Industrial Applications of Distributed AI

B. Chaib-draa

Distributed artificial intellegence helps far-flung, often stand-alone, application components work toward a common goal

Most work done in distributed artificial intelligence (DAI) has targeted sensory networks, including air traffic control, urban traffic control, and robotic systems. The main reason is that these applications necessitate distributed interpretation and distributed planning by means of intelligent sensors. Planning includes not only the activities to be undertaken, but also the use of material and cognitive resources to accomplish interpretation tasks and planning tasks. These application areas are also characterized by a natural distribution of sensors and receivers in space. In other words, the sensory data-interpretation tasks and action planning are interdependent in time and space. For example, in air traffic control, a plan for guiding an aircraft must be coordinated with the plans of other nearby aircraft to avoid collisions.

This interdependence results from possible overlaps in intercepted zones. The best way to take advantage of these overlaps to eliminate imprecision and uncertainty is to cooperate with the neighboring groups of sensors to evaluate and to interpret the available data. In addition to applications involving sensory networks, researchers have also investigated using DAI techniques in system automation projects, such as flexible workshops [13], and to help expert systems cooperate in engineering applications [2, 9]. These applications are motivated by the traditional positive aspects of distributed processing systems: performance, reliability, modularity, and resource sharing.

Today, ideas from DAI are becoming important in such research fields as distributed databases, distributed and parallel computing, computer-supported cooperative work, computer-aided design and manufacturing, concurrent engineering, and distributed decision making.

Success Stories

Real-time embedded applications. The Pilot's Associate program was a five-year (1985–1990) ARPA-funded effort to define, design, and demonstrate the application of DAI to helping pilots of advanced fighter aircraft [15]. The system was implemented as a set of five individual expert systems cooperating under the guidance of a sixth expert system—the mission manager. The goal was to provide the pilot with enhanced situational awareness by sorting and prioritizing data, analyzing sensor and aircraft system data,

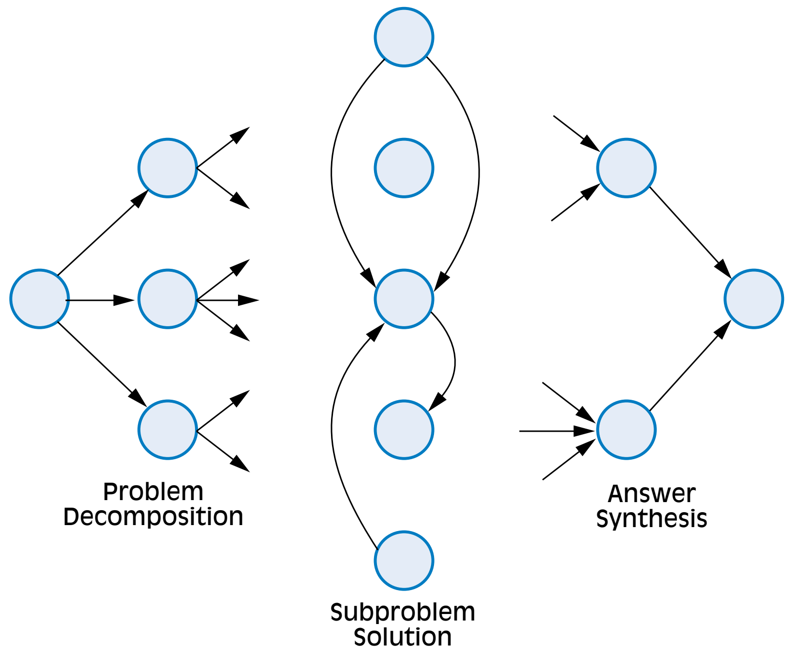

Figure 1. An example of distributed problem solving

distilling the data into relevant information, and managing the presentation of that information to the pilot. From this presentation, corrective measures or alternative plans for achieving mission goals can be developed and presented to the pilot for approval and execution.

Particle accelerator control and electricity distribution. This multiagent system is being developed as part of the ESPRIT-II project ARCHON (Europe's largest DAI project) [8], seeking to create an environment in which cooperative interaction is possible. The system controls a high-energy particle accelerator (for CERN) and was built using an ARCHON prototype system called Generic Rules and Agent Model Testbed Environment (GRATE). GRATE is a general purpose integrative DAI system that contains generic knowledge about cooperation and situation assessment. GRATE's generic knowledge can be divided into three broad categories:

- Controlling local activities;
- Controlling social activities; and
- Assessing the current problem with both local and global considerations.

The CERN laboratories used two earlier expert systems to diagnose problems in the accelerator's operation. These systems were successfully transformed from standalone expert systems to a community of cooperating agents under the control of GRATE. The benefits of a DAI approach to this application include:

- Decomposing a problem into interacting modules, yielding smaller problems that are much easier to tackle;
- Fitting a modularized approach more naturally into the existing organizational structure;
- Working with agents in parallel and producing results faster; and
- Removing some of the drudgery of the operator's job.

Another industrial application from the ARCHON project, called Cooperating Intelligent Systems for DIstribution System Management (CIDIM), is being developed as an aid for control engineers (CEs) who must ensure the electricity supply to electricity users. CIDIM was built to help CEs by automatically providing such services as fault diagnosis, user-driven restoration planning, and security analysis, as well as automatically collating much of the information CEs collate manually by reference to standalone systems. CIDIM consists of 10 agents, some containing conventional programs and some containing expert systems. Pursuing a multiagent approach, CIDIM allows each distinct function to be implemented using the most appropriate model, whether expert system, database, or conventional software.

Resource allocation in distributed factory scheduling. Sycara and coworkers at Carnegie Mellon University [17] view distributed scheduling as a process carried out by a group of agents, each characterized by:

- Limited knowledge of the environment;
- Limited knowledge of the constraints and intentions of other agents; and
- A limited amount of resources to produce a system solution.

A DAI Tutorial

With the steady progress of research in information technology over the past decade, it is now clear that many classes of complex problems cannot be solved in isolation. Research advances in DAI, however, have opened up many new avenues for solving such problems. Generally, the DAI field aims to construct systems of intelligent entities that interact productively with one another. More precisely, DAI is concerned with studying a broad range of issues related to the distribution and coordination of knowledge and actions in environments involving multiple entities [3]. These entities, called agents, can be viewed collectively as a society. The agents work together to achieve their own goals, as well as the goals of the society as a whole.

A major distinction in the DAI field is between research in distributed problem solving (DPS) and research in multiagent systems (MAS). Early DPS work concentrated on applying the power of networked systems to a problem as exemplified by the three-phase nomenclature in Figure 1. In the first phase, the problem is decomposed into subproblems. The decomposition process may involve a hierarchy of partitionings. The second phase involves solving the kernel problems through agents that communicate and cooperate as needed. Finally, the results are integrated to produce an overall solution. DPS work also addresses the robustness available from multiple sources of expertise, multiple views, and multiple capabilities [7]. Generally, multiple views refer to distributed applications, such as air traffic control and urban traffic control. In summary, all DPS work emphasizes the problem and how to get multiple intelligent entities (programmed computers) to work together to solve it in a efficient manner [7].

In MAS, the agents are autonomous, possibly preexisting, and typically heterogeneous. Research here is concerned with coordinating intelligent behaviors among a collection of autonomous agents—how these agents coordinate their knowledge, goals, skills, and plans to take action and to solve problems. In this environment, the agents may be working toward a single global goal or toward separate individual goals that interact. Like solvers in DPS, agents in MAS might share knowledge about tasks and partial works. Unlike the DPS approach, however, they must also reason about the process of coordination among the agents. Coordination is central to multiagent systems; without it, the benefits of interaction vanish, and the behavior of the group of agents can become chaotic.

Unlike DPS work, where the emphasis is on the problem, MAS focuses on the agent and its characteristics in multiagent environments. Three possible views of the relationship between MAS and DPS were recently identified [7]:

- DPS is a subset of MAS;
- MAS provides a substrate for DPS; or
- MAS and DPS are complementary research agendas.

The DAI community still debates which of these views is correct.

A central issue in DAI is how to allow autonomous agents to model each other to reason about the activities of other agents. Reasoning about other agents allows agents to coordinate their activities to produce elaborate but coherent solutions. Coordination can be analyzed in terms of agents performing interdependent plans that achieve goals [10,11]. The different system components—goals, agents, plans, and interdependencies—are associated with the coordination process. Table 1 summarizes these components and their associated coordination processes. All four components are necessary for a situation to be analyzed in terms of coordination. Indeed, it does not make sense to refer to a DAI system as being coordinated if no activities are performed or if the activities are completely independent.

Why choose a DAI approach? There are four main reasons:

- We need to address the necessity of treating distributed knowledge in applications that are geographically dispersed, such as sensor networks, air traffic control, and cooperation between robots. DAI can also be used to tackle large and complex applications;
- DAI can aid our attempts to extend human-machine cooperation.
- DAI can yield a new perspective in knowledge representation and problem solving through richer scientific formulations and more realistic representations in practice.
- DAI can shed new light on the cognitive sciences and on AI.

Table 1. Components of Coordination

Components	Associated Processes
Goals	Identifying goals, including goal selection
Agents	Mapping goals to agents, including goal allocation and negotiation
Plans	Mapping plans to goals, including planning
Interdependencies	Managing interdependencies, including resource allocation, sequencing, and synchronizing

Many agents can share these resources for making local decisions about assigning resources to specific activities at specific time intervals. Therefore, a complete order schedule is cooperatively created by incrementally merging agents' partial schedules. Cooperation is needed because no single agent has a global system view. This cooperation arrives at global solutions by interleaving local computations with the information exchange among agents. The system goal is to find schedules that not only are feasible but also optimize a global objective, such as minimizing order tardiness or work in process. The global objective to be optimized reflects the quality of the schedule produced.

Telecommunications systems. Using DAI techniques in telecommunications systems seems inevitable when we consider two trends in the design of such systems: distribution of functionality and incorporation of intelligence software that implements sophisticated services and decision making. The DAI literature includes a number of approaches that address the telecommunications field [18]. A noteable example is a system called LODES, which has been tried on operational networks. LODES detects and diagnoses problems in a segment of a local-area network [16]. Different LODES system copies—each acting as an agent—can monitor and manage different network segments. LODES includes components that let each agent cooperate with other agents. Although LODES was developed primarily as a research testbed, it has been tried on operational networks with some success. LODES's designers chose a distributed approach over a centralized approach to tap the physical and functional distribution of networks. A distributed approach also enables local problem solving, facilitating the communication of the results of analysis, rather than all the information needed for the process.

Database technologies for service order processing. Singh and Huhns from MCC [14] defined a distributed agent architecture for intelligent workflow management that functions on top of Carnot's environment. Their system consists of four agents that interact to produce the desired behavior, as well as databases that include the relevant data and the application programs that execute on them. The four agents are:

- The graphical-interaction agent;
- The transaction-scheduling agent;
- The schedule-processing agent; and
- The schedule-repairing agent.

Applications are executed by the schedule-processing agent. If the agent encounters an unexpected condition, such as a task failure, it notifies the transaction-scheduling agent, which asks the schedule-repairing agent for advice on how to fix the problem. Advice can involve how to restart a transaction, how to abort a transaction, and other operations. Finally, the graphical-interaction agent queries the systems to help users. Singh and Huhns implemented a prototype that executes on top of a distributed computing environment to help a telecommunications company provide a service that requires coordination among many operation support systems and network elements.

Applications Almost Here

Concurrent engineering. The Palo Alto Collaborative Testbed (PACT) is a concurrent engineering infrastructure encompassing multiple sites, subsystems, and disciplines [6]. Through PACT, investigators—from Stanford University, Lockheed Palo Alto Research Labs, and Enterprise Integration Technologies—are examining the technological and sociological issues of building large-scale distributed concurrent-engineering systems. PACT experiments have looked into building an overarching framework along three dimensions:

- Cooperative development of interfaces, protocols, and architecture;
- Sharing of knowledge among systems that maintain their own specialized knowledge bases and reasoning mechanisms; and
- Computer-aided support for negotiating and making decisions in concurrent engineering projects.

The PACT architecture is based on interacting programs, or agents, that encapsulate engineering tools. The agent interaction relies on three things:

- Shared concepts and terminology for communicating knowledge across disciplines;
- A common language for transferring knowledge among agents;
- A communication and control language that enables agents to request information and services.

This technology allows agents working on different aspects of a design to interact at the knowledge level, sharing and exchanging information about the design, independent of the format in which the information is encoded internally.

PACT is an ongoing collaboration. Its activities are intended to expand PACT into a broad-based engineering infrastructure. To achieve this, authors are upgrading the prototype software, which now handles low-level message passing between agents, to improve reliability, scalability, and ease of use. The next version will build on a commercial substrate, such as the Object Management Group's Common Object Request Broker Architecture (CORBA) [12], that can support multicast protocols in environments containing thousands of agents. In the next version, simulation and analysis will be transformed into generic engineering services with published interfaces and ontologies and made available on the Internet 24 hours a day.

Urban traffic control. Urban traffic is generally a highly interactive activity among various agents, which can include people, such as drivers, police officers, and pedestrians, and machines, such as vehicles and traffic-lights, that continuously adjust their actions to prevent conflicts like traffic jams and crashes. Today, many multiagent approaches can help investigate this task, particularly in Canada and Europe [1, 4].

Conclusions

This article argues that a DAI approach can be used to cope with the complexity of industrial applications. DAI techniques are beginning to have a broad impact; the current introduction of these techniques by an ESPRIT project, a Palo Alto consortium, ARPA, Carnegie Mellon University, MCC, and others are good examples. In the near future, other industrial products will emerge from the application of DAI techniques to other domains, including distributed databases, computer-supported cooperative work, and air traffic control. An important advantage of a DAI approach is the ability to integrate existing standalone knowledge-based systems. This factor is important because software for industrial applications is often developed in an ad hoc fashion. Thus, organizations possess a large number of standalone systems developed at different times by different people using different techniques. These systems all operate in the same physical environment, all have expertise that is related but distinct, and all could benefit from cooperation with other such standalone systems.

References

1. Bomarius, F. A. Multiagent approach towards modeling urban traffic scenarios. Tech. Rep. RR-92-47, DFKI-GmbH, Germany, 1992.
2. Bond, A.H. The cooperation of experts in engineering design. In *Distributed Artificial Intelligence*, Vol. 2. L. Gasser and M.N. Huhns, Eds. Morgan Kaufmann, Los Altos, Calif., 1989, 463–484.
3. Bond, A.H., and Gasser, L., Eds. *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann, Los Altos, Calif., 1988.
4. Chaib-draa, B. Coordination between agents in routine, familiar and unfamiliar situations. Rapport de recherche DIUL-RR-9401 du département d'informatique, Univ. Laval, Sainte-Foy, PQ, Canada, 1994.
5. Cockburn, D. Cooperating intelligent systems for electricity distribution. In *Proceedings of Expert Systems* 1992, Cambridge, UK, 1992.
6. Cutkosky, M.R., Engelmore, R.S., Fikes, R.E., Genesereth, M.R., Gruber, T.R., Mark, W.S., Tenenbaum, J.M., and Weber, J.C. PACT: An experiment in integrating concurrent engineering systems. *IEEE Comput. 26*, 1(1993), 28–37.
7. Durfee, E., and Rosenschein, J.S. Distributed problem solving and multi-agent systems: Comparisons and examples. In *Proceedings of the 13th International DAI Workshop*, (Seattle, Wash.), 1994.
8. Jennings, N.R. Cooperation in Industrial Multi-Agent Systems. *World Scientific Series in Computer Science, Vol. 43*. 1994.
9. Lesser, V.R., Pavlin, J., and Durfee, E. Approximate processing in real-time problem solving. *AI Magazine 9*, 1 (1988), 81–96.
10. Malone, T.W., and Crowston, K. The interdisciplinary study of coordination. *ACM Comput. Surv. 26*, 1, (1994), 87–120.
11. Martial, F.V. Coordinating Plans of Autonomous Agents. *Lecture Notes in AI*. Springer-Verlag, Berlin, 1991.
12. Object Management Group. *The Common Object Request Broker: Architecture and specification*. OMG Document 391.12.1, Framingham, Mass., 1991.
13. Parunak, H.V.D. Manufacturing experience with the contract-net. In *Distributed Artificial Intelligence, Vol. 2*. L. Gasser and M.N. Huhns, Eds. Morgan Kaufmann, Los Altos, Calif., 1989, 285–310.
14. Singh, M.P., and Huhns, M. N. Automating workflows for service order processing. *IEEE Expert 9*, 5 (May 1994), 19–23.
15. Smith, D., and Broadwell, M. The pilot's association—An overview. In *Proceedings of the SAE Aerotech Conference*, Los Angeles, Calif., 1988.
16. Sugawara, T., and Murakami, K. A multiagent diagnostic system for internetwork problems. In *Proceedings of the Internet Society — INET 92 1992 International Networking Conference*, Internet Society, Reston, Va., 1992, pp. 317–325.
17. Sycara, K.P., Roth, S.F., Sadeh, N., and Fox, M. Resource allocation in distributed factory scheduling. *IEEE Expert 6*, 1 (June 1991), 29–40.
18. Weihmayer, D. and Veltguijsen, H. Application of distributed AI and cooperative problem solving to telecommunications. In *Proceedings of the 13th International DAI Workshop* (Seattle, Wash., 1994).

About the Author:
B. CHAIB-DRAA is an associate professor of computer science in the computer science department at Laval University. His research interests include multiagent systems, distributed AI, and real-time systems.

Author's Present Address: Département d'Informatique, Pav. Pouliot, Fac de Sciences, Université Laval, Sainte-Foy, QC, G1K 7P4, Canada. email: chaib@ift.ulaval.ca