December 1993

Distributed Executive Information Systems: A Conceptual Framework

Robert Chi
California State University

Distributed Executive Information Systems - A Conceptual Framework

Robert T. H. Chi
Efraim Turban
Information Systems Dept., School of Business Administration
California State University, Long Beach

Abstract

Executive information systems (EIS) have been successfully implemented in a large number of organizations. Of all the various EIS commercial products, only one (Executive Edge) presents limited artificial intelligence (AI) capabilities. Yet the ability to include various problem solving agents for collaboratively information processing, filtering and presentation is highly desirable. It is possible that the successful EIS systems of the future will be built closely around AI components (expert systems, learning mechanisms and so on..) so that more efficient and effective information processing for the executives can be achieved. In fact, much of executive processing involves complicated problem domains. Therefore, individual AI agents' effort may be insufficient when the information is broad in scope and complicated in nature.

This paper proposes a framework of a distributed intelligent executive information system (DIEIS). It illustrates how multiple resources (consisting of knowledge learning, reasoning, filtering and presentation) can be combined for information processing in an EIS environment. For example, a particular piece of information may be refined and presented based on past experiences and current practices in a particular problem domain with the help of both an expert system and neural computing working independently of each other. Another issue involved in DIEIS is multiple agents working together collaboratively to help complex information processing.

Key words: Executive Information Systems, Artificial Intelligence, Distributed Intelligent Executive Information Systems.

1. INTRODUCTION

Executive Information systems (EIS) have been primarily designed to focus on, filter, and organize executives' information, so that the executives can make more effective use of computerized information. In general, the goals of EIS [Watson et al. 1992] are (1) to reduce the amount of data bombarding the executive, (2) to increase the relevance, timeliness, and usability of the information that does reach the executive, (3) to focus a management team on critical success factors (4) to enhance executive follow-through and communication with others and (5) to track the earliest of warning indicators: competitive moves, customer demands, and more. In short, it is a tool that supports the executive in identifying major problems (see phase I in Figure 1) and or opportunities and in taking appropriate actions (see phase II in Figure 1).

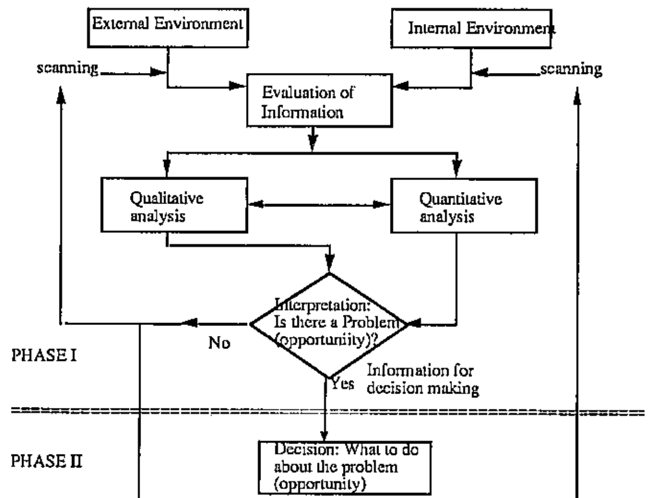


Figure 1. The Decision Making Process of an Executive

In other words, an executive information system is an interactive computer-based information system that provides executives with easy access to internal and external information that is relevant to their critical success factors [Watson, Rainer and Koh 1991]. The characteristics of EIS have been studied in recent research [Batkan 1991, Friend 1988]: For example, EIS are tailored to individual executive users; EIS extract filter, compress and track critical data; EIS provide drill down capabilities and exception reporting; and EIS present a very high quality graphical, tabular and textual information.

Preedy [1990] has defined EIS in terms of the following key characteristics:
"(1) An executive information system is used personally by the most senior managers of an organization, normally the executive directors in a company.
(2) It is used as a general tool, rather than to support one specific function; this means that it will be expected to cover several functions and to pick up data from several different sources often on different computers.
(3) It is used by executives whose main role is in managing other subordinate managers, rather than acting directly in an operational capacity themselves.
(4) Its main use is informative, offering insight into corporate data, rather than administrative; it is essential an information system, not merely an aspect of office automation."

Current EIS are modular in nature. A typical EIS provides:
(1) Drill down capabilities (an intelligent agent can help in identifying what is going wrong, saving drill down time. Thus, the agent can act as a director for drill down.
(2) Information monitoring: usually a CSF methodology is used to decide what information to track.
(3) Access to aggregate (global) information.
(4) Extensive use of external data.

(5) Written interpretations.
(6) Highlights problem indicators.
(7) Ad-hoc analysis.
(8) Information presentation in hierarchical form.
(9) Incorporating graphic and text in the same display.
(10) Providing management by exception reports.
(11) Showing trends, ratios and deviations.
(12) Providing access to historical and most current data.
(13) Being Organized around critical success factors.
(14) Having forecasting capability.
(15) Filtering, compressing and tracking critical data.
(16) Supporting open-ended problem explanation.

Recently, it has been recognized that the first generation of EISs, which intended to support mainly the identification of problems and opportunities (see phase I in Figure 1) should be enhanced with decision making capabilities. Thus, a second generation of EISs has been termed Executive Support Systems (ESS) [Rockart and Delong 1988] and it includes several analytical tools for decision support. Indeed, most EIS venders provide tools that are intended to build some kind of ESS (e.g., providing DSS capabilities, such as modeling in addition to the conventional EIS capabilities). For example, Commander EIS works with system W, Executive Edge with IFPSplus, Express/EIS with Express and Command Center EIS works with Advantage/G.

Executive decision are very complex and therefore they are frequently partitioned into subproblems. These subproblems are being analyzed by experts individually, or a task force of experts is formed to work on the problem collectively. Attempts to provide computerized support to this kind of approach falls under two titles: distributed decision making (DDM) and group decision support system (GDSS). The topic of DDM is used as a foundation to the framework proposal in this paper, The topic of GDSS will be only briefly mentioned here (for discussion see [Jessop and Valacich 1993]). Distributed decision making (DDM) is a coordinated decision making effort between communicating individuals which possess some specialized knowledge and can process the knowledge in a manner that contributes to performing some intelligent tasks involved in the decision process [Ching 1988].

As a potential extension of EIS, distributed information processing can be supported by a computer system with the characteristics of distributed participants and various expertises. Therefore, it is proposed that a distributed executive information system (DEIS) be defined as a "computer system support for executive distributed information processing with heterogeneous problem-solving participants and expertises".

A typical EIS framework offers single information processing mechanism, while DEIS focus on multiple problem solving mechanisms and heterogeneous expertises (see Table 1).

	Problem Solving Agents	Expertises
EIS	One	Homogeneous
DEIS	Many	Heterogeneous

Table 1. The characteristics of DEIS and EIS

Furthermore, the use of several problem solving agents usually implies specialization in a narrow domain. Therefore the framework of DEIS can provide an ideal opportunity for as a Distributed Intelligent Executive Information (or support) system, to be abbreviated DIEIS. EIEIS go beyond EIS by incorporating a distributed artificial intelligence (DAI) architecture into their information processing system. This architecture consists of heterogeneous agents with complementary skills cooperating to process information. The objective of this paper is to describe such a system. The paper is composed of the following parts: in section 2, a review of previous research is presented. In section 3, a conceptual model of DIEIS is introduced and in the last part conclusions are drawn and future research is outlined.

2. CURRENT RELEVANT RESEARCH IN DAI

2.1 Definitions

A number of research areas deal with the support of distributed problem solving processes. To better understand our proposal framework, we will redefine the concept of Distributed Problem Solving (DPS): "Distributed problem solving is the cooperative solution of problems by a decentralized and loosely coupled collection of knowledge sources (KS's) (procedures, sets of rules, etc.), located in a number of distinct processor nodes" [Smith and Davis 1981]. The focus of distributed problems solving systems research [Decker 1987] is the nature of the distributed problems and on the and multi-agent environment that is built to solve these problems. Such systems are also known as distributed artificial intelligence [Huhns 1987], distributed reasoning systems [Arni et al. 1990], cooperating knowledge-based systems [Croft & Lefkowitz 1988], and group problem solving systems [Shaw and Fox 1991].

2.2 Limitations of Current Executive Information Systems

Current EIS emphasize knowledge retrieving, knowledge filtering and knowledge presentation based on a single processing mechanism. The information used to support EIS has the characteristics of being deep in contents and broad in scope. Therefore, an intelligent agent which can help EIS in data retrieving, filtering and presentation can provide a significant contribution. In addition, since the EIS information is broad in scope, and since the supporting tasks for EIS are diversified in nature, more than one problem processing agents may be needed.

The theoretical background for employing a multiple intelligent information processing agents has been studied by Simon who stated that "The capacity of the human mind for formulating and solving complex problems is very small compared with the size of the problems whose solution is required for objectively rational behavior in the real world -- or even for a reasonable approximation to such objective rationality" [Simon 1957]. The limitation of a human mind's processing capacity was called by Simon "bounded rationality". Fox [1981] postulated that "bounded rationality implies that both the information that one person can absorb and the detail of control he may yield is limited. As tasks grow larger and more complex, means must be found to effectively limit the increase of information a person sees and the complexity of control".

The above limitations force the executive to conduct a tedious semi-automatic drill down search and/or to report a detailed DSS analysis from several experts. The distributed intelligent executive information system (DIEIS) framework proposed in this paper is designed to overcome the above limitations.

2.3 Distributed Intelligent Executive Information Systems

DIEIS are defined as executive information systems which have the scheme of heterogeneous intelligent agents and distributed information processing. Heterogeneous expertises support DIEIS to encompass a variety of problem domains, while distributed information processing allow the sharing of resources to improve the efficiency and effectiveness of the system. In distributed problem solving studies, intelligent agents can combine their resources so that the intelligence of the group is more than the sum of individual agents' intelligence [Durfee 1988]. The coordination mechanism between agents is a key issue in the success of DDM.

In the following sections, the architecture of distributed artificial intelligence is outlined and the relationship between distributed artificial intelligence and distributed intelligent executive information systems is discussed.

2.4 Distributed Artificial Intelligence (DAI)

Distributed problem solving appears in two forms (1) task-sharing and (2) result-sharing [Smith and Davis 1981]. In task sharing systems, nodes assist each other by sharing the computational load for the execution of subtasks of the overall problem, while in result sharing systems, nodes assist each other by sharing partial results which are based on somewhat different perspectives on the overall problem. Each form is discussed in following sections.

2.4.1 Task-Sharing Systems

In task-sharing systems, the overall problem to be solved is decomposed into several smaller subproblems (see Figure 2). Cooperation is achieved by sharing the computational load of the overall problem. Each subproblem is assigned to a particular agent that will asynchronously perform its own functions and submit a solution synchronously with other agents. The contract-net protocol developed by Smith [1980] proposed a framework that is designed to allow agents to submit bids for tasks. Any agent that receives a task announcement message can reply with a bid, indicating an information on how the task is to be accomplished. The coordinator that announces the task, collects the bids and awards the task to the bidder with the highest bid. Other examples of task sharing systems are the office information system proposed by Woo and Lachovsky [1986], the scheduling system of Shaw and Whinston [1988, 1989], and the object-oriented multiple agent planning system of Kamel and Syed [1989].

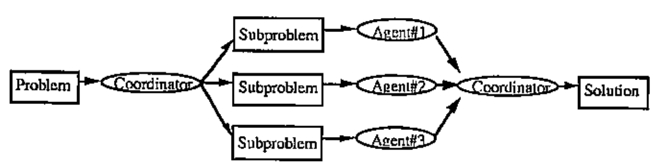


Figure 2. Framework Task-sharing Systems

Generally speaking, task-sharing based systems are most useful for problem domains in which it is appropriate to define a hierarchy of tasks or levels of data abstraction [Smith and Davis 1981]. Such problems can usually be decomposed into a set of independent subproblems. Many problem tackled by executives are of this nature. For example, a evaluation of a proposed acquisition may require advice from legal, financial, technological and organizational experts. The expertise is provided to the acquisition decision maker(s) who incorporates the expertise in determining the fate of the acquisition.

2.3.2 Result-Sharing Systems

Result-sharing is a form of cooperation in which individual nodes assist each other by sharing partial results, based on somewhat different perspectives of the overall problem [Smith and Davis 1981]. In this type of system, control is typically "data-oriented." At any point in time, the computation done by a certain agent is used to satisfy an information which is needed for the computation of another subtask done by another agent (see Figure 3). Thus, an explicit hierarchy of task-subtask relationships does not have to exist between individual nodes. Typically, one of the agents acts as the group planner (or the coordinator), and each of the other agents sends all pertinent information to this agent in order to form a global plan for problem solving. The main issue of such systems is how to guide and coordinate the interactions among the participating agents, so that the problem can be solved jointly by the group.

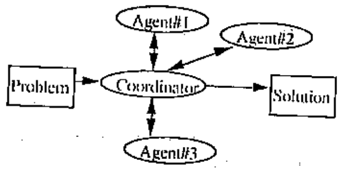


Figure 3. Framework of Result-Sharing Systems

An example of result sharing in an EIS context is the process of environmental scanning where the collected data are forwarded to interpreters who transfer the interpreted information to the decision maker. Another example is that where the results of some forecasts are forwarded to an analyst for interpretation. In general, result-sharing DAI systems are most useful in problem domains in which (1) results achieved by one node influence or constrain those that can be achieved by another node (i.e., the results are significantly relevant to each other), (2) sharing of results drives the system to converge to a solution of the problem (i.e., results received from remote nodes do not cause oscillation), and (3) sharing of results drives the system to a correct solution of the problem [Smith and Davis 1981]. Such situations are typical in what is known as sequential decision making [Sprague and Carleston 1982]. For example, a decision of how much to produce is interrelated with that of when to produce, which drives the machines and employees schedules, which drives cash flow and marketing plans.

To date, several result-sharing based DAI systems have been developed. One example is MACE .(Multi-Agent Computing Environment) [Gasser et al. 1987] which is an instrumented testbed for building a wide range of experimental distributed artificial intelligence systems at different levels of granularity. The dominant metaphor of MACE is a collection of intelligent, semi-autonomous agents interacting in organized ways. The computational units (agents) run in parallel, and communicate via messages for problem solving. Mason et al. [1989] proposed a distributed assumption-based truth maintenance system (DATMS) which interprets data from a seismic sensor network for nuclear test ban treaty verification. Each agent interprets data from a sensor site or geological region and relies on its communications lines to guide its search for an interpretation of its own data. DATMS is implemented under MATE (Multi-Agent Test Environment) using C and Common Lisp. Nii et al. [1989] incorporated two concurrent systems, Cage [Aiello 1986] and Poligon [Rice 1986], to solve problems based on the blackboard architecture. Both Cage and Poligon are designed to exploit multiprocessor hardware with the intent of achieving computational speedup. Shaw and Fox [1991] proposed a networked expert system testbed (NEST) which consists of a network of four expert systems. The architecture of NEST is based on a variation of the blackboard architecture, with "mailbox" areas added to the blackboard shared area for coordinating the agents. With three functional expert systems (marketing, production, and purchasing), and another expert system serving as the coordinator, NEST proposes a solution to determine the quantity of a new product that is about to be sent to the market.

2.4.3 Comparison for Task-Sharing and Result-Sharing Systems

Task-sharing is used to organize problem decomposition through the relationship of task-subtask connections between nodes. The result, which is typically a hierarchy, is used to structure answer synthesis. One important assumption made by task-sharing systems is subtasks can be accomplished by individual nodes working independently. This allows the improvements of problem solving efficiency by reducing internode communication. Result-sharing is used to facilitate problems which can not be solved by individual nodes working independently without significant communication with other nodes [Smith & Davis 1981]. Since result-sharing systems do not have the capability of problem decomposition, problem decomposition and distribution of subproblems to individual nodes are handled by an agent outside of these systems.

2.5 Coordination Mechanisms of Distributed Problem Solving

Coordination is the key component to the success of DEIS. The purpose of the coordination mechanism is to control problem solving so that cooperating nodes work together as a coherent team [Durfee 1988]. In distributed problem solving systems, coordination is achieved by exchanging data, partial solution plans, and constraints among agents. Several research projects have emerged in the area of designing coordination mechanisms for solution plan processes in a multi-agent environment. Shaw and Fox [1991] classified coordination mechanisms into the following seven categories for more recently developed distributed problem solving systems (See Table 2).

Coordination method	Features	Referencess
Coordination by Revising actions	Conflicts avoidence	Cammarata et al. [1983]
Coordination by Synchronization	Timing control, Interaction regulation	Corkill [1977], Georgeff [1983]
Coordination by Negotiation	Two-way communication for agreements	Croft and Lefkowitz [1988]
Coordination by Structured Group Mediation	Delphi technique, Nominal group technique	Nunamaker et al. [1988]
Coordination by opportunistic goal satisfaction	Blackboard model, Information sharing	Nii et al. [1989], Shaw et al. [1990, 1991]
Coordination by exchanging preferences	Game-theory	Silver et al. [1990]
Coordination by constraint reasoning	Constraint satisfaction among agents	Sathi and Fox [1989]

Table 2. Coordination mechanisms of Distributed Problem Solving

3. A CONCEPTUAL MODEL FOR DISTRIBUTED INTELLIGENT EXECUTIVE INFORMATION SYSTEMS

This section introduces the proposed framework of DIEIS. This is followed by a discussion of the problem solving processes of the system.

3.1 The Framework of Distributed Intelligent Executive Information Systems

In the DIEIS framework (see Figure 4), a decentralized group of agents cooperatively attempt to provide a solution to a complex problem through the coordinator. Information of decomposed subproblems and partial solutions is shared among agents. Each agent who works independently, and may even be at a different geographical location, is supported by the knowledge base which consists of different forms of knowledge (e.g. rules, cases, and data). In general, a DIEIS contains seven independent but closely related subsystems:

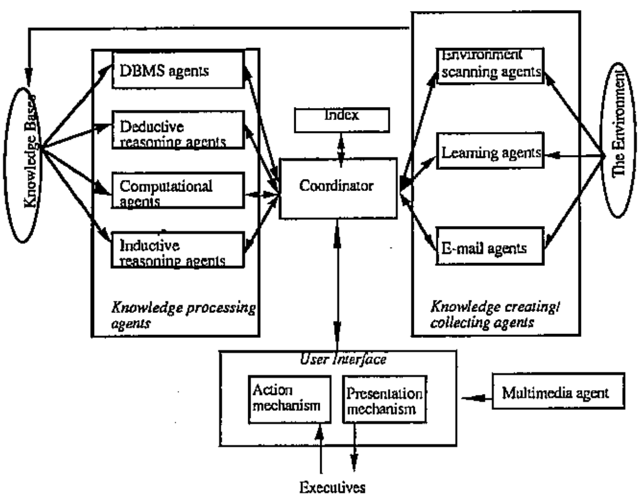


Figure 4. The Conceptual Framework of DIEIS

1. Knowledge Processing agents.
2. Knowledge Bases (the case base, the rule base and the data base).
3. Knowledge Creating/Collecting Agents.
4. User Interface.
5. Multimedia agent.
6. The Environment.
7. Coordinator.

Knowledge processing agents consist of DBMS, inductive reasoning agents, deductive reasoning agents and computational agents. Their responsibilities are to retrieve and organize data from the databases and refine them through the coordination of the coordinator. The refined data will then be sent to the presentation mechanism for executives. The inductive reasoning can be a case-based reasoning agent [Slade 1991; Sombe 1990; Vosniadou and Ortony 1989; Owen 1990; Riesbeck and Schank 1989] which uses past experiences for current problem solving. When the problem domain is poorly understood, or the domain theory is too weak to be acquired from the experts, case-based reasoning is an attractive reasoning approach [Porter 1989, Chi et al. 1991]. New problems can be solved by matching important features of previous cases that were successfully solved. Therefore, the major process of case-based reasoning involves remembering and adapting. Typical case-based reasoning research includes MEDIATOR [Kolodner & Simpson 1989], CHEF [Hammond 1986], and Casey [Koton, 1988]. The deductive reasoning agents are rule based mechanisms or expert systems. New solution is deduced from previously stored rules. Deductive reasoning agents employ the deductive reasoning method with an existing domain theory. The domain theory consists of production rules that are represented as if-then statements that define logical relations between concepts of the problem domain [Bratko 1986]. This type of system solves problems by applying the domain knowledge elicited from experts. The problem-solving process involves a search in the knowledge base that, hopefully, can guide the problem solver to the goal state. Many expert systems are also deductive reasoning systems. The premise is that the knowledge of an expert can be embodied in a set of rules. This premise was derived from Newell and Simon's pioneering work on the general problem solver (GPS) [Newell and Simon 1972], one of the first AI programs, and came to fruition in the DENDRAL project, one of the first rule-based expert systems [Feigenbaum et al. 1971].

The user interface is part of the dialog system (user interface) in a basic decision support framework. The user interface usually provides EIS with graphical capability so that executives' inquiries can be collected effectively and organized information can be presented in a more comprehensive format. The user interface is divided into two submechanisms based on the functionalities: (1) presentation mechanism and (2) action mechanism. Action mechanism is used to collect inquiries from executives, while presentation mechanism is responsible to present processed results (information) in a comprehensive way. In addition, raw information from different information processing agents can be refined and organized graphically since various sources of information may be collected by the coordinator and sent to the user interface. Furthermore, the user interface can be adjusted based on different executives' requirements. This subsystem can have its own intelligent agent who will determine, for example, the machine to be used in specific presentation [Sipior and Garrity 1992]. Generally speaking, user functions in EIS interface are designed in modules. Typical modules are

(1) Status report (with possibility of drill down and exception reporting): textual explanation and trend graph.
(2) Reminder: notes, calendar, tracking information about messages.
(3) Investigation: comparisons, calculations, drill down, personalized analysis, graphics.
(4) Electronic mail: alerts if mail is pending, monitoring mail, can transmit any screen of other modules.
(5) New service: both external and internal with capability to drill down for details. Hypertext capabilities are available.
(6) Detailed analysis: in this case, monitored results are going through a quantitative analysis. This is typically done in executive support systems (ESS).

Executives utilize the above user interface modules to issue inquiries so that multiple intelligent agents can be triggered for problem solving (information processing). For example, An "Investigation" query may involve using several DBMS agents to retrieve data from the different databases and employing a Spreadsheet agent for calculation; the results will then be sent to a Rule based agent for reasoning process.

Knowledge bases contain the relevant knowledge which is organized as a database, a case base, a rule base or a model base. The knowledge in the knowledge base is systematically organized for easy interfering and refining in order to support other information processing agents.

If no existing knowledge is available, knowledge creating/collecting agents are triggered. These agents can be of three types: inductive/deductive learning agents, environment scanning agents, and E-mail agents. The inductive [Michalski 1980; Michalski and Stepp 1983] and deductive learning agents [Mitchell et al. 1986] are used when new knowledge is needed or existing knowledge needs to be modified. Induction learning agent infers the description of a class from descriptions of individual objects of that class. Training examples are given as cases and described by a vector of attribute values. A general concept description is induced by inspecting specific instances of the concept. Since the concept description is generated by inspecting similarities among examples, it is a form of learning by examples. To arrive at a correct concept description in accordance with training examples, a hypothesis concept description is chosen to cover positive examples and exclude negative ones. As the process continues, new examples are fed in and the learner updates the hypothesis to keep it consistent with new examples, until all training examples are consistent with the learned concept description. Similarity based learning systems have been widely used to acquire knowledge for reasoning systems which perform classification tasks [Winston 1975; Michalski 1980; Michalski & Stepp 1983; Dietterich & Michalski 1983].

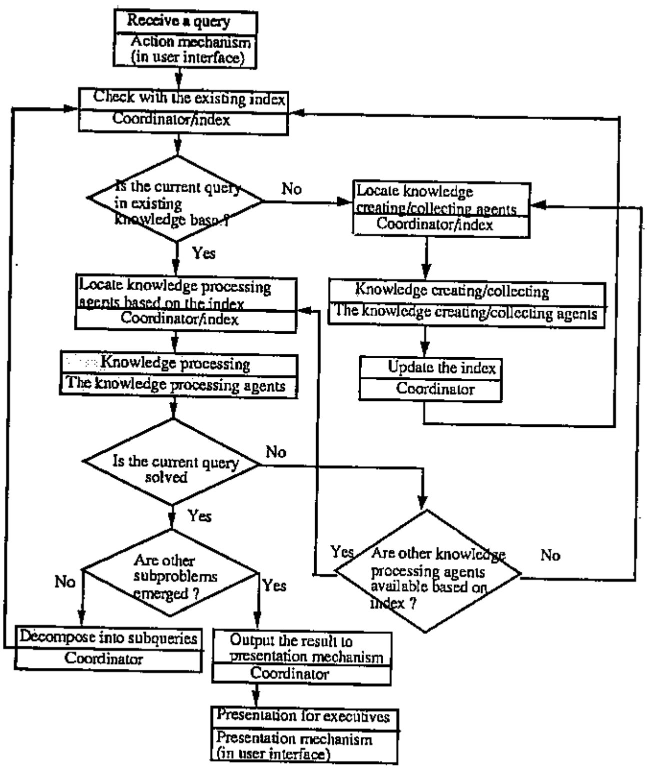
The deductive learning agent uses existing knowledge to explain and generalize a single example and thereby acquires an operational concept description and problem-solving knowledge [DeJong 1983; DeJong & Mooney 1986; Mitchell, et al. 1986; Mooney & Bennet 1986]. An explanation based learning (EBL) program takes a single positive instance as the training example, explains this example by an existing knowledge base (or the domain theory), and produces a generalized concept description as the final output. This class of learning method allows generalized concepts to be determined by only one instance, in contrast to multiple instances needed for the deductive learning agent. The deductive learning agents' construction and analysis of explanations require extremely detailed knowledge of the problem domain. The environment scanning agents collect available data from the environment for the knowledge base and other learning agents. The E-mail agents receive data or information from outside sources such as other decision support systems or agents.

The environment is where raw data can be collected. The environment can be classified as two categories: Internal environment and external environment. Internal environment has the raw data from the organization itself or internal sources, while external environment provides data source from any outside agents. For example, past instances of this organization are considered raw data from the internal environment. Historical statistical data from the government agency are considered as the raw data from the external environment. Both the internal and external environments provide the learning mechanism with data resources so that useful information (usually regularities and commonalties) is retrieved. The environment also provides the knowledge retrieving mechanism with raw data to be organized into the databases.

The coordinator, the heart of a DIEIS, regulates the inter actions among individual agents. All the communication activities among agents will be transferred through the coordinator. The meta knowledge of the coordinator is stored in the index which regulates the way agents communicate, problem decomposition, sub-problem assignments, and proposal evaluation.

3.2 The Information Flow Diagram in DIEIS

Information processing in DIEIS can be classified into two categories: (1). knowledge processing and (2) knowledge creating/collecting. Knowledge processing involves utilizing existing knowledge processing agents (e.g., DBMS, Inductive reasoning agent, Deductive reasoning agents...) to "reason, retrieve and filter" existing knowledge. A query such as "get the value of total_sales in region "A" during 1991" may require a DBMS with an existing database of annual sales account". If existing knowledge does not contain needed information, the coordinator will trigger the knowledge creating/collecting agents for knowledge collection form the environment (could be the internal environment or the external environment). For example, if "the rule to select good stocks" can not be retrieved from any rule-based agents based on existing knowledge bases, the learning agent will be used to implement a similarity-based learning process with the help of existing data base where historical stock information was collected. In general, DEIS achieves goals by collaborative team work to enhance the problem solving (knowledge processing) capabilities. Figure 5 summarizes the knowledge processing flow in DEIS. A query is received by the "action mechanism" of the user interface. The coordinator checks with the index if there is existing knowledge in the knowledge base. If there is knowledge available to solve this query, the knowledge processing agent(s) will be located and information processing is triggered. If the current problem is solved, the coordinator will then check if there is any subproblems emerged. That is to say the current problem is decomposed into more subproblems. For each subproblem, another iteration will be triggered and concurrent information processing is possible. However, if there is no existing knowledge available, knowledge creating/collecting agents will be employed to create/learn new concepts/knowledge from the environment. New learned concepts/knowledge is then stored in the knowledge base fore further processing. In the meantime, the index is also updated.



Shaded areas indicate concurrent processing is possible

Figure 5. The Knowledge Processing Flow in DIEIS

4. CONCLUSION AND SUGGESTED RESEARCH

In this paper, we propose a framework with distributed information processing agents. We introduce multiple intelligent agents in this system so that more complicated information can be processed by the cooperative efforts from various agents. In fact, many executive information contains various aspects of problem domains which can not be processed by a single data retrieving mechanism. In other words, current EIS retrieve data directly from the database and present it with no automated data processing. By employing this framework, a more intelligent yet automated EIS can be constructed by having a set of agents working cooperatively.

In order to implement a DIEIS, it is necessary to conduct further research. As a basis one can use the generic EIS research directions proposed by Watson et al. [1992]. These topics should be investigated as they related to DIEIS.

Specific DIEIS topics include:

(1) DIEIS architecture; especially its relationship to a blackboard structure.

(2) Learning mechanisms for DIEIS.

(3) Which agents will participate in the DIEIS and which role each of them is going to play

(for different possible scenarios).

(4) The economics of DIEIS; i.e. when would it be economically feasible to use a DIEIS.

(5) Development methodology; How DIEIS is going to be developed ? Can we use existing tools ?

(6) The interface of DIEIS with other CBIS and especially with DSS and intelligent DSS.

Multiple intelligent agents can support some of the most different issues in EIS/ESS implementation. They are:

(1) Finding the executive information requirements (e.g. see Wetherbe [1991] and Watson and Frolick [1992]).

(2) Managing the development process of EIS (e.g. see Watson et al. [1991]).

(3) Environmental scanning and interpretation (e.g. see Preedy [1990]; Watson et al. [1992]).

(4) Justification of EIS and especially DIEIS (e.g. see Batkan [1991]; Watson et al. [1992]).

(5) Integration of EIS with other computer-based information systems.

The investigation of these and related research issues could provide the insights which are needed to create powerful DIEIS which would support large number of executives in their complex job.

Reference

Aiello, Nelleke (1986),"User-Directed Control of Parallelism: The Cage System," Technical Report KSL-86-31, Knowledge Systems Laboratory, Computer Science Department, Stanford University, April.

Arni, N. et al. (1990), "Overview of RAD: A Hybrid and Distributed Reasoning Tool," MCC Technical Report No. ACT-RA-098-90, March.

Bratko, Ivan (1986), PROLOG Programming for Artificial Intelligence, Addison-Wesley publishing Co.

Batkan, Wayne C. (1991), Executive Information Systems,New York, Van Nosttand Reinholl.

Ching, Chee (1988), "A General Theory of Distributed Decision Making," Doctoral Thesis, Purdue University.

Croft, W. B. and Lefkowitz (1988), "Knowledge-based Support of Cooperative Activities", Proceedings of the 21st Annual Hawaii International Conference on System Science, Vol. III, pp. 312-318.

Corkill, D. (1979), "Hierarchical Planning in a Distributed Environment," Proceedings of the sixth International Joint Conference on Artificial Intelligence, Tokyo, Japan, August, pp. 168 - 175.

Chi, T. H. Robert, Minder Chen and Yihwa Kiang (1991), "A Generalized Case Based Reasoning System for Portfolio Management," Forthcoming in the Journal of Expert Systems with Applications.

Cammarata, S., D. McArthur and R. Steeb (1983), "Strategies of Cooperation in Distributed Problem Solving", Proceedings of the 8th International Joint Conference of Artificial Intelligence, pp. 767 - 770.

Decker, K. S. (1987), "Distributed Problem Solving: A Survey," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 17, No. 5, September.

DeJong, G. F. and R. J. Mooney (1986), "Explanation-Based Learning: An Alternative View," Machine Learning 1, 2, pp. 145-176.

DeJong, G. (1983), "Acquiring Schemata Through Understanding and Generalizing Plans," Proceedings of the Eighth International Joint Conference on Artificial Intelligence, pp. 462-464. Karisruhe, West Germany, August.

Dietterich, T. G., & Michalski, R. S. (1983), "Discovering Patterns in sequences of objects," Proceedings of the 1983 International Machine Learning Workshop, pp. 41-57.

Durfee, E. H. (1988), Coordination of Distributed Problem Solvers, Kluwer Academic Publishers, Boston, MA.

Feigenbaum, E. A., B. G. Buchanan, and J. Lederberg (1971), "On generality and problem solving: a case study involving the dendral program," In B. Meltzer and D. Michie, editors, Machine Intelligence, pp. 165-190, American Elsevier, New York.

Friend, David (1988),"EIS: Straight to the Point," Information Strategy: The Executive's Journal, v4n4, Summer 1988, 25-30.

Fox, Mark (1981), "An Organizational View of Distributed Systems," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 11, No. 1, January.

Gasser, Les, Carl Braganza, and Nava Herman (1987), "MACE: A Flexible Testbed for Distributed AI Research," in Distributed Artificial Intelligence, Michael N. Huhns (Ed.) , Morgan Kaufmann, Los Altos, CA.

Georgeff, M. (1983), "Communication and Interaction in Multi-agent Planning," Proceedings AAAI-1983.

Hammond, K. (1986), "CHEF: A model of case-based planning," Proceedings of American Association of Artificial Intelligence (pp. 267-271). Philadelphia, PA.

Huhns, Michael N. (1987), Distributed Artificial Intelligence, Vol. 1, Morgan Kaufmann Publisher, Inc., Los Altos, CA.

Jessop, L. M. and J. S. Valacich (1993), Group Support Systems, Macmillan, New York.

Koton, P. (1988), "Reasoning about evidence in causal explanations," Proceedings of American Association of Artificial Intelligence (pp. 256-261). St. Paul, MN.

Kolodner, Janet L. and Robert L. Simpson (1989), "The Mediator: Analysis of an early case-based problem solver," Cognition Science 13, pp. 507-549.

Kamel, M. and A. Syed (1989), "An Object-Oriented Multiple Agent Planning System," in Distributed Artificial Intelligence, Vol. II, Les Gasser and Michael N. Huhns (Eds.), Morgan Kaufmann, San Mateo, CA.

Mason, Cindy L. and Rowland R. Johnson (1989), "DATMS: A Framework for Distributed Assumption Based Reasoning," in Distributed Artificial Intelligence, Vol. 2, Morgan Kaufmann Publishers, edited by Les Gasser and Michael N. Huhns.

Mooney, R. J. and Bennett, S. W. (1986), "A Domain Independent Explanation-Based Generalizer," Proceedings of the National Conference on Artificial Intelligence, pp. 551-555.

Mitchell, Tom M. , Richard M. Keller and Smadar T. Kedar-Cabelli (1986), "Explanation-Based Generalization - A unifying view," Machine Learning 1, 1(January). 47-80.

Michalski, R. S. (1980), "Knowledge Acquisition through Conceptual Clustering : A Theoretical Framework and an Algorithm for Partitioning data into Conjunctive Concepts," International Journal of Policy Analysis and Information Systems, 4, 3, pp. 219-244.

Michalski, R. S. and Stepp, R. E. (1983), "Automated Construction of Classifications: Conceptual Clustering versus Numerical Taxonomy," IEEE Transactions on Pattern Analysis and Machine Intelligence, 5, 4, pp. 396-409.

Nunamaker, J. F., L. M. Applegate and B. R. Konsynski (1988), "Computer-aided Deliberation: Model Management and Group Decision Support," Operation Research, vol. 36, no. 6, pp.826-847.

Newell, A and Herbert Simon (1972), "Human Problem Solving," Englewood, N.J.:Prentice-Hall.

Nii, H.P., Aiello, N., and Rice, J. (1989), "Experiments on Cage and Poligon: Measuring the Performance of Parallel Blackboard Systems," Distributed Artificial Intelligence, vol. II., L Gasser and M. Huhns (Eds.), Pitman, London, pp. 319-384.

Owen, S (1990), "Analog for Automated Reasoning," New York, Academic Press, 1990.

Porter, Bruce W., E. Ray Bareiss, and Robert C. Holte (1989), "Knowledge Acquisition and Heuristic classification in Weak-theory domains," Technical Report AI-TR89-96, Artificial Intelligence

Laboratory, Department of Computer Sciences, University of Texas at Austin, February.

Preedy, D (1990), "The Theory and Practical Use of Executive Information Systems," International Journal of Information Management, 10(96-104).

Riesbeck, Christopher K. and Roger C. Schank (1989), Inside Case-based Reasoning, Lawrence relbaum Associates, Hillsdale, New Jersey.

Rice, James, "Poligon: A System for Parallel Problem Solving (1986)," Technical Report KSL-86-19, Knowledge Systems Laboratory, Computer Science Department, Stanford University, April.

Rockart, F. John and David W. DeLong (1988),"Executive Support Systems, The Emergence of Top Management Computer Use, Dow Jones-Irwin, Homewood, Illinois.

Shaw M. and M. Fox (1991), "Distributed Artificial Intelligence for Group Decision Support: Integration of Problem Solving, Coordination, and Learning", Working paper.

Shaw, M., B. Harrow and S. Herman (1990), "NEST: A Networked Expert Systems Testbed for Cooperative Problem Solving and Group Learning," the Proceedings of the Working Conference on Cooperating Knowledge Based Systems, University of Keele, Keele, UK.

Shaw, M., B. Harrow and S. Herman (1991), "Distributed Artificial Intelligence for Multi-agent Problem Solving and Group Learning," the Proceedings of Hawaii International Conference of Systems Science, IEEE Press.

Shaw, M. and A. B. Whinston (1988), "A Distributed Knowledge-based Approach to Flexible Automation: The Contract-net Framework," International Journal of Flexible Manufacturing Systems, pp. 85 - 104.

Shaw, M. and A. B. Whinston (1989), "Learning and Adaptation in Distributed Artificial Intelligence Systems," Distributed Artificial Intelligence, Vol. II, L. Gasser and M. Huhns (Eds), Pitman, London, pp. 413 - 430.

Sipior, J. C. and E. J. Garrity (1992), "Merging Expert Systems with Multimedia Technology," Database, Winter.

Silver, Bernard, William Frawley, Glenn Iba, John Vittal and Kelly Bradford (1990), "ILS: A Framework for Multi-Paradigmatic Learning," Proceedings of the 7th International Conference on Machine Learning , Austin TX.

Simon, H. A. (1957), Models of Man, New York: Wiley.

Smith, R. G. and R. Davis (1981), "Frameworks for Cooperation in Distributed Problem Solving," IEEE Transactions on Systems Man and Cybernetics, Vol. 11, No. 1.

Smith, R. (1980), "The Contract-net Protocol: High-level Communication and Control in a Distributed Problem Solver," IEEE Transactions on Computers, Vol. 29, pp. 1104 - 1113, Dec.

Sathi, A. and M. Fox (1989), "Constraint-Directed Negotiation of Resources Reallocations," Distributed Artificial Intelligence, Vol. II, L. Gasser and M. Huhns (Eds.), Pitman, London, pp. 163 - 194.

Slade, S. (1991) ,"Case-based Reasoning: A Research Paradigm" AI Magazine, Spring.

Sombe, L. (1990) ,Reasoning Under Incomplete Information in Artificial Intelligence, New York, John Wiley & Sons.

Sprague, R. H. and E. D. Carlson,"Building effective Decision Support Systems," Prentice-Hall, Englewood cliff, NJ., 1982.

Vosniadou, S. and A Ortony (1989), Similarity and Analogical Reasoning, Cambridge, Mass, Cambridge Press.

Wetherbe, J. C. (1991) ,"Executive Information Requirements: Getting it right," MIS Quarterly, March 1991.

Watson, H. J., et al. (1991) ,"Executive Information Systems: A Framework for Development and a Survey of Current Prctices," MIS Quarterly, March 1991.

Watson, H. J., et al. (1992), Executive Information Systems, Wiley, New York, 1992.

Watson, H. J., and M. Frolick (1992),"Determining Information Requirements for an Executive Information System," Information System Management, Spring.

Watson, H. J., and R. K. Rainer (1991) ,"Manager's Guide to Executive Support Systems," Business Horizons, March-April.

Watson, Hugh J., R. K. Rainer, and C. Koh (1991), "Executive Information Systems: A Framework for Development and a Survey of Current Practices," Management Science, March 1991.

Woo, C. and F. Lochovsky (1986), "Supporting Distributed Office Problem Solving in Organizations," ACM Transactions on Office Information Systems, July, pp. 185 - 204.

Winston, P.H. (1975), Learning Structural Description from Examples. Psychology of Computer Vision, (Winston, P.H. Ed.), (McGraw Hill, New York).