Verification of Multiple Agent Knowledge-based Systems

Daniel E. O'Leary
University of Southern California
3660 Trousdale Parkway
Los Angeles, CA 90089-1421

oleary@rcf.usc.edu

Abstract

The purpose of this paper is to extend verification tests to systems with multiple autonomous agent knowledge bases. Using a classic approach to verification, this paper focuses on tests concerned with consistency, completeness and correctness. In particular, this paper focuses on those unique issues that are generated as we go from single agent systems to multiple agent systems.

This paper is concerned with inter agent verification, since previous results can be used for intra agent verification. For example, consider one agent with the rule "if A then B" and a rule in another agent "if A then C". In such a setting, the agents would be constantly at odds. Alternatively we might find the following rules in one agent ("if A then B" and "if C then A"), while another agent that interacts with that agent might have the rule ("if B then C"). With those two interacting rule bases a dialogue starting with "A" could cycle indefinitely.

One potential approach to multiple agent systems is to compare the knowledge base of each subset of agents to determine the existence verification issues. Where the number of agents is small this approach is feasible. However, for even medium size systems this approach explodes computationally. This paper finds that many of the multiple agent verification tests can be conducted on a meta rule set generated from all the rules contained in each of the agents' knowledge bases thus minimizing computational effort. In addition, this paper finds that the property of agent "isolation" is an important verification criteria in multiple agent systems.

1. Introduction

Inter agent verification generates different problems than intra agent verification. Consider two agents with knowledge bases that include the following rules: Agent 1(if A then B) and Agent 2 (if A then C). In any dialogue between those two agents if "A" occurs the two will not come to closure. As another example suppose that agent 1 has the knowledge base (if A then B and if C then A) and agent 2 has the knowledge base (if B then C). Interaction between these two agents can cycle indefinitely. These unique issues are not addressed in traditional verification. As a result, the purpose of this paper is to elicit some of the issues in verification analysis for multiple autonomous agent systems.

This paper identifies these verification issues unique to inter agent verification. In so doing it identifies those situations where an aggregated set of rules across all agents can be used for determining issues of inter agent verification. The aggregated set of rules provides a computationally efficient way to facilitate verification of autonomous agent rule sets. In addition, the paper identifies some other unique issues such as "isolation" of agents.

This paper

The remainder of this paper proceeds as follows. Section 2 provides a brief discussion of the scope of the paper and previous research. In addition, section 2 briefly discusses three different design approaches for multiple agent systems, including "top-down" and "bottom-up." Sections 3, 4, and 5, each analyze the different aspects of verification, correctness, consistency, and completeness. Section 6 provides a brief summary of the paper.

2. Background, Scope and Previous Research

Knowledge Representation

I assume that each agent's knowledge base is represented using rules. This is not a critical assumption, however, it does facilitate explanation and presentation. Other forms of knowledge representation could be used, such as objects (e.g., Kandelin and

O'Leary, 1995) or multiple hybrid forms of knowledge representation could be used in different agents.

Verification

"Verification" has been defined as "building the system right" (e.g., O'Keefe and O'Leary, 1993). This view of view of verification is one where the concern is with implementation of the "technology" of knowledge-based systems (e.g., rules, weights on rules, etc.) in a correct manner. This view on the scope of verification tests is consistent with Nguyen et al.'s (1987) research on this topic.

Operationally, verification was defined by Adrion et al. (1982, p. 159) as "the demonstration of the consistency, completeness and correctness of the software." As a result, my analysis of verification tests in multiple agent systems will concern itself with each of those elements.

Multiple Agent Systems

I use multiple agent systems to refer to the existence of multiple independent agent rule bases. Agents are assumed to interact with each other sharing knowledge and information. Agent rule bases are likely to represent different actors and their capabilities. The agents may have the exact same rule bases or their rule bases may be completely independent of each other. Because agent rule bases interact we need to consider verification issues that result from those interactions.

Previous Research on Verification and Validation of Multiple Agent Systems

There has been only limited investigation of verification and validation of multiple agent knowledge-based systems. O'Leary (1993a) investigated procedures for determining the existence of differences in expert judgment. O'Leary (1993b) presented results on determining the existence of conflicts between probability estimates of multiple experts in an influence diagram. Kandelin and O'Leary (1995) discussed verification of object-oriented systems, that are often used in modeling multiple agent systems. Brown et al. (1995) investigated the problem of validating heterogeneous and competing knowledge bases using a blackbox approach. Throughout that literature there has been little concern given to verification tests of interacting knowledge bases, the focus of this paper.

Intra-Agent vs. Inter-Agent and Multiple Agent Systems

The focus in this paper is on some of the unique issues associated with "inter-agent," i.e., "between agent" verification problems. In particular, the concern is with unique verification problems deriving from the very nature of having n independent knowledge-based agents in the same system. The paper does not investigate "intra-agent" problems. In particular, it is assumed throughout that none of the anomalous verification situations (consistency, completeness or correctness) occurs in the individual agent knowledge bases. Intra-agent verification issues can be addressed using procedures such as those of Nguyen et al. (1987) and others.

Design Approaches for Multiple Agent Systems

Verification is inevitably dependent on the explicit or implicit design of the multiple agents. If each agent is generated completely independently of each other with no concern for relating ontologies then there is little hope for meaningful verification. However, under other circumstances discussed in this paper verification can provide important insights.

In particular, design of a multiple agent system typically would employ one of three basic approaches.

Case 1 "Bottom-up" -- Agents are designed independently of each other. However, I assume that if that is the case there is a common ontology across each of the agents so that each of the rule sets of different agents A_i, could be aggregated to form a meta rule set of all the rules, A'. Although the resulting rule set would have some redundancies, removal of those redundancies could be used to generate the reduced rule set A. This approach is assumed implemented with each agent being a source of rule development.

Case 2 "Top-Down" -- Agents are designed, so that each agent i's rule set A_i, is drawn from some meta rule set A, based on the same ontology, so that each A_i is a subset of A. Thus, the set is developed and from A, rules for each agent are chosen. In the case where the agents are homogeneous, each set A_i equals A. Ideally, this approach would be implemented by having each agent with the same rule drawing a referenced rule from a single source, rather than having rules independently, physically generated for each of the different agent knowledge bases.

Case 3 "Autonomous Chaos" -- Agents are designed independently using different ontology's. In this situation, it would be virtually impossible to determine similarity of rules because of semantic and syntactic differences in development environments.

The research presented here focuses on the first two cases, either where there is an "a priori" meta set of rules or "a postiori" meta set of rules, based on a single ontology that could be constructed from the different rule sets.

3. Correctness

Correctness between multiple agents is concerned with three primary issues: conflict, circularity and subsumption conditions. Each of the three correctness results that are generated, can be accommodated through analysis of the meta set of rules, A.

Conflict

A (single rule) conflict error occurs if two (or more) agents have different conclusions for the same "if". For example, if one agent has rule "If A then B" and another agent has the rule "If A then C" then there can be a conflict between the two agents. There are a number of disadvantages of agent conflicts. First, conflicts can make different agents produce different answers to the same queries given the same information. Second, conflicts can make it "difficult" for agents to negotiate, since no matter what is negotiated there is a difference of agent knowledge (opinion). Such a conflict is probably not as important if there is probability or certainty factor information carried along with each rule, however, it may still cause difficulties.

Conflicts can be identified by comparing the rules of different agents or by finding conflicts in the set of rules A. In particular, assuming that every rule in A is the rule set of some agent then we have the following result.

Result 1: There single rule inter-agent conflict if and only if there is conflict in the meta set of rules, A, using either a top-down or bottom-up design.

Circularity

Circularity can occur between a pair (or more) of agents in the following types of situations. If one agent has the rules "If A then B" and "If C then A" and another agent has the rule "If B then C" then dialogues between those agents can result in circularity, starting from agent one with A, to agent two with B, to agent one with C and then back to B, etc. This can result in "negotiations" that "don't go anywhere" or agent behavior can be described as being in a "rut."

Result 2: There can be circularity between agents if and only if there is circularity in the meta set of rules, A, using either a top-down or bottom up design.

Subsumption

Subsumption occurs in the following types of situations. If one agent has a rule "If A then C" and the other agent has the rule "If A and B then C" then there is subsumption. In multiple agent settings, subsumption indicates that the agents require differential amounts of information to draw conclusions. It may be that such subsumption is by design, e.g., one agent is designed to come to conclusions more rapidly than others. However, it can indicate a potential problem with negotiations between agents.

Result 3: There can be subsumption in the rules between agents if and only if there is subsumption in the meta set of rules, A, using either a top-down or bottom up design.

4. Consistency

In general, consistency between multiple agents relates primarily to ensuring that redundant rules, used by multiple agents, stay the same, in spite of activities like maintenance and development. It further requires that ontology and naming conventions between agents are the same, because otherwise it would be difficult or impossible for the agents to effectively communicate without consistency conventions.

Redundancy

Assume two (or more) agents (j and k) have identical rules, say rule r from A, in their knowledge bases, denoted, a_{jr} and a_{kr}, so that $a_{jr} = a_{kr}$. The existence of redundant rules in different agents is not an error, per se, (in fact in homogeneous agents there will be entire rule sets that are the same, and thus, redundant) but if there is maintenance to the rules then depending on the maintenance errors can be introduced. Redundancy is something that is to be preserved, and not eliminated in multiple agent systems. The question is "how to best preserve that redundancy between agent knowledge bases."

In the case of bottom-up system design, if there is maintenance to individual rule sets then that could result in a change of either a_{jr} and a_{kr} but not the other (and other possibilities). As a result, a change to individual rule sets A_j and A_k from a bottom up approach could result in a new A (assuming it is formed), which would include the original rule and the rule that was supposed to be changed. The resulting system would be contrary to original intentions and thus in error.

Alternatively, using the top-down approach, if there is required maintenance to an individual rule then it should be done on a rule in the meta set A. This would eliminate problems of changing the rule in one rule set, but not another, or other similar problems. The resulting A would be different than the A derived from a bottom-up approach.

As a result, there is an implication for the design of multiple agent rule-based systems in order to preserve multiple agent redundancies: If there is a rule that is used by more than a single agent then keep it in a repository and allow agents to reference the rule as part of their rule sets, whether the approach is top-down or bottom-up. This approach will limit errors due to changes in one agent but not in another for the same rule. Further, other additional redundancies are more likely to be preserved if virtually all the rules are maintained in a central repository.

Ontologies and Related Issues

In order to ensure consistency between agents there is also a need for the agents to function under the same ontologies and to represent model variables using the same, e.g., naming conventions. The lack of a consistent ontology (or even the same "naming conventions) can limit the effectiveness of agent communications, since agents would not be talking using the same "language." As a result, system design and testing should be implemented so that there is a consistency in the ontology. However, if there is a situation where there is no such similarity, then in order for effective multiple agent communication to take place there must be individual agent capabilities with multiple ontologies or some super agent who has the ability to provide multiple ontology communication between the agents.

There are a number of impediments to implementing ontologies in multiple agent settings. O'Leary (1997) noted that those impediments included ontology stationarity, scaling up, ontology interfaces and other issues.

Naming Conventions

Perhaps the clearest impact of a lack of naming conventions in knowledge-based systems is presented in Landauer (1990). He demonstrates the need for and importance of establishing consistent naming conventions in his analysis of the manned maneuvering unit. For example, Landauer found forty occurrences of "thrusters" and one occurrence of "thruster," suggesting that the use of "thruster" is incorrect.

Consistent naming conventions are also critical in multiple agent systems. One design approach to ensure consistency of naming conventions is to establish a list of variable names that is used by all of the agents, facilitating communication between the agents. Otherwise statistical tests of the occurrence of variable names, like those employed by Landauer (1990) can be used to generate anomalous variable occurrences.

5. Completeness

Completeness has been characterized in single knowledge base systems (e.g., Nguyen et al., 1987) as an issue of having unreferenced attributes (attributes that were declared as legal variables, but not used), illegal attributes (attributes that were not declared as legal variables but were used), unreachable conclusions (the conclusion should either match a goal or an if condition of another rule) or deadend conditions (attributes must be askable or matched by a conclusion in another rule).

Closely related questions need to be asked for multiple agent systems. For attributes, particularly in a top-down development environment, repositories of attributes can be generated and used in development to ensure that all attributes used are legal (on the list) and that all attributes on the list are used. Further, the existence of unreferenced or illegal rules (e.g., rules that are in A but are not used or rules that are used but are not in A, given a top-down approach) need to be determined. Otherwise, if no central repository of attributes is developed then there are no expectations, and it is difficult to test completeness by determining if there are any missing variables.

Finally, although it may be desirable to have agents that are virtually independent with no overlap, that situation should be criteria that is stated up front and otherwise tested. As a result, I define the "agency isolation" that tests the completeness of agent development through determination of extent of isolation.

Unreachable Conclusions

Next consider the situation of determining the existence of unreachable conclusions in different agents. In a manner similar to the correctness tests, the A matrix of all rules can be used to determine the existence of rules with unreachable conclusions.

Result 4: There are unreachable conclusions between agents only if there are unreachable conclusions in the meta set of rules, A, using either a top-down or bottom up design.

Agent Isolation

In some multiple agent systems, agents are treated as performing independent activities, with no common variables. However, in most multiple agent systems there is expected to be communication between agents about certain issues, e.g., with each agent bringing observation of different data to the discussions or negotiations. Accordingly, an important test of interagent consistency is the determination of whether or not there are any agents that are "isolated" from the other agents.

There are at least two "levels" of isolation, representing different extremes. I define "level 1 isolation" as occurring if there is an agent with a knowledge base that has no rules that are the same as any other agent. I define "level 2 isolation" as occurring if there is an agent with a knowledge base that has no variables that are the same as any other agents. Given a top-down approach, with level 1 isolation, agents can at least talk about or infer about the same variables. In level 2 isolation there is no basis of similarity between the agents. The occurrence of either form of isolation is likely to be anomalous in negotiation-based systems. Each form of isolation can be tested using a comparison of rules and variables of different agents.

6. Summary

This paper extended verification tests from single agent to multiple agent systems. Four results were presented that are designed to facilitate determination of conflict, circularity, subsumption and unreachable conclusions. These results are based on assuming that we can aggregate agent rules and investigate the resulting set of meta rules.

Acknowledgments. Some of the results summarized here were part of a presentation at the International Joint Conference on Multiple Agent Systems, San Francisco, June 11, 1995. The author wishes to thank Les Gasser for extensive comments on earlier versions of this paper.

References

Adrion, W., Branstad, M., and Cherniavsky, J. "Validation, Verification and Testing of Computer Software," ACM Computing Surveys, Volume 14, Number 2, pp. 159-192

Brown, C., N. Nielson, N., O'Leary, D. and Phillips, M., "Validating Heterogeneous and Competing Knowledge Bases Using a Black Box Approach," Expert Systems with Applications, 1995, pp. 591-598.

Kandelin, N. and O'Leary, D., "Verification of Object-Oriented Systems: Domain Dependent and Domain Independent Approaches," Journal of Systems and Software, Volume 29, 1995, pp. 261-269.

Landauer, C., "Correctness Principles for Rule-based Expert Systems," Expert Systems with Applications, Volume 1, Number 3, 1990, pp. 291-316.

Nguyen, T., Perkins, W., Laffey, T. and Pecora, D., "Knowledge Base Verification," AI Magazine, Volume 8, Number 2, Summer 1987, pp. 69-75.

O'Keefe, R. and O'Leary, D., "Expert System Verification and Validation," Artificial Intelligence Review, Volume 7, pp. 3-42, 1993.

O'Leary, D., "Determining Differences in Expert Judgment: Implications for Knowledge Acquisition and Validation," Decision Sciences, Volume 24, Number 2, March/April 1993a, pp. 395-407.

O'Leary, D., "Verification and Validation of Multiple Agent Systems: Combining Agent Probabilistic Forecasts," in Working Notes of the Workshop on Validation and Verification of Knowledge-based Systems, Eleventh National Conference on Artificial Intelligence, 1993b.

O'Leary, D. "Impediments in the Use of Explicit Ontologies for KBS Development," International Journal of Human Computer Studies, 1997, volume 46, pp. 327-337.