extensions and innovations of WIS design paradigms, techniques and tools.[1] Each stage of the WIS evolution process shifts the focus and the timing of design and user feedback analysis into a new context [Bauer et al.1999]. Figure 1 visualizes these changes and their relation to design (D), feedback analysis (A), and negotiation (N).

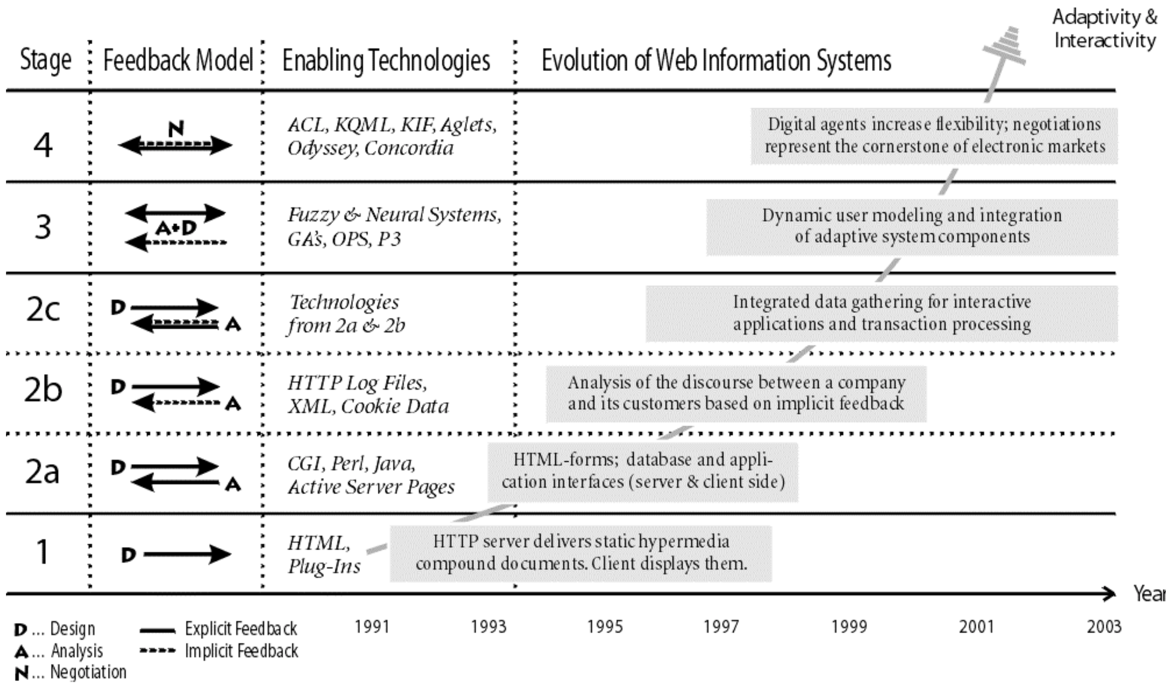

Figure 1. Evolution of the WIS infrastructure

2. Methodology

While technical implementation [Knapik & Johnson 1997, Lange & Oshima 1999], potential economic value, and social impact [Guttman & Maes 1998, Guttman & Moukas 1998] of digital agents have been the main focus of academic research so far, this paper examines strategies and processes to facilitate the commercial use of agent technology. In most organizations the introduction of digital agents into business processes and the "everyday" commercial life has not been accomplished yet and is not likely to happen in the short- to medium-term due to the technical complexity and the lack of commercially viable applications. This study is therefore not aiming at providing empirical validation, but looking at predicting future organizational agent scenarios and proposing strategies to extract the maximum commercial value from these scenarios.

The scenario planning for the prognosis of future applications of agent technology in commercial organizations (as utilized in section 5) is based on the interpretation of two main themes:

- The evolution of Web-based information systems from rudimentary, static Web sites to advanced interactive applications that support customized principal-agent relationships (see section 3).
- The early implementations of agent-mediated architectures that are already available via the World Wide Web today (see section 4).

Based on the hermeneutic approach and the qualitative "extrapolation" of these main themes, this research can be classified as "interpretive". Interpretive research is not concerned about formulating hypotheses with measurable dependent and independent variables, but recognizes that knowledge is acquired through human sense making and social constructs [Kaplan & Maxwell 1994]. Examples of interpretive research in developing and evaluating information systems are the work of Boland [Boland 1991] and Walsham [Walsham 1993]. The research objectives for this study can be summarized as follows:

- Specification of a framework for the deployment of digital agents in commercial, Web-based environments.
- Model-based description of the principal-agent relationship for such environments. This model has to incorporate strategic dimensions (if the principal is viewed in an organizational context), usability considerations, and visualization mechanisms (if the principal is seen as an individual member of that organization).
- Design of the business processes required for optimizing the commercial value of Internet agent deployment.

3. The Pre-Agent Era

3.1 A Static World

In the first stage, stand-alone servers deliver simple hypermedia compound documents, subsequently being displayed by the browser. Information flows in one direction only (from the server to the client), since user feedback is disregarded in this early stage of WIS development. Nevertheless, the design efforts necessary for such Web sites can be quite substantial in scale and require a planned, managed, and structured approach. Such Web site design methodologies were suggested in the works of Bichler and Nusser, Isakowitz et al., Nanard and Nanard, and Scharl [Bichler & Nusser 1996, Isakowitz et al 1995, Nanard & Nanard 1995, Scharl 1997, Scharl 1998]. A classification of various academic and commercial methodologies can be found in Bauer [Bauer 1998].

At least to some extent every user is able to specify the general appearance of documents by setting standard browser preferences, representing a rather low-level form of adaptability. However, with the progressive commercialization of the Internet and the integration of additional layout options companies have increasingly tried to determine the exact design of their documents due to strategic marketing considerations and in order to maintain a consistent corporate identity.

3.2 The Emergence of Interactivity

Planned analysis of user feedback is embedded into the underlying communication model in stage two. Depending on the sub-model (Stages 2a-2c), the analytical process covers methods for gathering implicit and/or explicit feedback. Although the design of questionnaires becomes critical to ensure their acceptance, the data from explicit user feedback is easily obtained by means of on-line forms [5]. Traditional fields such as demand-oriented market research or empirical social research together with the corresponding statistical processing provide a sound background for utilizing the acquired information. Gathering, reporting, and visualizing implicit feedback usually require more sophisticated approaches [Scharl & Bauer 1998] and are still underutilized in many commercial WIS despite the availability of an extensive array of suitable software solutions [Malchow & Thomsen 1997]. Not providing methods for the structured analysis of explicit and implicit sources of feedback represents a serious shortcoming of prevalent WIS design methodologies. WebMapper, a Java-based prototype for visualizing individual and aggregated user clickstreams, represents an effort to overcome this shortcoming by interpreting HTTP server log-file data and matching the results to the meta-model of the extended World Wide Web Design Technique ([Scharl 2000]; http://webdev.wu-wien.ac.at/).

WIS create new value for customers and deliver this value in an innovative and highly efficient way. During stage two, the connection between hypertext front-end and the underlying database architecture is established either on the server-side by means of the Common Gateway Interface (CGI) or Application Programming Interfaces (API), or on the client-side with technologies such as ActiveX or Java-Applets [Sommer & Zoller 1999]. The popularity of scripting languages for providing process control increases at the same time, again for both server (e.g. Perl) and client (e.g. JavaScript). However, there are no efficient mechanisms for interactive dialogs, user authentication, or the processing of transactions. As a consequence, little is known about the users themselves. Information about potential customers frequently remains incomplete and does not reflect economic reality. Therefore, more sophisticated markup languages such as Extensible Markup Language (XML), database connectivity, certification services, and application interfaces for integrated transaction processing gain popularity in stage two, overcoming the functional limitations of applications solely based on a loosely connected set of HTML documents (HTML = Hypertext Markup Language). HTML documents are designed for hypermedia representation and optimized for the document rendering requirements of early Web implementations (and have been extended with frames, tables, and other more advanced formatting constructs). As a consequence, HTML provides rich facilities for display, but no standard way to manage meta-data. The move from HTML to XML, a reduced subset of the Standard Generalized Markup Language (SGML), is likely to significantly affect the customizability of WIS documents [50]. SGML, an international standard (ISO 8879) published in 1986, is a meta-language formalism that facilitates the definition of descriptive markup languages for the purpose of electronic information encoding and interchange. HTML is an example of an SGML-based language. The International SGML Users' Group promotes its use and related standards such as DSSSL, HyTime, and XML.[1] XML is a simpler subset of SGML and allows an easier definition of specialized markup languages. Whereas HTML as presentational markup language imposes a lowest common denominator for document rendering, XML as a semantic markup language is extensible, validatable by external modules, and allows the definition of self-documenting tags [Ginsburg 1998]. Compare, for example, the *Annotate!* system described by Ginsburg and Kambil [21] which couples documents with XML annotations, adding value to the deployed WIS by increasing its overall semantics.

3.3 Adaptive Solutions

The concept of adaptability is extended beyond visual design by building systems with parameterized functionality. Available attributes and preferences of registered users may be stored in profile databases and incorporated into WIS using simple rule-based constructs. Granting different access privileges according to IP domain, personally addressing customers with dynamically generated documents, or determining purchase conditions according to user category are typical scenarios which require the server-side database and application interfaces of stage three. In this advanced scenario, user feedback is being processed instantly within the expected

WIS response time. Adaptive technologies like neural networks, genetic algorithms, natural language generation [Milosavljevic1998], case-based reasoning [Finnie & Wittig 1998], or related soft computing approaches are used to leverage the knowledge contained in such user models. Although they require considerable resources in terms of memory and computational speed during initialization and training, most of these technologies perform well when it comes to applying the knowledge to a particular case. Furthermore, completely accurate computations are rarely required in a business context. Heuristic solutions usually suffice. Incorporation of knowledge-based methods increases the functionality of deployed applications, independent of the complex infrastructurebeing necessary for the agent-mediatedsystems of stage four. They are still based on the same, slightly advanced network information infrastructure as the applications characteristic of the first two stages, but improve communication and transactions with dynamic responses generated on the fly.

Adaptive applications of stage three cannot be designed without a clear understanding of the appropriate parameters on which the adaptive behavior relies on. The definition of these parameters requires a detailed economic and socio-behavioral analysis as well as a feasibility study to gather and use the required information on a large scale. WIS developers demand visual development methods and tools for the creation of commercial applications. Such methods may be derived from existing WIS modeling methods by extending their functionality with dynamic site management. Tools should incorporate new additions to the repository of Web technologies, for example methods derived from research on artificial intelligence, data mining, user modeling, or advanced knowledge representations.

Keeping track of users interactions and reasoning about their intentions [Milosavljevic1998], dynamic solutions avoid redundant repetition, facilitate navigation, and increase the overall perceived value of the provided information or services. With reduced barriers between productive data processing (transactions) and dispositive data processing such as market analysis, WIS-tracking, or data warehouses, the wide-spread consideration of dynamic user models for customizing WIS will become a necessity for every serious commercial project and – in contrast to agent technology – does not require a complete redesign of the underlying IS architecture.

4. Agent-mediated Architectures

The focus on digital agents in stage four represents an ambitious effort, especially as far as information retrieval and complex negotiations are concerned.[1] Such agents, which promise to further increase flexibility and to radically change inherent characteristics of electronic commerce, are characterized by a number of attributes (see Table 1), which determine their usually cooperative behavior [Lange & Oshima 1999, Schubert et al 1998, Singh 1999]. The two orthogonal properties, interactivity and mobility, are not required by a number of agent definitions in the literature.

Table 1. Characteristic attributes of digital agents

Ability	**Description**
Proactivity, Reactivity	Initiate processes and react to internal or external events.
Intentionality, Goal-orientedness	Actively choose appropriate methods for pursuing a certain goal.
Adaptability	Learn and adapt to changes in the environment. Potential information sources include the observation of user actions, the analysis of user feedback, explicit user instructions, multi-agent collaboration, and so forth [1].
Autonomy	Act in an independent manner without direct intervention by the principal.
Interactivity	Communicate with each other, usually via an agent communication language (ACL). Some authors use the term multi-agent system to describe this attribute.
Mobility	Migrate between different information systems within realistic software environments. In contrast to their mobile counterpart, stationary agents are bound to the system on which they began execution.

The enhanced functionality of autonomous agents negotiating with each other over a network requires rethinking, reinventing and rebuilding World Wide Web communication models. Nevertheless, every innovation in this specific segment will have to provide backward-compatibility and -interoperability to enable seamless integration [Lindemann & Runge 1998]. Principal-agent relationships replace the traditional client-server approach with agent software acting as clients and servers at the same time [24]. Digital agents consider the principal's preferences, employ standardized coordination mechanisms, and automatically adapt to the situational requirements of their tasks. The "request-response" model predominant during the first three stages and the underlying protocol (HTTP) will be dissolved by the direct interaction of equal partners in an agent-driven communication network environment. This becomes most apparent in the automation of complex negotiation mechanisms and models for commercial applications[Bichler & Segev 1998]. The communication model dominant in stage four, therefore, no longer refers to design or analysis, but replaces these traditional concepts by the term negotiation (N) as depicted in Figure 1.

Software agents may be categorized according to their functionality and architectural attributes into information, co-operation and transaction agents [Schubert et al. 1998]. Information agents present information obtained from the Internet or specific WIS after applying filtering algorithms, investigating connections and interdependencies to related content material, and generating the preferred presentation style. For the customizability of WIS they are most useful as a "gateway" to information offerings. In the same context of adaptive WIS, co-operation agents are employed to synchronize, share and communicate the subjective preferences of user groups using analogies, correlations, and other statistical methods. Transaction agents are highly specialized programs that are used to perform the negotiation and settlement phases of electronic commerce [Schmid & Lindemann 1998]. The highly sophisticated algorithms and communication architectures enable customers and suppliers to participate in electronic marketplaces by setting strategies and endorsing transaction agents.

Currently, the hosting of agents is an issue of diverse discussions. Issarny and Sarodakis propose an example for an open architecture that allows customized remote execution of Web agents [Issarny & Saridakis 1999]. They discuss the host’s requirements for automating decision processes in agent execution environments. Many agent-based systems involve existing, dynamically created, or acquired knowledge. Knowledge about a certain domain is represented in a knowledge base. By the use of logical operators (i.e., Boolean, propositional, or first-order predicate calculus), inferencing can be performed on a knowledge base. Reasoning is important for the agent trying to ascertain the truth-value of its own beliefs or assertions or those made by other agents [Knapik & Johnson 1997].

4.1 Information Agents

Information agents are used for individualizing Internet-based communication and present each user with a personalized, intuitive interface that hides the more complex system architecture. They typically store a user profile together with the learning algorithm on the client side (for general Internet browsing) or server-side (for specific WIS). The personalization rules can either be specified by the user herself as explicit preference and/or by the system learning from the behavior of the user while navigating through it. The observed interests can be used to make recommendations, to personalize content and user interface, and to precisely target advertising.

Filtering agents are one of the simplest instantiations of information agents. Information filtering recognizes the interest of a user and delivers content that is based on the particular interests. Other systems can be considered as more proactive. They request particular pieces of information from *different* information sources and allow the user to deal with distributed sources of information effectively. It is obviously of importance that the requested information satisfies the user's query and goals. Thus the request sent by the retrieval agent – like that of the filtering agent – must be based both on the information obtained directly from the user and also upon any user model the system has managed to build up. Systems such as the BargainFinder[1] or All Direct Books[i] (performing comparison-shopping for books) are examples of this approach, but at the moment they are still lacking consistent user modeling.

Information agents show varying degrees of autonomy, being able to perform – to a certain extent – actions without the explicit consent of the owner. They acquire user preferences and use them to automate repetitive tasks or conduct searches on behalf of their principal. They are able to refine Internet-searches based on user-feedback, can automatically access HTTP servers and derive run-time navigational decisions based on what they find on the Web. In order to allow automated filtering, metadata that describe the contents of Web resources is essential. The Resource Description Framework[1] (RDF) is an approach for providing a standard way for using XML to represent metadata in the form of properties and relationships of items on the Web.

4.2 Co-operation Agents

Co-operation agents enable principals to establish a working relationship with others by sharing information and reaching agreements on business processes. The following detailed description of the application of co-operation agents for the implementation of recommender systems serves as an illustration for the potential of these systems.

Information agent systems that apply user profiling learn from WIS access patterns with or without the direct notice or action of the user. The user profile is automatically created and updated depending on the user's reaction to the delivered personalized content or recommendations. A problem of this approach is the initialization effort of the customer profile of new users. Co-operation agent systems, based upon the subjective evaluations of persons about the delivered content (social information), represent an approach to overcoming this barrier. Collaborative filtering systems correlate one customer's ratings with those of others in order to identify patterns of preferences resembling those of that particular customer. The system combines the ratings of these like-minded individuals to recommend items of interest [Balabanovic & Yoav 1997].

The reaction of a person to a delivered document is tracked either by explicit or by implicit ratings and is saved in the user profile for later information delivery (content-based filtering). Explicit ratings allow the user to directly express opinions about the delivered recommendations by evaluating the content as a number or on qualitative terms (e.g. poor | fair | average | good | excellent). Implicit ratings are collected by observing which products each customer chooses to look at, inquire about, or purchase. It is equally important to track which advertisements they click through and how much time they spend viewing different pages or media. In order to initialize the profile, new WIS visitors have to fill out a detailed evaluation form, where they are required to assess presented information objects. Only if enough individual evaluations are available, can collaborative filtering work.

However, only if the critical mass of users is surpassed, enough clusters of like-minded users can be built. Furthermore, if a new information object is added to the information system, no evaluations are available for that particular object. A collaborative system must then solve the problem of to whom this information object should be recommended.

Collaborative filtering is ideal for recommending opinion-based content such as books or music. Content-based filtering, on the other hand, can suggest personalized content even if there is only one visitor at a site. In practice, hybrid approaches that integrate collaborative and content-based approaches, are most efficient. Pioneers in collaborative Filtering Technologies were Firefly[1] with their CD Recommender System and Netperception[i] with their GroupLens[i] system (Amazon.com[i] has licensed the GroupLens technology in the meantime). Systems such as WiseWire[i] and OpenSesam[i] apply a hybrid approach.

To implement personalization techniques on the server-side it is important to ask visitors to register and provide basic information about their identity before access to protected parts of the WIS is granted. Each time a given customer revisits a site, he must identify himself either by specifying username and password or with the help of data that is saved on the client-side. Co-operative agents relying on cognitive information detect dependencies between objects and recommend items similar to those a given user has liked in the past. In analogy to an attentive shop assistant, the system observes customer habits and interests as they browse the product displays. Each individual's path is monitored and machine-learning techniques are used to infer conclusions about the rationale behind the behavior of the user. Frequent returns of the user result in a higher quality of the user profile. A content-based filtering system becomes more effective when additional visitor information is gathered. It takes several subsequent visits before the system can come up with a meaningful profile for a visitor.

4.3 Transaction Agents

Transaction agents complement information agents as they complete a business transaction by negotiating and fulfilling a business contract. Negotiation execution is the main operation area of a transaction agent. In a typical commercial negotiation process buyers and suppliers meet on a marketplace, express their interests and capabilities and adjust them until an agreement is reached. The subject of the negotiation is a product and its attributes (color, dimensions, quality, etc.) as well as other terms of a deal (e.g., price or terms of delivery). In a traditional negotiation situation the amount of negotiation partners as well as the number of negotiated attributes is kept low, as negotiation is a very time consuming and complicated process, especially if the traded products are complex. Software agents can help make the negotiation phase of a business transaction more efficient. They meet each other at a central marketplace where a negotiation protocol is provided that determines the valid actions of the trading partners during the negotiation process as well as the rules for conducting the matchmaking.

Multi-attribute auctions are an example of the implementation of a negotiation protocol. A buying agent that takes part in a multi-attribute procurement auction [Bichler et al. 1999] can express its utility function for multiple attributes of a deal and therewith start an auction process. The system on the marketplace determines and informs the suitable selling agents, which have registered their capabilities at the marketplace. They compose their bids according to the buyer’s utility function. At the end of the auction period the agent(s) that best satisfy the utility function of the buying agent are determined and the deal can be closed.

An early example of an agent-mediated transaction system is Market Maker (formerly known as Kasbah), an ongoing research project at the Software Agents Group of the MIT Media Laboratory.[1] The architecture of the Market Maker server includes a set of CGI-scripts running on an SSL protected HTTP-server and a database where the buying and selling marketplaces reside [Chavez & Maes 1996, Zacharia et al.1999]. Currently, the Market Maker system facilitates the trading of books, computer games, and various music media. The agents in the Market Maker project are not mobile. They are executing on a dedicated server that represents the market environment.

A user wanting to trade goods creates an agent, gives it some strategic direction, and sends the agent off into the agent marketplace. Market Maker agents proactively seek out potential buyers or sellers and negotiate with them on their creator's behalf. Each agent's goal is to meet a number of user-specified constraints, such as a desired price, a highest (or lowest) acceptable price, and a date by which to complete the transaction. In the Market Maker system, negotiation is straightforward, bilateral, and competitive. It provides buyers with three different negotiation strategies: anxious, cool-headed, and frugal – depending on the selected price raise function (linear, quadratic, or exponential) depicted in Figure 2. The simplicity of these heuristics makes it intuitive for users to understand the inherent structure of the marketplace [Brenner et al 1998, Guttman & Maes 1998]. After two agents are matched, the only valid action for buying agents is to offer a bid to selling agents with no restriction on time or price. Selling agents respond with either a binding "yes" or "no" [Guttman et al.1998].

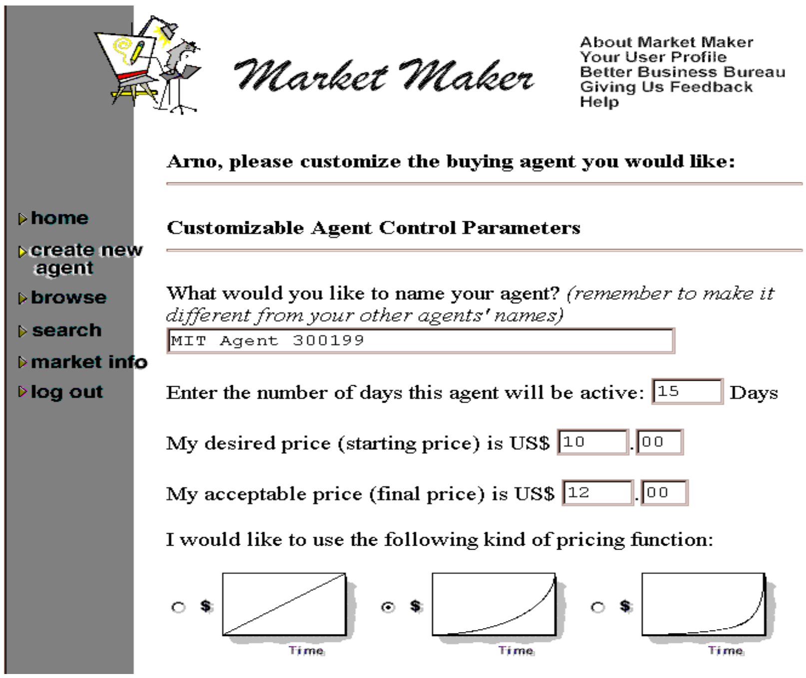

Figure 2. Customizable Market Maker Buying Agent

Negotiation processes are very interaction-intense. To compare and contrast products or suppliers as well as express preferences and needs, agents need to interact and exchange both syntactic and semantic information with other local or remote agents. They usually communicate through asynchronous message passing based on speech act theory (speech acts are actions performed via utterances; the circumstances surrounding the utterance, including other utterances, are called speech events[Yule 1996, Sommer & Zoller 1999]). For most computing scenarios [Singh 1999, Yule 1996], these communicative acts inform (*assertive*), request (*directive*), promise (*commissive*), give permission for another act (*permissive*), ban another act (*prohibitive*), cause events (*declarative*), or express emotions and evaluations (*expressive*).

The exchange of messages is established via an agent communication language (ACL), which provides agents with a means of exchanging information and knowledge. KQML[1] and the Foundation for Intelligent Physical Agents' FIPA ACL[i] are two examples of ACLs that separate the outer or package language from the inner or content language. The core ACL supplies the speech act that indicates the message type and is independent of the content language that supplies the domain-dependent part of the message. The content language is typically a formal language with precise semantics such as KIF.

KIF is a prefix version of first-order predicate calculus with extensions to support meta-operators and definitions [31]. As such it includes declarative semantics (i.e., the meaning of expressions in the representation can be understood without referring to an interpreter to manipulate those expressions), is logically comprehensive (i.e., it provides for the expression of arbitrary sentences in the first-order predicate calculus), and provides for meta-knowledge representation (knowledge about knowledge). Labrou, Finin and Peng [Labrou et al 1999] give the following example of a simple KQML message. Using lambda prolog[1] as the content language, they describe a query about the price of a share of IBM stock and the corresponding reply of the server:

(ask-one
:sender joe
:content (PRICE IBM ?price)
:receiver stock-server
:reply-with ibm-stock
:language LPROLOG
:ontology NYSE-TICKS)

(tell
:sender stock-server
:content (PRICE IBM 14)
:receiver joe
:in-reply-to ibm-stock
:language LPROLOG
:ontology NYSE-TICKS)

ACLs allow the agents to state which vocabulary they are presuming as the basis for their messages. As negotiation partners on the Web can be very heterogeneous and use different terms for describing domains, interoperation can be difficult. Therefore unambiguous semantic specifications must be provided. KIF can be used to represent so-called *ontologies*. These are a means of specifying a domain-specific vocabulary of entities, classes, properties, predicates and functions as well as a set of relationships that necessarily hold among those vocabulary items [Fikes & Farquhar 1999]. Ontologies are used as a basis for sharing a common terminology between multiple systems. The terms provided by the domain ontology can be used to assert specific propositions about a domain or a situation in a domain [Chandrasekaran 1999]. The aim is not the definition of one common vocabulary for all trading agents but the sharing of a foundational ontology that is used as the basis for interoperation among the agents. Therefore the merging of ontologies as well as the translation between them has to be supported. Standardization plays a very important role for interoperability and shared understanding. Ontology.org[1] is an organization that supports the development of reusable standard ontologies and associated XML schemas. *Ontolingua*, an ontology development environment, was developed for ontology authoring and translation. It provides a suite of tools and a library of modular reusable ontologies and helps constructing portable ontologies.

Other examples of ACLs are France Télécom's Arcol, FLBC (Formal Language for Business Communication), or Agent-0 [Labrou et al 1999, Moore 1999, Singh 1999]. However, most of them are not taking full advantage of Internet technologies. Internet standardization organizations tend to have other issues on their agenda [Labrou 1999]. RDF with XML as interchange syntax, for example, is a possible ACL content language. Although RDF is not a full predicate calculus language like KIF and therefore less expressive, it is much simpler to implement.

In conventional environments business-to-business commerce is conducted by the exchange of electronic business documents. There are standard syntax requirements for the different document types (e.g., invoices) called EDI message types (EDI = Electronic Data Interchange). They are typically used in the post-contract phase and are not appropriate for ad-hoc business contacts. Teich et al. proposed some specific agent architectures that are especially suited for business-to-business exchange markets. They developed simple multiple-issue algorithms and heuristics that can be used in electronic auctions to match businesses with other businesses and consumers based on dovetailing their underlying interests and preferences [49]. For business-to-consumer markets, efficient ways to implement the required functionality for each transaction phase have to be evaluated with special regard to standardized description models for user profiles such as the Open Profiling Standard (OPS) or the World Wide Web Consortium (W3C)'s Platform for Privacy Preferences (P3P). Users can delegate decisions to their personal agent whenever they wish. It is also a way for users to know what the corporate privacy policies are and to reconcile the preferences of the users with the policies of company [Lee & Speyer 1999]. P3P provides a framework for disclosing customer information during on-line interactions. By allowing users to specify what kind of personal information they are willing to divulge to WIS, P3P applications support tailored relationships with specific WIS [Reagle & Cranor 1999]. While OPS and P3P are somewhat similar, the specific focus of each technology is different. While OPS deals with secure transport and control of user data, P3P concentrates on enabling the expression of privacy practices and preferences. The commercially available product Cupcakes, for example, is one of the first products built upon the OPS foundation.[1]

Because of the ubiquity of the Java virtual machine and advanced class-loading mechanisms [Wong et al 1999], tools for developers to implement agent systems are often released in the form of Java classes that support their implementation. Popular examples are IBM's Aglets,[1] General Magic's *Odyssey*[i] *(the Java-successor of Telescript), or Mitsubishi Electric's Concordia [27, 30, 48].*[i]

4.4 Modeling Digital Agents for Commercial Applications

Negotiation processes between business entities are characterized by high degrees of unpredictability, complexity, and strategic importance to organizations [Beam & Segev 1997] and it can be assumed that a number of design methodologies for commercial utilization of digital agents will be made available with the progress of this technology. So far, most of these methodologies are still in their infancy. There are practically no user-friendly tools with graphical interfaces that intuitively implement the underlying theory [Bradshaw 1999]. With their Java-based Iconic Modeling Tool, Falchuk and Karmouch present one of the few exceptions where users can interactively model agents and their dynamic itineraries in a visual environment [16]. Novice users are able to create complex agents within minutes. The architecture is layered from simple iconic representation down to a textual itinerary language. Skilled users can therefore bypass the higher-level interface and program directly in the itinerary language.

The emergence of visual design methodologies for agent development works as an enabler for more transparency and concentration of the strategic approaches within electronic markets. However, the transformation to agent environments with bi- and/or multi-lateral interaction between independent participants in commercial scenarios requires the extension of previous modeling approaches to include the whole development process at either end. The increased adaptivity of user-centric, visual tool-sets and the reduced response times from electronic marketplaces suggest an evolutionary approach as the most obvious solution [Bauer et al 1999]. Figure 3 highlights these interdependencies graphically by connecting the (cybernetic) agent development feedback cycles of electronic market participants. Additionally, it portrays the principal-agent (user-system) relationship through the definition of distinct interaction phases and the acknowledgment of the overall importance of the user model in the center of both feedback cycles.

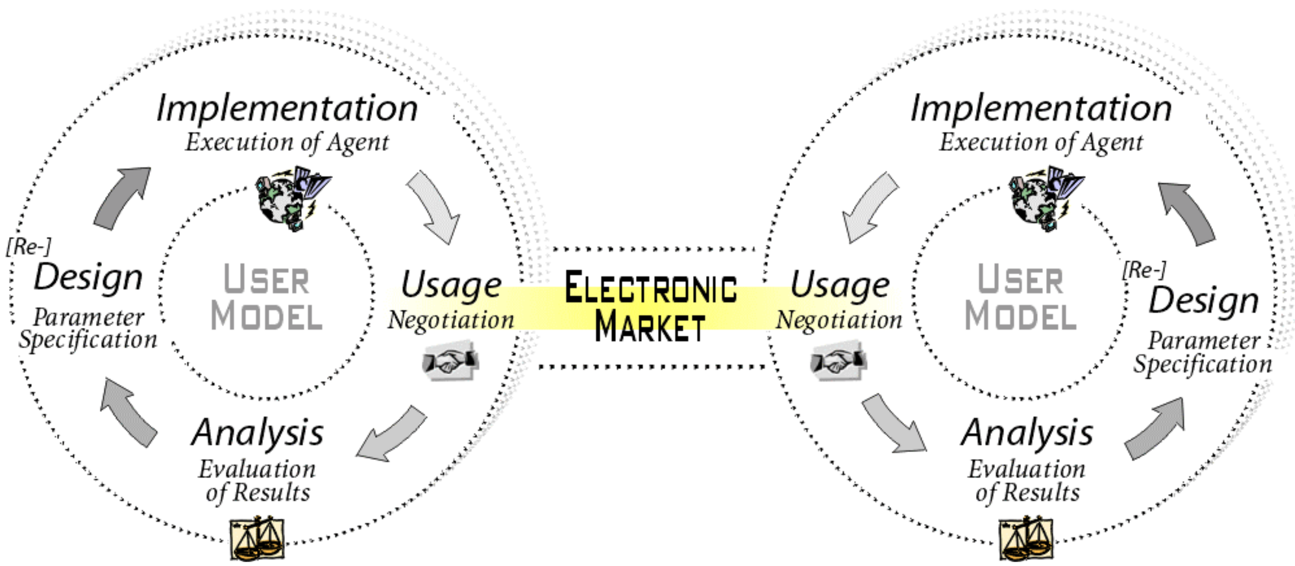

Figure 3. Evolutionary development of digital agents

The principals on either side constantly undergo the evolutionary development cycle, implicitly and/or explicitly, while participating in the electronic market. The degree of tool support and explicitness over the whole cycle is becoming a critical success factor for effective decision-making.

The distinction of design, implementation, usage, and analysis is based on experiences from Web information systems [6]. The design of an agent needs to address similar challenges as previous Internet systems. Visualization, user-centeredness and integration with implementation models remain critical success factors. The increased infrastructural and functional requirements of commercial agent systems further emphasize the importance of these design requirements. Fully integrated tools will implement fully functional agent systems based on the models derived from the design methodology. At this point in time speculation on prevalent future implementation *architectures for agent systems appear unjustifiable. Indications of the strengths, weaknesses, and opportunities have already been addressed in previous sections of this paper. In the usage phase agents are searching actively for negotiation partners in electronic market environments. It is the usage phase where the agents and their underlying development loops are tangent to each other, with their interaction facilitated by an electronic market. After the usage principals evaluate market transactions and review negotiation and information interactions. The outcomes and findings of this analysis phase are then fed back into the (visual) design of the updated agent.*

5. conclusion and outlook

Advanced Web-based systems will apply digital agent technologies, where buying and selling agents are independently discovering virtual spaces to find suitable business partners. Transaction agents are able to initiate and execute commercial transactions via the Internet. They have a predefined goal and keep track of their state information. Once they find a successful match – e.g., a product offered for sale by another agent – they either report back to their principal or directly negotiate without the intervention of the principal. In this way they help realize a fundamental transformation in the way people interact and transact goods. The support of customization and adaptability by these tools will enable (commercial) users to follow a strategy of evolutionary development of their agents through visualization and graphical modeling. Without broad support for standardization, however, agents will not be able to communicate with each other and to interpret relevant information in order to (re)act on behalf of the principal.

Even if the problem of standardization is solved, the support of semi-structured and unstructured tasks represents a fundamental challenge to the developers of agent-mediated architectures. Based on specific task characteristics, Gebauer and Scharl present a process model to evaluate Web-based infrastructures [Gebauer & Scharl 1999, Scharl 2000]. The model reveals a fundamental trade-off between long-term and short-term efficiency. It supports the assumption that there is an optimal degree of integration – i.e., a balance between complete automationand maximum flexibility with no structuring at all. In most cases, infrastructure investments are offset by drastically reduced costs of standard operations and – due to the increased flexibility – less need for the handling of exceptional operations. On the output side, the customer delivered valueis pushed to a higher level by increasing the quality of existing products and services, and by providing additional utility not available via traditional channels. Frequently, increasing customer delivered value is regarded more important than mere cost savings.

Web-based digital agents allow the development, definition, and management of customer relationships in ways that were not feasible before. Individuals and organizations will activate numerous independently acting agents with a predefined set of permissions. These agents will gather information, learn, and react in conformity with their embedded goals. They will meet other agents in order to exchange information, to coordinate their activities, to negotiate contractual terms, or to perform commercial transactions.

ENDNOTES / URLs

1. *http://www.ecn.curtin.edu.au/wis; http://scharl.wu-wien.ac.at/wis*
2. *http://www.isgmlug.org/isgmlug.htm*
3. *http://agents.www.media.mit.edu/groups/agents/projects/applications.html*
4. *http://bf.cstar.ac.com/*
5. *http://www.alldirect.com/*
6. *http://www.w3.org/rdf*
7. *http://www.firefly.com/*
8. *http://www.netperception.com/products/technology.html*
9. *http://www.grouplens.org/*
10. *http://www.amazon.com/*
11. *http://www.wisewire.com/*
12. *http://www.bowneinternet.com/*
13. *http://maker.media.mit.edu/*
14. *http://www.cs.umbc.edu/kqml/*
15. *http://www.fipa.org/*
16. *http://www.cse.psu.edu/~dale/lProlog/*
17. *http://www.ontology.org/*
18. *http://www.cupcakes.com/*
19. *http://www.trl.ibm.co.jp/aglets/index.html*
20. *http://www.genmagic.com/technology/odyssey.html*
21. *http://www.meitca.com/HSL/Projects/Concordia/*

nmn

REFERENCES

Akoulchina, I. and J.-G. Ganascia. SATELIT-Agent: An Adaptive Interface Based on Learning Interface Agents Technology. in 6th International Conference on User Modeling (UM-97). 1997. Chia Laguna, Italy: Springer.

.Balabanovic, M. and S. Yoav, Fab: Content-Based, Collaborative Recommendation. Communications of the ACM, 1997. **40**(3): p. 66-72.

Bauer, C. Using Reference Models to Develop WWW-Based Applications. in 9th Annual Australasian Conference on Information Systems (ACIS-98). 1998. Sydney, Australia: University of New South Wales.

.Bauer, C., B. Glasson, and A. Scharl, Evolution of Web Information Systems: Exploring the Methodological Shift in the Context of Dynamic Business Ecosystems, in Doing Business on the Internet: Opportunities and Pitfalls, C.T. Romm and F. Sudweeks, Editors. 1999, Springer: London. p. 53-64.

.Bauer, C. and A. Scharl. Acquisition and Symbolic Visualization of Aggregated Customer Information for Analyzing Web Information Systems. in 32nd Hawaii International Conference on System Sciences (HICSS-32). 1999. Maui, USA: IEEE Computer Society Press.

Bauer, C. and A. Scharl. Advanced Design of Web Information Systems Based on Dominant and Emerging Web Communication Patterns. in 5th Australian World Wide Web Conference (AusWeb-99). 1999. Ballina, Australia: Southern Cross University Press.

Beam, C. and A. Segev, Automated Negotiations: A Survey of the State of the Art. Wirtschaftsinformatik, 1997. **39**(3): p. 263-268.

Bichler, M., M. Kaukal, and A. Segev. Multi-attribute Auctions for Electronic Procurement. in IAC First Workshop on Internet Based Negotiation Technologies. 1999. Yorktown Heights, USA.

Bichler, M. and S. Nusser. Modular Design of Complex Web-Applications with W3DT. in 5th Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE-96). 1996. Stanford, USA: IEEE Computer Society Press.

Bichler, M. and A. Segev. Brokerage in E-Commerce: State-of-the-Art and Open Issues. in Workshop on Information Technology and Systems (WITS-98). 1998. Helsinki, Finland.

Boland, R.J., Information Systems Use as a Hermeneutic Process, in Information Systems Research: Contemporary Approaches and Emergent Traditions, H.E. Nissen, H.K. Klein, and R.A. Hirschheim, Editors. 1991, North-Holland: Amsterdam. p. 439-464.

Bradshaw, J.M., et al., Agents for the Masses? IEEE Intelligent Systems, 1999. **14**(2): p. 53-63.

Brenner, W., R. Zarnekow, and H. Wittig, Intelligente Softwareagenten: Grundlagen und Anwendungen. 1998, Berlin: Springer.

Chandrasekaran, B., J.R. Josephson, and R. Benjamins, What Are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems, 1999. **14**(1): p. 20-26.

Chavez, A. and P. Maes. Kasbah: An Agent Marketplace for Buying and Selling Goods. in 1st International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM-96). 1996. London, UK: Practical Application Company.

Falchuk, B. and A. Karmouch, Visual Modeling for Agent-Based Applications. Computer, 1998. **31**(12): p. 31-38.

Fikes, R. and A. Farquhar, Distributed Repositories of Highly Expressive Reusable Ontologies. IEEE Intelligent Systems, 1999. **14**(2): p. 73-79.

Finnie, G. and G. Wittig. Electronic Commerce via Personalised Virtual Electronic Catalogues. in 2nd Annual Collaborative Electronic Commerce Technology and Research Conference (CollECTeR-98). 1998. Sydney, Australia: University of Wollongong.

Gebauer, J. and A. Scharl, Between Flexibility and Automation: An Evaluation of Web Technology from a Business Process Perspective. Journal of Computer-mediated Communication, 1999. **5**(1): p. http://www.ascusc.org/jcmc/.

Ginsburg, M. Annotate! A Tool for Collaborative Information Retrieval. in 7th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. 1998. Stanford, USA: IEEE Computer Society Press.

Ginsburg, M. and A. Kambil. Annotate: A Web-based Knowledge Management Support System for Document Collections. in 32nd Hawaii International Conference on System Sciences (HICSS-32). 1999. Hawaii, USA: IEEE Computer Society Press.

Guttman, R. and P. Maes. Cooperative versus Competitive Multi-Agent Negotiations in Retail Electronic Commerce. in 2nd International Workshop on Cooperative Information Agents (CIA-98). 1998. Paris, France.

Guttman, R., A. Moukas, and P. Maes, Agent-mediated Electronic Commerce: A Survey. Knowledge Engineering Review, 1998. **13**(2): p. 147-160.

Hansen, H.R. and M.F. Tesar. Die Integration von Masseninformationssystemen in die betriebliche Informationsverarbeitung. in Fachtagung "Data Warehouse" an der Gerhard-Mercator-Universität GH Duisburg. 1996. Duisburg, Germany.

Isakowitz, T., E.A. Stohr, and P. Balasubramanian, RMM: A Methodology for Structured Hypermedia Design. Communications of the ACM, 1995. **38**(8): p. 34-44.

Issarny, V. and T. Saridakis, Defining Open Software Architectures for Customized Remote Execution of Web Agents. Autonomous Agents and Multi-Agent Systems, 1999. **2**(3): p. 237-249.

Jamali, N., P. Thati, and G.A. Agha, An Actor-based Architecture for Customizing and Controlling Agent Ensembles. IEEE Intelligent Systems, 1999. **14**(2): p. 38-44.

Kaplan, B. and J.A. Maxwell, Qualitative Research Methods for Evaluating Computer Information Systems, in Evaluating Health Care Information Systems: Methods and Applications, J.G. Anderson, C.E. Aydin, and S.J. Jay, Editors. 1994, Sage: Thousand Oaks. p. 45-68.

.Knapik, M. and J.B. Johnson, Developing Intelligent Agents for Distributed Systems: Exploring Architecture, Technologies, and Applications. 1997, New York: McGraw-Hill.

Koblick, R., Concordia. Communications of the ACM, 1999. **42**(3): p. 96-97.

Labrou, Y., T. Finin, and Y. Peng, Agent Communication Languages: The Current Landscape. IEEE Intelligent Systems, 1999. **14**(2): p. 45-52.

Lange, D.B. and M. Oshima, Seven Good Reasons for Mobile Agents. Communications of the ACM, 1999. **42**(3): p. 88-89.

Lee, K. and G. Speyer, Platform for Privacy Preferences Project (P3P) & Citibank, . 1999, Citibank Advanced Development Group.

Lindemann, M.A. and A. Runge. Electronic Contracting within the Reference Model for Electronic Markets. in 6th European Conference on Information Systems (ECIS-98). 1998. Aix-en-Provence, France: Euro-Arab Management School.

Malchow, R. and K. Thomsen, Web-Tracking. Screen Multimedia, Sep 1997, 1997: p. 57-61.

Milosavljevic, M. Electronic Commerce via Personalised Virtual Electronic Catalogues. in 2nd Annual Collaborative Electronic Commerce Technology and Research Conference (CollECTeR-98). 1998. Sydney, Australia: University of Wollongong.

Moore, S.A. KQML & FLBC: Contrasting Agent Communication Languages. in 32nd Hawaii International Conference on System Sciences (HICSS-32). 1999. Hawaii, USA: IEEE Computer Society Press.

Nanard, J. and M. Nanard, Hypertext Design Environments and the Hypertext Design Process. Communications of the ACM, 1995. **38**(8): p. 49-56.

Reagle, J. and L.F. Cranor, The Platform for Privacy Preferences. Communications of the ACM, 1999. **42**(2): p. 48-55.

Scharl, A., Referenzmodellierung kommerzieller Masseninformationssysteme: Idealtypische Gestaltung von Informationsangeboten im World Wide Web am Beispiel der Branche Informationstechnik. 1997, Frankfurt: Peter Lang.

Scharl, A. Reference Modeling of Commercial Web Information Systems Using the Extended World Wide Web Design Technique (eW3DT). in 31st Hawaii International Conference on System Sciences (HICSS-31). 1998. Hawaii, USA: IEEE Computer Society Press.

Scharl, A., Evolutionary Web Development: Automated Analysis, Adaptive Design, and Interactive Visualization of Commercial Web Information Systems. 2000, London: Springer.

Scharl, A. and C. Bauer. Informational Requirements for Participating in Electronic Business Ecosystems. in Western Australian Workshop on Information Systems Research (WAWISR-98). 1998. Perth, Australia: Curtin University of Technology.

Schmid, B.F. and M.A. Lindemann. Elements of a Reference Model for Electronic Markets. in 31st Hawaii International Conference on System Sciences (HICSS-31). 1998. Hawaii, USA: IEEE Computer Society Press.

.Schubert, C., R. Zarnekow, and W. Brenner. A Methodology for Classifying Intelligent Software Agents. in 6th European Conference on Information Systems. 1998. Aix-en-Provence, France: Euro-Arab Management School.

Singh, M.P., Agent Communication Languages: Rethinking the Principles. Computer, 1999. **31**(12): p. 40-47.

Sommer, U. and P. Zoller. WebCon: Design and Modeling of Database Driven Hypertext Applications. in 32nd Hawaii International Conference on System Sciences (HICSS-32). 1999. Hawaii, USA: IEEE Computer Society Press.

Tai, H. and K. Kosaka, The Aglets Project. Communications of the ACM, 1999. **42**(3): p. 100-101.

Teich, J., H. Wallenius, and J. Wallenius, Multiple-issue Auction and Market Algorithms for the World Wide Web. Decision Support Systems, 1999. **26**(1): p. 49-66.

Vasudevan, V. and M. Palmer. On Web Annotations: Promises and Pitfalls of Current Web Infrastructure. in 32nd Hawaii International Conference on System Sciences (HICSS-32). 1999. Hawaii, USA: IEEE Computer Society Press.

Walsham, G., Interpreting Information Systems in Organizations. 1993, Chichester: Wiley.

Wong, D., N. Paciorek, and D. Moore, Java-based Mobile Agents. Communications of the ACM, 1999. **42**(3): p. 92-102.

Yule, G., Pragmatics. Oxford Introductions to Language Study, ed. H.G. Widdowson. 1996, Oxford: Oxford University Press.

Zacharia, G., A. Moukas, and P. Maes. Collaborative Reputation Mechanisms in Electroinc Marketplaces. in 32nd Hawaii International Conference on System Sciences (HICSS-32). 1999. Maui, USA: IEEE Computer Society Press.