Applying Intelligent Agents in the Scanning and Interpretation of Strategic Data

Shuhua Liu

Åbo Akademi University,
Institute for Advanced Management Systems Research,
Lemminkäisenkatu 14 A, FIN-20520 Turku, Finland

Turku Centre for Computer Science
TUCS Technical Report No 85
December 1996
ISBN 951-650-928-2
ISSN 1239-1891

"What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention, and a need to allocate that attention efficiently among the over abundance of information sources that might consume it." -- *Herbert Simon*

(Agent News, Volume 1, Number 6, Baltimore, May 11, 1996, available at: http://www.cs.umbc.edu/agentnews/96/06/)

Abstract

The scanning and interpretation process of executive managers has often featured by intuitive and fragmented activities, as well as both information overload and information ignorance. Messages and signals from the highly complex, dynamic and unpredictable business world are characterized by invariable ambiguity and subject to multiple interpretations. In this work, a software agent based support system is proposed as a helpful solution. We consider that computerized system be able to contribute a lot to a systematic, extensive while intensive, and faster scanning process than human process alone. The system is expected to offer managers with personalized information services they need, to help them in distinguishing useful information implicit in large volumes of data, acquiring the right, limited information, evaluating the accessed information and extracting meaningful insights from it. This paper reviews agents fundamentals: agent concept, architecture, attributes and properties, languages and tools, and its various applications. It also identifies the agents needed in the support system for strategic scanning and interpretation. Tasks of these agents are analyzed and described. A layered, federated multi-agent system structure is designed.

Keywords: Strategic Scanning and Interpretation, Strategic Issue Management, Software Agents, Agent Based Support System

TUCS Research Group
IAMSR Management Support Systems Group

1. Introduction

Scanning and interpretation, in a general sense, concerns about seeing, perceiving and sense-making. Scanning is both looking at information (viewing) and looking for information (searching). And interpretation is the process of translating the viewed and searched events, of developing models for understanding, of bringing out of meaning, and of assembling conceptual schemes (Weick and Daft 1983). Strategic data is "anything that has potentially larger impact for the company" (El Sawy, 1985). More specifically, strategic data refer to information about events, trends, and relationships in an organization's business environment[1], the knowledge of which will be of assistance to top executives in identifying and understanding strategic threats and opportunities (Aguilar, 1967). No type or class of information is intrinsically strategic. It is strategic only if it relates to a strategic opportunity or problem (Aguilar, 1967), that is, when it unveils opportunities to exploit the firm's strengths, or highlights the firm's weaknesses or potential threats facing the firm (Lozada and Calantone, 1996).

Scanning and interpretation of strategic data provide the information basis for strategy formulation, and it typically forms the first phase of an organization's strategic process and decisions[2]. Through the process of strategic scanning and interpretation (SSI), it is expected for the managers to grasp how the environment is changing and to develop a shared understanding of the ways these changes will have major impacts on their business; to recognize the threats and opportunities, to identify the strategic issues that they need to attend to, and then to take proper actions.

Although being informed of the business situation has always been important to strategy makers, the globalization of markets, fast developing technologies and intensified competition have increased the scope, the dynamics and the interconnectedness of the external environment. Many organizations are undergoing significant changes inside themselves too. Discontinuities and surprises in the business environment often catch a company easier than before, and speedier communications makes the time span of important changes critical. Furthermore, in a complex, dynamic and unpredictable business environment, the messages and signals are invariably ambiguous and subject to multiple interpretations. As a result, reaching the relevant and timely information, resolving the equivocality, discerning the most significant changes and early warning about discontinuities, interpreting their meaning for developing appropriate responses becomes the central information activities. While traditionally managers' understanding of their business situation has been largely intuitive and fragmented (Nanus, 1982), today's environment would seem extremely obscure and difficult to be comprehended if still little help and guidance is available on where they should focus their attention. In order to base their decisions on the most up-to-date information while not out-dated information, continuously or constantly active and systematic acquisition and sense-making of strategic information becomes highly critical.

---

[1] In this work, we use the term business environment to refer to both the external and internal settings of an organization.
[2] The two terms - "The Scanning and Interpretation of Strategic Data" and "Strategic Scanning and Interpretation" - will be treated as synonyms that can be used interchangably.

In a more general context, with the advances in the collection of electronic scanner data, the amount and recency of environmental data has been considerably enhanced. However it has become a common problem that the increased information availability through computers threatens to exceed our abilities (especially intuitive managing and understanding capability) to quickly sort and choose, to find and review the right information, and more importantly, to absorb and understand properly its implication. Most companies still do not have the formal collection and analysis system in place, let alone making effective use of the information in strategic decision making (AKA Inc.).

So, on one hand, a continuous stream of events and trends keeps bombarding the managers and deluge of information could make them feel overwhelmed and cost them a lot of valuable time and effort. On the other hand, what coexists with information overload is the problem of information ignorance, which means that managers often do not have the information they really need at hand. We believe that, to cope with this situation, managers need a number of powerful information and service management tools to help them to distinguish information implicit in the data, to get the *limited, right* information, to help them to evaluate the accessed information and extract meaningful insights with guidelines for future actions. A natural question is thus, can a computer based support system contribute to the strategic scanning and interpretation activities and how to construct such an interactive software environment to enhance or stretch managers' capability for discovering, among the larger number of comprehensively interrelated business environmental factors, strategic opportunities and threats. This constitute the main concern of our research.

As exciting as it is, the building of a computer based system to support the strategic scanning process of top executives is also a challenging task. To offer valuable assistance, the system is expected to be able to facilitate executives with a broad and comprehensive first approximation to environmental information, alert managers to weak signals so that they will not to be surprised or less surprised by sudden changes. It is expected to attract top management's attention upon critical issues and to have their attention refreshed frequently, make information recalling easily, and encourage the sharing and communicating of managers' views and perspectives. The ultimate goal will be to scan more effectively than current approaches, to make the SSI process more systematic, more intensive while extensive, relatively inexpensive and faster than human process alone.

Research in automatic gathering, filtering and searching of information has exploded in recent years. With the proliferation of electronically available information, an additional burden has been placed on. The information that represents the best response to a user's need is quite often the synthesizing of data from a variety of different, distributed, or heterogeneous sources (Decker et al). In dealing with these problems, the software agent approach and technology represents one of the most popularly exploited solutions, which we consider to have much to offer in building a SSI support system.

Intelligent agents originates from the domains of artificial intelligence (AI) and distributed artificial intelligence (DAI), and are applied in many areas. Although the notion of an agent was generated much earlier, it is only recently that the term intelligent software agent has become quite popular. Significant interest in agent reassert and development has mushroomed and can be found scattered across different forums such as AI and mainstream computer science, robotics, software development, information mining in general and in network environment, and many others (Riecken, 1994b; Isbister and Layton, 1995; Wooldridge and Jennings, 1995; Song and Franklin, 1996). In management studies, agent concept has been utilized in designing agent oriented organizations (Unland, 1995).

The transition from agent research to practical applications has also occurred at a very brisk pace and agent-based computing (ABC) is said likely to be the next significant breakthrough in software system development (Sargent, 1992; Robin Smith, "The Practical Impact of Agent Technology", PAAM'96). It is even felt increasingly that, both within academia and industry, intelligent agents will be a key technology as computing systems become more distributed and interconnected. A broad spectrum of companies are investing in the research and development of intelligent agents and probably all the major software manufacturers are exploring the use of them (Norman, 1994; Nwana, 1996). Although presently most users of the agent technique are still limited to researchers and a small part of the WWW user population (Hermans, 1996), agents have been applied to assist in a range of daily, administrative activities such as meeting scheduling, email messaging, news filtering, papers sending, transportation arranging, tracking the whereabouts of people, information locating in multiple databases and so on. They also have the potential to contribute to many other contexts such as system and network management, workflow and industry process management, electronic commerce, business management such as portfolio management and marketing intelligence.

As with anything, when we look at one side of software agents, we need also to know the opposite. As Norman (1991, 1993, 1994) has pointed out,

> "Agents occupy a strange place in the realm of technology - generating fear, fiction, and extravagant claims. ... Agents may well have numerous positive contributions to our lives. They can simplify our use of computers, allowing us to move away from the complexity of common languages or the tedium of direct manipulation toward intelligent, agent-guided interaction. ... Agents promise to hide complexity, to perform actions we could not or would prefer not do it ourselves. ... But along with the potential comes potential danger. ... Along with their benefits and capabilities[3] come the potential for social mischief, for systems that run amok, for a loss of privacy, and for further alienation of society from technology through increasing loss of the sense of control. ... Myths, promises, and reality are all colliding."

obviously agent technology is not problem-free. Lanier even claimed that agents make people diminish themselves and start to think of themselves as being like the computer, as a consequence of unavoidable psychological algebra. *"They (will) start to limit themselves to the categories and procedures represented in the computer, without realizing what has been lost."* All these highlight a question of how users should trust their software agents and how to develop agents that are trustable. So, the agent technology and applications are still new and much remains to be explained, however it

[3] "The kernel concept of an agent implies human-like autonomy to some extent: some level of intelligence, some form of self-initiated, self-determined goals; working without supervision on tasks thought to be for our benefit, but not necessarily to our liking." (Norman, 1994)

seems clear that the development of agent software is coming anyway because of its potentially great help and benefits and demand from industries. When we try to use it, we have the chance to bear in mind the questions and potential problems and to find out the answers and solutions in our practice.

This paper presents our first effort towards the application of agent approach in strategic scanning and interpretation. In Section 2, a review of software agent fundamentals is made to understand what an agent is; what it can do and can not do; what are the components of an agent and how an agent can be constructed. In Section 3, useful software agents needed to constitute a SSI support system are identified. The tasks for these agents are analyzed. On the basis of existing principles and research findings, primary design considerations of a multi agent support system for strategic scanning and interpretation (MASCAN) are drawn out in Section 4. The system structure is then proposed. Discussions and conclusions are made in Section 5.

2. Intelligent Software Agent Fundamentals

As we will see later, central to the notion of an agent is its autonomous and goal-directed behavior. However, automata are not new concepts. Intelligent machines have existed in fact or fiction for centuries. The interest in having an agent to do certain tasks for people has always been there, and it is in fact regarded as the primary driving force of the agent domain (Minsky, 1994; Norman, 1994). The reason why this field is revitalized today is to a large extent attributed to the developing of information technology and the increasing of computational power. In one aspect, applications thought to be impossible to automate in the past are rapidly becoming a possibility; in another aspect, the widespread computer systems create new extra tasks for people. Computers can do more to help people to exploit most from the computerized systems and they are also expected to do so because people need help to deal with the ever-increasing information overload and work overload.

Within this rapidly growing field, it is well acknowledged that there is relatively little consensus about what an agent is after all. Researchers have offered a variety of definitions and descriptions of the term agent, with each growing directly out of the set of examples of agents the definer had in mind and hoped to explicate their specific use of it (Franklin and Graesser, 1996). The situation has aroused caution. Wooldridge and Jennings (1995) pointed out the danger of the term being subject to both abuse and misuse, and to the potential confusion of the research community. Many efforts have been made to clarify the concept (Wooldridge and Jennings, 1995; Jennings, Applying Agent Technology, PAAM'96; Franklin and Graesser, 1996; Hermans, 1996), still it is quite contentious. Obviously it is not an easy task to work out exactly what is the essence of the agent concept. But we believe that we can make our contribution by first comprehending its basic implication and then incorporating new perspectives through utilizing it in the scanning and interpretation activities. The resulted description of an agent would be a shared requirement and understanding of both the researchers and the users.

2.1 The Mysterious Agent

2.1.1 Agents and Their Attributes

Agents bear multiple names and forms in different contexts: robots, daemons, knowbots, softbots, and so on. One dictionary definition of the term "agent" describes it as *a person or business or an entity authorized to act on another's behalf* (Minsky, 1994). This suggests first that an agent has some *specialized purpose* and accomplish *delegated tasks.*

A software agent is defined analogously as *a computational program that acts to accomplish tasks on behalf of its user with some degree of autonomy, and in so doing, employ some knowledge or representation of the user's goals or desires.* The basic idea is that the *user needs only specify a high-level goal instead of issuing explicit instructions, leaving the 'how' and 'when' decisions to the agent* (The Software Agents Mailing List FAQ; IBM Agent: http://activist.gpl.ibm.com:81/WhitePaper/ptc2.html). In this sense, agents participate actively in accomplishing tasks, rather than serving as passive tools as do conventional applications (Smith et al, 1994). It makes goal-orientation and autonomy an important property of software agents.

From users' point of view, an agent at least reduces complexity and increases efficiency, and the delegation of functions supports user mobility. An agent is also seen as its user's personal digital assistant which provide personalized assistance in a specialized task (Maes, 1994). An agent's capability of personalization or customization of assistance comes form its learning ability or adaptivity. According to Maes (1994), agents not only perform on user's behalf; they can also train or teach the user and help different users to collaborate.

When considering a multi-agent system, agents are not only active and autonomous programs, but also *concurrently and cooperatively running* entities to *provide acceptable solutions to certain problems at an affordable price.* Some people believe that in many cases the individual agents of a system are not that intelligent at all, but the combination and cooperation of them lead to the intelligence and smartness of an agent-system (Hermans, 1996).

The concept of an agent is also closely related to the concept of an object. From an object oriented view, an agent is an object which encapsulates some private internal state, operations which may alter this state, and an external view which describes the behavior of the system in terms of the user interface level events and makes aspects of its internal state visible to the environment (Fields et al, 1994; Fisher, 1995).

Most often, agents are defined or described by giving a list of attributes they enjoy. The literature has proposed a host of properties of agents, which are summarized in Table 1. It is by these properties and behavior that an agent differs from just any computer programs.

Table 1 Agent Attributes and Properties

(Sources: Maes, 1994; Etzioni and Weld, 1995; Jennings and Wooldridge, 1995; King, 1995; Moon et al, 1996; Franklin and Graesser, 1996; etc.)

(1) Autonomous
Central to the definition of an agent is the ability to act autonomously (Kearney, 1996). Unlike standard programs which are directly invoked by the user, an agent is able to take initiative and exercise a non-trivial degree of control over its own actions and internal state. It can perform the majority of its tasks without the direct intervention of the user or other agents (Wooldridge and Jennings, 1995). The autonomy of an agent is also presented as that it does not blindly obey commands, but has the ability to modify requests, ask clarification of questions, or even refuse to satisfy certain requests. Autonomy requires not only autonomous execution, but also autonomous goals. An agent's autonomy is necessarily limited and bounded, otherwise autonomy would be contradictory with its situated reactiveness and its collaborativity (Castelfranchi, 1995).

(2) Collaborative or Socially-Able
A software agent inhabits certain computational environment, which may compose of, for example, its user, a collection of other agents, the Internet or other data sources. It interacts or collaborates, when it deems appropriate, with its user or other agents to accomplish its own goals or to help other agents in accomplishing their goals.

(3) Reactive or Responsive
An agent perceives its environment and responds in a timely fashion to environmental changes. An agent's actions are not scripted, rather, an agent is able to dynamically choose which actions to invoke and in what sequence, in response to the state of its external environment.

(4) Proactive or Goal-Directed
An agent does not simply act in response to its environment. It also anticipates the environmental changes and creates goals proactively for its specialized purpose. An agent accepts high-level requests indicating what a human wants and is responsible for deciding how and when to satisfy the request. It does not need explicit action instructions from its user. An agent can have one single goal or multi conflict goals that may be set forth in the design or given to the agent at run time. When an agent has no specific goals, it explores the environment with the very general goal of "satisfying curiosity" or "learning new things" (Farnk, 1990).

(5) Adaptive and Learning
Adaptivity and learning enable an agent to observe and extract its user's interests, preferences and habits over time, to automatically customize its assistance according to what it learned about the user, and to change its behavior based on previous experience.

(6) Concurrently Execution
The notion of an agent is seen as a natural development of the object based concurrent programming paradigm. An agent behaves like a self-contained concurrently executing software process that encapsulates some state and is able to communicate with other agents via message passing. Actions are concurrently executed within an agent and between agents (Agha, 1986; Agha et al, 1993).

(7) Rationality
An agent will act in order to achieve its goals and will not act in such a way as to prevent its goals being achieved, at least in so far as its beliefs permit. (Goodwin, 1993)

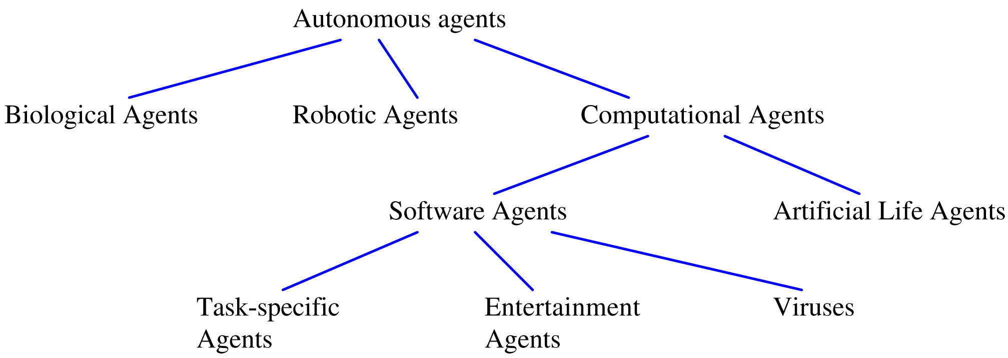
(8) Mobile
A mobile software agent carries out delegated tasks on a network. It can transport itself from one machine to another, navigate around LANs, WANs or the Internet to perform complex processes, traveling from desktop-to-desktop at the discretion of the sender and report. An agent with mobility is called mobile agent.

(9) Veracity: an agent will not knowingly communicate false information.

<u>(10) Character:</u> an agent has a well-defined, believable "personality" and emotional state.

2.1.2 Agent Taxonomy

The agent world is of great richness and diversity. To get a general picture about it, Franklin and Graesser's (1996) agent tree is very helpful. It clearly outlines the agent world as follows:



Agents can be further categorized according to their specialized attributes, control structures and their applications (Franklin and Graesser, 1996). In terms of their property, there are for example learning agents and mobile agents. In terms of their architecture or control structures, there are reactive agents, deliberative agents and hybrid agents (Brustoloni, 1991; Franklin and Graesser, 1996), which we will discuss in the next section. By the tasks they perform agents can be classified into many different kinds of domain specific agents, depending very much on the application context. For our purpose we will focus on two general types of software agents: *interface agents and information agents.*

<u>Interface Agents</u>
An interface agent is usually modeled closely after the metaphor of a personal assistant. When an agent is used to provide personalized assistance to a user engaging in the use of a particular computer application or service, it is called an interface agent (Maes, 1994).

An interface agent is also a learning agent. It gradually learns how to better assist the user by observing and imitating the user, understanding user's interests and needs, inducing and building up user profile, receiving positive and negative feedback from user, receiving explicit instructions from user, or asking other agents for advice. The basic idea is that machine learning can lead to self customizing software assistants (Maes, 1994; Mitchell et al, 1994). Interface agents help users with tasks such as calendar scheduling, email messaging, news filtering, web watching, etc..

Information Agents

Information agents help people in information access and assimilation. They take the roles of information brokers: gathering, filtering, searching and interpreting information for the user.

Information access has traditionally been very syntactic, keyword-based searching. A user needs to specify a set of words as a part of the search command and the search is conducted for the occurrence of these words. In order to get a better result, he should know exactly what information he is looking for be able to express his information need clearly with the right keywords (Hermans, 1996). However, users could be uncertain and imprecise. The keywords often could not exactly reflect what he is looking for and the retrieved information could be too large a volume and unfocused. The user may have to repeatedly refine his query to find a satisfactory answer. The process can be long and discourage the user in performing further searches.

The storage and retrieval of data is undergoing substantial changes in recent years. More recently the development of Internet and WWW has turned the classical view of databases as a centralized collection of structured homogeneous data into distributed heterogeneous ones. A datum is not just a structured entity any more, but it could be almost any kind of representable item such as text, graphics, image and sound. And the unstructured storage of data makes the search complex since the user could get lost in the information space (D'Aloisi and Giannini, 1995). Obviously the syntactic approach is difficult to meet the requirements under these circumstances. An alternative approach to it is semantic information retrieval, by which a system keeps track of what information is available where at a semantic level, i.e., in terms of what the data really means instead of just a serious of words, interprets queries similarly, and provides user with the access based on this understanding. This approach does not depend on the information bearing object being textual so it should be able to deal with the multiplicity of data formats (Guha and Lenat, 1994). Yet there are still other needs from users that are not satisfied with search-only approaches.

Software information agents represent a new approach to information access service. They are invented to be able to combine intelligently disparate streams of information from multiple distributed resources to respond to a specific request from users with relevant, synthesized results; to answer queries that require logical or heuristic deductions or induction, or require numerical computation using algorithmic models; to extract multiple cause-and-effect relationships from large data set, and to uncover findings that could be missed by human experts. At the same time, as an ongoing process, an agent can remain active all the time to visit periodically the information sources, monitor information, looking for trends, identify new information, extract relevant data, fuse the information with other relevant data, and distribute results to its user. This is becoming an invaluable service as such tasks would be very difficult or simply impossible to be carried out by human alone. With agents, the user's time is not wasted at manually searching for information anymore. The information agents help people to *reduce information overload and increase awareness* (Woelk and Tomlinson, 1994; AKA Inc.)

Information agents found great potentiality in many applications. For example they could be used for providing opportunity advice through tracking share prices or checking stock levels for inventory control planning.

2.2 Agent Architectures

As mentioned earlier, an agent situates in a dynamic environment. Within such an environment, it continuously performs three functions: sense, perception and action. It senses, and perceives the environment, determines its actions and act autonomously to accomplish its goals and tasks. In doing so it potentially also causes changes in the environment (Hayes-Roth, 1995; Moffat and Frijdat, 1995; Wooldridge and Jennings, 1995; Franklin and Graesser, 1996). An agent could exhibit different behavior, for example reactive or proactive behavior or both, which is affected by the environment, but is determined by its internal mental state and the way its elements are structured and interrelated (Maes, 1994; Rao and Georgeff, 1992). So, *what is meant by the mental state of an agent, what are the ways to create certain behavior and what kind of architectures an agent can have?*

2.2.1 The Mental State of Software Agents

The mental state of an agent represents the mentalistically conceptualized model of it, especially when a stronger notion of agents is adopted (Wooldridge and Jennings, 1995). It is in turn described and represented by certain mentalistic notions or variables such as belief and knowledge, goals, desires, intentions, plans and obligations (Rao and Georgeff, 1992; Bates, 1994; Thomas, 1995; Wooldridge and Jennings, 1995; Weerasooriya et al, 1995).

The mentalistic variables are also called attitudes. According to Wooldridge and Jennings (1995), agent attitudes fall into two categories: *information attitudes and pro-attitudes*. Information attitudes are related to the information an agent has about the world it occupies. Belief and knowledge are information attitudes. Pro-attitudes are those that in some way guide the agent's actions, such as desire, intention, obligation, commitment, and choice. Information attitudes and pro-attitudes are closely related. An agent will make choices and form intentions on the basis of the information it has about the world. Precisely which combination of attitudes is most appropriate to characterize an agent is an issue of debate. Different works tend to take it differently. One general rule is that an agent must be represented in terms of at least one information attitude and at least one pro-attitude (Wooldridge and Jennings, 1995). In this paper we introduce four attitudes: belief, intention, goal and plan.

(1) Belief:
Belief represents an agent's information about the world it occupies. It is the synonym of knowledge that is used in more traditional AI field. We understand belief as knowledge in a broad sense, which means that it can encompass both facts and knowledge. Perceptions give the agent beliefs about the current state of the domain. Predications or inferences give the agent beliefs about certain aspects of the domain

that it should next consider (Norman and Long, 1995; Thomas, 1995). So, perceptions and predication all contribute to belief revision.

(2) Intention:

Intentions represents desires that an agent will attempt to bring about. The tasks an agent has decided to accept become that agent's intentions. *An intention contains a goal part, a motivation part, and a rating part* (Norman and Long, 1995). The goal represents the purpose of the intention and the motivation represents what must be believed in order for the intention to be maintained. An intention will be maintained until either its goal is believed to be satisfied or its motivation is no longer present. So, agent intentions are the goals that it has committed itself to accomplish. The rating of an intention represents its priority. The higher the rating the more important the intention. The highest rate intention is the current intention and will guide the action of the agent. The state of motivation is dependent on both the internal state of the agent and the state of the external world. The motivations of an agent are derived from both the predicted behavior of the domain and significant unpredicted events detected by the agent. In complex environments which are subject to unexpected changes, an agent may be obliged to revoke a prior intention by dropping its commitment to the current intention because unanticipated changes make it irrational for the agent to continue to act for its intention (Rao and Georgeff, 1992; Dougha, 1995).

(3) Goal:

Goals are the function of an agent's desires and its concerns. Goals are generated through environmental stimulus, self monitoring (Sloman; Norman and Long, 1995), or external force; reactively, proactively or externally. Reactive goals are created in response to beliefs about the current state of the domain. It is essential for an agent to create goals in reaction to certain environmental conditions because there will always be situations in any real world domain that can never be predicated and can only be reacted when they occur. Proactive goals are created in response to predications about how the domain will change over time, and typically satisfied through planning. If an agent is capable of creating effective proactive goals in an efficient way, the undesirable situations that can be predicated by the agent may be avoided through purposeful actions. An agent will react to proactive goals if the predicated situation is of immediate relevance. In this way the behavior of the agent changes as the context changes. An agent is also capable of planning for goals provided by an external source, i.e. instructions given by a human operator. These external provided goals can be suspended when the agent needs to react to an emergent situation.

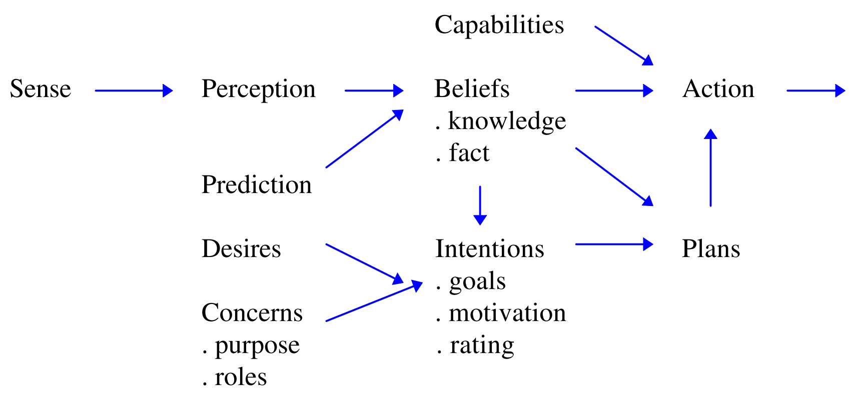
Concerns are the function of an agent's fundamental purposes, its role in life and its reason for having goals. The difference between concerns and goals is that, goals are usually transient and concerns are relatively long term. Concerns also contribute to the formulation of goals. When an event is perceived relevant to one of the agent's concerns, an internal signal is emitted and a goal is generated. The concerns are definitely not active all the time while goals will endure until they are achieved. The concerns can instead be thought of as demons, lying dormant until concern-relevant events arouse them (Moffat and Frijda, 1995).

(4) Plan:

Once an agent has adopted an intention it may need to develop a plan for achieving that intention. The content of an intention is a statement to be made true, not an action to be done. Plans, however, come into two kinds: plans to do -- plans to perform some action, and plans to achieve some state of the world -- plans to achieve. Intentions are linked to actions by rules or plans which represent information about how to achieve intention (Wooldridge, 1995).

Based on the above concepts, an internal view of agents is illustrated in Figure 1.

Figure 1. An Internal View of Agents



2.2.2 Different Approaches and Architectures for Software Agents

The concept analysis of an agent put up requirements for its architecture. An agent architecture concerns issues surrounding the construction of a computer system that satisfies the desired attributes. It is, first, a balanced choice of agent properties and behavior; and then, a careful modeling to produce the desired behavior, which involves choosing and organizing of various mental attitudes (Muller et al, 1995; Wooldridge and Jennings, 1995). For certain desired behavior, there can be different architectures to generate it. The different architectures reflect different approaches for agent building.

The different approaches for agents are generally summarized into three categories by Wooldridge and Jennings (1995): *deliberative approach, reactive approach and hybrid approach.* Correspondingly there are three types of agent architectures: *deliberative architecture, reactive architecture and hybrid architecture.*

1. Deliberative Approach (Planning Architecture)

A deliberative agent or architecture is defined as one that contains an explicitly represented, symbolic model of the world, and in which decisions an agent makes (for example what actions to perform) are based on logical (or at least pseudo-logical reasoning) pattern-matching and symbolic manipulation. This approach represents a classical way to build agents, which is also known as planning approach or symbolic AI paradigm. By deliberative approach, agents are viewed as a particular type of knowledge based systems.

The deliberative approach results in planning centered agents whose primary component is a planner, the focus of which is principally concerned with the creation of good plans to satisfy a conjunction of goals in an efficient way. Agent actions are predefined or preplanned (Jennings, 1993a, 1993b; Jennings et al, 1993; Jennings, 1995).

2. Behavior Approach (Reactive Architecture)

With deliberative approach, it typically assumes that neither the goals of an agent nor the importance of the goals change with time. However, this assumption can not be justified for an agent acting in a real world domain where changes happen over time and present a continuous flow of contexts to it. An effective agent must be capable of identifying changes that are relevant to its purposes and creating appropriate goals in response to these changes. This calls for an alternative approach which is called behavior approach or reactive approach. Reactive architectures have their basis on Brooks' three key theses:

> *First,* intelligent behavior can be generated without explicit representation of the kind that symbolic AI proposed;
> *Second,* intelligent behavior can be generated without explicit abstract reasoning of the kind that symbolic AI possesses;
> *Third,* intelligence is an emergent property of certain complex systems.

The reactive architecture does not include any kind of explicit knowledge representation and central symbolic world model, and does not use complex symbolic reasoning. It relies completely on the world for information and built-in competence. It enables a tight connection of sense and perception to action and exhibits intelligent behavior based on situated actions. The main property of an reactive agent is its reflexivity which means that it is highly reactive to the changes in its environment (Brooks; Hayes-Roth; Maes; Wavish and Graham, 1994; Sloman, 1996).

3. Hybrid Approach (Hybrid Architectures)

Comparing a purely deliberative approach with a purely reactive approach, many researchers believe that neither of them is suitable for building agents. A combination of the two is advocated, which results in the third class of agent architectures: hybrid architectures. A hybrid agent is expected to generate both the above two types of behavior. It synthesizes reactive action and deliberative reasoning, and unifies reaction with goal-deliberation. It can react immediately to certain stimuli and at the same time it can also be carrying out a longer term planning process in the background (Maes, 1990; Fisher, 1995; Ekdahl et al, 1995; Franklin and Graesser, 1996; Wooldridge and Jennings, 1995).

A variety of hybrid architectures have been proposed by researchers. The Procedural Reasoning System (PRS) by Georgeff and Lansky (1987), Touring Machines by Ferguson (1992a, 1992b), COSY by Burmeister and InterRRap by Muller (Muller, 1994; Muller and Pischel, 1994; Muller et al, 1995), are well known examples.

As an important class of approaches to modeling systems that have to behave in a goal-directed manner in a complex changing environment, the layered approach has also been used by the agent community in formulating hybrid architecture for software agents. The layers serve as a connection mechanism or an intermediary between the perception (input) and action (output) of an agent. The resulted *layered hybrid architecture* describes an agent as consisting of several hierarchical functional modules, representing different requirements on an agent, such as reactivity (with a behavior based reactive layer), goal-directed behavior (with a plan based planning layer), coordination with others (with a cooperation layer), belief representation (with a modeling layer). All the layers work concurrently.

Layered hybrid architectures are furtherly manifested as *horizontal layer architectures and vertical layer architectures.* In horizontal layer architectures, all the layers of an agent have access to both perception and action components; while in vertical layer architecture only one and normally the lowest layer has a direct interface to perception and action components. The different forms of layered architectures is illustrated in Figure 2. Ferguson's Touring Machines, which takes a horizontal layered architecture, and Muller et al's InteRRap, which takes a vertical layered architecture, are shown in Figure 3 and Figure 4 respectively.

Figure 2 Layered Agent Architectures
(Source: Muller et al, 1995)

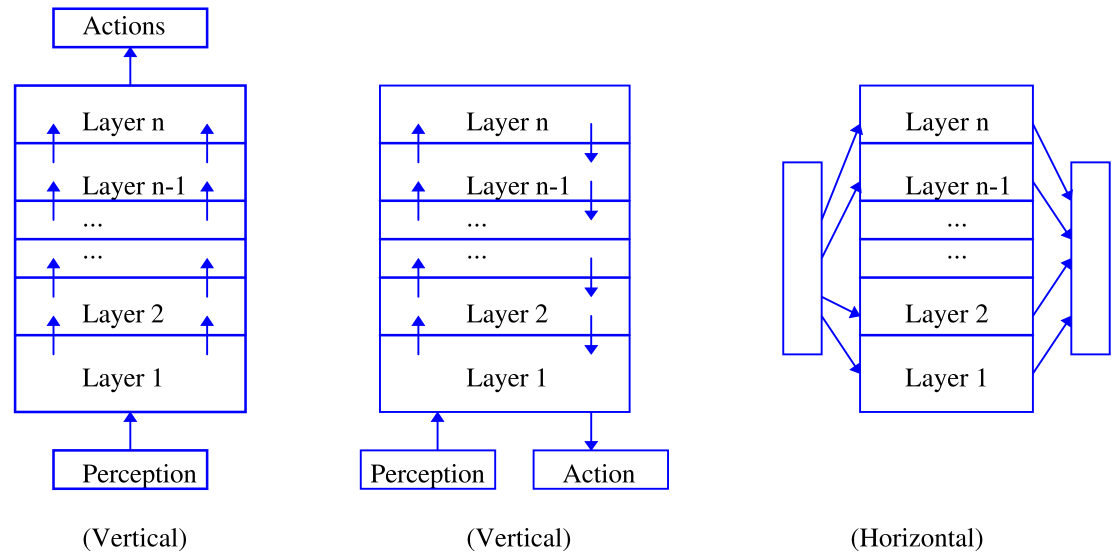


Figure 3 TouringMachine's Control Framework
(Source: Ferguson, 1995)

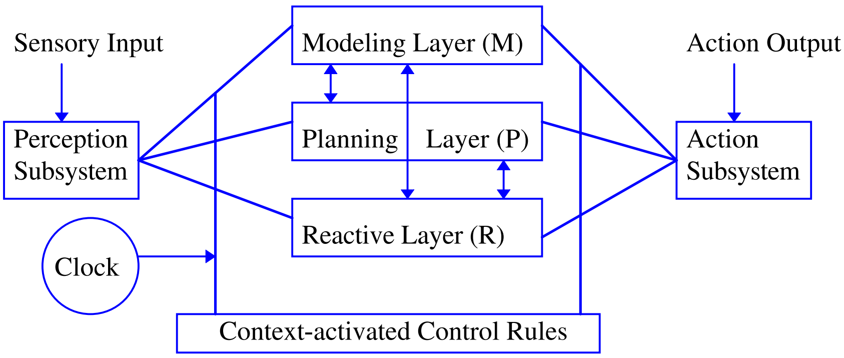
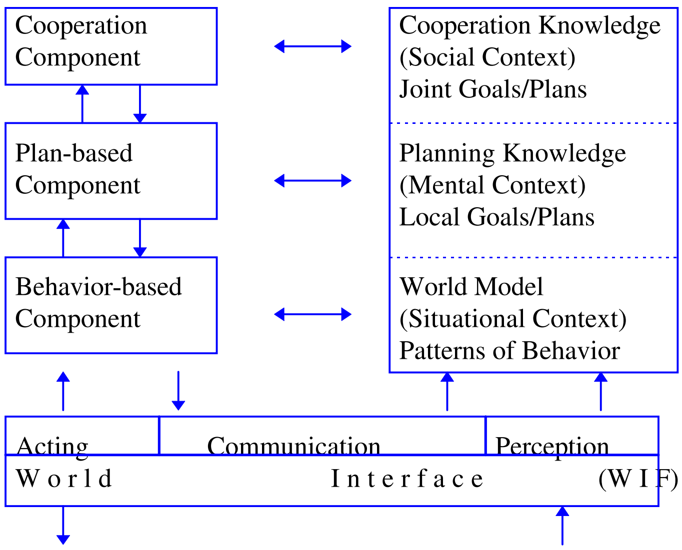


Figure 4 The InteRRap Agent Model
(Source: Muller et al, 1995)



2.3 Research on Agents: the State of the Art

Research on software agents or agents in general has involved various theoretical and practical issues associated with the design and construction of intelligent agents and has been exhibited in three areas: agent theory, language and architecture. Based on the development in these areas, a variety of agent application research are being carried out currently. An extensive survey of the three areas has been conducted by Wooldridge and Jennings (1995). In this section a short introductory review of the theoretical and language research is presented, and special attention is paid to the application researches.

2.3.1 Theoretical Research

The theoretical research in agent field has mainly concerned with agent concept and the developing and use of mathematical formalisms for representing, reasoning, and executing logical agent specifications. Other heavily researched areas include agent architectures, learning techniques, agent models based on the view of an agent as the subject of its mental states, and the formalizing of its attitudes (Clark & Niblett, 1989; Bell, 1995; Quinlan, 1993; Payne & Edwards, 1995; Wooldridge and Jennings, 1995). Significant works have been done in investigating the relationship between an agent's information attitudes and pro-attitudes, its environment and its mental state, as well as its mental attitudes and its actions (Moore, Cohen and Levesque, Rao and Georgeff, Singh, Werner, Wooldridge).

2.3.2 Agent Language and Tool Research

Agent language studies focus on how to program and experiment with agents. Software agents are currently implemented in general programming languages and tools such as C, Java, Prolog, Lisp, Microsoft Visual Basic and Borland's Delphi. At the same time, language researchers in the field have also tried to work out new programming tools that allow one to program computer systems in terms of the concepts developed by agent theorists, and enable the exchange of complex information and goals, commands and scripts. (Genesereth et al; Wooldridge and Jennings, 1995). Their efforts have resulted in quite a number of agent specification and communication languages and tools. A collection of languages and tools for agent building is shown in Table 2. Among these tools, Java, Tcl, Telescript and ACL are very popularly used.

Table 2 Agent Languages and Tools (Sources: Wooldridge and Jennings, 1995; http://www.cs.umbc.edu/agents/)
A. Agent Oriented Programming Languages: (1) AGENT0 by Shoham (1990, 1993) is the first attempt at agent oriented programming, to illustrate AOP principles. In AGENT0, an agent is specified in terms of three types of mentalistic notions: a set of capabilities, a set of initial beliefs and commitments, and a set of commitment rules. (2) PLACA (Planning Communicating Agents) by Thomas (1993), is a refined implementation of AOP principles based on AGENT0. It was intended to address one drawback of AGENT0: the inability of agents to plan. (3) The Concurrent MetateM by Fisher (1994) is again a refinement of both AGENT0 and PLACA. A concurrent MetateM system contains a number of concurrently executing agents, each of which is able to communicate with its peers via asynchronous broadcast message passing. Each agent is programmed by giving it a temporal logic

specification of the behavior that it is intended the agent should exhibit. An agent's specification can be executed directly to generate its behavior (in Wooldridge and Jennings, 1995).

(4) ACL (Agent Communication Language) by the DARPA knowledge sharing effort (Patil et al, 1992), are the best known agent communication language. It consists of three parts: its vocabulary, an inner language called KIF (Knowledge Interchange Format) and an outer language called KQML (Knowledge Query and Manipulation Language). An ACL message is a KQML expression in which the "arguments" are terms or sentences in KIF formed from words in the ACL vocabulary. The efficiency of communication is enhanced by providing a linguistic layer in which context is taken into account, which is the function of KQML (Genesereth and Ketchpel, 1994).

(5) APRIL (Agent Process Interaction Language) and MAIL, by the IMAGINE project at Imperial College, are two languages for developing multi-agent applications. The two languages were intended to fulfill quite different roles. April was designed to provide the core features required to realize most agents architectures and systems, and the generality of it comes at the expense of powerful abstractions. An agent system builder has to implement it from scratch using the primitives of APRIL. In contrast, the MAIL language provides a rich collection of predefined abstractions including plans and multi-agent plans (Haugeneder and Steiner, 1994; Haugeneder et al, 1994; in W&J, 1995).

(6) Telescript by General Magic Inc., is recognized as the first commercial agent language. It is a language based environment for constructing agent societies. It provides a platform that enables the creation of active, distributed network applications using an agent-based paradigm. Telescript technology actually refers to a family of concepts and techniques developed by General Magic to underpin their products (White, 1994). The Telescript Development Environment (TDE) is free and can be downloaded via the web.

(7) AgenTalk, a cooperation effort of NTT Communication Science Laboratories and Ishida Laboratory, Department of Information Science, Kyoto University, is a coordination protocol description language for multi agent systems. AgenTalk allows coordination protocols to be defined incrementally and to be easily customized to suit application domains by incorporating an inheritance mechanism.

8. AgentSpeak is a concurrent agent-oriented language. It can model distributed autonomous agents, situated in dynamic environments, that are reactive as well as proactive towards the environment (AgentSpeak: Weerasooriya et al, 1995).

B. Concurrent Object Languages

Concurrent object languages are in many respects the ancestors of agent languages. The notion of a self-contained, concurrently executing object, with some internal state that is not directly accessible to outside world, and responding to messages from other such objects, is very close to the concept of an agent as we have explained (W&J, 1995).

The Agents Kernel Language (AKL) is a concurrent constraint programming language developed at the Swedish Institute of Computer Science (SICS).

C. Java and Java Extensions:
Java is generally considered to be an important new technology for software agents because of its object-orientation, portability, built-in mobility, and good Internet classes are available, etc.

(1) Java Intelligent Agent Library (commercial) (by Bits & Pixels) provides components for building intelligent agents, implemented entirely in Java. The library is a collection of over 150 Java classes covering various aspects of building intelligent agents. In addition to agent communication and data handling facilities, the library also contains rule-based and neural net based processing modules. There is also an extensive modeling facility for prototyping agent interactions and graphical classes for creating static and animated interactive displays. 7/21/96

(2) Java-To-Go (by William Li) is an experimental infrastructure that assists in the development and experimentation of mobile agents and agent-based applications for itinerative (site-to-site) computing. Sites are usually traversed in sequence by a single mobile agent or in parallel by a group of agents. This infrastructure is based on two components and their interfaces: the mobile agents (Java programs) that actually move to remote sites and performs all the interactions, and the stationary servers that provide the necessary computing resources for the agent to perform their duties. Agents are given the freedom to initiate computations at its volition at one or more remote agent servers. In contrast, standard Java applets can only be invoked passively. 5/11/96

(3) JavaAgent Template! (by Rob Frost at Stanford University) allows one to build KQML-speaking agents in Java. It provides basic agent functionality packaged as a Java application. This agent can be executed as a stand alone application or as an applet via a WWW browser.

(4) Aglets (by IBM) are agents written in Java. Aglets can move from one host on the Internet to another. An aglet that executes on one host can suddenly halt execution, dispatch to a remote host, and resume execution there. When the aglet moves, it brings along its program code as well as its state (data). IBM's Aglets Workbench is claimed a first-of-its-kind visual environment for building network-based applications that use mobile agents to search, access, and manage corporate data and other information.

(5) CyberAgent Software Development Kit (by Ftp Software Inc.) provides numerous agent classes designed to expedite the development of Java-based mobile agents. The CyberAgent classes include templates to create an intelligent agent, start an agent, stop an agent, define a travel plan, allow access to OLE-enabled applications, and support secure agent communications. CyberAgents are autonomous software programs that carry out delegated tasks on a network. Traveling from desktop-to-desktop at the discretion of the sender, these intelligent agents collaborate, execute remotely and report. Password security and access restrictions prevent agents from unauthorized use and travel 5/11/96

D. Tcl/Tk and Tcl Plug-Ins

(1) Tcl/Tk (Tool Command Language / Tool Kit) (by SunLabs), is an Internet scripting language. First it provides most powerful way to build graphical user interfaces. Second it complements Java for scripting and system integration. It is a glue language for making existing applications work together and making it easy to add exciting features to existing applications (Dr. Eric Schmidt, Dr. Bert Sutherland).

(2) The Tcl Plug-in (by SunLabs), is one of the Tcl/Tk applications. It makes it possible to create Web pages that include Tcl/Tk scripts. Before plug-ins, non-html Web content had to be displayed in local viewers outside the Web browser. Tcl Plug-ins are external programs that are loaded into Netscape Navigator on demand to extend its capabilities to display Web content that Navigator itself does not know how to format and render. With Tcl plug-in for Netscape Navigator, the Tcl/Tk scripts can be executed to provide custom user interfaces and simple animation inside the browser. Each plug-in is designed to display content of a different type and enables Navigator to display pages containing elements it would not otherwise know how to format and display. Tcl Plug-ins can be used to create pages that contain compelling applications based on Tcl/Tk and view the applications via Netscape Navigator. The beginning version of the Tcl plug-in runs only with Netscape Navigator. Support for other browsers, such as Microsoft Internet Explorer, has also been planned.

(3) Agent Tcl (by Robert Gray at Dartmouth University) is a transportable agent system in which the agents are written in Tcl 7.4 and Tk 4.0. Agent Tcl is under continuous use at Dartmouth in a range of information-retrieval and information-management applications. It is roughly analogous to Telescript except that it uses Tcl.

2.3.3 Application Research

As we have mentioned earlier, agents have found great potentiality in a broad range of applications and people do seem like to use the agent concept and technology in more and more contexts. The field has attracted much attention from researchers at universities, research centers and consulting firms, as well as a considerable number of companies such as IBM, Microsoft, Xerox PARC, MCC. Prior research on agent applications has produced a large number of prototype systems, and some have resulted in commercial products (Maes, 1994). Overall, the current agent applications are still of a rather experimental and basic nature (Hermans, 1996).

Among the nowadays agent based systems are softwares that can schedule meetings, filter and respond automatically to in-coming emails, filter and search through Usenet news articles or Newswire news on the Internet or WWW to offer personalized news service. Agents are also applied in system and network management, cooperative information systems, industry and control process. Agents for business applications are just emerging in such areas as business process and workflow management as well as electronic commerce. All these form a good basis for more integrated applications, where various single, basic agents are used as the building blocks (Green; Bayer, Edwards; PAAM'96; Turner and Jennings). Here the various kinds of agent systems are organized into five groups: *interface agent applications, information agent applications, Internet and web based applications, management applications, and system and network management applications.*

A. Interface Agent Applications

Research on interface agents has studied learning based adaptive human-computer interactions, collaborative interface agents, and interface agents to the Internet (Maes and Kozierok, 1993; Pattie Maes and Yezdi Lashkari; Pattie Maes, Max Metral et al; Etzioni and Weld; Bayer et al; Hermans, 1996). Interface agents help people in many ways:

(1) An Interface agent acts as its user's personal assistant in dealing with daily administrative tasks. In addition to meeting scheduling, traveling arrangement, email handling, electronic news filtering and selection of entertainment, it can also serves as a reminder or notifier, or an eager assistant (Maes, 1994; IBM). Some people call this kind of agents office automation agents.

Mail agent interface aids a user in sorting incoming electronic mail messages. Features extracted from messages are used by learning algorithms to induce a user profile, which is then used to predict and automate user actions on new messages (for example MAGI by Bayer et al at ABDN). A personal newspaper can include headlines and summaries of articles, examples of those service that are currently being offered include The Wall Street Journal, The Times and IBM's InfoSage (Hermans, 1996).

Firefly by Maes and Metral et al at MIT Media Lab, is a personal software agent capable of communicating with other users and recommending music that it knows its user will enjoy. It learns about the user and leverage the information to best serve the user's needs. It uses the tastes, opinions, preferences and idiosyncrasies of those that are most similar to its user in order to suggest new music that the user might like too. The more the user trains its agent, the more useful and accurate it gets. The more other people use the system, the smarter the Firefly community becomes (Pattie Maes, Max Metral et al.).

(2) Although the user interfaces was transformed by the advent of graphical user interfaces, for many computer remain difficult to use. An interface agent can be attached to such computer applications and acts as a kind of intelligent assistant to users with these systems (Lieber 1995).

Open Sesame! is a commercial software agent that learns the way users work with their Macintosh applications. It uses a learning agent which observes user's activities and learns which tasks are repeated again and again. It then offers to perform those repetitive tasks for the user automatically. Open Sesame! can also automate crucial maintenance tasks the user may easily forget, such as rebuilding the desktop. It learns work patterns and generates instructions that automate tasks. It can automatically perform two or more tasks that the user would otherwise have to perform separately. It uses two types of triggers: time-based and event-based. Time-based triggers will cause it to execute certain instructions at a given time, whereas event-based triggers cause it to execute an instruction in response to a desktop action such as opening a folder, quitting an application, start-up, shutdown and so on (Hermans, 1996)

The softbot based interface to the Ineternet by Etzioni and Weld at the University Washington, is another very good example of such an application. It will be introduced in the third group.

(3) An interface agent can serve as a coordinator in an agent based system. The multiple specialized agents can be combined through it and it can delegate sub-tasks or queries to these specialized agents. The interface agent (called Info Agent) of a multi-agent cooperative information system by D'Aloisi and Giannini (1995) has played this role.

B. Information Agent Applications (Information Access and Management)

Information access and management is an area of great activity for software agents, which is also the major concern of our research. In this field, intelligent agents are supporting users not only with search and filtering of information, but also categorization, attention prioritization, annotation, mining, collaborative sharing and understanding, as well as selective dissemination[4]. There are already consulting firms such as Sageware Inc., Anderson Consulting, SRI and AKA Inc., that provide or prepare themselves for agent based information service. The Sageware Inc. provides commercial information agents, agent components, and the tools to build and maintain these agents. Their customers has included Knight-Ridder, Hewlett-Packard and the U.S. Government.

(1) Search, Filtering and Dissemination Agents

SandPoint's Hoover is a commercial service which provides a single user interface to multiple information media, including real-time newswires, on-line databases, field intelligence, and corporate computing resources. Hoover automatically organizes selected information according to the context of the user's need or function (Hermans, 1996). It offers such services as: *Current Awareness* - an agent delivers two types of current awareness: real-time news and full-text premier publications; *Search* - based on the type of information the user wants, such as information on companies, people, places, and markets, the agent will do the search based on the appropriate context.

[4] Here a distinction between search, filtering and datamining is as follows:
Searching differs from filtering and datamining in that, searching is retrieval, is to acquire certain information upon requests, whereas filtering and datamining are based on general viewing need and information intaking. Both information retrieval and data mining deal with relatively static databases, whereas filtering applies to transiently occurring information on a network. The aim of an filter is to route through to a user those source information deemed relevant to his needs, and sift off those deemed not to be relevant. The user's needs can be expected to change or evolve and the filters are expected to be able to adjust accordingly. Finally, while both search and filtering only capture what is existing, datamining explores patterns and discovers the unknown.

Searching through news feeds and on-line databases in real-time is a further possibility. Then the retrieved information can be updated automatically as often as necessary. Further information-enabled applications of it include decision support systems, corporate intelligence systems, which enable the sharing of volumes of important ideas and observations and being neatly integrated with authoritative external sources (in Hermans, 1996).

The Information Lens research project at MIT's Sloan School of Management, addresses information overload problem in multiple ways. In addition to filtering and retrieval, the Information Lens is also an information-sharing system that help users in properly disseminating information to the right persons.

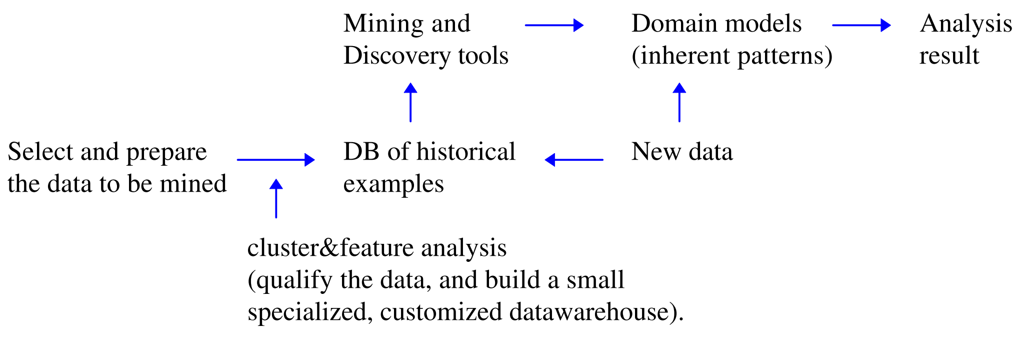
(2) Cooperative Information Agents

Agents that cooperate to locate information sources, access and filter information from distributed heterogeneous databases, seamlessly integrate the information and then present the results to the user has been the central concern for many. The InfoSleuth project of MCC (Microelectronics and Computer Technology Corporation, 1995) is a quite famous one. In addition, the InfoAgent by D'Aloisi and Giannini (1995), CIG Searchbots by UMASS, are also examples of cooperative information agent system.

(3) Agents in Data Mining and Knowledge Discovery

Datamining and Knowledge Discovery were invented to deal with the problems of explosive data volume and complex data structure of corporate information systems. It tends to help by being able to extract useful information, identify interconnections among pieces of information, discover the hidden patterns and relationships from large datasets. The mining results are often used as input for further applications. A datamining and knowledge discovery process is illustrated below:

Datamining and Knowledge Discovery Process



Datamining typically uses a variety of algorithms or their combinations. The idea is that if one algorithm can not find something, another will. Datamining techniques are generally AI based techniques for knowledge discovery or statistical methods based post-data analysis for reporting purpose. Hall (1996) identified seven types of mining and discovery techniques existing in commercial data mining tools: *rule or decision* *tree,* neural network, conventional statistics, advanced visualization, fuzzy logic, knowledge based system and combination of multiple techniques.

The neural net based technique makes natural clustering of data, while conventional statistical data analysis can only group data into canned clusters. While canned clusters forces data into predefined groups, natural clustering derives clusters according to underlying patterns in the data. So, statistics based tools are good at analyzing fixed patterns, and neural network based tools do better in analyzing and modeling nonlinear data or data that are noisy or spotty (local, incomplete) and they do it quickly. Knowledge base system technique is opposed to mining tools that use machine learning techniques. And the knowledge base development is very focused at specific domains. Combined mining tools are functionally abundant. However they are usually also complex environments that are expensive and tend to have a high learning curve (Hall, 1996).

Whichever the algorithm is or whatever combinations they are employed, they are based on four kinds of underlying approaches: *predictive modeling, database segmentation (clustering), link analysis (association discovery and sequence discovery), and deviation detection*. They are defined as follows (Hall, 1996):

> *Predictive modeling:* Predicative modeling in datamining is inductive reasoning while in traditional KBS it is deductive reasoning. It can be implemented via neural networks or induction algorithms.
>
> *Database segmentation:* the automatic partitioning of a database into clusters. There are statistical clustering and natural clustering.
>
> *Link analysis:* identifying connections between records, based on association discovery and sequence discovery.
>
> *Deviation detection:* the detection of and explanation for why records cannot be put into specific segments.

The mining and discovery techniques hold great potentiality in making effective use of corporate data, while it seems that they are not meant for corporate end user though. When the powerful capability of mining and discovery techniques and software agent approach are incorporated with each other, it presents new techniques for data-mining over distributed databases and makes them suitable for mangers' personalized use.

C. Internet or Web based Applications

The Internet and WWW provide attractive testbeds and playground for agent researchers. It hosts the world's largest network of open sources of information. It also present us with an explosion of new computer-based tasks and services, and demand a new style of human-computer interaction. A large number of software agents have been developed for the Internet and web. Here we grouped them into four collections: web searching and filtering agents, web monitoring agents, internet interface and information filtering agents, as well as mobile access and management agents.

(1) Web Searching and Filtering Agents

Right following the information and information resource explosion on the network come the search engines. The MetaCrawler by CSD of the University of Washington is described as a multi-threaded web search service. It sends users' queries to nine different search services at once: Open Text, Alta Vista, Lycos, WebCrawler, InfoSeek, Excite, Inktomi, Yahoo and Galaxy, integrates the results and verifies the hits, and presents them to the user in a unified format.

The homepage finder -- Ahoy!, is a free web service also developed by CSD of the University of Washington. It helps to find the people's home pages on the Web. It uses Metacrawler to create a large list of candidate web pages and then employs various techniques to weed out the least promising ones, presenting a manageable number of choices to the user. It adds filtering into searching.

Resume Robot wanders the web looking for web pages containing resume information. If it identifies a page as highly probable of being a resume, it attempts to extract pertinent information from the page, such as email address, phone number, skills descriptions, and location. The resulting database is used to connect job seekers with recruiters (by James Stakelum).

(2) Web Monitoring Agents

The diffAgent (http://www.cs.umbc.edu/agents/agents.shtml) is a web monitoring system developed by Carnegie Mellon University and Industry.Net Corporation. It continuously watches web sites for specific changes and will send an email to the user when it detects a triggering event. As an example, it can be asked to track a specific package at Fedex and to notify when the package has been delivered, where the triggering event is "Package has been Delivered".

(3) Internet Interface and Information Filtering Agents

There are quite some interface agent based applications on the web. The Webhound from the MIT Media Lab recommends new WWW documents based on a user's past choices. The WebWatcher by David Zabowski and others at Learning Lab of CMU, is a web filtering system which learns the user's preferences and highlights interesting links on web pages that the user visit. It also includes some collaborative filtering functionality, suggesting known pages to the user. WBI (Web Browser Intelligence) by IBM, can remember wherever you've been on the web, what you found there, and can help one recall any word on any page that have been visited. It can alert the user before he goes to a page, whether the site is not available or the access time will be slow and helps you navigate more productively through the web by learning your preferences and patterns for searching for information. WBI can also alert you if a page you have an interest in has had a content change. IBM Activist is a webwatcher who monitors user's web activities and builds a personal profile.

Another group of interface agent applications automatically prioritize and organize emails, pulling messages of interest from common repositories such as newsgroups discussions, mailinglists, newsletters, news services. Examples include IBM Alert Ego, an email messaging agent; IBM Globenet, a newsreader of information from newsgroups, and heterogeneous WAN sources.

The Internet Softbot, again by Etzioni and others at CSD of the University of Washington, is a fully developed, planning based software agent, serves as an integrated interface to a wide variety of Internet services and utilities, including structured information services such as stock quote servers or library databases. A user can make a high-level request indicates what the human wants or wants to accomplish, and the softbot will use search, inference and knowledge to determine how to satisfy the request. A user filled-in form is automatically translated into a softbot goal. Softbot uses semantics based approach and can answer focused queries with relatively high reliability and high relevance. A request does not have to be a complete and exact specification of the human's goal, but a clue or a hint that the softbot attempts to decipher and then satisfy[5]. The agent accepts such incomplete specified goals and searching for missing information whenever possible. It will balance the cost of finding information on its own against the cost of postering the human with questions. The Softbot is also able to alter or refuse a request if it is unethical or dangerous. It is able to limit its use of valuable resources, and it is able to block human actions that have unintended consequences (Etzioni and Weld, 1994; Hermans, 1996).

A similar product, InfoTicker by Erik Mueller, is a web-watching robot with filtering functionality, a real time agent for extracting, from web pages and Usenet groups, stock market quotes, weather information, classified advertisement, and so on (in Hermans, 1996).

The Amalthaea by MIT Media Lab (Pattie Maes and Alexandros Moukas) is a multi-agent system that discovers, monitors and filters information and information resources distributed among heterogeneous sources on the web. The system consists of a society of smaller and simpler specialized agents that try to solve the problem collectively. Each filtering agent acts as a very specialized filter that is applied only in a narrow sector of the user interests domain, while each discovery agent does the same in the information sources domain proactively exploring new domains that may be of interest to the user.

The group of news filtering agents is big. For example, the *InfoScan:* filters email and USENET news and includes document visualization features; *InfoSage:* a news service from IBM, filters newswire articles based on an explicit profile constructed by the user. *SIFT (Stanford Information Filtering Tool)* by Stanford University, searches a 6+ month archive of more than 13,000 news-groups and a large (>1000 lists) archive of publicly-accessible mailing lists. *INFOSCOPE* agents aid to create personalized

[5] In general, the softbot's goal language allows the human to state three kind of goals: ground goals, existential quantified goals (e.g. some), and constrained goals (e.g. the location of a computer, ...). And they found that constrained goals strike a useful balance between burdening the user with endless questions and sending the softbot on a massive search of the Internet.

structure in USENET news and assist users by suggesting filters based on observed interest patterns (Stevens, 1992b); *Synview:* by David Lowe of the University of British Columbia, is a collaborative USENET news filter; *UNA:* by Bayer et al at ABDN, is also a USENET News Agent that helps a user by identifying interesting USENET news article -- the user rates articles on a scale of 1 (dull) to 6 (very interesting) and features extracted from articles are used to generate a user profile, then new articles are filtered by the agent based on the predicted level of user interest; *AARON:* also by Bayer et al, is for automatic retrieval of on-line newspapers. It constructs a personalized electronic newspaper by gathering stories from various Web-based newspapers. The personalized newspaper has three sections: Headlines (extracted based on analysis of frequency of stories occurring in on-line newspapers), Interest, and Gazette (a collection of random stories). The Smart NewsReader by Intel is a similar Windows application that provides access to Usenet newsgroups. It can read through the articles and score each thread of conversation based on user's past interests. The articles are then sorted using this score so the most interesting articles stay at the top of the list. As the user reads articles he tells the agent which articles interest him and which articles he found boring. In 50 or more feedback articles, the agent will be able to converge on user's particular interests.

(4) Mobile access and management

This is another focused area of IBM's agent research. As computing becomes more pervasive and network centric computing shifts the focus from the desktop to the network, users want to be more mobile. Intelligent agents reside in the network rather than on the users' personal computers, can persistently carry out user requests despite network disturbances. They process data at its source and ship only compressed answers to the user, rather than overwhelming the network with large amounts of unprocessed data (IBM).

D. Agents in Business Management

Agent applications in business management is just starting and is very limitedly reported in the literature. However the potentially promising areas for agent applications can be seen in Internet based financial and marketing information services, credit management, electronic commerce agents that help users search and autonomously monitor new product announcements in the industrial domain; agents for financial decision making support; agent application in executive information systems and decision support systems in general. The Intelligent Agents for Financial Portfolio Management project at Carnegie Mellon University (http://www.cs.umbc.edu/agents/agents.shtml) is applying an agent framework to portfolio management. The idea is to provide an integrated financial picture for managing an investment portfolio over time, using the information resources available over the Internet.

In conducting electronic commerce, buyers need to find sellers of products and services, they need to find product information, or obtain expert advice both prior to the purchase and for service and support afterward. Sellers need to find buyers and they need to provide expert advice about their product or service as well as customer service and support. Both buyers and sellers need to automate handling of their transactions. Intelligent agents can assist in electronic commerce in a number of ways. Agents can "go shopping" for a user, taking specifications and returning with recommendations of purchases which meet those specifications. They can act as "salespeople" for sellers by providing product or service sales advice, and they can help troubleshoot customer problems. The Softbots group at the University of Washington has developed an experimental prototype shopping agent on the web: ShopBot. It helps to shop for computer software or CDs. The user fills in the name and manufacturer of the software product that he is looking for, or the title and artist of the CD, and then lets the ShopBot go to work.

Workflow management is another field of great interest where a workflow is a system of activities related to one another by a trigger relation and triggered by external events. It represents a business process starting with a commitment and ending with the termination of that commitment. Workflow Management (WFM) is the computer assisted management of business processes through the execution of software whose order of execution is controlled by a computerized representation of the business processes. Help is needed to build and manage collaborative teams of people, and manage their work products. Intelligent agents can be used to ascertain, then automate user wishes or business processes. The IBM's Virtual "Office buildings" house agencies and agents that can pursue a variety of goals.

E. Systems and Network Management

Systems and network management is one of the earliest application areas to be enhanced using agent technology. They can help filter and take automatic actions at a higher level of abstraction, and can even be used to detect and react to patterns in system behavior. Furthermore, they can be used to manage large configurations dynamically.

3. Strategic Scanning and Interpretation Support: Agents and Their Task Analysis

Delegation of tasks and functions has been desired and expected as a response to the problems of work overload and information overload. Agents present one of the approaches and technologies for such delegation. However, the study of El Sawy (1985) has shown that, for perceptual filtering reasons, executives do not and will not delegate the scanning and interpretation activities. This is because that when the work is delegated to other people, the peculiar perspectives of executives on certain events or changes are usually lost. *So, can executives delegate their scanning and interpretation tasks to their software agents? What an agents system can do for them and what would be better kept by themselves?*

With an agent system, tasks will be carried out on behalf of the managers whether they are present or absent, with guidance rather than direct control by the managers. Yet, with such a system, we would expect more. We expect not only that it can take over certain responsibilities from the managers, but it can also incorporate managers' perspectives in fulfilling these responsibilities. For such a system, what agents are needed and how they can work together as an integrated support system to serve the managers? These form the focus of this and the next section.

Agents and the properties they should include and the behavior they exhibit, depend very much on the context they will be in. The strategic scanning and interpretation process has been shown to be a continuous seeing, perceiving, understanding and conceptual assembling process, and to be an integral part of a strategic issue management process. It encompasses at least three scanning modes: surveillance, tracking, and probing (El Sawy and Pauchant, 1988). Surveillance is an exploratory mode of scanning that is not directed to a specific problem; tracking involves bounding the environmental scope by limiting the number of issues that are being monitored; and probing involves scanning with diagnostic intention in response to a signal indicating a specific threat or opportunity (Huber, 1984; El Sawy and Pauchant, 1988). It is this scanning process with all its various modes which drives the perception of the evolution of strategic issues; and the evolution of strategic issues also directs the scanning and interpretation process. This scanning and interpretation process involves continuous engaging in three types of activities: *locating and choosing information sources, scanning and interpretation of relevant information, diagnosis of strategic issues and managing of the strategic issue agenda.* Correspondingly we identify three types of agents here: the Information Resource Agent, the Scanning and Interpretation Agents, and the Interface Agent.

3.1 Information Sources: the Information Resource Agent

Managers rely on many sources, formal and informal, for the information they need and the interpretation of certain information they receive. Some of the sources to be used or the accessibility of the sources are likely unknown at the time a support system is developed and might vary over time. For example, it was only until a few years ago that most informal information was embedded in type-written documents, hand-written notes, reference books, and people's minds, while not in machine readable form. As more and more information, even informal information is more likely to be accessible directly on line, managers can expect to be able to directly access more through one common media. As the number of potential electronic sources of information, local or remote, formal or informal, is increased rapidly, effective scanning activity become closely related to effective managing of these sources.

The Information Resource Agent can make an inventory of the kinds of services and information that are being offered by certain sources. At the same time, it watches and learns about the changes of the information sources, add new ones to and delete obsolete one from the system. It can also maintain the different sources as an information food chain, for example, distinguishing sources from where to identify what is the relevant information (high-end of the chain, e.g. consulting sources) and sources from where you get the information (low-end of the chain, e.g. data sources). The Information Resources Agents suggests carefully chosen sources to the scanning and interpretation agents or to the user directly, to inform which data sources to check for what information, and to help to make the appropriate contacts and asking the right questions. In summary, it is useful in the following ways:

(1) information advertising: to inform the user, and scanning and interpretation agents, that information is available by certain sources of information;

(2) to seek new information sources actively and to inform the user, and scanning and interpretation agents of useful sources and source features;

(3) help with user activated information gathering for specific purpose: when the user want to collect certain information directly from some sources, the agent can send out the predefined or user profile directed questionnaires.

3.2 User-System Interaction and Strategic Issue Agenda: the Interface Agent

As we have described, to the managers the scanning and interpretation system is more a tool by which they capture strategic issues, track the issues and discuss with other managers about the issues. So, the interface agent will act as both an intermediary between the user and the task agents and a front-end for managing strategic issue agenda. It will take the following responsibilities:

(1) The Interface Agent acts as a personal assistant in such applications as email filtering, news filtering, meeting scheduling and traveling arrangement. The basis of these functions is the agent's ability to handle a user's personalized needs, which requires the agent be able to assist the user in compiling his profile, or be able to induce the user profile by keeping track of the evolution of his interests and preferences in accomplishing these tasks. The agent is then able to reason about the user's requests and to understand what type of need he is expressing according to the current state of affairs and the dynamic user model. One user can set many different profiles, with each reflecting one specific interesting area.

(2) The Interface Agent presents to the user the methods of operation and activities of task agents in an appropriate level, so that the user can understand easily the agents' behavior at a conceptual level, and do not have to involve in too much detail unnecessarily. The interface agents also ensures to present the retrieval information or interpretation results to the user in a unified way that can be managed very easily for displaying or printing.

(3) The interface agent, as a query manager and the highest level facilitator of the system, is responsible for translating the user's request or strategic issue profiles, developing an initial high-level scanning and interpretation plan, select the agents or agent-groups that are able to accomplish the task, and coordinate their (domain specific facilitators) activities. It takes care of the user specific of the system so that the system can be updated when needed only by modifying itself. In this way the system structure is kept flexible.

(4) The interface agent will communicate with other managers' interface agent to share certain information, call for multiple interpretation, and participate in the group discussion of the strategic issues. This could be seen as a more advanced feature of it.

(5) Manager of the Strategic Issue Agenda

One of the most important features we have expected from the system is its ability to incorporate managers' perspectives in scanning and interpretation of the accessed information, and in identifying and managing the strategic issues. The interface agent takes such a responsibility, as a manager of the strategic issue agenda.

El Sawy and Pauchant (1988) have showed that strategic issues can be operationalized through the concepts of issue-templates, template-triggers and template-twitches. Templates are frames of reference through which a particular issue domain is perceived. Triggers are stimuli which impinge upon templates and which might cause them to shift. Twitches are template modification (a short movement with a sudden motion) caused by a trigger (El Sawy and Pauchant, 1988). They emphasized the continuous dynamic interplay between scanning and interpretation of environmental signals, and cognitive frames of reference and assumptions. On one hand, the frames of reference determine how to perceive the data and how to understand the data. On the other hand, as people scan, the new information and interpretation they acquire also tend to change their cognitive frames of reference, which in turn affect what they will perceive in their environment and how they will scan. With frames they perceive the world and with the perception they change the frames[6]. They also stressed that, the template-twitches are often more informative than the template-triggers. This forms a general rule that directs issue based probing and tracking.

As issues evolve dynamically and they can be viewed as going through a lief cycle, which is practiced through the notion of strategic issue agenda (El Sawy and Pauchant, 1988). El Sawy and Pauchant described the issue life cycle as four stages: *discovery, emergence, maturity and fading,* which is interpreted as: “specific strategic opportunities and problems matures through a strategic issue, the issue can be viewed as passing from emergence into maturity”. Re-discovery and re-emergence are possible and an issue can have “many lives” and is often reincarnated. Depending on the stages in the issue lifecycle, it calls for a design that includes a large issue structuring component and repository support services. As issue deliberations are stored for future access or as issues fade, they need to be cataloged into some form of repository that reorganizes them appropriately for useful recollection. There is also an need to track attention indicators longitudinally across the life of an issue and to signal any lifecycle shifts. As is indicated in Figure 3, to identify a strategic issue attention needs to be paid

[6] The shift of frames of reference is seen as being generated through the perception of new environmental information or the occuring of new learning and ideas. El Sawy and Pauchant (1988) describe four ways for the change of frames of reference: (1) *assimilation:* where new information is assimilated into old frames; (2) *accomodation:* where old frames are modified because the new data do not fit the old catogories; (3) the *formation* of new frames; (4) *tension and conflicting processes* between competing frames are seen as underlying the dynamics of frame shifting.

to the capturing of triggers or leading indicators. After an issue is detected, the attention is shifted to the continuous reformulation of templates and interpretation of the template twitches. When sunset indicators are detected, the issue start fading or get solved. In this way, a systematic scanning and interpretation support system actually makes it possible to capture and organize the key inferences, assumptions, and indicators during the strategic issue managing cycles for future access for the benefits of cumulative learning across participants.

The strategic issue agenda (SIA) is in fact a means for maintaining a set of dynamic strategic issue profiles. Issues can be structured in terms of, for example, their scope, urgency, criticality, templates, triggers, and twitches of frames of reference, lifecycle stage, stakeholders, emergent objectives, scenario analysis, and so on. As the manager of SIA, the interface agent need to understand and reason about the user's mental models during a continuous SSI process, translate this model into queries to the scanning and interpretation agent. It is also expected to manage the issue dissemination so that specific issues reaches those people to whom they are of concern without interfering with those to whom they are not.

Figure 3 The Strategic Issue Lifecycle
(Source: El Sawy and Pauchant, 1988)

Issue Lifecycle	Issue Discovery →	Issue Emergence →	Issue Maturity →	Issue Fading
Issue Based for **Requisite** organizational **Functionality**	Trigger Capture	Template& Twitch definition	Providing squeezing heuristics	Cataloging memory
Attention Indicators for Envelop (issue)	Leading indicators	Critical indicators	→	Sunset indicators
Domain Design and Delivery Activities	→	Structuring the issue → defining information support services	Restructuring and managing issue scope	defining repository support services
		defining decision support services		

3.3 The SSI Process: Scanning and Interpretation Agents

The scanning and interpretation agents will conduct the surveillance and probing, or tracking tasks on behalf of the managers. For example, they may work together as gatekeepers and commentors for large social and cultural context, regulatory and legal environments, technological developments, economic situation, competition, organization resources and so on. Just as individual people have different specializations, classes of agents with different specializations are needed in the scanning and interpretation support system. Here we introduced four kinds of task agents: *scanning agents, interpretation agents, datamining and knowledge discovery agents, and coordinating agents.* They are tasks specific while not necessarily domain specific, which means that one scanning agent can contribute to the scanning of both the regulatory environment and the technological environment, when directed by different frames of reference and access to different sources of information.

(1) Scanning Agents: A scanning agent is responsible for finding the right information. It scans, filters and advertises incoming information. It searches and retrieves required information from multiple sources. It probes around specific signals, and tracks specific issues. A scanning agent's activity may also involve aggregating, joining or abstracting of information to produce an appropriate result for interpretation. A scanning agent keeps alertly watchful for the weak signals indicating pending threats and opportunities. It also captures the template-triggers. Central to a scanning agent are its sensor and filter. The sensor continuously identifies new information and the filter ensures selective attention. When passing through the filter, only relevant articles are presented while irrelevant contributions are screened out.

(2) Datamining and Knowledge Discovery Agents (DKDA): the mining and discovery agent can remain active in the computational environment, working on the accessed datasets to discover underlying patterns, predict changing trends, create domain models and derive unknown facts. It helps to examine the assumptions underlying certain information and fresh the frames of reference.

(3) Interpretation Agents: An interpretation agent is responsible for giving the right comment or explanation about the implications of the accessed information. It is meaning developing and impact analysis. The interpretation agent works cooperatively with scanning agent and mining agent to make sense of the scanned and searched information. Template-twitches are important information for its examination.

(4) Coordinating Agents -- the Facilitators: The Facilitators are domain-specific coordinating agents. They are further coordinated by the Interface agent to get the best possible general picture of the business environment, and to identify and manage strategic issues. A Facilitator agent coordinates the scanning, mining, and interpretation activities. It enables a group of agents works cooperatively under certain constraints of time and resource boundedness, supervises task agents' refinement, maintains the

subsystem coherence, and ensures that the task agents will succeed in attaining their common goals.

4. Scanning and Interpretation of Strategic Data: A Multi-Agent Support System

4.1 System Requirements and Design Considerations

As a support tool for the scanning and interpretation activity of managers, the system (MASCAN) should provide a manager with seamless interaction with remote information resources (including human resources), expedite the process of searching for pertinent information in a distributed and constantly growing network of information resources, and dynamically match the needs and sources in the best possible way. The systems has a cooperative and supportive role. The goal is to disengage the user from learning complex tools and from performing tedious actions.

As have been mentioned earlier, the very nature and properties of agents incurs susceptible concerns on their practical applications. The situation calls for some special considerations in design of software agents or agent based systems.

Trust between users and their agents has been cited as one of the major problems facing the development of agents. On one hand, not trusting when trust is warranted leads to excessive monitoring of or non-use of intelligent agents. On the other hand, trusting when trust is unwarranted holds the potential danger (Slyke and Collins).

Trust comes slowly. According to Norman (1994) and Slyke and Collins, trust comes on the basis of competence, consistency and predictability of agent performance on frequent tasks. It is also based on user's feeling of control, and feeling of protection in safety and privacy. So one of the critical success factors of agent system would be that, it can allow users to assume that agents are reliable and predictable and the human user remains in ultimate control (Kautz et al, 1994). Based on these concerns, the design of an agent based system needs to pay great attention to the following issues:

(1) One way for the user to be conformable with the actions performed for them by automata is through a feeling of understanding. How the agent system will interact with the user become the key here. Norman (1994) suggested that, *"The best way is to develop an appropriate conceptual model of the underlying actions. In such a model, the actions of agents are understood in context so users feel comfortable in knowing that they can find out what actions have been taken on their behalf, private matters remain private, expensive or unwanted actions will not be taken without explicit permission, and it is always possible to trace back to the actual sequence of acts and undo any that are seen as unwarranted."* The design has to decide properly how much of the agent's actions should be visible to the user (Slyke and Collins).

(2) The second way for the user to accept an agent system is through confidence in the system which is related with system competence, performance consistency, as well as the difference between what is claimed and what is offered. Norman argues that people

will be more accepting of intelligent agents if their expectations are consistent with reality. Minimizing overblown expectations and false hopes should be another important consideration.

(3) The safety concerns calls for built-in safeguards to prevent runaway computation; to make sure that the agent does not do things that would jeopardize the physical, mental, or monetary well-being of the owner; and to confirm that the information is true not false. Safety plays a part in feeling of control (Norman, 1994).

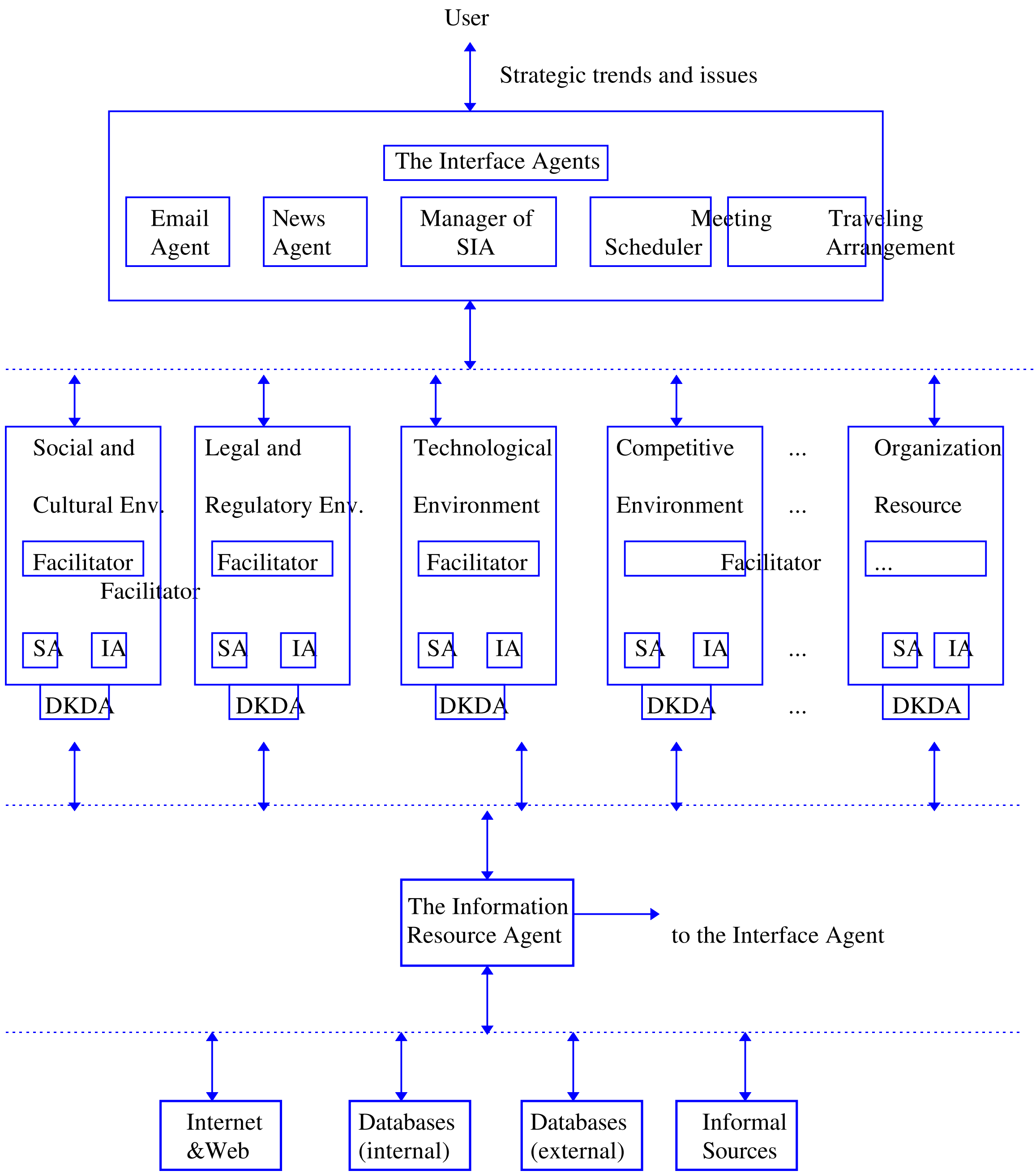
(4) Privacy concerns are also a subset of the feeling of control problem. The privacy issue may be disturbing to many users, no matter how helpful the agents might be (Norman, 1994).

(5) Other concerns may include: effectively handling of textual and document-based information sources (El Sherif and El Sawy, 1988; Frank, 1990; AKA Inc.), the need for constraining the design so that it is realizable by simple and fast hardware (Malec, 1995) and for being less expensive.

4.2 MASCAN: the Structure

In this section we present a principal structure for a Multi-Agent Based Support System for Strategic Scanning and Interpretation (MASCAN). It is designed as a concurrently executing agent system consisting of a collection of task-agents and domain specific coordinating agents, an information resource agent and an interface agent. To the user, it is a personal assistant and a strategic issue managing front-end integrated with a dynamic executing back-end. For example, upon logging into the system, the manager may be presented with a list of email messages sorted according to the order of importance (Email Agent); a list of news articles, with the system draws his attention to a particular one (News Agent); a list of strategic issues in a prioritized sequence, with the system alerts him newly detected signals and presents alternative interpretations (by Scanning and Interpretation Agents). The structure provides an extensible platform from which a new domain specific agent group can be added to the interface agent without affecting the existing agent groups. This new group of agents can also make use of the basic task agents (scanning, mining and interpretation agents) already in the system.

Figure 4 MASCAN: the Structure



The system takes a layered federated structure (Genesereth and Ketchpel, 1994). By this structure, the various agents are structured as groups coordinated by their facilitators. They do not communicate directly with each other. Instead, they communicate only with their facilitators. Facilitators communicate with each other or with a higher level facilitator. Agents document their needs and capabilities for their local facilitators. In addition to such metalevel information, they also send application level information and requests to their facilitators and accepts application level information and requests in return (Genesereth and Ketchpel, 1994). In such a system, a group of agents form a federation in which they surrender part of their autonomy to their facilitators and the facilitators take the responsibility for fulfilling their needs.

The federated structure facilitates communication within multi-agent systems (Genesereth and Ketchpel, 1994; Franklin, 1996). The two approaches for agent coordination has been *direct communication (point-to-point message-passing) and broadcast communication (point-to-many).* In point-to-point message-passing, messages are sent to a specific address (the receiver) which must be known by the sender. Cost and complexity are two main problems with this approach. It is widely used in concurrent object based systems. The point-to-many message-passing involves sending a message, not to a specific address but to all agents at once. It is becoming more and more widely used in distributed computer systems. A third alternative is to structure the agent space into groups, with each agent as a member of at least one group. When an agent broadcasts a message, it will be sent to all members of one or more of the agent groups. In this approach, full broadcast message-passing is replaced by *multicast message-passing* (Fisher, 1995), which is implemented as a federated system structure.

5. Discussions and Conclusions

Agent technology seems a potential powerful tool to make great contributions to computerized management support applications. So, how agent based systems are similar with and different from others?

Comparing with conventional DSS/EIS/OLAP, an agent based system can support continuous surveillance of environment while the others do not (traditional EIS monitors but do not scan actively; data is mostly aggregated but not interpreted). Conventional DSS/EIS/OLAP applications are usually based on existing databases and predefined or preassumed data needs, while an agent based system can incorporate varying sources of information, dealing with changing information needs and provides current awareness service to inform users regularly about information changes. It is based on its combined reactive and proactive behavior that an agent based system provide such kind of active support.

An agent based system is sometimes seen as a particular type of knowledge based systems. However, while traditional knowledge based approach is based on knowledge acquisition through knowledge engineer, the agent approach is more based on another AI technique: machine learning. In this aspect, it takes an approach similar to datamining and knowledge discovery. This makes it more adaptable in contrast to traditional KBS.

An agent based system also implies that it is a distributed or collaborative system, not standalone KBS. To some, this is seen as a paradigm shift for building intelligent systems: from grappling with problems of knowledge representation and acquisition, to establishing distributed, not particularly, intelligent systems, and hope that intelligence will emerge in effective collaboration. The knowledge acquisition problem gets solved by being declared to be a ‘non-problem’. Learning and communicative become a must for agent based systems (Hermans, 1996). The collaborativity is also what agents differ from search engines as the latter usually do not cooperate and cause the waste of a lot of time and bandwidth by working in isolation.

In this paper we identified three types of software agents needed in constructing an agent based strategic scanning and interpretation support system: the Resource Agent, the Interface Agents and Strategic Issue Agenda Manager, and Scanning and Interpretation Agents. We draw from existing research findings some useful guidelines for the design of a multi agent system and come up with a principal system framework, which takes a layered, federated, hybrid architecture.

Agents seem to possess the properties that are exactly needed by SSI support systems, however the real answer has to be found through practice. The next step work will involve the building of the various specialized agents we defined here, which will in turn involves the defining and choosing of a set of attributes and properties of these agents. We will need to find out what will be the core concept or attributes of the various agents in assisting scanning and interpretation of strategic data. For this purpose it is very necessary for us to first obtain practical experience in building and using some concrete, basic agents before committing to any particular theoretical propositions. We also need to define what kind of mentalistic models would be appropriate to describe the agents in our system.

References:

Aitken J. S., F. Schmalhofer and N. Shadbolt, “A Knowledge Level Characterization of Multi-Agent Systems”, in Wooldridge M. J. and N.R. Jennings (eds.), Intelligent Agents - Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, August 8-9, Springer-verlag, 1995, pp. 179-190

Appleton E., “Smart Document Retrieval”, Datamation, pp. 20-23, January 15, 1992

Artificial Intelligence, 1995, Special Issue on Agents and Interactivity

Baclace P. E., “Competitive Agents for Information Filtering”, Communications of the ACM 35(12) 50, December 1992

Bayer D. P., C. L. Green, R. Mackenzie, T. R. Payne and P. Edwards, “Learning Agents for Information Filtering/Gathering”, (http://www.csd.abdn.ac.uk/~pedwards/res/filter.html)

Bell J., “Changing Attitudes”, in Wooldridge M. J. and N.R. Jennings (eds.), Intelligent Agents - Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, August 8-9, Springer-Verlag, 1995, pp. 40-54

Bird S. D., “Toward A Taxonomy of Multi-Agent Systems”, International Journal of Man-Machine Studies, Vol. 39, pp. 689-704, 1993

Boden M. A., “Agent and Creativity”, Communications of the ACM, Vol. 17, No. 7, 1994, pp. 117-121

Bratman M. E., “What is Intention”, in Cohen P. R., J. L. Morgan and M. E. Pollack (eds), Intentions in Communication, pp. 15-32, The MIT Press, 1990;

Brooks R. A., “A Robust Layered Control System for A Mobile Robot”, IEEE Journal of Robotics and Automation, Vol. 2, No. 1, pp. 14-23, 1986;

Brooks R. A., “Intelligence without Reason”, Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91), pp. 569-595, Sydney, Australia, 1991;

Brooks R. A., “Intelligence without Representation”, Artificial Intelligence, Vol. 47, pp. 139-159, 1991;

Brown C., L. Gasser, D. O’Leary and A. Sangster, “AI on the WWW: Supply and Demand Agents”, IEEE Expert, Vol. 10, No. 4, August 1995, pp. 50-55

Castelfranchi C., “Guarantees for Autonomy in Cognitive Agent Architecture”, in Wooldridge M. J. and N.R. Jennings (eds.), Intelligent Agents - Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, August 8-9, Springer-Verlag, 1995, pp. 56-70

Cheryl Gerber, “Excavate Your Data”, Datamation, May 1996.

Cohen P. And Levesque, “Intention is Choice with Commitment”, Artificial Intelligence, Vol. 42, pp. 213-261, 1990

Crowston K. and T. W. Malone, “Intelligent Software Agents”, Byte (December1988) 267-271

D’Aloisi D. And V. Giannini, “The Info Agent: An Interface for Supporting Users in Intelligent Retrieval”, November 1995, (http://www.cilea.it/nir-it-95/giannini/giannini/giannini-nir-95.html)

Davies W. H. E., T. R. Payne and P. Edwards, “Novel Learning Techniques for Agent Systems”

Daviss B., “Knowbots”, Discover, pp. 21-23, April 1990

Decker K., V. Lesser, M. V. N. Prasad and T. Wagner, “Cooperative Multi-Agent Information Gathering”, available at: http://dis.cs.umass.edu/research/cig/current-status.html

Dongha P., “Toward a Formal Model of Commitment for Resource Bounded Agents”; in Wooldridge M. J. and N.R. Jennings (eds.), Intelligent Agents - Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, August 8-9, Springer-Verlag, 1995, pp. 86-101

El Sawy O. And T. C. Pauchant, “Triggers, Templates and Twitches in the Tracking of Emerging Strategic Issues”, Strategic Management Journal, Vol. 9, pp. 455-473, 1988

Elofson G. and Konsysnki B., “Supporting Knowledge Sharing in Environmental Scanning”, HICSS-23 Preceedings, Vol. 4, pp. 281-288, 1990

Etzioni O. and D. S. Weld, “Intelligent Agents on the Internet - Fact, Fiction, and Forecast”, IEEE Expert - Intelligent Internet Services, No. 4, page 44-49, August 1995

Etzioni O. and D. Weld, “A Softbot-Based Interface to the Internet”, Communications of the ACM 37 (72-76), July 1994

Ferguson I. A., “Integrated Control and Coordinated Behavior: A Case for Agent Models”, in Wooldridge M. J. and N.R. Jennings (eds.), Intelligent Agents - Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, August 8-9, Springer-verlag, 1995, pp. 203- 218

Fields B, M. Harrison and P. Wright, “From Information Requirements to Agent-Based Specification”, SIGCHI Bulletin 26(2), 65-68, April 1994

Fisher M., “Representing and Executing Agent-based Systems”, in Wooldridge M. J. and N.R. Jennings (eds.), Intelligent Agents - Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, August 8-9, Springer-verlag, 1995, pp. 307-323

Frank M., “Strategies for Intelligent Agent Exploration of Complex Environments”, Term paper, March 1990. (http://medg.lcs.mit.edu/mpf/papers/Frank/Frank-90a.html.)

Franklin S. and Graesser, “Is It An Agent, or Just A Program?: A Taxonomy for Autonomous Agents”, Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.

Genesereth M. R. and S. P. Ketchpel, “Software Agents”, Communications of the ACM, Vol. 37, No. 7, pp. 48-53, 1994

Gilbert D. et. al, White paper on intelligent agents, IBM Corporation, Research Triangle Park, NC.

Guha R. V. And D. B. Lenat, “Enabling Agents to Work Together”, Communications of the ACM, Vol. 37, No. 7, pp. 127-142, 1994

Hall C., “Intelligent Data Mining Tools”, Intelligent Software Strategies, Cutter Information Corporation, July 1996

Hardy D. R., M. F. Schwartz and D. Wessels, “Harvest: Effective Use of Internet Information”, Technical Report CU-CS-743-94, University of Colorado at Bulder, April 1995.

Harman D., “Relevance Feedback Revisited”, in N. Belkin, P. Ingwersen and M. Pejtersen (eds.), Proceedings of the 15th Annual International Conference on Research and Development in Information Retrieval. ACM Press, 1992, 1-10.

Harmon P., “Software Agents”, Intelligent Software Strategies, January 1995

Hayes-Roth B., "An integrated architecture for intelligent agents," SIGART Bulletin 2, 1991, 79-81.

Hayes-Roth B., "Evaluation of integrated agent architectures," SIGART Bulletin 2, 1991, 82-84.

Hayes-Roth, B. (1995). "An Architecture for Adaptive Intelligent Systems," Artificial Intelligence: Special Issue on Agents and Interactivity, 72, 329-365

Hermans B., Intelligent Software Agents on the Internet: An Inventory of Currently Offered Functionality in the Information Society and A Predication of (Near-)Future Developments, Thesis of Tilberg University, Tilberg, The Netherlands, July 1996

Jackson S. and J. Dutton, “Discerning Threats and Opportunities”, Administrative Science Quarterly, Vol. 33, No. 3, pp. 370-387, September 1988

Jaron Lanier, “Agents of Alienation”, (http://www.voyagerco.com/consider/agents/jaron.html)

Jennings N. R., "Commitments and Conventions: the Foundation of Coordination in Multi-Agent Systems", Knowledge Engineering Review, Vol. 8, No. 23, pp. 223-250, 1993

Jennings N. R., "Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems Using Joint Intentions", Artificial Intelligence, Vol. 74, No. 2, 1995;

Jennings N. R., "Specification and Implementation of a Belief Desire Joint-Architecture for Collaborative Problem Solving", Journal of Intelligent and Cooperative Information Systems, Vol. 2, No. 3, pp. 289-318, 1993;

Jennings N. R., Varga L. Z., Aarnts R. P., Fuchs J. and P. Skarek, "Transforming Standalone Expert Systems into a Community of Cooperating Agents", International Journal of Engineering Applications of Artificial Intelligence, Vol. 6, No. 4, pp. 317-331, 1993

Kaplan S. J., M. Kapor, E. Belove, Landsman R. and T. Drake, "Agenda: A Personal Information Manager", CACM, Vol. 33, pp. 105-116, July 1990

Kimbrough S. And S. Moore, "Message Management Systems: Concepts and Motivations", Proceedings of HICSS-25, Vol. 4, pp. 654-665, 1992

King J. A., "Intelligent Agents: Bring Good Things to Life", AI Expert, Feb.&Mar., 1995

King W. R., "Strategic Issue Management", in King W. R. and D. I. Cleland (eds.), Strategic Planning and Management Handbook, VanNostrand Reinhold Cor., New York, 1987, pp. 252-264

King W., "Integrating Strategic Issues into Strategic Management", OMEGA, Vol. 12, No. 6, pp. 529-538, 1984

King W., "Using Strategic Analysis", Long Range Planning, Vol. 15, No. 4, pp. 45-49, August 1982

Konstadt P., "Information Sources", CIO, pp. 73-80, November 1990

Kozierok R. and P. Maes, "A Learning Interface for Scheduling Meetings", ACM SIGCHI, 1993

Lashkari Y., M. Metral and P. Maes, "Collaborative Interface Agents", Proceedings of the 12th National Conference on Artificial Intelligence (AAAI 94), Seattle (WA), 1994, 444-449.

Maes P., "Agents that Reduce Work and Information Overload", Communications of the ACM, Vol. 71, No. 7, pp. 31-40, July 1994;

Maes P., "Situated Agents Can Have Goals", Robotics and Autonomous Systems, Vol. 6, No. 49-70, 1990

Maes P., "The Agent Network Architecture (ANA)", SIGART Bulletin, Vol. 2, No. 4, pp. 115-120, 1991;
Maes P., Designing Autonomous Agents, Cambridge, MA, MIT Press, 1990

Malone T. W., K. R. Grant, F. A. Turbak, S. A. Brobst and M. D. Cohen, "Intelligent Information-Sharing Systems", Communications of the ACM 30(5) 390-402, May 1987

Malone T., K. Grant, F. Turbak, Brebet S. and M. Cohen, "Intelligent Information Sharing Systems", CACM, Vol. 30, may 1987

March J. and J. Olsen, “Garbage Can Models of Decision Making in Organizations”, in Weissinger-Baylon R. And J. March (eds.), Ambiguity and Command: Organizational Perspectives on Military Decision Making, Pitman Publishing, New York, 1986

McCabe F. and Clark K., April: Agent Process Interaction Language.

Mitchell T., Caruana R., Freitag D., McDermott and D. Zabowski, “Experience with A Learning Personal Assistant”, Communications of the ACM, Vol. 17, No. 7, 1994, pp. 81-91

Moffat D. and N. H. Frijfa, “Where there is a *Will* there is an Agent”, in Wooldridge M. J. and N.R. Jennings (eds.), Intelligent Agents - Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, August 8-9, Springer-verlag, 1995, pp. 245-260

Muller J. P., Pischel M. And M. Thiel, “Modeling Reactive Behavior in vertically layered Agent Architectures”, in Wooldridge M. J. and N.R. Jennings (eds.), Intelligent Agents - Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, August 8-9, Springer-verlag, 1995, pp. 261-176

Murray D., “Developing Reactive Software Agents”, AI Expert, March 1995

Musto D. and K. Konolige, “Reasoning about Perception”, AI Communications, Vol. 6, No. 3-4, pp. 207-212, 1993

Norman D. A., “How Might People Interact with Agents”, Communications of the ACM, Vol. 37, No. 7, pp. 68-71, 1994

Norman T. J. And Long D., “Goal Creation in Motivated Agents”, in Wooldridge M. J. and N.R. Jennings (eds.), Intelligent Agents - Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, August 8-9, Springer-verlag, 1995, pp. 277-289

Ousterhout J. K., "Tcl and the Tk Toolkit", Addison Wesley, 1994

Rasmus P. W., “Intelligent Agents”, PC AI, Vol. 9, pp. 27-32, 1995

Rickens D., “M: An Architecture of Integrated Agents”, Communications of the ACM, Vol. 17, No. 7, 1994, pp. 107-116

Riecken D. “A Conversation with Marvin Minsky about Agents”, Communications of the ACM 37(7) 22-29, July 1994

Riecken D., “Intelligent Agents”, Communications of the ACM 37(7) 18-21, July 1994

Robinson M., “Through the Lens Smartly”, Byte 16 177-87, May 1991

Selker T., “COACH: A Teaching Agent that Learns”, Communications of the ACM, Vol. 17, No. 7, 1994, pp. 92-99

Shoham Y., “Agent-Oriented Programming”, Artificial Intelligence, Vol. 60, No. 1, pp. 51-92, 1993

Sloman A., “What sort of architecture is required for a human-like agent?” Invited talk at Cognitive Modeling Workshop, AAAI96, Portland Oregon, Aug. 1996.

Smith R., “The Practical Impact of Agent Technology”, PAAM 1996

Steinberg D., “Demon Knowbots”, PC Computing, 135-136, January 1990

Stevens C., “Automating the Creation of Information Filters”, Communications of the ACM 35(12) 48 December 1992

Stoffels J. D., “Strategic Issue Management: A Comprehensive Guide to Environmental Scanning”, Pergamon, Oxford, 1994

Thomas S. R., “The PLACA Agent Programming Language”, in Wooldridge M. J. and N.R. Jennings (eds.), Intelligent Agents - Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, August 8-9, Springer-verlag, 1995, pp. 356-369

Walls J. G., G. R. Widmeyer, and O. A. El Sawy, “Building an Information Systems Design Theory for Vigilant EIS”, Information Systems Research, Vol. 3, No. 1, pp. 36-59, March 1992

Wavish P. And M. Graham, “Roles, Skills and Behavior: A Situated Action Approach to Organizing Systems of Interacting Agents”, in Wooldridge M. J. And N. R. Jennings (eds), Intelligent Agents, Proceedings of ECAI-94 Workshop on Agent Theories, Architectures and Languages, Amsterdam, The Netherlands, August 8-9, 1994, pp. 371-385

Weerasooriya D., Rao A., and Ramamohanarao K, “Design of a Concurrent Agent-Oriented language”, in Wooldridge M. J. and N.R. Jennings (eds.), Intelligent Agents - Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, August 8-9, Springer-verlag, 1995, pp. 386-401

White J. E., "Telescript Technology: The foundation for the Electronic Marketplace", General Magic White Paper, General Magic Inc., 1994

Wooldridge M. J. and N. R. Jennings, “Agent Theories, Architectures and Languages: A Survey”, in Wooldridge M. J. and N. R. Jennings (eds), Intelligent Agents, Proceedings of ECAI-94 Workshop on Agent Theories, Architectures and Languages, Amsterdam, The Netherlands, 1995, pp. 1-39

Wooldridge M., “This is MyWorld: The Logic of an Agent-Oriented DAI Testbed”, in Wooldridge M. J. and N.R. Jennings (eds.), Intelligent Agents - Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, Springer-verlag, 1995, pp. 160-175

Young L., “Mess Management Support Systems: Computer Support for Managing Dynamic Networks of Concerns”, in El Sawy (ed.), DSS-Transactions, pp. 95-106, 1987

Commercial Information Filtering Systems (http://www.enee.umd.edu/medlab/filter/commercial.html)

Cooperation Among Heterogeneous Agents (http://dis.cs.umass.edu/research/team.html)

Freely Available Information Filtering Systems (http://www.enee.umd.edu/medlab/filter/software.html

WebCrawler (http://webcrawler.cs.washington.edu/WebCrawler/WebQuery.html)

Harvest (http://harvest.cs.colorado.edu)

WebAnts (http://thule.mt.cs.cmu.edu:8001/webants)

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN-20520 Turku
Finland

http://www.tucs.abo.fi/

University of Turku
• Department of Mathematical Sciences

Åbo Akademi University
• Department of Computer Science
• Institute for Advanced Management Systems Research

Turku School of Economics and Business Administration
• Institute of Information Systems Science