Natural Language Interface for a Multi Agent System

Victoria Yoon  
*University of Maryland, Baltimore County*

Bonnie Rubenstein-Montano  
*Georgetown University*

Teresa Wilson  
*UMBC*

Stuart Lowry  
*Science Applications International Corporation*

Follow this and additional works at: [http://aisel.aisnet.org/amcis2004](http://aisel.aisnet.org/amcis2004)

Recommended Citation


This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
Natural Language Interface for a Multi Agent System

Victoria Yoon
Department of Information Systems, UMBC
yoon@umbc.edu

Bonnie Rubenstein Montano
The McDonough School of Business, Georgetown University
brm7@georgetown.edu

Teresa Wilson
Department of Information Systems, UMBC
twilson@umbc.edu

Stuart Lowry
Science Applications International Corporation, Annapolis, MD
stuart.lowry@saic.com

ABSTRACT

This paper presents the natural language interface for an agent-based system, called MACS (Multi-Agent Contracting System). The MACS system is designed to provide advice in the pre-award phase of a defense contract. The original version of MACS allowed the users to query using only a list of predetermined keywords using pull-down menus. The second version of MACS incorporated a natural language interface in order to increase its usability. This study further enhances the natural language interface of MACS with a more robust parser. The paper presents the overall architecture of MACS and the parser used to further enhance the natural language interface.

Keywords

multi agent system, user interface agent, natural language interface, defense contracting.

INTRODUCTION

Multi-Agent Systems (MAS) have successfully demonstrated their capabilities of tackling a variety of complex problems and in turn have become a viable alternative for solving many real problems (Jennings, 2001; Papazoglou, 2001; Wang, 2002). One factor affecting the performance of MAS is the user interface, which is responsible for interacting with users to receive instructions and to provide the results of its actions. However, most user interface agents in MAS have suffered from limited flexibility when interfacing with the users. The user agent often responds only to direct command manipulation, forcing users to memorize a set of commands and the proper sequence of issuing commands. A natural language interface eliminates the necessity to remember commands and provides much flexibility in terms of user interactions. Users can input a string of words or queries and the natural language processing agent can parse out possible commands from the natural statements.

MACS (Multi-Agent Contracting System) was successfully developed to provide advice during the pre-award phase of defense contracting (Liebowitz, Adya, Rubenstein Montano, Yoon, Buchwalter, Imhoss, Baek, and Suen, 2000). The first version of MACS allowed the user to query using only a list of predetermined keywords from pull-down menus. In order to increase the flexibility in the first version of MACS, the natural language interface was incorporated into the MACS architecture. The objective of this paper is to present the enhanced capability of a natural language interface for MACS. The next section describes the architecture of MACS. Then the development of the natural language (NL) interface is presented, and performance results of the NL interface are given. Lastly, conclusions and future research directions are given.

ARCHITECTURE OF MACS
The MACS architecture implements a typical three-tiered brokered architecture and contains nine agents — a User agent, a Facilitator agent, a Natural Language Process (NLP) agent, a Bayesian Learning (BL) agent, and five specialty agents. The User agent is the highest level, interfacing with users through natural language queries. The Facilitator agent is responsible for parsing communications between the User agent and all of the other agents in the MACS system and coordinates agent activities. The seven remaining agents interface with the Facilitator agent and are responsible for resolving user queries. The NLP agent translates English into Interagent Communication Language (ICL). The BL agent implements Bayesian learning for learning which specialty agent(s) can best respond to user questions over time. The BL agent is described in detail in Montano, et al. (Rubenstein Montano, Yoon, Lowry, and Wilson, 2004). The remaining five agents are the specialty agents which are encoded with domain knowledge about the five general areas of expertise required of contracting officers: Forms, Justification, Evaluation, Synopsis, and Type of Contract agent. The Forms agent identifies the forms needed to complete a procurement request package. The Justification agent indicates situations where a justification and approval is required to complete a procurement request. The Evaluation agent provides guidelines for evaluating proposals. The Synopsis agent identifies the type of synopsis for a given procurement request. Lastly, the Type of Contract agent identifies the type and nature of a contract based on conditions such as the source of contract, the nature of the work, etc.

Users interact with MACS by submitting an English phrase or sentence into a form displayed by the User agent. The User agent receives the sentence and passes it to the Facilitator agent with a request to translate the sentence into ICL to be understood by the MACS agents. The Facilitator agent forwards this request to the NLP agent, which then creates the corresponding ICL. The resulting ICL is returned to the Facilitator agent. The Facilitator agent then passes the ICL back to the Facilitator agent, which submits the action plan to the Facilitator agent for completion. In MACS, the action plan indicates which specialty agent(s) should be contacted to respond to an incoming query. The Facilitator agent completes the action plan by performing the necessary low-level communication between the specialty agents. These communications lead to solutions being returned from a specialty agent(s) to the Facilitator agent and then back to the User agent.

The architecture of MACS has been implemented with a publicly available software agent architecture called Open Agent Architecture (OAA), developed by the Artificial Intelligence Laboratory of the Stanford Research Institute (SRI). OAA was selected to implement the NL interface in MACS because it supplies the NLP agent to translate English sentences into the ICL.

**NATURAL LANGUAGE PROCESSING**

**Attain**

The NLP agent used in the previous phase is the Definite Clause Grammar-Natural Language (DCG-NL) in OAA, which is designed to parse incoming natural language queries. Through the use of a logic-based grammar called Definite Clause Grammar (Pereira and Warren, 1981), the DCG-NL translates incoming English queries into expressions using Interagent Communication Language (ICL). The Definite Clause Grammar (DCG) is a type of grammar formalism that is expressed as a series of facts and rules.

The NLP agent used in the current phase is ATTAIN. ATTAIN is a package of natural language OAA agents which provides parsing and translation of English sentences into ICL messages that other agents can use. The difference between ATTAIN and the DCG-NL is that ATTAIN employs a unification grammar whereas the DCG-NL utilizes a Definite Clause Grammar. Unification grammar means that grammatical categories incorporate features that can be assigned values; thus, when grammatical category expressions are matched in the course of parsing or semantic interpretation, the information contained in the features is combined, and if the feature values are incompatible the match fails. This parser technology provides a more expressive ICL representation (OAA, 2000).

In general terms, ATTAIN allows for both active and passive voice constructions, extensive use of modals (should, could, would), and longer verb predicates (longer lists of noun phrases and prepositional phrases after the verb). These seemingly simple features greatly enhanced the ability of the MACS system to handle the type of queries that are encountered in the contracting domain.
Information Retrieval

Once a user’s query is parsed, the next step is information retrieval (IR) that uses the parsed output to retrieve knowledge from the five specialty agents. IR deals with the representation, storage, organization of and access to information items (Baeza-Yates, 1999). The widely used IR techniques are Boolean (Sparck Jones, 1979), vector (Yu and Salton, 1976), and probabilistic models (Robertson and Sparck Jones, 1976; Sparck Jones, 1979). More sophisticated models include alternative set theory models, such as fuzzy set (Radecki, 1981), and extended Boolean (Salton, Fox and Wu, 1986), the algebraic models, such as generalized vector space (Yu and Salton, 1976), latent semantic indexing (Furnas, Deerwester, Dumais, Landauer, Harshman, Streeter, and Lochbaum, 1999), and neural network (Kwok, 1995), and alternative probabilistic models, such as the Bayesian inference (Callan, 1996), and belief networks (Ribiero_eto and Muntz, 1996). Each exhibits unique strengths and weaknesses; the selection of a suitable model is dependent on the domain data set, available technical resources, and usage constraints such as speed, storage, trade-off between recall and precision, and user requirements. MACS uses a combination of basic Boolean, pattern matching, and fuzzy matching algorithms of IR. This combination is suitable for our purposes, since the knowledge base, which is relatively small, is not indexed, and speed is not an issue at this juncture.

The information retrieval process for MACS is comprised of three steps. First, the ICL resulting from the NLP agent is stripped down as follows:

(a) All of the special characters are stripped out of the ICL (parentheses, commas, double & single quotations, brackets).

(b) All of the prolog style lists are expanded to a series of terms. (e.g. ['up the hill','Ed','Cameron'] gets turned into “up the hill Ed Cameron”).

The second step is to remove the stop words. The stop words are managed by the MACS System Administrators. A web form was developed to allow system administrators to add and remove words from the stop word list. The stop words are those deemed by the Administrators of the MACS system as being statistically insignificant. Therefore, whenever these words are present in the resulting ICL, they are removed from further consideration. Examples of stop words include articles, leading “Wh-” words, pronouns, and certain forms of verbs and modals. The third step performs the Boolean matching function to each rule in the specialty agent, and this IR process occurs in the specialty agent itself. This process occurs here because it is considered to be the “value added” by the specialty agent. Each agent is capable of applying unique criteria to match the word list to its individual rules. At this time the distinction between the specialty agents is the rule-based knowledge and not their individual functionalities.

The matching algorithm consists of a Boolean matching percentage and a whole word match requirement. Instead of an implicit Boolean “AND”, a percentage scale was used. 1% was equivalent to the implicit “OR” and 100% was equivalent to the implicit “AND”. Although performance using the implicit “AND” was good, it was too inflexible because the IR was performed only exact matches. But, an implicit “OR” was too unconstrained, resulting in too many irrelevant rules to be fired. This was the motivation behind the percentage scale.

The whole word-matching requirement assisted in narrowing the result set to the proper rule. As each word in the word list was matched in the clause portion of a particular rule, it was flagged as matched. Each term in the word list was treated with the same weight. At the end of the matching processes, the number of hits was divided by the number of terms present in the word list and multiplied by 100. If this value is greater than or equal to the pre-set Boolean percentage, then the rule will be considered an answer to the question.

Demonstration

Screen shots below illustrate NL processing in MACS. The user is presented with a web page that will manage a “session”, as shown in Figure 1. The user controls a session in order to drill through the knowledge base to discover an answer to their question.
A user submits a query, “What justification type do I need if I am working with a sole source contract?” The NL agent successfully parses and processes the query. Figure 2 shows the result of the user query.
RESULTS

Domain experts provided 26 natural language queries to test the performance of NLP and IR. Originally, 36 queries were provided by the experts, but ten did not correlate to rules in the MACS rulebase. All 26 test queries (100%) were successfully parsed. Moreover, fewer queries needed to be modified than in earlier versions of the MACS system because ATTAIN is more robust than the DCG-NL.

The performance of NLP was tested on the basis of the percentages of parsed words presented in the condition part of a rule. It was determined to use 50% as the threshold, whereby half of the parsed words needed to be present in the rule base to fire a rule. Ground truth rules, which are those that the domain expert had determined to be the “correct” rules, were identified and used to evaluate performance. Table 4 presents the summary of NLP/IR performance results.

<table>
<thead>
<tr>
<th>Types of Result</th>
<th>Hit Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries returning a rule</td>
<td>26 out of 26, 100%</td>
</tr>
<tr>
<td>Queries returning only the ground truth rules</td>
<td>8 out of 26, 31%</td>
</tr>
<tr>
<td>Queries returning more rules in addition to ground truth</td>
<td>18 out of 26, 69%</td>
</tr>
</tbody>
</table>

Table 4: Summary of Results
Conclusions and Future Directions

This study has applied a more robust natural language parser to a multi-agent system, MACS, which was constructed to assist users in defense contracting. The Natural Language Processing agent successfully parses the natural language queries and selects the specialty agent(s) to answer the user’s query by firing the rules in its knowledge base.

This research is an important step toward providing a natural language interface for MAS and contributes to the existing MAS literature by incorporating NLP. The purpose of investigating the NL capability was to provide the user with more flexibility in query formulation. The results of this study indicate that this was sufficiently accomplished. The future directions of this research are to further enhance the capability of the natural language interface by automatically updating vocabulary/synonyms in the MACS system. Further, in order to improve the practical usability of MACS, a group of potential system users will be identified to use the system and provide feedback about problems and necessary enhancement.

REFERENCES
