KNOWLEDGEABLE AGENTS FOR SEARCH AND CHOICE SUPPORT IN E-COMMERCE: A DECISION SUPPORT SYSTEMS APPROACH

Susan Sproule
McMaster University
sproulsm@mcmaster.ca

Norm Archer
McMaster University
archer@mcmaster.ca

ABSTRACT

Software agents are a major innovation in how people use information systems, and they have parallels with how Decision Support Systems (DSS) support human decision-making. A DSS approach to the development of software agents suggests a highly interactive and flexible interface between the agent and its user, and addresses some potential barriers to the successful adoption of agent technologies. Within a DSS model, agents can be classified as providing search, choice or interface support. Each of these classifications uses techniques originating from separate disciplines and requires different performance measures. We use a real estate agent as a metaphor to examine the descriptive, procedural and semantic knowledge bases that agents can use to support search and choice activities in an e-commerce domain.

1. Introduction

Software agents are computer programs that run in the background and perform tasks autonomously, as delegated by the user. Although there has been much research on this topic, usable software agents are at an early stage of development, and are only now starting to appear in real applications. A fruitful application area for software agents is in the area of electronic commerce where agents can help buyers and sellers deal with the flood of information that can be exchanged and processed. Related research has examined how agents can support purchasing activities using traditional buyer behaviour models [Maes et al. 1999; Nissen 1999] and identified basic knowledge requirements for agents supporting integrated electronic commerce systems [Wang 1999].

In this paper, we look at commerce as a series of decision-making processes where problems are identified, alternative solutions are considered and choices are made. A Decision Support Systems (DSS) approach to software agent development provides insight into how interactive systems can provide flexible and adaptive ways of approaching the complex decision-making processes involved in electronic commerce. The DSS approach also suggests a functional classification system for agents based on their reference disciplines and provides effective ways to evaluate agent performance.

The knowledge that human agents use to provide their services is a useful metaphor for the knowledge bases that software agents may employ. Real estate agents are used as an example to illustrate the types of knowledge that may be useful within a specific e-commerce domain. We describe how a number of different knowledge representation techniques may be required for different parts of the commerce decision-making process, and how the choice of technique may depend on the nature of the information and the level of interactivity that is designed into the system.

By examining the knowledge that agents will require in e-commerce we are able to identify a number of design challenges. Current research and development activities such as Extensible Markup Language (XML) and Knowledge Query and Manipulation Language (KQML) promise to enhance the knowledge-acquisition and knowledge-sharing abilities of future agent-based systems.

2. Theoretical Foundations

2.1 Approaches to Software Agents

In reviewing research and development activity in software agents it is helpful to acknowledge some related areas of research and their different approaches to this new field of information technology. Software agents were originally conceived and developed within the Artificial Intelligence (AI) research community. Reasoning and learning capabilities developed within AI provide the autonomous and adaptive behaviour that we want for agent applications. However, traditional AI systems are designed as "black box" systems that focus on results. [1]

Within the field of Management Information Systems, agents are most closely related to the study of Decision Support Systems (DSS). The study of DSS examines how information systems can be used to help decision-makers make better decisions. Decision-making involves activities such as collecting relevant information from the environment, modeling the problem domain and generating alternative solutions, employing a decision strategy to choose between alternatives, testing and justifying the decision, and effecting the necessary changes in the environment to implement the decision. DSS have been developed to support human users across all of these activities [Turban et al. 1998].

Expert systems (ES) have been the most successful practical application of AI technologies and can be viewed as a hybrid of AI and DSS. These systems apply rule-based reasoning, developed in AI, to assist human decision-makers in solving real life problems. According to Wooldridge [1999], the main distinction between ES and software agents is that ES do not generally receive information from, or act directly on, their environment. The human user of an ES acts as a middleman in these information exchanges. DSS researchers have studied ES as they evolved from research projects into successful practical applications and have identified design characteristics that facilitate the way that users interact with these systems. We might expect to see agent technologies follow a similar path to adoption, and examining how DSS principles have been applied to ES may provide clues for the successful implementation of intelligent agents in decision support.

2.2 A Decision Support Systems Approach

Decision-making is a complex, multi-staged process. DSS research recognizes that computers can complete certain parts of this process faster and more accurately than people can. People, however, bring abilities such as creativity, intuition, and experiences that enable them to complete other parts of the process more effectively than machines. The DSS approach is to structure parts of an ill-structured problem. These structured parts can then be performed by the system. Humans interact with the system, using their own knowledge to "join" the structured parts together and develop a complete solution to the problem.

By segmenting the overall problem, components can be defined to require very specific domain knowledge and reasoning capabilities, making them well-suited to the limitations of current AI technologies. Current research in DSS is incorporating AI to add structure to larger and more complex areas of the decision-making process. AI techniques can be used to build systems that learn from experience, deal with ambiguity and uncertainty, apply logical reasoning and inference, and adapt to new situations [Siskos et al. 1999].

The DSS approach demands a lot of interaction between the decision-maker and the system. Some experienced agent developers propose an approach to the design of software agents which is remarkably similar to DSS design, describing a level of interactivity very different from the "black box" model that is found in an AI approach. From their experience developing the *Information Lens* agent system, Malone, Grant and Lai [1997] propose two principles for agent design that fit well within the DSS paradigm:

> "Don't build agents that try to solve complex problems all by themselves... Build systems where the boundary between what the agents do and what the humans do is a flexible one. We call this the principle of *semiformal systems*…"
>
> "Don't build agents that try to figure out for themselves things that humans could easily tell them. Instead try to build systems that make it as easy as possible for humans to see and modify the information and reasoning processes their agents are using. We call this the principle of *radical tailorability*…" [Malone et al. 1997, pg. 110]

The development process is another area where DSS research may be useful in developing agent systems. DSS are often built to support individual decision-makers, with one-time or ad hoc problems, and DSS developers recognize that their human users learn during the development process and while using the system. A fast and highly interactive development process is necessary and DSS tools allow changes to be made quickly and flexibly during the process [Turban et al. 1998]. Similar problems arise in the design of software agents. A defining characteristic

---

[1] The classic "Turing test", where an AI system is expected to produce behaviour that is indistinguishable from that of a human being, is evidence of this focus [Turing, A. M. "Computing Machinery and Intelligence." *MIND - A quarterly review of Psychology and Philosophy* LIX(236)October, 19501950. http://www.abelard.org/turpap/turpap.htm

of software agents is the ability of an agent to be "personalized" for each user. Agents must be able to satisfy the needs of users with different levels of experience, different perceptions of risk, and different decision-making preferences. This will require tools comparable to those used in DSS development, where users can experiment with "prototype" agents and change their agent's characteristics as they gain experience and trust in the agent's abilities.

Finally, DSS research pays a lot of attention to the system's usefulness, as defined by the user. While other organizational information systems, such as transaction processing systems and management information systems are usually "mandated" into use, the use of a DSS is generally considered to be optional [Turban et al. 1998]. Similarly, we assume that people will choose to use an agent, and will do so only if its usefulness is clearly evident. Table 1 summarizes the contributions of a DSS approach in agent design and development.

Table 1 – Contributions from a DSS Approach

The DSS approach promotes…	In the development and design of software agents, this accommodates…
the segmentation of a large ill-structured decision problem into smaller components	… the limited problem domains that AI applications can adequately address … the need for different representations and reasoning systems in separate parts of the problem
flexible boundaries between the user and the system allowing for many levels of interaction	… the development of trust … user learning … dynamic situational factors … constructive search and choice behaviour
an interactive development process with tools that allow the user to adapt and customize the system	… the need for agents to be personalized for each user
"usefulness" as a critical characteristic of the system	… the need to consider the voluntary nature of agent use.

In related work, Bui and Lee [1999] take a DSS approach to developing a system of collaborative agents to assist in crisis management. Their development process involves deconstructing the overall problem-solving process into primitive tasks, specifying the required functionality and behaviour of agents for these tasks, and deciding if use of an agent is justified. Coordination and collaboration mechanisms are then designed so that humans and software agents can integrate their activities into an overall workflow.

Cuena and Ossowski [1999] provide a framework for the design of distributed decision support for control systems using multi-agent systems. They argue that knowledge modeling is often difficult when systems are designed using functional decomposition and object-modeling methods. Agent-based models provide a higher level of modularity that can combine knowledge about the problem type and the environment. They believe that this is a more intuitive approach for both modeling and organizing knowledge. It lets the DSS designer balance the "level of specialty" and the "level of autonomy" by integrating a significant set of functions, but restricting the scope of the environment in which they are applied.

2.3 Knowledge–based Systems

The roles of human agents can serve as useful metaphors to derive models of what software agents may do [Jennings et al. 1998]. Some of a human agent's knowledge replicates the client's knowledge. In this case the agent is valued for being able to reduce the time that the client must spend in the process. Software agents that allow the user to build and add to the knowledge base or where the agent learns from the user's actions, would be examples of systems that attempt to replicate this type of support. Human agents also possess knowledge that the client may not have, and in this case they are valued for their expertise. Corresponding software agents are those based on the traditional class of rule-based expert systems (ES) and collaborative agents that combine the knowledge of a number of different users to arrive at decisions or make recommendations.

2.3.1 Knowledge Representation

AI research has explored a number of different theories of intelligent reasoning. Davis et al [1993] classify five of these theories according to the disciplines from which they originate as follows: pure logic-based systems (mathematics), probabilistic reasoning systems (statistics), frames (psychology), connectionist systems such as neural networks and genetic algorithms (biology), and utility theory and rational agents (economics). A fundamental concept in any knowledge-based system is knowledge representation - using symbols to build a model of the portion of the real world that is of interest. Knowledge representation techniques include predicate logic, frames, production rules and semantic nets. The choice of representation will determine the type of reasoning that the system employs, how the knowledge base is processed, and the responses that the system allows [Davis et al. 1993]. It is important to choose a representation technique that meets the needs of the problem situation. In many e-commerce applications a combination of representation techniques, each for different parts of the overall problem, may be required.

If the user must interact with the system it is also important to use representations that compliment the way that the user conceptualizes the problem. The user's conceptual model provides the "predictive and explanatory power for understanding the interaction" [Norman 1983]. Designers must start with a conceptual model that will be easily understood by users. They must then ensure that the system's appearance, responses and documentation lead users to develop an appropriate conceptual model of the system as they interact with it [Norman 1990]. The degree of users' involvement and interaction with the system should therefore be an important consideration in the choice of representation. In ES, "explanation" capabilities have been shown to improve performance, learning and user's perceptions and should be considered an important component of any interactive, intelligent system design [Gregor et al. 1999]. If "explanation" is a design requirement for a part of the process that we want an agent to handle, representations based on connectionist systems like neural networks should not be used because they provide "black box" solutions and no explanations.

2.3.2 Types of Knowledge

Using the human agent metaphor, we can see that human agents in the commerce domain know "facts" about products, buyers, sellers and the market. We call this type of knowledge *descriptive* knowledge. Human agents also know what to do with this information – how to process it to arrive at and implement decisions. We will call this *procedural* knowledge. Finally, human agents know what facts are important, both in general and to their individual clients, and how various facts relate to each other. This allows them to evaluate and assimilate new information and communicate by exchanging knowledge in a meaningful way to others. We will call this *semantic* knowledge.

In the context of building knowledge-based decision support systems, Holsapple and Whinston [1996] define three primary types of knowledge (descriptive, procedural, and reasoning) and three secondary types of knowledge (assimilative, linguistic and presentation). Our definition of descriptive knowledge is consistent with their classification. We use the category of procedural knowledge to discuss knowledge that can be represented in the processing code of systems, including reasoning capabilities, or procedural knowledge that can be stored and retrieved from a knowledge base. Our classification of semantic knowledge combines the three secondary types of knowledge (assimilative, linguistic and presentation), as they can all be considered meta-knowledge, or knowledge about knowledge.

3. Proposed Development Framework

3.1 Agent Classifications from a DSS Approach

DSS are commonly considered to include a data subsystem, a model subsystem and a dialogue subsystem. Turban [1988] suggests that AI can be embedded into DSS to support the model, data or dialogue subsystems, the complete system, or the user. We propose a classification of agents according to whether they support search functions through the data subsystem, choice functions through the model subsystem, or interface functions through the dialogue subsystem. This is shown in Figure 1.

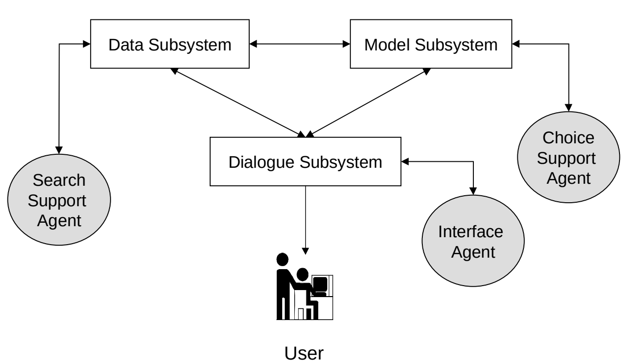

Figure 1 - Agents in a Decision Support System
[adapted from Turban, 1988]

We can distinguish between search, choice and interface agents according to the disciplines from which they borrow their techniques and how their performance should be measured. Table 2 summarizes these disciplines and measures. Agents that support the search function use techniques and measures developed within the information retrieval (IR) community. Agents that support the choice function borrow their techniques from economics, psychology, management science and other disciplines that describe how people make choices between alternatives and how to improve decision quality. The different theories proposed by these disciplines result in a variety of evaluation criteria. Interface support is based on principles developed in the study of human-computer interaction, where various measures can be used to evaluate a system's "usability".

Table 2 – Agent Reference Disciplines and Measures

Type of Agent	Reference Discipline(s)	Potential Performance Measures
SEARCH SUPPORT	Information Retrieval	Precision Recall
CHOICE SUPPORT	Decision theory from: Economics Psychology Management Science	Consistency of decisions Compare choice to optimal Amount of information used or processed Time to make decision
INTERFACE SUPPORT	Human Computer Interaction	Usability measures such as: User satisfaction Errors Learning time

The boundary between the dialogue subsystem and the other subsystems is often not clearly defined. For example, natural language processing (NLP) is an important area of development for improving user-system dialogue, however NLP techniques also have important applications in information retrieval, which forms part of the data subsystem.

A very active area of current software agent research focuses on improving the dialogue subsystem. It will be important to find a more natural way for users to communicate with systems as they become more pervasive in our everyday activities. We also need to find appropriate ways for users to deal with software agents that incorporate high-level concepts such as goals, beliefs and intentions. To date, much of the activity in interface agents follows a black box, AI approach, trying to simulate human behaviour with anthropomorphic characteristics such as emotion and personality. While it is desirable to delegate certain activities to such agents, we believe that users will want to retain control of other parts of the process. The DSS approach suggests a more flexible and configurable interface model that, at times, allows the user to take over and interact directly with the data and model subsystems. The boundary between these areas may vary with the user, the task and the situation. How this boundary varies, and how to design systems that accommodate these variations, are important areas for future research. However, to remain focussed on a DSS approach the following discussion concentrates on search and choice support functions.

3.2 A Model of Search

Figure 2 shows a basic search model that contains an information source and its representation, an information need and its representation, and a method for comparing these representations.

Both the information source and the information need may change over time. Information retrieval deals with a "static" set of sources and a "dynamic" set of one-time needs (queries). Information filtering deals with "dynamic" sources and a "static" need (a profile) [Belkin et al. 1992].

Information sources can be unstructured (e.g. full text), semi-structured (e.g. integrated catalogues) or structured (e.g. databases), and the degree of structure will affect the kind of representation used. Full text sources may be represented by sets of index terms. Catalogue items may be represented by minimal information (e.g. a product name and a supplier) with a link to the full information source. The records or objects in a database represent structured information. Similarly, queries or profiles can be unstructured (e.g. a natural language request), semi-structured (e.g. a list of key words or phrases, possibly enhanced by logical operators), or structured (e.g. a SQL command to a database).

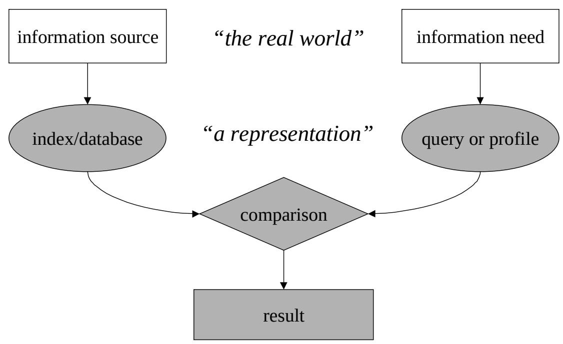

Figure 2 - Model of Search
[adapted from Belkin & Croft, 1992]

3.3 A Model of Choice

In the real world, the choice problem can be described as interrelated sets of alternatives, criteria and consequences that are processed and analyzed by the decision-maker [White 1975]. To model the problem (see Figure 3) each alternative can be represented by a set of variables. Parameters are set to represent the selected criteria and any assumptions about the problem situation. A decision model is used to process each alternative, returning a result that represents the consequences of that choice. Prescriptive models compare results and determine the best choice of alternatives. Descriptive models present the results associated with each alternative to the decision-maker.

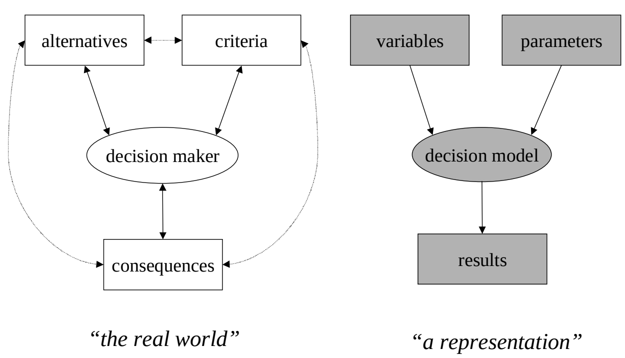

Figure 3 - Model of Choice
[adapted from White, 1975]

4. Knowledge Requirements for Search and Choice in Electronic Commerce

To show how this framework can be applied within the electronic commerce domain, we first look at the descriptive, procedural and semantic knowledge that a human agent may use to support search activity. We then provide examples of how these knowledge requirements have been built into software agent systems, identify some of the major design challenges, and describe technologies and research areas that show promise in meeting these challenges. Choice support is examined in the same way.

4.1 A Real Estate Agent Metaphor

We will use real estate agents to illustrate the various types of knowledge that a human agent may possess and how this knowledge is linked to the perceived value of their services in supporting search and choice activities. Real estate agents were chosen as our example because they may act for either the buyer or the seller. The real estate agent metaphor can be related to the previous discussion of knowledge-based systems in two ways.

Some of the services that a real estate agent performs are valued because they save their client's time. For other services, the client relies on the real estate agent's expertise. To provide these services, the real estate agent uses both knowledge that the client provides and expert knowledge.

A real estate agent is able to use different ways of reasoning and processing this knowledge. If we ask a real estate agent what effect a proposed price will have on our mortgage payments, or if the current zoning is consistent with proposed use of the building, we want a logically sound, correct answer. However, many decisions involve uncertainty and an answer that is "probably" true may be preferable to no answer at all. If we ask our real estate agent to identify the best neighborhoods or schools, we want an informed but necessarily subjective answer.

4.2. Real Estate Agents and Search Support

4.2.1 Descriptive Knowledge

Real estate agents know where to obtain information about properties, buyers, sellers and market conditions. They have access to directories and catalogues such as Multiple Listing Services (MLS) and gather additional information from first-hand observation and discussions with colleagues. Using these sources, the agent collects "facts" or descriptive knowledge about properties. For an experienced real estate agent, this knowledge covers both the current and past states of the market.

When a new buyer arrives, the real estate agent determines the client's needs. If an initial query that represents these needs fails to find a satisfactory property, it is stored as a "profile" that can be compared to any properties that are subsequently listed. The real estate agent will continually try to clarify a client's information needs, by probing or observing the client's reactions to the information presented.

4.2.2 Procedural Knowledge

We expect a real estate agent to develop an efficient strategy that will determine the information sources to be used and the order in which they are used. A seller's agent will construct a listing for the MLS and perhaps develop an information sheet with supplementary information. A buyer's real estate agent will construct appropriate queries to search the MLS catalogue or may discuss the client's needs with other agents. The agent must then compare their client's needs with the information obtained from these sources to find potential matches and produce a "reasonable" number of alternatives.

4.2.3 Semantic Knowledge

A real estate agent knows the relationships between objects and concepts, and can therefore determine the relevance and importance of facts. For example, knowing the age of a heating system, the real estate agent can estimate when the cost of replacement will occur and the impact this may have on the purchasing decision.

4.3 Software Agents and Search Support

Table 3 summarizes the knowledge requirements, design challenges, applicable technologies and research areas for software agents that provide search support.

4.3.1. Descriptive Knowledge

Software agent system designers must address the "connection problem" – how does the agent find information sources and other agents to assist in achieving its goals. In controlled systems, collaborative agents can request and obtain descriptive knowledge that has been collected by other agents [Ackerman et al. 1997]. In open environments, most agent systems use directories, matchmakers, and brokers to identify potential information sources [Brenner et al. 1998].

Information retrieval often consists of finding structure in predominately free text documents, such as those that make up the Web. Structure can be inferred from features such as hyperlinks [Arocena et al. 1999], header tagging [Guan et al. 1999], or question-answer formats [Burke et al. 1997]. An important area of development involves Extensible Mark-up Language (XML). XML allows creators to encode additional structure into their Web-based information sources, producing more "searchable" documents by allowing more complex and complete representations to be built [Glushko et al. 1999].

Software agents can ask clients to state their information needs.[2] A form or questionnaire can be used to elicit the representation requirements where information is highly structured. However, where information needs are complex and ill-structured, more open processes of collection may be required and these processes can be time-consuming and inaccurate. Significant efforts have been made to design software agents that use proxy information to develop profiles [Rucker et al. 1997] or learn their user's preferences by observing behaviour [Lieberman ; Ngu et al. 1997]. Collaborative filtering compares profiles to find users with similar information needs so that information judged relevant by one user can be shared with others. This is another way to reduce the profiling effort required by each user [Balabanovic et al. 1997].

[2] While the client's information needs may indicate potential decision criteria, this is not necessary at the search stage.

Table 3 – Knowledgeable Agents for Search Support

Type of Knowledge	Knowledge Requirements	Design Challenges	Applicable Technologies and Research Areas
DESCRIPTIVE	Location of information sources	Distributed sources and the “connection problem”	Multi-agent architectures with directories, matchmakers, and brokers
	Data extracted from information sources	Heterogeneous sources with varying levels of structure	XML coding within information sources
	Data describing information needs	Reduce profiling effort	Use of proxy information, learning systems and collaborative systems
PROCEDURAL	Search strategies	Distributed, dynamic environment	Adaptive search strategies that optimize time, cost or quality of search.
	Creating representations	Heterogeneous sources and users	Information extraction and query formation using linguistic analysis and natural language processing
	Matching algorithms	Balance between precision and recall	Probabilistic techniques for information retrieval
SEMANTIC	Ontology	Standardization	Base and domain ontology development
	Communications protocols	Open systems, heterogeneous agents	KQML

4.3.2. Procedural Knowledge

Most software agents have pre-defined search strategies. Some attempts to design adaptive strategies have examined query optimization [Duschka et al. 1997], the efficient use of network resources [Howe et al. 1997], or balancing source cost against quality [Lesser et al. 2000].

Software agents are able to create representations and translate between source and need representations. Information extraction techniques such as automated indexing systems are used to create feature-based representations of Web documents. Within a specified domain, systems that use more sophisticated linguistic analysis can create structured databases out of information extracted from full text sources [Cardie 1997]. Some meta-search agents are able to translate phrase-based requests into either keyword or phrase-based queries acceptable to popular Web search engines [Etzioni 1997]. “Virtual service representatives” can extract key words and patterns from natural language queries [Neuromedia].

The agent must be able to compare the source and need representations to find potential matches and produce a “reasonable” number of alternatives. Simple agents may use traditional Boolean systems of information retrieval to match queries to documents, but many agents use more advanced probabilistic systems that weight index terms or look at the statistical distribution of terms within a document. These systems also allow document to document comparisons, creating clusters of sources or user profiles that can be used in retrieval and filtering operations [Pao 1989; Belkin et al. 1992].

4.3.3 Semantic Knowledge

Computers can store vast amounts of descriptive knowledge, and process this knowledge at speeds greatly beyond human capabilities. However, it is semantic knowledge that will produce what we consider to be intelligent and adaptive systems. By using semantic knowledge, unexpected information can be assessed and the agent can broaden or narrow the search if the expected information is missing or the amount of information retrieved is overwhelming.

An ontology is a formal description of the relationships between objects and concepts within a domain. These formal descriptions provide a common vocabulary, allowing agents to exchange information in a meaningful and unambiguous way [Gruber 1993]. Frames and semantic nets are knowledge representation techniques that have been specifically developed to model such relationships.

The objects and concepts in a commercial transaction or relationship can be described at many levels. A base ontology covers terms common to all transactions such as those for finance, measurement, and standard contractual conditions. Domain ontologies describe objects and concepts within a product category. Individual suppliers or intermediaries can create a translation ontology that relates proprietary terms to the domain ontology [Keller et al. 1996]. Spurred by the potential of XML-based e-commerce, many inter-industry and intra-industry groups are actively developing base and domain ontology [Glushko et al. 1999; Smith et al. 1999].

The e-commerce environment is envisioned as an open, decentralized environment where agents must be able to communicate with other heterogeneous agents and systems. The Knowledge Sharing Effort (KSE), a project of the University of Maryland (Baltimore) has developed the Knowledge Query and Manipulation Language (KQML) to facilitate this type of communication [KSE (Knowledge Sharing Effort)]. KQML provides communications protocols and has been adopted for use in many multi-agent systems, including matchmaking and brokering systems for information retrieval and filtering. KQML specifies the "intent" of the message, based on speech act theory. The message content can be written in any knowledge representation language that is understood by the recipient [Finin et al. 1994].

4.4 Real Estate Agents and Choice Support

4.4.1 Descriptive Knowledge

Through the search process, a real estate agent has gathered descriptive knowledge of the alternatives - a set of attributes that describe each property. There may be information about buyers or sellers that will influence the decision process and the agent may use knowledge about market conditions to help define the problem space. The real estate agent has also collected and refined information about the client's decision criteria including the relative importance of the various attributes, acceptable trade-offs, and threshold levels on specific attributes.[3] To assist the client, a real estate agent is able to select information that is relevant, transform it into the form required, and provide reasonable assumptions about missing information.

4.4.2 Procedural Knowledge

A real estate agent is expected to facilitate and assist in decision-making, suggesting different ways of processing information about the alternatives. Experienced agents are likely to have a number of different decision-making techniques that they can match to the situation and the client's individual preferences. A real estate agent often handles transactions where there is more than one decision-maker (such as a husband and wife or a logistics department within a large corporation). An understanding of the information flows and decision-making processes employed within these groups can be used to ensure that the appropriate information is conveyed to each party at each stage in the process. A real estate agent also knows how and when to negotiate.

4.4.3 Semantic Knowledge

A real estate agent is expected to know the "rules" of negotiation, and how to communicate with other parties during the negotiation process in a series of offers and counter-offers. Finally, a real estate agent is expected to be able to communicate the results of a decision in a manner that ensures that the transaction is completed.

4.5 Software Agents and Choice Support

Table 4 summarizes the knowledge requirements, design challenges, applicable technologies and research areas for software agents that provide choice support.

4.5.1 Descriptive Knowledge

Software agents have access to descriptive information about the alternatives collected through the search process. "Restructuring" refers to functions that edit, transform, and infer information so that the chosen decision model can be populated with alternatives [Coupey 1994]. To restructure information, software agents must rely on an ontology to standardize attribute values, eliminate redundant information, and infer missing information. Restructuring can also be seen as a constructive process. Transforming attribute data into standardized values,

---

[3] In the search process the client's profile represented the information needed to identify a set of alternatives. While a search query or profile may indicate something about the way that a choice will be made, it may be important that other parties not be able to determine the choice criteria from the information request as this could jeopardize future negotiating strategies.

eliminating redundant or irrelevant information, and rearranging information, may reveal patterns and regularities that suggest the use of a particular choice model [Coupey 1994]. The constructive nature of restructuring is another indication that an interactive process may be preferred by the decision-maker. By restructuring and presenting information in different ways, the system can help decision-makers to choose models they are comfortable applying in particular situations. An agent should be able to handle market requests in both surplus and shortage situations. Widemeyer and Lee [1986] describe the ontological requirements for an AI system that can broaden the search to include substitute products in a shortage situation. The system can also apply increasingly stringent criteria to represent the need in a surplus situation.

Table 4 – Knowledgeable Agents for Choice Support

Type of Knowledge	Knowledge Requirements	Design Challenges	Applicable Technologies and Research Areas
DESCRIPTIVE	Attributes to describe alternatives	Restructuring	Base and domain ontology
	Decision criteria (weights, thresholds, trade-offs, etc.)	Constructive Choice	Learning and interactive systems
		Situational factors	Case-based reasoning and learning
PROCEDURAL	Decision models and algorithms	Individual preferences and use of more than one model	Multi-model systems and model management
		Sequential decisions	Dynamic decision-making models
	Process and workflow knowledge	Adaptive processes	Learning and reasoning systems
	Negotiating strategies	Non-cooperative environments and multi-dimensional solution spaces	Learning systems using probabilistic networks or genetic algorithms
SEMANTIC	Negotiation protocols	Mechanisms that encourage appropriate agent behaviour	Research from micro-economics and game theory
	Transaction protocols	Standardization	Adapting EDI-type messages for agent systems using KQML

Software agents can ask the decision-maker to weight the importance of attributes or to set threshold levels for various attributes. Some theories of consumer choice argue that buyers often do not know these preferences in advance [Bettman et al. 1998]. These theories again support the need for a highly interactive system, where users can see results and vary their criteria in an iterative process.

The complexity of some commercial transactions and relationships arises from the many outside factors that may or may not warrant consideration. Resource limitations, potential risk and reward, goals, time-pressure, and many other factors can change from one transaction to another. In this context, case-based systems that collect a number of features describing a situation or "case" may be the most effective way to represent the parameters involved in complex purchasing or selling situations.

4.5.2 Procedural Knowledge

Theories of consumer choice have developed out of research in economics and psychology. Economic theories of choice assume a perfectly rational decision-maker, able to state clear preferences at the beginning of the choice process. These preferences are used to develop a utility function that can be optimized to form the decision model. Psychological theories of choice have developed out of the belief that humans have limited information-processing capabilities and often use heuristics to reduce the amount of information processing required in decision-making. Heuristic models of decision-making use a series of constraints to eliminate alternatives until a decision can be made with minimal effort [Meyer et al. 1991; Bettman et al. 1998].

We can find examples of agents from both of these paradigms. *Personalogic* uses a heuristic approach [Maes et al. 1999], asking the user to specify both hard and soft constraints on the attributes describing alternative brands of a product. It eliminates brands that do not meet the specified hard constraints and presents the remaining alternatives ranked in order of how they compare on the soft constraints. *Tete-a-Tete* is based on a rational model of decision-making [Maes et al. 1999], using weighted-averages and a utility function to recommend a product choice. Consumers often use a combination of decision models [Bettman et al. 1998]. In an interactive system, if the agent is to follow a process that is familiar and recognizable to the user, information may need to be passed between coordinating agents with different modeling capabilities.

The models of buyer behaviour described above are static models that assume that a buyer's choice is independent of previous purchases. Market researchers have also developed models that represent the dynamic nature of consumer decision-making, incorporating factors such as learning, loyalty, novelty seeking, or inertia [Meyer et al. 1991]. Today's technologies make it possible to collect large amounts of time-series data for individual consumers. An agent that is able to predict behaviour from historical purchase information could make timely suggestions based on the loyalty, inertia or variety-seeking tendencies in that consumer's behaviour.

In a business-to-business environment, agents can use procedural knowledge to integrate activities within buying or selling organizations. While not an e-commerce application, Bui and Lee's [1999] crisis management system shows how procedural knowledge can be used to coordinate the activities of specialized agents. Agent systems designed to assist in organizational purchasing may require similar procedural knowledge. Reasoning and learning techniques will be required to provide adaptive systems that can handle exceptions and special circumstances.

Negotiating strategies are procedural knowledge in that they describe a plan of action that can be employed to change the set of attributes describing the alternatives. Simple, one-dimensional (price) time-dependent negotiating strategies have been used by buying and selling agents in an electronic marketplace [Chavez et al. 1996]. More sophisticated theories of negotiation can include cooperative and non-cooperative situations and multi-dimensional solution spaces. Negotiating agents must agree to use a common ontology and there must be a way to represent buyer and seller preferences as a utility function. Agents can be preprogrammed with negotiating strategies or equipped with ways to learn effective strategies through techniques such as probabilistic networks or genetic algorithms [Beam et al. 1996].

4.5.3 Semantic Knowledge

A negotiation protocol defines the rules for an economic mechanism and the form of communications between parties. Negotiating agents must have knowledge of these rules in order to communicate with systems, other humans, or other agents. While a protocol is defined for a particular environment, individual agents can have different strategies as they act within the environment. Users must ensure that the chosen strategy is effective with the given protocol [Brenner et al. 1998] and that the strategy cannot be inferred by other parties [Beam et al. 1997]. Many electronic auctions allow participants to "instruct" agents that can monitor for certain events and act on their behalf according to the rules defined for the auction. Multi-agent systems developers are applying research from microeconomics and game theory to more sophisticated negotiation systems. These systems employ mechanisms and protocols that encourage appropriate agent behaviours and consider social welfare, efficiency and market stability [Sandholm 1999].

There are also rules that must be followed to complete a transaction. EDI messages enable systems to exchange information and create contractual agreements between parties in a transaction. Moore [1998] has shown how standard EDI messages can be interpreted in terms of speech act theory. Covington [1998] examines how KQML, based on speech act theory, can provide a way for software agents to exchange similar messages. Both Covington [1998] and Genesereth [1997] suggest improvements or modifications to simple KQML message protocols so that they can convey the level of detail necessary for commerce transactions.

5. Discussion

Software agents may be an important innovation in how people deal with distributed, complex and ubiquitous systems [Jennings et al. 1998] such as those envisioned for e-commerce. We have shown how a DSS approach to software agents leads us towards flexible and interactive systems that accommodate the capabilities of AI systems and adjust to the user's individual and changing needs. The DSS approach also suggests a classification system according to whether agents support search, choice or interface functions. The techniques used in these functions have different reference disciplines, and suggest that agent performance should be measured differently in each function. Future research will be directed at determining whether a system image that uses these functional classifications can help the user develop an improved conceptual model of the agent system. Similarly, we would like to determine if function-specific measures could improve the way agent systems are evaluated.

Within the DSS framework, we have provided examples of the knowledge bases that agents may use to duplicate the services of a human agent in search and choice functions. We identified some of the many design challenges that these systems will encounter and highlighted some promising research areas.

Some of the design challenges can best be addressed through continuing multidisciplinary efforts. The capabilities of new information and communications technologies are redirecting research efforts in many related areas. The Information Sciences community continues to work on more effective linguistic analysis and probabilistic techniques for information retrieval. Management Science can contribute with innovative and dynamic decision models, economists by continuing to develop and adapt mechanisms for non-cooperative environments and multi-dimensional solution spaces. Computer Science will need to develop, design and implement the systems architectures where agents can interact. Continued multidisciplinary communication and collaboration will be important in meeting these challenges.

Other design challenges reflect the need for effective industry cooperation and coordination. Base and domain ontology and transaction protocols require broad support across and within industry groups. History tells us that the market will ultimately determine the success of XML, KQML and other potential standards.

Norman cautions that the main problems facing widespread agent implementation will be “social” and not “technical”. In order to develop trust in the agent’s capabilities, users will need to understand what the agent is doing, and receive appropriate reassurance that it is behaving as expected [Norman 1997]. From the user’s point of view, agent performance in e-commerce will not be likely be satisfactory until we can develop rich user profiles and incorporate relevant situational factors and the user may have to play an active role while this knowledge is acquired. Constructive choice theories and individual decision-making preferences suggest that more than one decision model should be available and that the user may need to interact with the system in order to choose the model that they are comfortable with for the given task. The DSS approach to software agent development and design addresses these challenges by promoting highly interactive, user-centered, systems.

REFERENCES

Ackerman, M., D. Billsus, S. Gafney, S. Hettich, G. Khoo, D. Kim, R. Klefstad, C. Lowe, A. Ludeman, J. Muramatsu, K. Omori, M. Pazzini, D. Semler, B. A. Starr and P. Yap, “Learning Probabilistic User Profiles,” *AI Magazine,* Summer: 47-56, 1997.

http://www.ics.uci.edu/~pazzani/Publications/AI-MAG.pdf (Accessed June 5, 1998)

Arocena, G. O., A. O. Mendelson and G. A. Mihaila, "Applications of a Web Query Language," University of Toronto.

http://www.cs.toronto.edu/~websql/www-conf/wsql/PAPER267.html. (Accessed August 4, 1999.)

Balabanovic, M. and Y. Shoham, “Fab: Content-based, collaborative recommendation,” *Communications of the ACM,* 40(3): 66-72, March 1997.

Beam, C. and A. Segev, "Electronic Catalogs and Negotiations," CIMT Working Paper 96-WP-1016, CITM Berkley, 1998.

http://haas.berkeley.edu/~citm/wp-1016-summary.html. (Accessed July 3, 1998.)

Beam, C. and A. Segev, "Automated Negotiations: A Survey of the state of the Art," CITM Working Paper 97-WP-1022, CITM Berkeley, 1998.

http://www.haas.berkeley.edu/~citm/auction/index.html. (Accessed July 2, 1998.)

Belkin, N. and W. B. Croft, “Information Filtering and Information Retrieval: Two Sides of the Same Coin?” *Communications of the ACM,* 35(12): 29-38, December 1992.

Bettman, J. R., M. F. Luce and J. W. Payne, “Constructive Consumer Choice Processes,” *Journal of Consumer Research,* 25: 187-217, December 1998.

Brenner, W., R. Zarnekow and H. Wittig, Intelligent Software Agents: Foundations and Applications , Springer-Verlag, Berlin Heidelberg, Germany, 1998.

Bui, T. and J. Lee, “An agent-based framework for building decision support systems,” *Decision Support Systems,* 25: 225-237, 1999.

Burke, R. D., K. J. Hammond, V. Kulyukin, S. L. Lytinen, N. Tomuro and S. Schoenberg, “Question Answering from Frequently Asked Question Files,” *AI Magazine,* Spring: 57-66, 1997.

Cardie, C., “Empirical Methods in Information Extraction,” *AI Magazine,* Winter: 65-79, 1997.

Chavez, A. and P. Maes, "Kasbah: An Agent Marketplace for Buying and Selling Goods," First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK, April 1996.

Coupey, E., "Restructuring: Constructive Processing of Information Displays in Consumer Choice," *Journal of Consumer Research,* 21: 83-99, June 1994.

Covington, M. A., "Speech acts, electronic commerce, and KQML," *Decision Support Systems,* 22: 203-211, 1998.

Cuena, J. and S. Ossowski, "Distributed Models for Decision Support," Multiagent Systems, G. Weiss (ed.), MIT Press, Cambridge, MA, 1999.

Davis, R., H. Shrobe and P. Szolovits, "What is Knowledge Representation?" *AI Magazine,* 14(1): 17-33, 1993.

http://medg.lcs.mit.edu/ftp/psz/k-rep.html. (Accessed August 10, 1998.)

Duschka, O. M. and M. R. Genesereth, "Infomaster - An Information Integration Tool," Stanford University, 1999.

http://infomaster.stanford.edu/infomaster-info.html. (Accessed May 11, 1999.)

Etzioni, O., "Moving Up the Information Food Chain," *AI Magazine,* Summer: 11-18, 1997.

Finin, T., R. Fritzson, D. McKay and R. McEntire, "KQML - A Language and Protocol for Knowledge and Information Exchange," Technical Report CS-94-02, Computer Science Department, University of Maryland , UMBC, Baltimore MA., 1998.

http://www.cs.umbc.edu/kqml/papers/kbkshtml/kbks.html. (Accessed September 17, 1998.)

Genesereth, M. J., "An Agent-based Framework for Interoperability," Software Agents, J. M. Bradshaw (ed.), AAAI Press/MIT Press, Menlo Part, CA, 1997.

Glushko, R. J., J. M. Tenenbaum and B. Meltzer, "An XML Framework for Agent-based E-commerce," *Communications of the ACM*, 42(3): 106-114, March 1999.

Gregor, S. and I. Benbasat, "Explanations from Intelligent Systems: Theoretical Foundations and Implications for Practice," *MIS Quarterly,* 23(4): 497-530, December 1999.

Gruber, T., "What is an Ontology?" 1999.

http://www-ksl.stanford.edu/kst/what-is-an-ontology.html. (Accessed July 5, 1999.)

Guan, T. and K. F. Wong, "KPS --- a Web Information Mining Algorithm," 1999.

http://www8.org/w8-papers/4a-search-mining/kps/kps.html. (Accessed July 5, 1999.)

Holsapple, C. W. and A. B. Whinston, Decision Support Systems - A Knowledge-based Approach, West Publishing Company, St Paul, Minnesota, 1996.

Howe, A. E. and D. Dreilinger, "SavvySearch: A Metasearch Engine That Learns Which Search Engines to Query," *AI Magazine,* Summer: 19-25, 1997.

Jennings, N. R. and M. Wooldridge, Applications of Intelligent Agents, Springer-Verlag.

http://www.cs.umbc.edu/agents/introduction/jennings98.pdf. (Accessed June 17, 1998.)

Keller, A. M. and M. R. Genesereth, "Multivendor Catalogs: Smart Catalogs and Virtual Catalogs," 1999.

http://infomaster.stanford.edu/infomaster-info.html. (Accessed May 11, 1999.)

KSE (Knowledge Sharing Effort), University of Maryland (Baltimore), 1998.

http://www-ksl.stanford.edu/. (Accessed August 20, 1998.)

Lesser, V., B. Horling, A. Raja, X. Zhang and T. Wagner, "Resource-Bounded Searches in an Information Marketplace," *IEEE Internet Computing,* 4(2): 49-58, March - April 2000.

Lieberman, H. "Autonomous Interface Agents", 1998.

http://lieber.www.media.mit.edu/people/lieber/Lieberary/Letzia/AIA/AIA.html. (Accessed July 2, 1998.)

Maes, P., R. H. Guttman and A. G. Moukas, "Agents that Buy and Sell," *Communications of the ACM,* 42(3): 81-91, March 1999.

Malone, T., K. R. Grant and K.-Y. Lai, "Agents for Information Sharing and Coordination: A History and Some Reflections," Software Agents, J. M. Bradshaw (ed.), AAAI Press / The MIT Press, Menlo Park, CA, 1997.

Meyer, R. J. and B. E. Kahn, "Probabilistic Models of Consumer Choice Behaviour," Handbook of Consumer Behaviour, T. S. Robertson and H. H. Kassarjian (eds.), Prentice-Hall, Englewood Cliffs, NJ, 1991.

Moore, S. A., "Categorizing automated messages," *Decision Support Systems,* 22: 213-241, 1998.

Neuromedia., 2000.

http://www.neuromedia.com. (Accessed May 15, 2000.)

Ngu, D. S. W. and X. Wu, "SiteHelper: a localized agent that helps incremental exploration of the World Wide Web," *Computer Networks and ISDN Systems*, 29: 1249-1255, 1997.

Nissen, M. E., "Procurement Revolution with Intelligent Agent Technology," *PRACTIX,* December 1999.

http://www.capsresearch.org/research/best-practices.html. (Accessed January 22, 2000.)

Norman, D. A., "Some Observations on Mental Models," Mental Models, D. Gentner and A. L. Stevens (eds.), Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.

Norman, D. A., The Design of Everyday Things, Doubleday/Currency Edition, New York, NY, 1990. (Reprint. Originally published: The Psychology of Everyday Things, Basic Books Inc., New York, 1988.)

Norman, D. A., "How People Might Interact with Agents," Software Agents, J. M. Bradshaw (ed.), AAAI Press/The MIT Press, Menlo Park, CA, 1997.

Pao, M. L., Concepts of Information Retrieval, Libraries Unlimited Inc., Englewood, CO, 1989.

Rucker, J. and M. J. Polanco, “Personalized Navigation for the Web,” *Communications of the ACM,* 40(3): 73-75, March 1997.

Sandholm, T. W., "Distributed rational decision-making," Multiagent Systems: a modern approach to distributed artificial intelligence, G. Weiss (ed.), MIT Press, Cambridge, MA, 1999.

Siskos, Y. and A. Spyridakos, “Intelligent multicriteria decision support: Overview and perspectives,” *European Journal of Operational Research*, 113: 236-246, 1999.

Smith, H. and P. Poulter, “Share the Ontlogy in XML-based Trading Architectures,” *Communications of the ACM,* 42(3): 110-111, March 1999.

Turban, E., Decision Support and Expert Systems, MacMillan, New York, NY, 1988.

Turban, E. and J. E. Aronson, Decision Support Systems and Intelligent Systems, Prentice Hall, Upper Saddle River, NJ, 1998.

Turing, A. M., “Computing Machinery and Intelligence,” *MIND - A quarterly review of Psychology and Philosophy,* LIX (236), October 1950.

http://www.abelard.org/turpap/turpap.htm. (Accessed August 10, 1999.)

Wang, S., “Analyzing Agents for Electronic Commerce,” *Information Systems Management,* 16(Winter): 40-47, 1999.

White, D. J., Decision Methodology - A Formalization of the OR Process, John Wiley and Sons, London, UK, 1975.

Widemeyer, G. R. and R. M. Lee, "Preference Elicitation in Decision-Aiding: Application to Electronic Shopping," Decision Support Systems: A Decade in Perspective, E. R. McLean and H. G. Sol (eds.), Elsevier Science Publishers B. V., North Holland, 1986.

Wooldridge, M., "Intelligent Agents," Multiagent Systems, G. Weiss (ed.), MIT Press, Cambridge, MA, 1999