COOPERATIVE MULTI-AGENT INFORMATION GATHERING *

Keith Decker[†], Victor Lesser, M. V. Nagendraprasad, and Thomas Wagner
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
DECKER@CS.UMASS.EDU

Overview

With the proliferation of electronically available information, an additional burden has been placed on the implementors of information gathering systems. The set of data that represents the best response to a user's query may be the aggregation of data acquired from distributed, heterogeneous information sources. We have begun a project to build a cooperative multi-agent information gathering system in the initial domain of computer product queries. Top level user queries drive the creation of partially elaborated information gathering plans, resulting in the employment of multiple cooperative agents for the purpose of achieving goals and subgoals within those plans. The agent will use existing information resources available on the Internet such as newsgroups, archives, magazine databases, and corporate WWW sites. Such a system takes a "distributed problem solving" approach where the agents are autonomous, socially situated, and can react to a dynamically changing environment. For example, agents may explicitly coordinate their actions when it is desirable or necessary. Agents may share information with one another that affects what plans they pursue, what order they execute actions, and when they execute them. Another feature is that the agents are time-aware and are free to develop and follow different information gathering plans that depend on the amount of time they have to produce an answer. Eventually, some agents will use technologies like natural language form-filling programs and the INQUERY indexing scheme to make decisions about how to focus their search without constantly requiring detailed user feedback.

One goal is to provide seemingly replicated access to known information sources without excessive redundant retrieval of information by multiple agents. A more interesting goal is for the agents to redirect their processing resources and information gathering plans as information is retrieved during query processing. For example, a web page might be retrieved that contains links to new pages that have a high probability of containing useful information but that are not a part of the current retrieval plan. Partial search results may be presented to the user in order to get feedback to focus the remaining search. For example, if the user is looking for information on color printers, the system may discover that there are probably too many models and would then present the user with a list of features (such as price or print quality) that the user can interact with to narrow the search.

*This work was supported by DARPA contract N00014-92-J-1698, Office of Naval Research contract N00014-92-J-1450, and NSF contract CDA 8922572. The content of the information does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

[†]Currently at The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213–3890

Architecture

From an organizational point of view, the agents form a matrix organization consisting of both functional and query-answering units. A functional unit provides access to some particular type of information resource such as current and archived news feeds, or magazine databases. Each functional unit has a functional manager agent who is responsible for task assignments to agents within the unit. Functional units may comprise relatively undifferentiated agents that have similar knowledge and capabilities, or differentiated agents with fairly focussed skills. A query-answering unit comprises a query-manager agent and a set of agents drawn from the various functional units to respond to a particular user's query. The query manager is responsible for developing an initial high-level information gathering plan, recruiting agents from the necessary functional units (via the functional unit managers), and monitoring the plan's execution (including keeping the user up-to-date and able to steer the plan as it is carried out). The user interacts with a query manager via a web browser and HTML forms. Agents interact with one another using KQML (Knowledge Query & Manipulation Language, part of the ARPA Knowledge Sharing Effort) and a task structure specification language based on the TÆMS (Task Analysis, Environment Modeling, and Simulation) representation framework. TÆMS represents the structure of tasks (like information gathering), including the hard and soft relationships between tasks, for the purpose of coordinating and scheduling agent actions.

Each individual agent has an architecture consisting of several components: a task structure database that contains the agent's current view of its tasks and their relation to other agent's tasks; a coordination module that reacts to certain patterns in the task structure database and causes the creation of new local scheduling constraints, called commitments, which can be communicated to other agents; a planning module that can elaborate the task structure representation; a local scheduler that takes as input the current task structure, local commitments, and non-local commitments (from other agents) and produces several possible schedules (what actions to take, in what order, and at what point in time); and an execution monitor that oversees action execution to anticipate problems and slipped deadlines. This architecture is based on the GPGP (Generalized Partial Global Planning) architecture described elsewhere.

More information and a complete set of papers are available at our WWW site, `http://dis.cs.umass.edu/` under the subtopic "Cooperative Information Gathering".