Conflict Resolution Among Heterogeneous Reusable Agents in the PARI DSS

Murray J. Bentham

Supervisors: Dr. Jim Greer and Dr. John Cooke

Abstract

Large complex problem-solving systems often require the expertise of multiple specialized agents working together to produce comprehensive solutions. Where diverse expertise must be integrated, conclusions or recommendations can be inconsistent from one expert to another. In addition, different experts generally have different goals and different methods for achieving these goals. Consequently, conflicts in the knowledge and goals being represented are likely to occur. Although integrated artificially intelligent systems of heterogeneous reusable agents are potentially adaptable, maintainable and affordable, conflict resolution plays a major role in the integration and forms the basis for communication and coordination protocols. Research has resulted in the development of several approaches to conflict management within the paradigm of cooperating expert systems. In this paper, we discuss distributed artificial intelligence focusing on multi- agent systems and examine existing approaches and techniques for conflict resolution among cooperating expert agents. We describe the application of conflict management techniques in our attempt to utilize "off-the-shelf" reusable agents, within the application context of an agricultural decision support system called the PARI DSS. Integrating heterogeneous reusable agents in the collaboratively developed PARI DSS provides a natural, flexible development methodology, providing the capability to add agents as they become available and to delete or substitute agents in response to changes in specifications and technology.

1 Introduction

Present day problem-solving systems are large and complex reflecting rapid and constant changes to objectives in application domains. Often these systems require the expertise of multiple expert agents cooperating together to produce comprehensive solutions to complicated problems. These agents might be developed from scratch for each application system, a costly and timely process, or more recently, application systems are being built with heterogeneous and reusable agents. The context of our application, explained in Section 1.1, necessitates a further challenge to utilize existing "off-the-shelf" heterogeneous reusable agents, not only to minimize costs and development time, but to create an open system. *Open systems* receive new knowledge and information from outside themselves at any time, and hence, their operation is always subject to unanticipated outcomes [11].

An *agent* is defined as a logically independent computational process representing a specific area of expertise, such as an expert system, developed to deliver a specific knowledge set [2]. Information represented in a modular and logically independent method is potentially *reusable* [23]. A reusable agent system is computationally analogous to a pool of human specialists. Hence, a *multi-agent* system is composed of agents dynamically selected from an existing library and integrated with no, or minimal, customized implementation [15].

Heterogeneity among agents can be classified as logical and implementational. Logically, or knowledge (as they are sometimes called), heterogeneous agents are characterized by differences in declarative knowledge, solution evaluation criteria, goals, capabilities and priorities [15]. Implementationally heterogeneous agents can be characterized by differences in architectures, algorithms, languages, inference engines or hardware requirements. These differences may also be classified as representation heterogeneity [15].

Our goal of an open system not only enables existing agents to be utilized "off-the-shelf", but provides collaborators with the freedom to develop their agents for their own specific use. Furthermore, integrating heterogeneous reusable agents provides a natural and flexible development methodology whereby new agents can be added and existing agents deleted in response to system requirements. Other advantages to using "off-the-shelf" heterogeneous reusable agents which fulfill our goals include [15]:

- the uniformity of expertise within each agent make them easier to design, develop and maintain than single large systems;
- consistency of knowledge can be maintained within local boundaries without necessitating agreement across boundaries;
- software modifications can be focused at the problem or enhancement within the agent, without requiring changes to be disseminated throughout the entire system.

In the paradigm of cooperating expert systems, heterogeneous reusable agents must be integrated to provide effective technical interaction. Within these systems, the state of problem-solving must be communicated and agents' actions must be coordinated to arrive at mutually acceptable solutions. However, in these cooperative environments, conflicts must be resolved as a result of incomplete or inconsistent knowledge and/or incorrect assumptions, different problem-solving techniques, and different solution evaluation criteria [20]. Anticipating and removing potential conflicts through software engineering at agent-development time is not possible since it is not known what knowledge will be contained in an open system. Software reuse in any form is hindered by the absence of technical tools and techniques to support information sharing [23]. However, collectively joining diverse knowledge from the viewpoints of cooperating experts provides an extremely important source of balance and robustness in many real-world situations. This is characterized by carefully selected human project teams and work groups. Teams and groups can solve problems which are normally beyond the comprehensiveness of individual experts, and in doing so, provide potentially creative and innovative solutions resulting from a rich and varied body of knowledge [16].

In this paper, we discuss distributed artificial intelligence, focusing on multi-agent systems in terms of the formulation, description, and allocation of problems, and the interaction, communication, coordination, and general conflict resolution among agents. Traditionally, multi-agent systems do not acknowledge conflict resolution as playing a major role in agent-integration, nor forming the basis for communication and coordination protocols among agents to control problem-solving activity [19]. Next, we examine, in more detail, existing approaches and techniques for conflict resolution among cooperating expert agents. We then describe our application of conflict management techniques which are used for controlling and directing problem-solving activity of our integrated "off-the-shelf" reusable agents, within the application context of an agricultural decision support system called the Parkland Agriculture Research Initiative (PARI) Decision Support System (DSS). Conflict is managed through the exchange of information among the agents. What to exchange, when to exchange, how to exchange and who to exchange it with are questions that are addressed in this paper.

1.1 Application Context

Farmers and ranchers in the Prairie Region of Western Canada are currently facing many challenges of which soil degradation is one of the key issues. Conservation production systems must be implemented to ensure the long-term sustainability of the soil resource and in turn the sustainability of agriculture. Farm managers need sound advice and accurate information on all aspects of their farm operations in order to make appropriate decisions that will conserve the soil, land and water resources while also maintaining the economic viability of their operations. In order to deal with the many complex issues and numerous alternatives available, they need access to various knowledge sources to support their production decisions.

A vast amount of technical information, both published and unpublished, resides in various locations which is not readily available nor utilized at present. In addition, many "experts" in various disciplines, at research and educational institutions and at extension agencies, have considerable knowledge and experience that is not being fully utilized. Innovative multiple objective support systems are required to make information and knowledge available to farm managers to support their production decisions.

Agriculture and Agri-Food Canada, Research Branch, under the Parkland Agriculture Research Initiative (PARI), is developing a conservation production decision support system, comprising multiple agents to facilitate multi-objective decision making focusing on encouraging adoption of sound economically viable soil conservation practices. The primary client of the PARI Decision Support System (DSS) will be farm managers interested in adopting conservation farming practices, although other clients may include extension specialists, research scientists, and private industry.

The PARI DSS is being collaboratively developed with producers, governments, universities and private industry. A 4 level framework, 0 to 3, enables flexible natural growth while providing a "single window" to decision support. Level-0 acts as a clearing house, providing services and administrative support to PARI DSS clients. Level-1 consists of basic decision support based on databases and management information systems. Level-2 provides advanced decision support consisting of standalone analyzers. Level-3, which provides the most sophisticated support through the integration of multiple heterogeneous reusable expert systems utilizing a blackboard architecture, is the focus of this paper.

2 Distributed Artificial Intelligence

Historically, researchers divided Distributed Artificial Intelligence (DAI) into two main areas [2]. *Distributed Problem Solving* (DPS) undertakes the work of problem-solving by dividing the tasks among a number of cooperating modules that share knowledge about the problem and the developing solution [29]. Alternatively, research in *Multi-agent Systems* concentrates on coordinating the knowledge, goals, skills and plans of autonomous intelligent agents to facilitate problem-solving through their joint intelligent behavior. Multi-agent systems, like the PARI DSS, will be the focus of attention in Section 2.1.

2.1 Multi-Agent Systems

The agents in a multi-agent system might work together toward a single global goal or toward separate individual goals that cooperatively interact to meet a common goal. In the PARI DSS, the agents are heterogeneous, and in most cases "off-the-shelf" (i.e. reusable) with separate individual goals, which are integrated through a cooperative environment to provide comprehensive decision support. As in a DPS system, agents in a multi-agent system need to share knowledge about problems and solutions but must also reason about the coordinating processes used among the agents in the cooperative environment. Coordination in a multi-agent system, particularly an open system [11], can be difficult, as situations can arise where there is no possibility for global control, globally consistent knowledge, globally shared goals, or global success criteria, or sometimes, not even a global representation of a system [9, 10]. Many of the coordination difficulties in multi-agent systems can be attributed to several main problems which are discussed in the following sub-sections.

2.1.1 Formulation, Description and Allocation of Problems

In the same way that work is divided among human experts, there must be a division of labour and organization in order to distribute tasks among agents [2]. In order to accomplish this distribution, tasks must be formulated and described in such a way as to facilitate their distribution [2]. Complicated tasks requiring more resources or knowledge than is held by a single agent must be decomposed in order that they can be accomplished. Tasks must be allocated or assigned to particular agents that are capable of performing them. All these requirements are very interdependent.

The concepts and methods used for task description and formulation will influence how tasks can be decomposed, and what dependencies explicitly exist among tasks. For instance, the same task described from different perspectives may necessitate different decomposition and different resources. Decomposition affects how tasks can be allocated, since the resources and knowledge of agents performing the tasks must match the task requirements [22]. Alternatively, the dependence and communication among tasks can affect the way a task is decomposed. As the context of the overall system changes, decomposition decisions may need to be revised. In addition, the availability of resources may affect the choices of distribution and aggregation.

In multi-agent systems, in order to establish and maintain problem- solving alliances, agents may need to negotiate the appropriate formulation of problems and appropriate responsibility for the description, decomposition and allocation decisions. Historically, research has focused on creating flexible task allocation methodology, to be used after a problem has been described and decomposed, rather than automated dynamic problem formulation and description [2].

2.1.2 Interaction, Communication and Coordination Among Agents

In order for several intelligent agents to combine their expertise and efforts, interaction and communication are necessary underlying concepts of multi-agent systems. Interaction in a multi-agent system, means a type of collective action in which agents take action or make decisions based on influence from the presence or knowledge of other agents [2]. Unlike agents' perceptions, beliefs, and goals, which may or may not be distributed, interaction is inherently distributed and is necessary for the successful coordination of agents' actions.

A *rational action* is taken by an agent in response to goals it has, believing that the goal will be satisfied by the action [2]. Communication normally takes place whenever there is the intention to communicate, as agents know that this intention to communicate will provide additional information. It is important to note, particularly in the case of the PARI DSS, that although agents may communicate coherently in order to structure coordination, communication between each other is not necessarily required in order to effectively coordinate interaction among agents. Agents may coordinate their actions utilizing models or metalevel information they have of each other, without interaction [28, 33].

Durfee [4] indicates that metalevel information should enable an agent to reason about the past, current and future actions of another agent. It is thought that agents that utilize redundant data are more efficient when they exchange metalevel information about the goals and plans of other agents, because it enables them to avoid redundant work. Metalevel information permits agents to more accurately assess the effects of communications on other agents' activities [5]. Metalevel communication has two advantages, fewer messages need to be exchanged reducing communication, and a decrease in the computation required to interpret messages and predict future actions in other agents. A disadvantage is that unless obsolete metalevel information is managed, it can be counterproductive.

Although integrated artificially intelligent systems of heterogeneous reusable agents are potentially adaptable, maintainable and affordable [20], conflict resolution plays a major role in the integration and forms the basis for communication and coordination protocols. General issues surrounding conflict resolution are introduced and followed with more in-depth discussion and analysis.

2.1.3 Conflict Resolution Among Agents

Intelligent problem-solving agents must externalize parts of their world in order to reason about them [2]. This externalization is subject to problems of abstraction and incompleteness due to the problem of attempting to fully represent an object or process. Therefore, intelligent agents need to contend with differences and uncertainty between their externalized representations and the affairs actually represented. When multiple agents are integrated they must externalize one another into representations which must then be aligned to coordinate with each other. Alignment does not mean that agents share representations but rather that the representations must allow them to act so as to fulfill their individual goals. Developing common representations, such as standards for communication protocols or information exchange [4, 5, 6] appear to be workable, but it depends very much on the agents' compatibility to interpret these protocols [35]. Furthermore, agents may have knowledge bases which contain differing beliefs, or they might have the same beliefs but with different confidence levels. Differences like these lead to conflicts among the agents.

Not all differences should necessarily cause conflict [2]. Agents need to have different knowledge and structure to avoid duplication of work. If agents have redundant or overlapping data, they must be able to determine which data are relevant to them and which to another agent, to again avoid duplication. As for reasoning about differences in belief among agents, it is necessary to be able to represent other agent's beliefs [27]. In essence, it is important to understand each agent's identity and manage their differences appropriately.

Not all conflicts can be resolved, particularly in open systems [11]. To resolve conflicts, agents must have some basis on which to agree. This basis is often established using a central or global controller with decision-making authority. Agents can also resolve conflicts by relaxing constraints or by redefining a problem to eliminate them [8]. If it is thought that conflict arose from inconsistent assumptions, backtracking is sometimes used to discover if the roots of the disparity lie in the assumptions [30]. Differences can then be resolved at any point in the backtracking procedure. Naturally, this approach necessitates that an agent has knowledge of the supporting assumptions and their relationships. This might possibly be accomplished by incorporating an ATMS (Assumption-Based Truth Maintenance System) which can represent all the relationships that have ever been considered at the same time and thereby check which assumptions might cause the discrepancy.

3 Conflict Resolution in the Paradigm of Cooperating Expert Systems

There are several existing approaches to conflict management within the paradigm of cooperating expert systems. Klein and Lu [13] manage conflict by establishing an abstraction hierarchy of conflict classes each with an associated applicable conflict resolution strategy. When conflict is detected, a conflict resolution expert uses the global conflict hierarchy to resolve it. Similarly, Werkman et al. [34] within their Design Fabricator Interpreter system for distributed cooperative problem solving among construction agents, utilize a third- party arbitrator agent to resolve conflicting recommendations based on global conflict resolution knowledge. Sycara [31] addresses conflict resolution by incorporating a negotiation model, which integrates case- based reasoning (CBR) and multi-attribute utilities. CBR retains the process and the results of its decision, and thereby can shorten similar reasoning chains by referring to the stored process and results as well as recognizing and avoiding similar failures in advance. These approaches will be elaborated upon in Section 3.1.

Lander and Lesser have developed several methods for managing conflict which will be introduced and discussed in more detail in subsequent sections. One of their methods is based on a generic cooperating experts framework to support cooperative problem-solving among sets of knowledge-based systems [16, 17]. Within this framework, conflicts are resolved by employing compromise and integrative negotiation. Compromise negotiation involves a concession of requirements on the part of each agent in order to satisfy other agents' requirements. Integrative negotiation identifies the most important goals of each party to use in formulating a basis for a new solution. In an extended variation of the same cooperating experts framework, Lander et al. utilize conflict resolution strategies that are appropriate to the problem-solving context [21]. Lander and Lesser have also developed a negotiated search algorithm which can be used in different variations of conflict management [18]. Finally, Lander and Lesser in a different approach, use meta-information to resolve conflict and improve joint results [20]. Each reusable agent shares meta-information with other agents, assimilates shared information from other agents, and then uses this information to improve their local understanding of the global solution space to achieve global solutions.

3.1 Conflict Classes and Third-Party Conflict Resolution Agents

Klein and Lu's [13] run-time model is oriented towards cooperative conflict resolution based on the notions that conflict resolution expertise can be captured explicitly and organized usefully into a taxonomy of conflict classes each with an associated applicable conflict resolution strategy. They support and instantiate this model in a study of conflict resolution among human experts in the domain of architectural design. In contrast to competitive conflict situations, where each agent has solely their own interest and benefits in mind, cooperative conflict resolution strategy strives to achieve a globally optimal solution, sacrificing personal goals in the interest of increased global benefit. Hiding information from the adversary and making threats are common strategies in competitive conflict situations. Whereas strategies for cooperative conflict situations usually include compromise or surrendering less important goals in order to reach as mutually beneficial solution as possible.

In Klein and Lu's [13] abstraction hierarchy of conflict classes, the more general conflict classes are arranged near the top of the hierarchy and represent domain independent classes, whereas the more specific classes are near the bottom and relate to domain dependent classes. Conflict resolution strategies within the domain dependent classes will have a narrower range of applicability, but usually greater efficiency, than the general strategies contained in the domain independent classes. When a conflict occurs, this hierarchical structure facilitates the search for the most specific conflict class matching the conflict, and tries the conflict resolution strategies associated with that class. If these strategies are unsuccessful, progression can proceed up the hierarchy to the next conflict class where the more general and less efficient strategies associated with that class are tried. Hence, if no specific conflict class for a particular conflict was encoded, the hierarchy can be searched to locate the closest conflict class associated with that kind of conflict. There is a trade off between the main two segments of the hierarchy with the domain independent segment being more applicable but less efficient and the domain dependent segment being more efficient and less generally applicable. General conflict classes would be difficult and time consuming to apply, and hence, Klein and Lu [13] recommend developing enough specific conflict classes and strategies to avoid the use of these expensive general strategies.

Although the domain independent levels of the conflict hierarchy are generated once during the development of the design system, based on acquired knowledge from domain experts, the domain dependent levels need to be recreated for each substantially new domain. Within Klein and Lu's [13] cooperative architectural design system, the conflict hierarchy works as follows. All design agents cooperate with each other by contributing to a shared representation of the design and criticizing other design agents contributions. A conflict results when two incompatible design contributions exist or when an agent's commitment is criticized. A description of the conflict is then communicated to the conflict resolution expert which utilizes the conflict hierarchy to determine a solution for the conflict. Due to the complexity of the PARI DSS, all the different kinds of conflicts cannot be pre- established, and hence, could not be arranged into such an abstraction hierarchy of conflict classes. However, this approach could be useful in conjunction with case-based reasoning discussed later in this section.

In a similar approach, Werkman et al. [34] within their Design Fabricator Interpreter (DFI) system, also utilize a third-party arbitrator agent to resolve conflicting recommendations based on global conflict resolution knowledge. The DFI is a system for distributed cooperative problem-solving between construction agents. This system resembles the distributed nature of the construction industry by incorporating a multi-agent architecture which models design, fabrication and erection processes [34]. The architecture, by using negotiation, considers each participant's important issues in the connection design process and produces a cooperative solution. The representation of specialized construction process knowledge in each agent's framework of models, facilitates the addition of new knowledge as it is acquired. The modular nature of the architecture enables new agents with their new construction expertise to be added with little customization required.

The DFI system evaluates multi-agent viewpoints and suggests alternative connection designs. During the proposal process, the agents form their viewpoints with both cooperative and competitive behavior. They work jointly toward a common goal by suggesting alternative configuration connections which resemble the connection design originally specified by the user, but simultaneously, compete with each other, attempting to maximize their own positions within the problem- solving domain. An agent formulates its viewpoint from several unique issues based on different connection aspects such as economics, feasibility and material type. Within a connection evaluation process, the importance of a particular agent issue is determined by which agent viewpoint is taken, designer, fabricator or erector. When making a proposal, an agent will evaluate the connections previously proposed by other agents for any affected issues. During the agents' proposal process, a third-party arbitrator monitors the proposals and mediates by accessing an abstract level shared knowledge base containing agent's issues. The negotiation process produces a final set of alternative connections which is usually acceptable to all agents.

3.2 Case-Based Reasoning

Sycara [31] uses case-based reasoning (CBR) extensively in the development of the PERSUADER, a decision support system which provides enhanced conflict resolution and negotiation support for group problem solving, specifically labour management disputes. CBR is an AI approach to machine learning where the process and results of decisions (i.e. cases) are retained and subsequently used to solve future related problems (i.e. case-based reasoning). As experiences and sessions are accumulated, system performance improves providing more intelligent support, and in essence, the system learns from experience. There are several advantages of using CBR as a learning methodology [31]. Incremental learning can be pursued without attention to the order of presented cases. Previous successful decisions can be exploited providing short cuts to current problem solving. Similarly, previous failures and reasons for failure can be used to recognize and avoid similar failures in advance, as well as support recovery from current problem solving failures. If the repair of a failure is also stored in the database, then the repair can be accessed and re-used for a similar problem. At the conclusion of each problem-solving session, the case database is updated with the process and results of the decision-making session.

In the PERSUADER, a case is solved by first accessing relevant precedent cases from the case database and selecting the most appropriate cases. Using the most important features from the current case, a baseline solution is constructed which is adapted to the details of the current problem using heuristics and CBR. The baseline solution is then evaluated for applicability to the current case and modified accordingly to obtain a candidate solution.

Within the case database, cases are structured hierarchically based on important concepts in the problem domain. Cases are retrieved based on whether they share salient attributes with the current case. Sycara [31] uses a high-level knowledge structure called generalized episode [14] to organize similar concepts in the case database. Concepts within a generalized episode structure are organized in a hierarchical discrimination network where nodes are either an individual case or another generalized episode [31]. Within a generalized episode, there are norms which are collections of features that represent the abstraction of all the cases which are organized under this generalized episode. Indices are used to connect a generalized episode with other generalized episode trees and cases organized below it. In the generalized episode, cases are indexed based on the features which differentiate them from the norm. As cases are introduced and added to the case database, the generalized information describing the collection of features in the norm part of an existing generalized episode is improved and additional case indices are added. In this method, salient attributes are learned and incorporated into the case database structure [31]. Within the PARI DSS, CBR will not be used for negotiation and mediation per se as with the PERSUADER, but will contribute to the problem-solving activities particularly when a similar problem has arose previously.

4 The Paradigm of Cooperating Experts

When human experts cooperate on a single problem, they contribute their multiple diverse viewpoints in the problem-solving process. Merging diverse knowledge is common in real-world situations and brings with it a source of robustness and balance [16] which is required to develop comprehensive credible solutions. A plant breeder and an agrologist work together to develop crops with high quality and good yields. A chemist and an machinery engineer will cooperate to produce effective and environmentally safe herbicides. In general, the greater the complexity of problem-solving, the more cooperating experts required. Cooperation and diversity can be advantageous, providing increased problem-solving abilities beyond the individual expert and promoting creativity and innovation [17]. However, conflicts result from cooperation which must be resolved among the experts through information exchange. Conflict resolution among heterogeneous reusable agents in the PARI DSS pose some special problems which are addressed in Section 4.1.

4.1 Cooperating Expert Systems

The PARI DSS is being developed collaboratively with cooperation of specialists from federal and provincial governments, universities and private industry. All of the expert systems within the PARI DSS either exist "on-the-shelf" as reusable agents or are being developed as reusable agents. A specialized multi-agent system which shares expertise has several advantages over a large monolithic system developed to solve the same problem(s) [17]. The multi-agent system approach provides the PARI DSS with a natural, flexible development framework in which to integrate heterogeneous reusable agents. This approach provides the capability to implement agents independently, with minimal time and resource requirements, add agents to the PARI DSS as they become available, and to delete or substitute agents in response to changes in specifications and technology. Agents developed for multiple uses are generally more reliable and their development cost can be amortized over multiple uses [20]. In addition, a modular expert approach provides the ease and simplicity associated with developing, debugging, testing and maintaining small systems.

The PARI DSS is to utilize reusable expert systems, and with minimal modification to them, if any, integrate them into a cooperating multi- agent system. There are two main approaches to developing cooperative expert system environments. A popular approach is to establish a global structure containing information specifically about each agent and corresponding conflict strategies [13, 17, 21, 34]. A contrasting approach is to develop each agent with its own conflict knowledge, separate from its domain level knowledge, and conflict management knowledge, which is not accessible or known to other agents [26, 25]. There are tradeoffs between these two approaches. With the global structure approach, information must be specifically defined about each agent and corresponding conflict strategies. However, with the contrasting approach, each agent must be developed specifically with its own set of conflict resolution schemes and translation capabilities, in order for the system to interact. Dependencies exist in both approaches. One approach has dependence on the global conflict structure and the other approach has dependence on the agent itself.

The approach, in which each agent has a built-in set of conflict resolution schemes and translation capabilities, defies the term *reusable agent*, and hence, is not applicable to the PARI DSS. The global conflict structure approach holds more promise for conflict management within the PARI DSS. Although each agent in a cooperating expert system is independent and a fully functional knowledge-base system, Lander and Lesser [16], do not believe that "off-the-shelf" agents will work effectively, even with minor modifications. Some of their major reasons include [16]:

- a shared communication language does not exist within a reusable agent;
- there is no mechanism to handle inconsistency and unique beliefs among reusable agents;
- inadequate internal knowledge representations do not permit sufficient goal and history information (i.e. cases) to be retained for cooperative solution revision;
- there is no mechanism for sharing timely information during problem-solving;
- there is no provision for resolving conflicts.

Section 5 addresses these concerns and describes the application of conflict management techniques in the PARI DSS to accomplish the use of "off-the-shelf" reusable agents.

5 Application of Conflict Management Techniques in the PARI DSS

Within the PARI DSS, it is necessary to integrate several conflict management techniques to use the "off-the-shelf" reusable intelligent agents in this cooperative problem-solving system. This is not an easy task as stated by Lander and Lesser [20]; "There is a high degree of complexity inherent in building heterogeneous agents that can understand each other well enough to positively affect mutual work." Figure 1 illustrates the general system architecture for level-3 of the PARI DSS. A Global Control Expert (GCE) is being developed to interact with a Global Knowledge Base (GKB), which will be implemented using a blackboard-based problem-solving approach. The blackboard provides the communication medium since a common communication language among reusable agents is not possible. Meta-knowledge about the reusable agents, accessible by the GCE, will be utilized to handle inconsistency and unique beliefs among the agents. The GKB will contain the meta- knowledge about each reusable agent including declarative knowledge, assumptions, beliefs, problem-solving techniques, solution evaluation criteria, minimum and maximum constraints, if applicable, conflict threshold limits, and other relevant meta-knowledge. Reusable agents post their information to the blackboard via a translation data structure which temporarily holds the data while the GCE, utilizing agent meta-knowledge, determines the required subset of data required to be placed on the blackboard for each agent and the problem-solving process, and then distributes this data to the blackboard. Case-based reasoning, utilizing a case history database, may be incorporated to solve future related problems. Compromise negotiation and weighted constraints are evoked, utilizing the GKB, when conflict thresholds are reached. In some cases, an agent may need to relax some requirement on the solution in order to resolve conflict. Flexibility values are used which indicate the degree of relaxation permitted. The following discusses the combination and extension of these approaches to conflict management, in order that a successful methodology can be developed to use "off-the-shelf" reusable intelligent agents in the PARI cooperative problem-solving paradigm.

5.1 The Blackboard Model Approach to Problem-Solving

The blackboard model is a relatively complex problem-solving model providing the conceptual organization of information and knowledge, and prescribing the dynamic control and use of it pertaining to the problem-solving behavior within the overall system [7, 12]. The blackboard data structure, in its most general form, consists of a global data area containing problem-solving state data. Agents incrementally make changes to the blackboard based on their problem-solving ability at a given time. The agents communicate and interact solely through the blackboard. A controlling mechanism, the GCE in the case of the PARI DSS, is responsible for determining which agent to activate, when, and using which part of the blackboard. In addition to organizing information and knowledge, blackboard systems have a particular reasoning behavior associated with them as a result of incremental problem-solving. At each solution step or control cycle, any type of reasoning step can be initiated in response to the part of the emerging solution being addressed. Hence, the control mechanism selects and applies agents dynamically and opportunistically rather than by fixed and preprogrammed methods. In some cases, the control mechanism might activate an *a priori* determined set of knowledgeable sources, as in the PARI DSS.

The development of the blackboard model was motivated by the need for flexibility in reasoning and for information sharing. The blackboard model is useful in a broad range of applications and is able to facilitate different problem-solving methods. In particular, it is being used within the PARI DSS because of its capability to handle the following problem characteristics [12]:

- a huge solution space;
- various input data;
- a need to integrate diverse information;

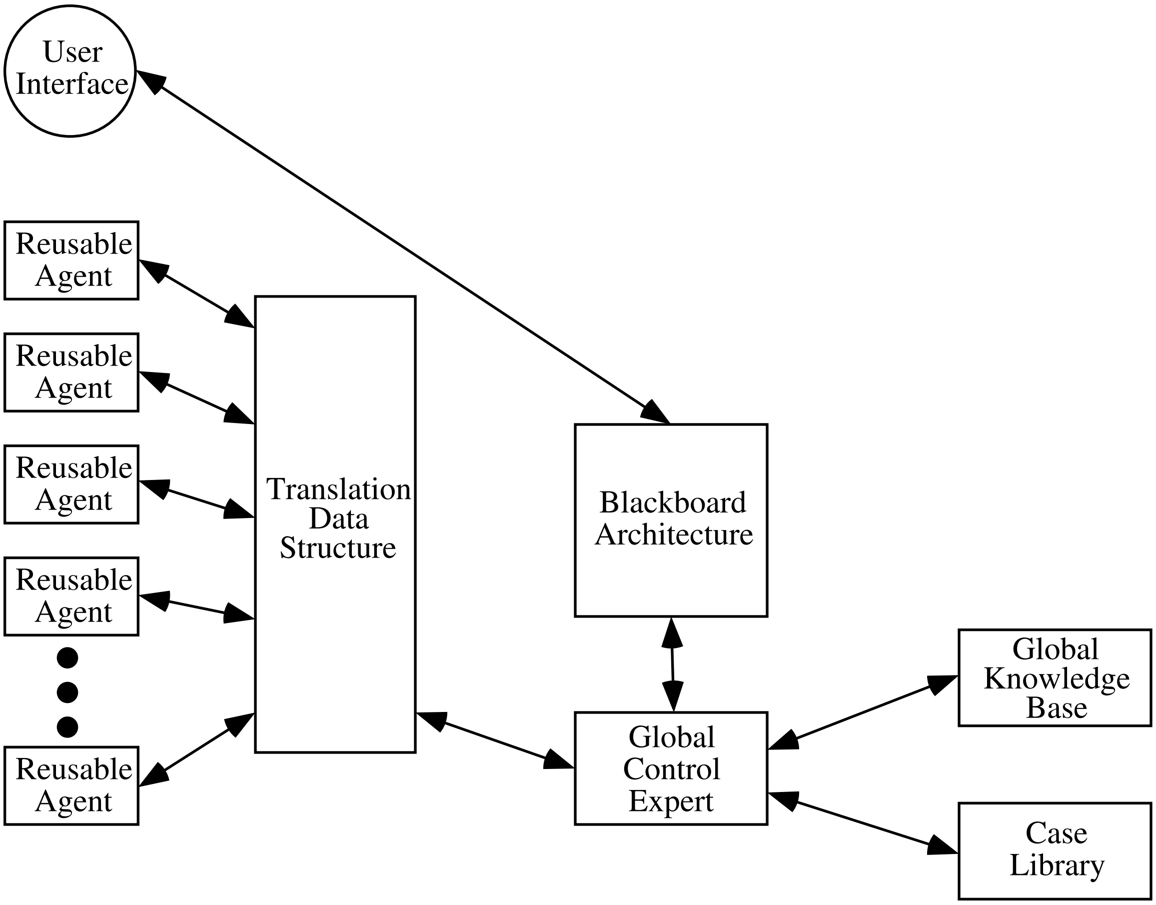

Figure 1: PARI DSS Level-3 System Architecture

- a need for cooperation and coordination among heterogeneous reusable agents in forming credible solutions or recommendations;
- the requirement for an evolutionary solution.

5.1.1 Recognition-Oriented Blackboard System

Within the problem-solving paradigm of the blackboard model, there are two methods for determining the next state in the solution process; search and recognition. At any solution step or state, the search method generates and evaluates the possible next states, whereas the recognition method simply knows what the next state should be through a pattern matching process that involves scanning the knowledge base for knowledge that can be applied to a state [12]. In the case of the PARI DSS, the recognition-oriented approach is taken (i.e. the knowledge base is the GKB and the knowledge that can be applied to a state is either activation rules or conditional rules). Naturally, there may be several pieces of knowledge (i.e. rules) applicable for a given state and a decision is necessary to determine what knowledge (i.e. rules) to apply. This decision is also knowledge based, or in the case of the PARI DSS, rule-based. In a recognition-oriented blackboard system, this decision, results in triggering either an activation rule or a conditional rule. The triggering of an activation rule will fire and control a set of agents, based on the predetermined rules for a given situation, resulting in a change to the solution space on the blackboard. The triggering of a conditional rule will fire a predetermined rule set for a given situation resulting in a contribution to the solution space on the blackboard. In the PARI DSS, the activation rule sets, located in the GKB, trigger and control the reusable agents in their search for a solution. The conditional rule sets provide less complex problem-solving, where the superior intelligence of agents is not required.

5.2 The Global Control Expert

The Global Control Expert (GCE) provides the global control necessary for interaction, communication and coordination among agents. In addition, the GCE has access to meta-knowledge about the reusable agents, which is contained within the GKB, and used to handle inconsistency and unique beliefs among the agents. In this way, the GCE is able to reason about the coordinating processes used among the agents in the cooperative environment. The GCE, by utilizing the GKB, allocates tasks to the agents by matching the task requirements with the agent's knowledge and resources. The GCE triggers agents into action based on influence from the agents' contributions, the knowledge of other agents, and the problem-solving process. Since there is no common communication language shared among the agents, the GCE monitors the agents' processing of tasks, utilizing the GKB, providing the necessary interaction and communication among the agents via the blackboard and the translation data structure. Reasoning on the part of the GCE is aided by meta-knowledge, and possibly case-based histories, enabling the GCE to determine future actions of an agent.

As the agents interact through the GCE, they post contributions to the shared portion of the blackboard. The GCE detects conflicts when agents post contributions which are judged to be in disagreement with each other based on the constraints and goals of the original task. In resolving conflict among heterogeneous reusable agents, the GCE has sufficient artificial intelligence, and access to required knowledge, so as to have a global view of the problem-solving activity with the power to make global decisions that are coherent and in the best interest of the agents involved. In so doing, it may be necessary to settle for only partial goal satisfaction. On the other hand, possibly goals can be relaxed, so that all goals can be fulfilled. In the extreme case of conflict resolution, perhaps some goals need to be fully relinquished in order to find a solution. The GCE must be able to make decisions about what goals to emphasize, and localize revisions to these goals as much as possible.

5.3 Meta-Knowledge

As the PARI DSS changes, with the addition of new agents and deletion of agents no longer required, the meta-knowledge within the GKB needs to be updated. The objective of the GCE, when using meta-knowledge, is not to locate a specific solution, but to focus the GCE in its search for a solution [as in [20]]. In order for the GCE to guide the processing towards an acceptable result, the GCE must be able to determine the agents' requirements for solutions. This is accomplished by utilizing the agents' meta-knowledge. Agent meta-knowledge will consist of knowledge including:

- what are the agents' goals;
- what implicit heuristics are employed;
- what results is the agent capable of providing;
- what are the agents' input parameters;
- what are the agents' conflict thresholds;
- what are the agents' weighted constraints;
- how are constraints imposed.

One of the key entities of the meta-knowledge will be constraints. The GCE must be able to assimilate constraining information of each agent in order to guide the problem-solving activities. Constraints are discussed in more detail in Section 5.4.

5.4 Compromise Negotiation and Relaxation

In heterogeneous reusable agent problem-solving, each agent can determine the status of its local solution, but there is no effective method for an agent to determine the status of the global solution, as no agent has knowledge of the constraints of other agents [18]. Although researchers [32] have addressed this problem to some extent in their literature on constraint-directed negotiation, the investigations they conducted on this subject in a multi-agent environment necessitated that the agents share a global consistent problem-solving methodology and agent architecture. These agents are heterogeneous, based on their resource requirements, but homogeneous in their implementation, and implicitly use their homogeneity for control.

Although conflict resolution can be attempted using any of several methods of negotiation, compromise negotiation holds the most promise for dealing with negotiation among heterogeneous reusable agents in the PARI DSS. Compromise negotiation uses a methodology where values are iteratively revised by sliding them along some dimension until a mutually agreeable position is found [17]. An example of compromise negotiation is demonstrated when a customer is purchasing a car. The car buyer and salesman iteratively propose prices (i.e. slide a value along a monetary scale) until the two proposals converge. For compromise negotiation to be effective, a small number of dimensions should be involved, a method should exist for determining whether the proposed values are moving toward each other, and the values should be within an acceptable range [16]. The GCE will utilize weighted constraints from the agent meta-knowledge in the GKB and hence, will know the agents' "bottom line" constraints, leaving the main task of identifying and compromising the salient constraints. By establishing a set of conflict attributes as part of the meta-knowledge for each agent, and defining threshold values for these attributes, the GCE can reason as to when to try compromise negotiation.

An alternate approach to conflict resolution among heterogeneous reusable agents in the PARI DSS is relaxation. Relaxation methods are invoked when the problem-solving activity is believed to be over constrained [18] resulting in no or little progress in solution building. Relaxation is invoked when one or more agents relax some requirement on a goal or solution, thereby enabling problem-solving to advance. Sometimes through relaxation, a case stored in the case history database which previously failed, may become immediately acceptable. Within each agent's meta-knowledge, weighted constraints are assigned to each information type or variable, which define the degree to which that agent is willing to relax that information [18]. When managing conflict, the GCE will evaluate the weighted constraints of the agents and the constraints being imposed by the solution analysis and thereby determine if relaxation is feasible.

5.5 A PARI DSS Scenario

Through the use of a third-party agent, a shared global blackboard, agent meta-knowledge, and several conflict resolution techniques, we are attempting to utilize "off-the-shelf" reusable agents. Integrating heterogeneous reusable agents in the collaboratively developed PARI DSS provides a natural, flexible development methodology, providing the capability to add agents as they become available and to delete or substitute agents in response to changes in specifications and technology. Agents developed for multiple uses are generally more reliable and their development cost can be amortized over multiple uses [20]. Conflict resolution plays a major role in the success of integrated reusable agents and forms the basis for communications and coordination protocols. The following scenario will help to illustrate the role of conflict resolution among heterogeneous reusable agents in the PARI DSS. The following examples of PARI DSS reusable agents will be used: 1) CROPMAN, a crop management expert system (ES), which focuses on crop production including crop selection and rotation, herbicide requirements, fertility, and potential plant disease [3]; 2) a Stepwise Technology Adoption Risk Reduction Tool ES called STARRT, which evaluates the economic impact of different cropping decisions [1]; and 3) a planting and residue ES called PARMS, which determines the level of success of planting a crop given the planting equipment, current production practices and conservation goals [24].

Even without considering other reusable agents in the PARI DSS, such as weed and insect agents, the complexity of integration and conflict resolution can be illustrated. A producer, during pre-growing season planning, requests the PARI DSS to advise on the crop(s) to grow with the best economic net return. The producer's field-specific database is queried and required data is placed on the global blackboard. The GCE triggers CROPMAN which determines crop candidates that can be grown based on crop rotations, fertility, soil moisture, potential plant disease, herbicide residue and agro-climatic zone. These results are posted to the blackboard via the translation data structure, and the GCE triggers STARRT which uses the information on the blackboard and determines the best crop(s) to grow on each field based on present market outlooks and input costs, and posts its results to the blackboard. The GCE invokes PARMS which uses the information from the blackboard, machinery inventory, and current levels of crop residue to determine that the crop(s) recommended cannot be seeded without equipment modifications due to the high level of crop residue present. PARMS posts the applicable information to the blackboard including the costs for equipment modification. The GCE recognizes a conflict and accesses the case history database to realize that this decision process and results have not occurred previously and updates the case-based history database. The GCE, with the aid of the GKB, determines that STARRT needs to reevaluate the net return given the cost of equipment modifications. If the net return indicates the recommended crops are not cost effective, then CROPMAN is triggered again with imposed crop constraints. The process continues until a viable net return results with no conflicts.

6 Conclusions

Lander [15] states; "An agent can be either an existing piece of software, modified to work within an agent set, or it can be a piece of software specifically designed as a reusable agent to work within an agent set. The first case is more problematic." Neches et al. [23] indicate that sharing and reusing knowledge sources in any form is difficult, but feasible, if knowledge sharing technology and infrastructure tools can be created with which to facilitate knowledge-based system development and operation. The PARI DSS is challenged with the integration and utilization of "off-the-shelf" reusable intelligent agents in this cooperative problem-solving system.

We believe that the PARI DSS has an advantage when utilizing "off-the-shelf" reusable agents in that the recognition-oriented blackboard system method being employed will simplify the problem-solving paradigm over the search method. The GCE, currently under development, plays a major role in the PARI DSS. As the control mechanism, the GCE is comprised of problem-solving heuristics as opposed to application heuristics. The GCE's responsibilities include monitoring the status of the blackboard to determine if the current solution state necessitates the triggering of an activation rule, or a conditional rule, and then controlling these predetermined rule sets, including the predetermined set of agents, if an activation rule is fired. The GCE, as the knowledge source control executive, must be able to assimilate and apply information from agents in order to guide the problem-solving activities. Therefore, agent information must be consistent with the GCE's local knowledge and thereby, the GCE relies heavily on agent meta-knowledge. The GCE's most difficult task will be guiding the problem-solving activities of "off-the-shelf" reusable agents in a cooperative environment to provide comprehensive decision support. We have discussed the role of conflict resolution in the integration of heterogeneous reusable agents and how it forms the basis for communication and coordination protocols.

Although there are several conflict resolution approaches and techniques that exist and are currently being researched, we believe that our investigation and consideration of the available conflict management methodologies has resulted in the development of a design which holds the most promise for integrating heterogeneous reusable agents. In addition to our current complement of conflict resolution methodologies and techniques, we believe the incorporation of case-based reasoning would assist in the solution of future related problems. The GCE, GKB and blackboard architecture are currently being developed. Our plan is to utilize case-based reasoning in the future and possibly other approaches depending on the result of our prototype development. The PARI DSS is being built as an open system and hence, there will be ongoing development.

References

[1] Alexiev, V., Duhaime, K., and Dzikowski, P. STARRT — Stepwise Technology Adoption Risk Reduction Tool (v1.3) [Computer software]. Alberta Agriculture, Edmonton, AB, 1995.

[2] Bond, A. H., and Gasser, L., Eds. *Readings in Distributed Artificial Intelligence.* Morgan Kaufmann Publishers, San Mateo, California, 1988.

[3] de Gooijer, H. C. CROPMAN — Crop Management (v2.0) [Computer software]. 20-20 Agricultural Services, Kelliher, SK, 1995.

[4] Durfee, E. H. *A Unified Approach to Dynamic Coordination: Planning Actions and Interactions in a Distributed Problem Solving Network.* PhD thesis, University of Massachusetts, Amherst, MA, 1987.

[5] Durfee, E. H., Lesser, V. R., and Corkill, D. D. Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers 36*, 11 (Nov 1987), 1275–1291.

[6] Durfee, E. H., Lesser, V. R., and Corkill, D. D. Cooperation through communication in a distributed problem solving network. In *Distributed Artificial Intelligence*, M. N. Huhns, Ed. Pitman Publishing/Morgan Kaufman Publishers, San Mateo, CA, 1987.

[7] Engelmore, R., and Morgan, T., Eds. *Blackboard Systems.* Addison-Wesley Publishing Company, New York, 1988.

[8] Goldstein, I. Bargaining between goals. In *Proc. 1975 International Joint Conference on Artificial Intelligence* (Tbilisi, USSR, 1975), pp. 175–180.

[9] Hewitt, C. The challenge of open systems. *Byte 10*, 4 (1985), 223–242.

[10] Hewitt, C. Offices are open systems. In *Readings in Distributed Artificial Intelligence*, A. H. Bond and L. Gasser, Eds. Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[11] Hewitt, C. Open information systems semantics for distributed artificial intelligence. *Artificial Intelligence 47*, 1 (1991), 79–106.

[12] Jagannathan, V., Dodhiawala, R., and Baum, L. S., Eds. *Blackboard Architectures and Applications.* Academic Press, San Diego, CA, 1989.

[13] Klein, M., and Lu, S. C.-Y. Conflict resolution in cooperative design. *The International Journal for AI in Engineering 4*, 4 (1990), 168–180.

[14] Kolodner, J. L. *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model.* Lawrence Erlbaum, Hillsdale, NJ, 1984.

[15] Lander, S. E. *Distributed Search and Conflict Management Among Reusable Heterogeneous Agents.* PhD thesis, University of Massachusetts, Amherst, Mass, 1994.

[16] Lander, S. E., and Lesser, V. R. A framework for cooperative problem-solving among knowledge-based systems. In *Proc. 1989 MIT-JSME Workshop on Cooperative Product Development* (1989), pp. 1–14.

[17] Lander, S. E., and Lesser, V. R. A framework for the integration of cooperative knowledge-based systems. In *Proc. 1989 IEEE International Symposium on Intelligent Control* (Washington, DC, 1989), IEEE Computer Society Press, pp. 472–477.

[18] Lander, S. E., and Lesser, V. R. Customizing distributed search among agents with heterogeneous knowledge. In *Proc. 1992 First International Conference on Information and Knowledge Management* (1992), pp. 335–344.

[19] Lander, S. E., and Lesser, V. R. Understanding the role of negotiation in distributed search among heterogeneous agents. In *Proc. 1993 International Joint Conference on Artificial Intelligence* (1993), pp. 438–444.

[20] Lander, S. E., and Lesser, V. R. Sharing meta-information to guide cooperative search among heterogeneous reusable agents. Tech. Report CMPSCI 94-48, Department of Computer Science, University of Massachusetts, Amherst, MA, 1994.

[21] Lander, S. E., Lesser, V. R., and Connell, M. E. Conflict resolution strategies for cooperating expert agents. In *Proc. 1990 International Working Conference on Cooperating Knowledge Based Systems* (Berlin, Germany, 1991), Springer-Verlag, pp. 183–200.

[22] Lesser, V. R., and Erman, L. D. Distributed interpretation: A model and experiment. *IEEE Transactions on Computers 29*, 12 (Dec. 1980), 1144–1163.

[23] Neches, R., et al. Enabling technology for knowledge sharing. *AI Magazine 12*, 3 (Fall 1991), 36–56.

[24] Pavlik, C., Smith, E., and Lindwall, C. W. PARMS — Planter And Residue Management System (v2.1) [Computer software]. Agriculture and Agri-Food Canada, Lethbridge, AB, 1995.

[25] Polat, F., Shekhar, S., and Guvenir, H. A. Distributed conflict resolution among cooperating expert systems. *Expert Systems 10*, 4 (Nov. 1993), 227–236.

[26] Polat, F., Shekhar, S., and Guvenir, H. A. A negotiation platform for cooperating multiagent systems. *Concurrent Engineering: Research and Applications 1*, 3 (Sep. 1993), 179–187.

[27] Rosenschein, J. S. Formal theories of knowledge in ai and robotics. Tech. Report CSLI-87-84, Center for the Study of Language and Information, Stanford, CA, 1987.

[28] Rosenschein, J. S., Ginsberg, M., and Genesereth, M. R. Cooperation without communication. In *Proc. 1986 Conference for the American Association for Artificial Intelligence* (1986), pp. 51–57.

[29] Smith, R. G., and Davis, R. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man and Cybernetics 11*, 1 (1981), 61–70.

[30] Sycara, K. P. Arguments of persuasion in labor mediation. In *Proc. 1985 International Joint Conference on Artificial Intelligence* (Los Angeles, CA, 1985), pp. 294–296.

[31] Sycara, K. P. Machine learning for intelligent support of conflict resolution. *Decision Support Systems 10*, 2 (Sep. 1993), 121–136.

[32] Sycara, K. P., et al. Distributed constrained heuristic search. *IEEE Transactions on Systems, Man and Cybernetics 21*, 6 (Nov./Dec. 1991), 1446–1461.

[33] Tenney, R. R., and Sandell Jr., N. R. Strategies for distributed decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics 11*, 8 (1981), 527–538.

[34] Werkman, K. J., et al. Design and fabrication problem solving through cooperative agents. Report No. 90-05, ATLSS Engineering Research Centre, Lehigh University, Bethlehem, PA, Jul. 1990.

[35] Winograd, T., and Flores, F. *Understanding Computers and Cognition.* Alblex, Norwood, NJ, 1986.