Source: www.mwftr.com/charlesk.html
Citation Info: Proc. of 14th International Conference on Computers in Industry and Engineering, Nov 27-29, 2001. pp. 130-1333, Las Vegas, NV.

Internet-Based Global Decision Making System Architecture

Charles J. Kim and Jamshid Goshtasbi
Department of Electrical Engineering
Howard University
Washington, DC 20059

Abstract: The far-reaching Internet and the relevant technologies on intelligence shed a light of hope for those problems that seek solutions that are suitable for global perspectives, supported and updated by solutions everywhere. An architecture for global decision making system is proposed. The architecture is composed of with distributed LDMSs, which monitor and draw conclusion based on the local data collected around the area. The global decision making (GDM) architecture centers on two elements: Internet-based agent software and information entropy-based decision-making mechanism. An example implementation of GDM using Concordia agent framework is suggested.

Keywords: Decision Making, Mobile agent, Inductive Reasoning, Global Decision Making System

1. Introduction

In many scientific problem solving and industrial applications from weather forecasting to global warming disputes to energy information infrastructure, substantial efforts have been devoted to developing complex and sophisticated systems. To be globally suitable, these complex and global systems should have been justified by and tuned to the other decision-making systems, scattered all around the globe.

For an example, the study of climate changes is one of the global factors controlling local-level decision. The prediction of El Nino and La Nina can significantly help local decisions such as what kind of crop should be planted in El Nino Years. This is a typical situation in need of a decision making system with global implications: a remote local station is monitoring data to draw a local conclusion, and it wants to revise and update the conclusion to be adaptive to global changes. However, these changes cannot be monitored unless provided by other local stations in other part of the region undergoing the same decision-making functions.

In two recent U.S. Global Change Policy Symposiums held in 1993 and 1994, there were many concerns on the way data were collected and employed for better development of environmental policy. It was reported that the current U.S. global change research program would not provide decision-makers with information they required to respond to global change [1, 2]. Also raised were the needs to focus on hard science of global change, to include such issues as the effects of climate change and how to adapt to them. In other words, the research community on global implication needs to focus on a new tool that provides with global changes and effects, preferably in an intelligent and wide-reaching fashion.

Recently, the Internet has dramatically promoted the growth of enabling technologies such as databases, artificial intelligence, and intelligent agent [3]. The far-reaching network infrastructure and the relevant technologies on intelligence shed a light of hope for those problems that seek solutions which are suitable for global perspectives, supported and updated by solutions elsewhere. For example, grand challenge problems like global warming disputes, weather forecasting, epidemic studies, or complex energy network management, can only be solved satisfactorily by accumulated information from a wide area of region, which may cover the whole world [4]. Nonetheless, these global problems also seek solutions suitable to the local region; a globally fit solution is necessary only to provide a better and tested solution for the local area.

To realize the knowledge-sharing global decision making infrastructure, we have to provide two important elements to each local decision making stations: intelligence mechanism to learn from other resources and communication platform to share knowledge and resources. The objective of this paper is to propose an architecture for realization of the global decision making infrastructure by providing intelligence and interconnection to each and between local decision making stations. We devise ways to provide intelligence and adaptability to local stations and to coordinate inter-stations communication via Internet with mobile agent framework.

II. Global Decision Making Architecture

The global decision making (GDM) infrastructure architecture centers on two elements: Internet-based agent software and intelligent decision-making mechanism. In a sense, most decisions and conclusions of global issues are irrelevant and statistically insufficient unless they include all possible data values available. However, the global availability and inclusiveness, which were previously considered almost impossible, can now be realizable if we utilize the vast Internet and pervasive computers. Data monitoring local decision making station (LDMS), connected via Internet, can share information with other LDMS and update the decision rule it previously generated, by the acquired rules and data. This global decision making with local relevance can only be possible by an innovated infrastructure such that it provides intelligence to LDMS for suitable decision making and communication platform for the LDMS, by means of agent through Internet.

A. GDM Infrastructure

In global decision-making situation, local decision-making stations (LDMS) are located in remote and geographically separated places from each other, monitoring the surrounding conditions, making decisions, and drawing conclusions using the monitored parameters. However, since they are physically separated, each LDMS cannot monitor all the parameters that may affect the local decision making processes. In other words, each LDMS deals with only a subset of the whole parameters.

Therefore, each LDMS's decision and classification of a status are suitable only for a limited local situation and leave rooms for improvement. In other cases, some parameters obtained at a LDMS may never be used when other parameters are dominantly used in the decision process. However, these unused data and parameters may be valuable sources for other LDMS which experiences rather low confidence in the rules, caused by lack of parameters. If each LDMS is able to share their expertise and rules and parameters with other LDMSs, then the local decision process will be greatly enhanced and will bring up a conclusion which is not only locally suitable but also globally justified.

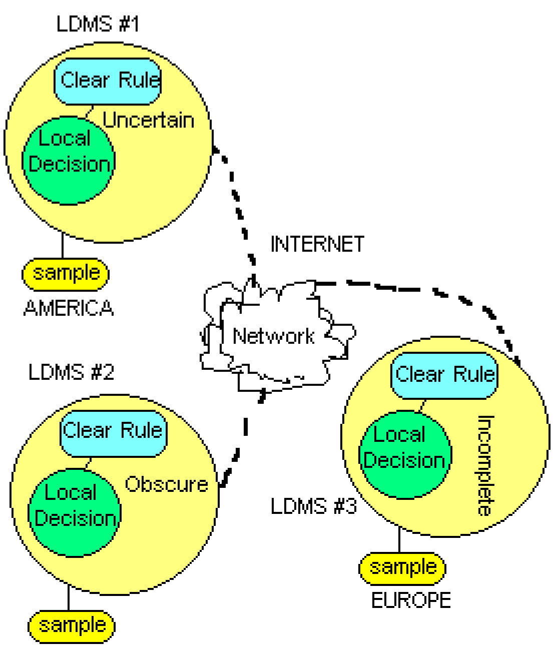

Fig.1. Global Decision-Making Infrastructure with Three Local Decision Making Stations (LDMS).

The GDM architecture shown in Fig. 1 displays why we need the global infrastructure for local stations: one local station may not be able to monitor some parameters that are available only to other local stations. Moreover, a solution will be more relevant when it is drawn from more and diverse data. It is apparent from the illustration that each local rule and decision may be valid only in a limited area and the rule should be tuned and updated from the samples available only from other local stations. Therefore, when the rule confidence is low and there is uncertainty, the local station is in need of consultation with other local stations. Then, the knowledge sharing is initiated from the local station to other stations. Since other local stations may include the samples of different variables, it is highly likely to get valuable information from those local stations.

The suggested architecture envisions an itinerant and autonomous agent software that roams from LDMS to LDMS, shares intelligence and knowledge, and collects information.

B. Intelligent Decision-Making

This section discusses the intelligent decision making system that will be residing in LDMS for rule generation and rule revision. Agent framework is discussed in next section. An intelligent system is characterized by an effective analysis and synthesis of information, and a learning ability that makes the system's reaction appropriate to each situation [5]. One of the most important aspects of an intelligent device is that it has something capable of adjusting to its environment. It has been shown that inductive reasoning with entropy minimum principle can be used as a rule extractor and learner to adapt to the changing environment [6].

Entropy is a measure of the disorder in classifying data. The key goal of entropy minimum principle is to determine the quantity of information in a given data set. This entropy is smallest when the amount of information that we can expect to gain from further observation is least. In entropy minimum state, all of the information has been extracted from the available sample data [7]. This observation is very important to derive a classification rule: when samples are the only source of information, maximum extraction of information is essential for an automated process.

In the classification problem, a rule derivation assumes the values of interests are in one of two classes. Therefore, a certain value must be set to divide the sample events into two classes, for example, planting crop A or crop B. This is called the threshold value. To find the threshold value to separate two classes efficiently, we try and pick a certain value x, in the range of the minimum and the maximum sample values, $[x_{min}, x_{max}]$, that results in the minimum entropy.

Therefore, final form of the generated rule will show the steps of the sequence, and this rule structure resembles a decision tree. For an example, a rule structure derived the from imaginary three weather variables (V_1, V_2, and V_3) is formed as shown in Fig. 2. As indicated in the figure, one important by-product of this decision structure is the rule confidence at each step or branching point (indicated inside the box). The rule confidence plays an essential role for a LDMS to determine if it is confident to draw a conclusion from the rule or it should get some more information from other LDMSs. This process forms a self-learning loop by the aid of itinerant agent that travels the Internet to other LDMSs. Using the data and knowledge obtained by the agent, the LDMS resumes the whole process to obtain a new decision rule.

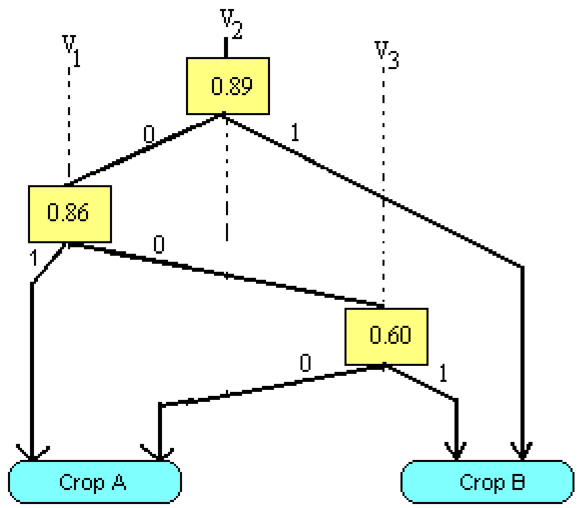

Fig.2. An Example Rule Structure

III. Knowledge Sharing via Agent over Internet

A. Agent and Mobile Agent

An agent is said to be a software or embedded-software that observes its environment through sensors, and acts upon that environment through effectors. An intelligent agent is, therefore, an agent that carries out some set of operations on behalf of a user or another program with some degree of intelligence. Such a system might, like a human assistant, discover new relationships, connections, or concepts independently from the human user, and exploits these in anticipating and satisfying user needs [8].

The type of agent we consider in the proposed architecture is an itinerant agent that, dispatched from a remote or local station, roams among networked servers of another local stations until it accomplishes the tasks of sharing and collecting information and knowledge, and returns to the original station. The coordination of agent-to-LDMS communication is made of the following procedures: the agents travel through the Internet to a LDMS server computer, which recognizes and identifies the agent as sent from a member LDMS, and share resources and knowledge to the server computer.

Once the agent was sent by a local station for a mission to get data or information over the network, it becomes on its way to other local stations, and the sending station forgets about it until the agent returns with results. In this architecture, the local station is not consumed keeping track of what the agent is doing; instead, it allocates its resources to collect data and make local decisions continuously, and updates the decision by the shared knowledge from other distributed LDMS.

To accommodate all the requirements of the agent communication and for an easy implementation of the GDM, an agent framework like Concordia is the first choice of the authors.

B. Mobile Agent and Concordia

Mitsubishi Electric's framework, called Concordia, allows the creation of Mobile Agent programs written in the Java language. These programs use Concordia services to move about a network of distributed machines and to access services available on them [9]. A Concordia system, at its simplest, is made up of a Java VM (virtual machine), a Concordia Server, and at least one agent. The Concordia Server is a Java program which runs there, and at any other local stations on the network where agents may need to travel.

The advantages of developing applications using Concordia are numerous. Since Concordia is written in Java, therefore it's portable, even ubiquitous. It runs on platforms large and small, and integrates easily with existing applications and frameworks. In addition, Concordia agents are securely transmitted across the network, and no additional code is required to provide for secure, distributed operation. The Concordia architecture is shown in Fig.3.

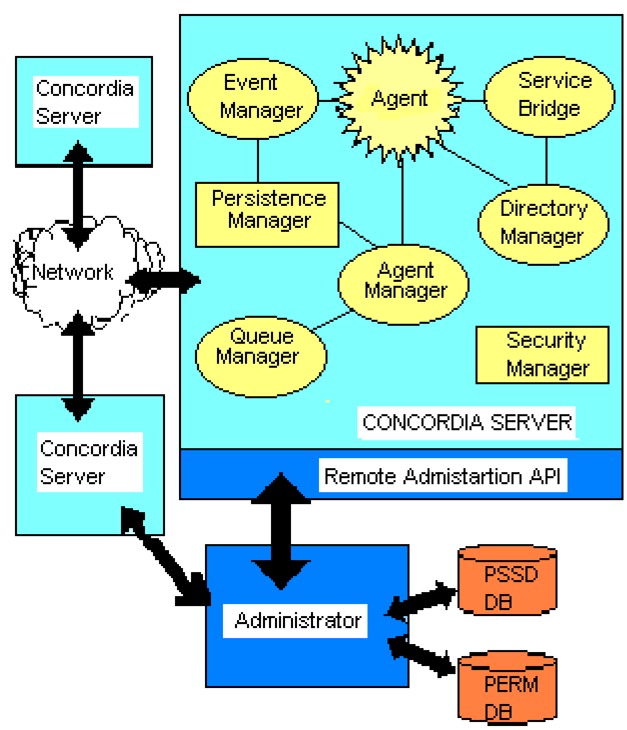

Fig.3. Concoridia Framework Architecture

The Concordia agent is also a Java program that the Concordia Server manages, including its code, data, and movement as shown in Fig.4. As objects, agents are composed of both data (the agent's state) and executable code (the agent's Java bytecodes). Furthermore, moving an agent around a network requires the movement of the agent's state as well as its bytecodes.

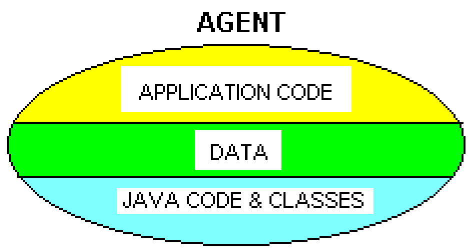

Fig.4. Agent in Concordia

C. GDM Infrastructure with Concordia Frame work

In GDM architecture with Concordia framework, there are many Concordia Servers, one on each of the various LDMSs. The Concordia Servers of the LDMSs are aware of one another and connect on demand to transfer agents. Each LDMS is equipped with Concordia server and Java VM along with data acquisition and inductive reasoning algorithm for minimum entropy, as indicated in Figure 5.

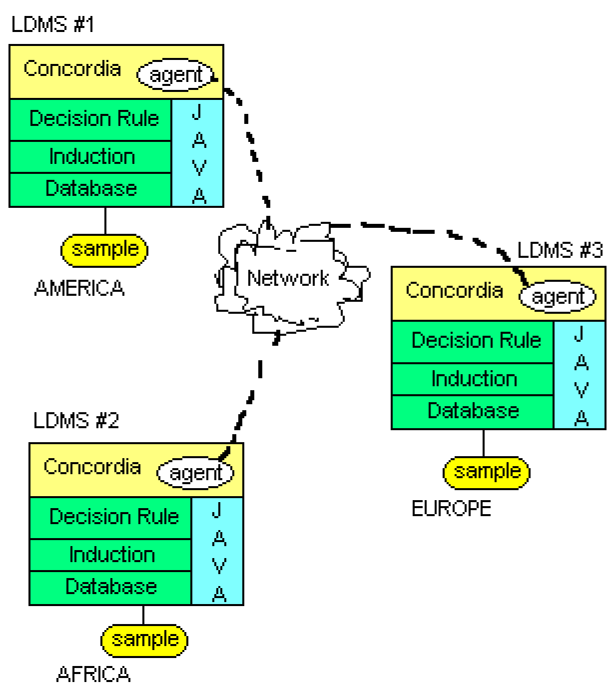

Fig.5. GDM infrastructure with Concordia Framework

The agent in a LDMS initiates the transfer by invoking the Concordia Server's methods. This signals the Concordia Server to suspend the agent and to create a persistent image of it to be transferred. After being transferred, the agent is queued for execution on the receiving LDMS. When the agent again begins executing, it is restarted on the new LDMS according to the method, and it carries with it those objects such as data and decision rules.

The work that the agent performs is determined by the code which it was programmed to execute. The agent starts interactively, by prompting the LDMS for search and retrieval information, then travels to another LDMS to perform the query for rules. As its methods complete, the itinerary causes the agent to be moved to other Concordia nodes of LDMSs.

V. Conclusions

An architecture for global decision making system was proposed. The architecture is composed with distributed LDMSs, which monitor and draw conclusion based on the local data collected around the area. The LDMS is, in hardware, a local server computer or an Internet-embedded microcomputer with data acquisition capability. The operating system software in the LDMS controls the operation of the hardware. An example of GDM architecture using Concordia is illustrated and a full implemenation of it is the immediate future work. Once completed, the GDM infrastructure will substantially enhance the decision making process of global implications in performance and relevance.

VI. References

[1] Pamela Zurer, "Policy focus urged for global-change research," *Chemical and Engineering*, Vol. 72, pp. 39-40, April 25, 1994.

[2] Pamela Zurer, "Refocusing of research on global change urged," *Chemical and Engineering,* Vol. 71. p.8, April 5, 1993.

[3] David King and Daniel O'Leary, "Intelligent Executive Information Systems," *IEEE Expert*, Vol. 11, No. 6, pp. 30-35, December 1996.

[4] R. M. Stein, *Real-Time Multicomputer Software Systems*, Ellis Horwood, 1992.

[5] S. L. Tanimoto, *Elements of Artificial Intelligence*, Rockville, MD: Computer Science Press, 1987.

[6] C. J. Kim and B. D. Russell, "Classification of Faults and Switching Events by Inductive Reasoning and Expert System Methodology," *IEEE Transactions on Power Delivery*, vol.4, no.3, pp. 1631-1637, July 1989.

[7] Ronald Christensen, *Entropy Minimax Sourcebook*, Lincoln, MA: Entropy Limited, 1984.

[8] Katia Sycara, et al, "Distributed Intelligent Agents," *IEEE Expert*, Vol. 11, No. 6, pp. 36-45, December 1996.

[9] "Mobile Agent Computing - A White Paper," Mitsubishi Electric ITA, January 19, 1998.