Experiences Building a Distributed Sensor Network

Victor Lesser

Department of Computer Science, University of Massachusetts/Amherst
Amherst, MA 01003-9264 USA
Lesser@cs.umass.edu

1 Extended Abstract

A central challenge in building advanced sensor networks will be the development of distributed and robust control for such networks that scales to thousands of intelligent sensors [8]. Appropriately structuring where and when control and interpretation activities are done is key to the effective operation of the network. This structuring must be adaptive to changing network conditions such as new sensors being added, existing sensors malfunctioning, and communication and processor resource modifications. Together with this adaptive re-structuring of long-term roles and responsibilities, there is also a need for short-term adaptivity related to the dynamic allocation of sensors. This involves allocating the appropriate configuration of sensing/processing resources for effectively sensing the phenomena but also the resolution of conflicting resource assignments that may occur when there are multiple phenomena occurring in the environment that need to be tracked concurrently. More generally, this structuring can be thought of as organizational control. Organizational control is a multilevel control approach in which organizational goals, roles, and responsibilities are dynamically developed, distributed, and maintained to serve as guidelines for making detailed operational control decisions by the individual agents. The parameters guiding the creation and adaptation of the organization can have a dramatic impact on the performance of the sensor network. We have recently completed work on a small-scale sensor network (approximately 36 low-cost, adjustable radar nodes) for multi-vehicle tracking [5,7], that exemplifies in a simplified form many of the issues discussed above (see Fig. 1). This lecture will discuss how we approached the design of the sensor network and what technologies we needed to develop.

The sensor network hardware configuration consists of sensor platforms that have three scanning regions, each with a 120-degree arc encircling the sensor (see Fig. 1, top left). Only one of these regions can be used to perform measurements at a time. The communication medium uses a low-speed, unreliable, radio-frequency (RF) system over eight separate channels. Messages cannot be both transmitted and received simultaneously regardless of channel assignment, and no two agents can transmit on a single channel at the same time without causing interference. The sensor platforms are capable of locally hosting one or more processes, which share a common CPU (in this case a commodity PC and signal processing hardware). The goal of this application is to track one or more targets that are moving through the sensor environment (in this case model railroad trains traveling on railroad tracks whose pattern is unknown, see Fig. 1: top right). The radar sensor measurements consist of only amplitude and

Y. Xiang and B. Chaib-draa (Eds.): AI 2003, LNAI 2671, pp. 1-6, 2003.

frequency values, so no one sensor has the ability to precisely determine the location of a target by itself. The sensors must therefore be organized and coordinated in a manner that permits their measurements to be used for triangulation.

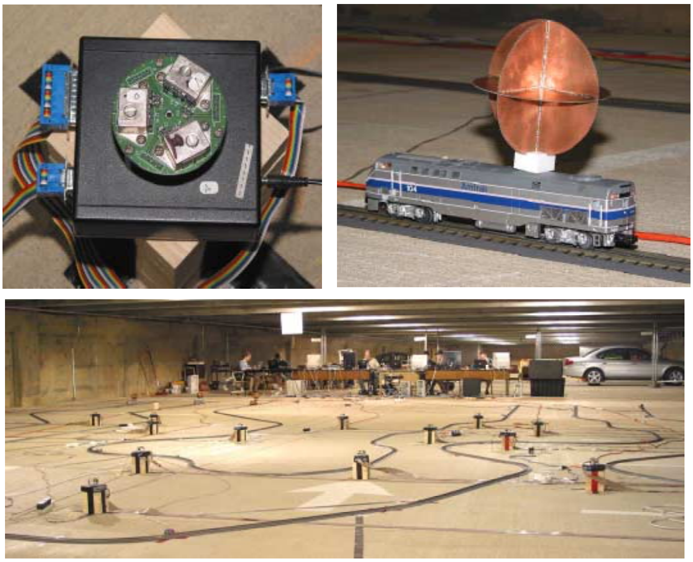

Fig. 1. Sensor Network. Top left: radar unit with three sensing heads. Top right: vehicle being tracked. Bottom: an example configuration with 35 sensors and 3 vehicles

The need to triangulate a target's position requires frequent, closely coordinated actions amongst the agents, ideally three or more sensors performing their measurements at the same time. In order to produce an accurate track, the sensors must therefore minimize the amount of time between measurements during triangulation, and maximize the number of triangulated positions. Ignoring resources, an optimal tracking solution would have all agents capable of tracking the target taking measurements at the same precise time as frequently as possible. Restrictive communication and computation, however, limits our ability to coordinate and implement such an aggressive strategy. Low communication bandwidth hinders complex coordination and negotiation, limited processor power prevents exhaustive planning and scheduling, and restricted sensor usage creates a trade-off between discovering new targets and tracking existing ones. These considerations led us to an overall design philosophy that includes the use of an agent organization and satisficing behavior in all aspects of problem solving.

Our approach is built upon a soft, real-time agent architecture called SRTA, which we constructed as part of this effort [6]. The SRTA architecture provides a robust planning, scheduling and execution subsystem capable of quantitatively reasoning over deadlines and resource constraints. This provides a useful layer of abstraction, enabling the agent's higher level reasoning components to operate at a more tractable level of granularity, without sacrificing fine-grained control and reactivity.

Built upon this agent architecture is a virtual agent organization based on partitioning the environment into geographically self-contained *sectors* each with its own local management. Each of these sectors has a *sector manager*, a role in the organization which has several responsibilities associated with information flow and activity within the sector. Among these responsibilities is the dissemination of a scan schedule to each of the sensors in its sector, specifying the rate and frequency that should be used to scan for new targets. This information is used by each sensor to create a description of the scanning task, which is in turn used by the SRTA architecture to schedule local activities. When a new target is detected, the sector manager selects a *track manager*, a different organization role responsible for tracking that target as it moves through the environment. This allocation process uses an abstract view of what activities are presently being conducted in the sector to make a choice that load balances processor and communication requirements. Track manager activities entail estimating future location and heading, gathering available sensor information, requesting and negotiating over the sensors, and fusing the data they produce. Upon receipt of such a commitment to perform tracking, a sensor takes on a *data collection* role. Like the scan schedule, these commitments are used to generate task descriptions used by SRTA to schedule local activities. If conflicting commitments are received by a sensor that imply that the agent has been asked to perform multiple concurrent data collection roles, SRTA will attempt to satisfy all requests as best possible. This provides a window of marginal quality in which a conflict can be detected and then potentially resolved through negotiation with the competing agent to find an equitable long-term solution. As data is gathered, is it fused and interpreted to estimate the target's location, which allows the process to continue. We call this a virtual agent organization since a particular sensor/processor node may be multiplexing among different roles, e.g. sector manager and data collection. The SRTA architecture does the detail scheduling of activities associated with different roles based on their priority and deadline. The planning and scheduling ability of the SRTA architecture also allows us to approach the dynamic allocation of sensors to tracking tasks at an abstract level. Commitments made at this abstract level are then mapped into detail allocations of sensor resources and data processing activities.

The organizational structuring we have discussed so far involves setting up long-term patterns of control and information processing. There is also a need for setting up more short-term and dynamic patterns involving the allocation of groups of sensors (sensor platforms and sensor heads) to the tracking of the movement of a specific vehicle. Since sensor heads have limited sensing range and orientation and the vehicle is moving, this allocation process must be repeated as the current group of sensors become inappropriate for tracking the vehicle. Further, the need for this allocation process may be occurring simultaneously in different parts of the sensor network when there are multiple vehicles moving in the environment. Finally, this allocation process is intimately tied with information fusing activities that are tracking the current locations of vehicles and predicting where they are likely to be going. The real-time ability to do this prediction accurately is key to having sensing resources appropriately allocated to sense the vehicle when it arrives in their sensing region. Resource contention is introduced when more than one target enters the viewable range of the same sensor platform.

This type of resource allocation can be too complex and time consuming to perform in a centralized manner when the environmental characteristics are both distributed and dynamic, because the costs associated with continuously centralizing the necessary information are impractical. Negotiation, a form of distributed search [12] has been viewed as a viable alternative to handling complex searches that include multi-linked interacting subproblems [1]. Researchers in this domain have focused primarily on resource allocation scenarios that are formulated as distributed constraint satisfaction problems [11,13]. In our approach, we extend this classic formulation in two ways. First, we introduce soft, real-time constraints on the protocol's behavior. These require the negotiation to adapt to the remaining available time, which is estimated dynamically as a result of emerging environmental conditions. Second, we reformulate the resource allocation task as an optimization problem, and as with the distributed Partial Constraint Satisfaction Problem (PCSP) [2,3,4], we use constraint relaxation techniques to find a conflict-free solution while maximizing the social utility of the tracking agents. Of course, when more than one tracking agent desires a particular resource these two goals may contradict each other.

Our approach, called SPAM (The Scalable Protocol for Anytime Multi-level negotiation [9,10]), is a real-time, distributed, mediation-based negotiation protocol that takes advantage of the cooperative nature of the agents in the environment to maximize social utility. By mediation based, we are referring to the ability of each of the agents to act in a mediator capacity when resource conflicts are recognized. As a mediator, an agent gains a localized, partial view of the global allocation problem and makes suggestions to the allocations for each of the agents involved in the mediation. This allows the mediator to identify over-constrained subproblems and make suggestions to eliminate such conditions. In addition, the mediator can perform a localized arc-consistency check, which potentially allows large parts of the search space to be eliminated. Together with the fact that regions of mediation overlap, the agents rapidly converge on solutions that are in most cases good enough and fast enough. Overall, the protocol has many characteristics in common with distributed breakout [14], particularly its distributed hill-climbing nature and the ability to exploit parallelism by having multiple negotiations occur simultaneously.

In summary, the use of a sophisticated agent architecture (that includes capabilities for planning and scheduling) and distributed resource allocation mechanisms for short-term agent control and resource allocation, together with an organization structure for long-term agent control, create a powerful paradigm for building the next generation of large scale and intelligent sensor networks. More generally, we see these techniques as applicable to the building of advanced multi-agent applications.

Acknowledgements

This represents the combined work of a number of researchers in the Multi-Agent Systems Laboratory at the University of Massachusetts. The main contributors were Bryan Horling, Roger Mailler, Jiaying Shen and Dr. Regis Vincent.

This effort was sponsored in part by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory Air Force Materiel Command, USAF, under agreements number F30602-99-2-0525 and DOD DABT63-99-1-0004. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. This material is also based upon work supported by the National Science Foundation under Grant No.

IIS-9812755. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), Air Force Research Laboratory or the U.S. Government.

References

[1] S.E. Conry, K.Kuwabara, V.R. Lesser, and R.A. Meyer. Multistage negotiation for distributed constraint satisfaction. IEEE Transactions on Systems, Man, and Cybernetics, 21(6), Nov. 1991.

[2] E.C. Freuder and R.J. Wallace. Partial constraint satisfaction. Artificial Intelligence, 58(1-3):21-70, 1992.

[3] K. Hirayama and M. Yokoo. Distributed partial constraint satisfaction problem. In G. Smolka, editor, *Principles and Practice of Constraint Programming* (CP-97), volume 1330 of Lecture Notes in Computer Science, pages 222-236. Springer-Verlag, 1997.

[4] K. Hirayama and M. Yokoo. An approach to overconstrained distributed constraint satisfaction problems: Distributed hierarchical constraint satisfaction. In *International Conference on Multi-Agent Systems* (ICMAS), 2000.

[5] B. Horling, R. Vincent, R. Mailler, J. Shen, R. Becker, K. Rawlins, and V. Lesser. Distributed sensor network for real time tracking. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 417-424, 2001.

[6] B. Horling, V. Lesser, R. Vincent, and T. Wagner. The soft real-time agent control architecture. University of Massachusetts/Amherst Computer Science Technical Report 2002-14, 2002.

[7] B. Horling, R. Mailler, J. Shen, R. Vincent, V. R. Lesser. Using Autonomy, Organizational Design and Negotiation in a Distributed Sensor Network. Accepted for publication in *Distributed Sensor Nets: A Multiagent Perspective*.

[8] V. Lesser, C. Ortiz, and M. Tambe. *Distributed Sensor Networks: A multiagent perspective*. Kluwer Publishers, 2003 (to appear).

[9] R. Mailler, R. Vincent, V. Lesser, J. Shen, and T. Middlekoop. Soft real-time, cooperative negotiation for distributed resource allocation. In *Proceedings of the 2001 AAAI Fall Symposium on Negotiation*, 2001.

[10] R. Mailler, V. Lesser, B. Horling. Cooperative Negotiation for Soft Real-Time Distributed Resource Allocation. Accepted for publication in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Melbourne, Australia, 2003. Also available as University of Massachusetts/Amherst Computer Science Technical Report 2002-49.

[11] P. J. Modi, H. Jung, M. Tambe, W.-M. Shen, and S. Kulkarni. Dynamic distributed resource allocation: A distributed constraint satisfaction approach. In J.-J. Meyer and M. Tambe, editors, Pre-proceedings of the Eighth International Workshop on Agent Theories, Architectures, and Languages (ATAL-2001), pages 181-193, 2001.

[12] T. Moehlman, V. Lesser, and B. Buteau. Decentralized negotiation: An approach to the distributed planning problem. *Group Decision and Negotiation*, 1(2):161-192, 1992.
[13] M. Yokoo, E. H. Durfee, T. Ishida, and K. Kuwabara. The distributed constraint satisfaction problem: Formalization and algorithms. *Knowledge and Data Engineering*, 10(5): 673-685, 1998.
[14] M. Yokoo and K. Hirayama. Distributed breakout algorithm for solving distributed constraint satisfaction problems. In *International Conference on Multi-Agent Systems* (ICMAS), 1996.