

# Global Sensor Web Coordination and Control in a Multi-agent System

**John S. Kinnebrew**

Department of Electrical Engineering and Computer Science & ISIS,  
Vanderbilt University, Nashville, TN 37203, USA  
john.s.kinnebrew@vanderbilt.edu

## 1 A Multi-agent System for Sensor Webs

The availability of a variety of sensors around the globe and the ability to quickly make this data available in remote locations give today's scientists an unprecedented advantage in studying and predicting weather, natural disasters, and climate change. Selecting and coordinating an appropriate subset of these heterogeneous and distributed sensors and associated computational resources for such large-scale tasks is complex. For example, sensors must first be located and may have to be reconfigured and recalibrated to collect the needed data. Moreover, large-scale tasks may require the cooperation of multiple component sensor networks. Coordination among these distributed sensor and computational resources is required for efficient and effective execution of complex tasks.

In a joint project with the Lockheed Martin Advanced Technology Center, we have developed an initial design for the *Multi-agent Architecture for Coordinated Responsive Observations* (MACRO) [Suri *et al.*, 2007]. MACRO provides a powerful computational infrastructure for enabling the deployment, configuration, and operation of large sensor webs that are composed of many constituent sensor networks. MACRO is divided into a broad, two-level hierarchy of agents: (1) the *mission level*, where agents interact with users to allocate high-level science tasks to sensor net resources and coordinate to create plans and schedules to achieve these tasks, and (2) the *resource level*, where local server and field agents employ decision-theoretic planning and scheduling to translate tasks into actions and application deployments related to sensor configuration and data collection, analysis, and transmission [Suri *et al.*, 2007].

My dissertation research focuses on the MACRO mission level, which is comprised of *User agents*, *Mission agents*, and *Broker agents*. User agents are the primary providers of high-level tasks to the system. Typically the User agents are interfaces to mission scientists or wrappers for applications and legacy systems (*e.g.*, weather modeling and natural disaster prediction applications) that can request execution of sensor web tasks. The Mission agents take on overlapping roles in the allocation and planning aspects of the sensor web, and their interactions with each other and the User agents ultimately provide the global resource allocation and distributed planning/scheduling necessary for coherent operation of the sensor web. Broker agents are responsible for matchmaker

services between User agent requests and applicable Mission agents. They also mediate allocation negotiations between User agents and Mission agents. As opposed to most User and Mission agents, Broker agents are implemented by the MACRO system designers, providing them with a way to directly influence overall system performance.

## 2 Research Challenges and Solutions

Sensor web users may have differing priority in the system and consequently may be assigned a different "share" of sensor web resources. The finite resources available to execute sensor web tasks ensure that, at times, the tasks of its heterogeneous users will be in conflict for control of certain resources. This research addresses the sensor web coordination problem of efficiently achieving fair, high-utility allocations of complex tasks to sensor web resources.

The complex nature of tasks and plans at the mission level of a sensor web MAS makes hierarchical analysis important for dealing with this complexity, both for problem/task representation by domain experts and for coordinated planning among multiple agents. MACRO employs a modified implementation of the Task Analysis, Environment Modeling, and Simulation (TÆMS) [Horling *et al.*, 1999] language, which provides a hierarchical task tree representation for multi-agent planning and scheduling. Further, MACRO incorporates the OGC SensorML [Botts and others, 2007] standard for sensors and data processing with TÆMS hierarchically-decomposable tasks to provide standardized descriptions of task/subtask requirements and effects.

For task allocation, MACRO extends the Contract Net Protocol (CNP) [Smith, 1980], which uses computationally simple, single auctions. The CNP is a well-studied multi-agent negotiation mechanism for distributed task/resource allocation, where agents have different, or even changing, resources, capabilities, preferences, and performance. In particular, the CNP and its derivatives can allow effective allocation of tasks/resources without restricting the criteria individuals agents can apply to determine their preference for tasks or bids.

One of the major advantages of using the contract net protocol for allocation is that it provides flexibility in defining the utility or preference structure that User agents employ for evaluating task bids, which depends on the needs of the people or application they represent. However, this flexibility

also presents a challenge in designing an allocation architecture that provides fair and efficient (high-utility) task allocations. The standard contract net protocol does not provide a method to influence allocations other than by the definition of task/bid creation and preferences in announcers and bidders. This is insufficient for MACRO because User agents are designed and implemented independently of the MACRO infrastructure. Therefore, a significant challenge in achieving appropriate allocations is providing a framework in which system designers can influence the contract net negotiations to yield allocations that provide fair and efficient use of resources.

MACRO extends the traditional contract net protocol with Broker agents to influence the allocation of tasks. In addition to providing an efficient matchmaking/locator service (*i.e.*, determining Mission agents capable of performing all or part of an announced task and forwarding messages appropriately), Broker agents are responsible for assigning utility values to task announcements based on User agent priority and past system performance. These task valuations help guide Mission agents' choice of tasks on which to bid in the contract net. To calculate these values, the announcing User agent first assigns a personal utility value to the task, such that the sum of currently announced tasks and previously contracted tasks, over a given time window, is equal to a pre-defined value, such as 100. This utility value is then multiplied by the agent's pre-assigned fair share of the sensor web to define a system utility value for the announced task. The Broker then weights this system utility value by the inverse of the agent's current deviation from fair share of sensor web resources, based on all contracts assigned over the current time window. The resulting value assigned by the Broker to an announced task represents a trade-off between the importance of efficiency (system-wide utility) and fairness.

Another difficulty in applying the contract net protocol to a sensor web MAS is its emphasis on two-party contracts. In MACRO, a User agent may require the resources of multiple Mission agents to achieve a complex task. Therefore, MACRO uses subcontracting, allowing resources of multiple Mission agents to be assigned to a task through a primary contract and additional subcontracts between Mission agents. However, with multiple possible decompositions of a task, there may be a large number of Mission agents who could be the primary contractor for the task, and each Mission agent could choose among various sets of subcontractors. If each of these Mission agents announces subtasks and receives bids before bidding on the task, a great deal of communication and computation must be performed before each can generate a complete bid on the task. On the other hand, if each Mission agent bids on the task before soliciting subcontracts, there is a great deal of uncertainty in the accuracy of resulting bids. Finding a trade-off between subcontracting overhead and completeness of bids for decomposable tasks presents a major challenge for efficient and effective allocation.

The MACRO CNP addresses this challenge by separating initial bidding from final bidding, extending the approach in the two-phase contract net protocol proposed by Akinine et al. [Akinine et al., 2004]. This variation of the CNP breaks the contract net negotiations into an initial pre-

commitment phase and a final commitment phase. During pre-commitment, the task is announced and initial bids are made. The announcing agent then pre-accepts what it determines to be the best initial bid, beginning the commitment phase. In the MACRO CNP the pre-accepted agent can then announce subtasks it may not be able to, or want to, perform. After receiving bids on its announced subtasks, it makes a final bid on the task to the original announcing agent. This final bid includes relevant information on potential subcontracts and better estimates of applicable quality measures and time-to-completion. The announcing agent then chooses whether to accept the final bid or pre-accept a different agent.

MACRO employs this two-phase CNP, including subcontracting, to limit subtask negotiations to a subset of initial bidders. Rather than allowing all potential contractors for the high-level task to announce subtasks, the MACRO CNP allows Mission agents to announce subtasks only after receiving a pre-accept from the User agent that announced the high-level task. Further, by limiting the number of pre-accepts a User agent can issue for a given task announcement, the MACRO CNP significantly reduces the total amount of communication and computation overhead in the contract net. The preliminary results of our subtask allocation experiments verify the efficiency of MACRO limited pre-commitment subcontracting in terms of scalability, which was linear for the major factors of number of subtasks, number of task decompositions, and agent capability overlap. Further, these results illustrate significant reduction in messages over the baseline, pre-bid subcontracting and identify a reasonable upper bound on pre-accept cutoffs for a variety of system configurations and operating conditions.

## References

- [Akinine et al., 2004] S. Akinine, S. Pinson, and M.F. Shakun. An Extended Multi-Agent Negotiation Protocol. *Autonomous Agents and Multi-Agent Systems*, 8(1):5–45, 2004.
- [Botts and others, 2007] M. Botts et al. Sensor Model Language (SensorML). Technical Report OpenGIS Implementation Specification Document 07-000, Open Geospatial Consortium, July 2007.
- [Horling et al., 1999] B. Horling, V. Lesser, R. Vincent, T. Wagner, A. Raja, S. Zhang, K. Decker, and A. Garvey. The TAEMS White Paper. *Multi-Agent Systems Lab, University of Massachusetts*, 1999.
- [Smith, 1980] R.G. Smith. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, 29(12):1104–1113, 1980.
- [Suri et al., 2007] Dipa Suri, Adam Howell, Douglas C. Schmidt, Gautam Biswas, John Kinnebrew, Will Otte, and Nishanth Shankaran. A Multi-agent Architecture for Smart Sensing in the NASA Sensor Web. In *Proceedings of the 2007 IEEE Aerospace Conference*, Big Sky, Montana, March 2007.