A Knowledge-Based Approach to Coalition Formation

Michal Pěchouček, Vladimír Mařík, and Jaroslav Bárta,
Gerstner Laboratory, Czech Technical University

Planning humanitarian relief operations is a challenge for several reasons, not the least of which is that the involved players are often both vaguely linked and hesitant to share information. Typically, coalitions are organized through a central planning component that distributes the plan with collaborative agents. We suggest an alternative knowledge-based approach where all agents collaborate in forming coalitions and planning humanitarian and peace-keeping missions.

Establishing coalitions among humanitarian aid organizations is a challenge. The CPlanT system uses a knowledge-based approach to reduce complexity and communication traffic, while also protecting stakeholder privacy.

We developed our CPlanT multiagent knowledge-based system with two goals in mind:

- Simplify coalition formation, and thus make it more efficient
- Maintain the confidentiality of agents' private information

To reduce complexity, we base our system on *alliances*—a set of agents that agree to share some private information and eventually cooperate. In CPlanT, each agent represents a complex, organized entity—such as a non-governmental organization (NGO), humanitarian organization, or army unit—that plays an active role in mission planning. These agents group themselves in various, temporary coalitions, each of which solves a specific mission or part of a larger mission.

Using alliances reduces the coalition-formation complexity by splitting the larger agent community into disjunctive alliances and creating coalitions, preferably within alliances, to carry out work. Furthermore, the system reduces communication traffic and preserves information privacy by combining classical negotiation mechanisms, such as contract net protocol (CNP), with acquaintance models that store an agent's social knowledge.[1,2]

In this article, we offer a domain and CPlanT overview, followed by a more detailed discussion of the system architecture. We then discuss results from our experiments with CPlanT on a humanitarian relief scenario.

Domain overview

Coalition-formation research is commonly associated with war avoidance operations such as peacekeeping, noncombatant evacuations, or disaster relief operations. Unlike classical war operations, where decision-making technology is strictly hierarchical, operations other than war (OOTW) are typically cooperative efforts involving several vaguely organized groups of people (often volunteers). These NGOs and humanitarian groups often work alongside military and other government organizations. OOTW operations also differ from more hierarchical planning in that it is relatively flexible and dynamic in how it groups individual organizations. New entities can freely join an operation autonomously and get involved with planning according to their capabilities. Given this, any organization framework must be essentially "open."

Because OOTWs do not necessarily have a single shared goal or operation metric (such as political, economical, or humanitarian), plan evaluation occurs from multiple perspectives.[3] Thus, the goals of individual groups within a coalition might conflict. Even when community members share the same goal, it can be easily misunderstood due to different cultural backgrounds.

In addition to planning and negotiation issues, a further domain challenge involves information sharing. Planning and organizing operations under a central authority is difficult because NGOs are typically reluctant to provide detailed information about their intentions, goals, and resources. Consequently, we must address organizations' need for assurances in this area. Many institutions will readily share resources and information within a well-defined community, but will refuse to do the same with a central planning system and also refuse to follow centralized commands. However, many will participate in executing a plan if they've played an active role in its formulation.

Basic CPlanT operations

The CPlanT system has four consecutive operational phases: registration, alliance formation, coalition formation, and team action planning. The system uses three primary communication techniques: a central communication agent, a contract net protocol, and acquaintance models. Because it is typically a communications bottleneck and centralizes information (leaving it vulnerable to disclosure), we minimize the role of the central communication component, using it only in the registration phase.

Registration

As we describe in more detail below, each agent contains two basic types of knowledge: private knowledge, which contains its alliance-formation restrictions, and public knowledge, which typically includes organization type, objectives, and country of origin. In the registration phase, new agents register their public knowledge (stored in their acquaintance model) with the community's central communication unit, or facilitator. The facilitator then informs all existing agents about the new agents, and informs new agents about existing agents. (Agents can also deregister with the facilitator.) A registered agent stores the public knowledge about other agents in its acquaintance model. Because agents register only their public knowledge, this phase poses no threat to the confidentiality of their private information.

Alliance formation

In this phase, agents analyze the agent information that they've stored in their acquaintance model and attempt to form alliances with other agents. In principle, each agent is expected to compare its own private knowledge with the public knowledge of other alliance members. When the agent detects a possible future collaborator, it proposes joining the alliance. The agent typically bases its collaboration preferences on existing alliances. If it fails to find a good option, it might start a new alliance by itself.

We define an alliance's quality in terms of maximizing the individual agent's contribution to the alliance. That is, the agent's contribution covers significant services that other alliance members cannot implement. This process does not guarantee an optimal alliance allocation. Each agent simply joins the most profitable alliance given the existing alliance configuration. If agents register with the alliance in a different order, CPlanT's formation algorithm might create different alliances.

Coalition formation

In this phase, agents group together to perform a single, well-specified task. In a single alliance, agents typically know each other and can thus suggest a coalition with foreseeable properties. To coordinate the coalition, CPlanT chooses an agent at random. The coordinator then detects the most suitable collaborators. To do this, the coordinator first sends a subscribe-type of message to solicit potential collaborators based on the services they provide. If their capabilities change, collaborators update the coordinator by sending an inform-type message. When the coordinator triggers the coalition-formation phase, it parses the prepared service offers and selects the best collaborators without further negotiation. Once the coordinator sends a request, the collaborator updates its resources and confirms the contract. Any change in collaborator resources is advertised to all subscribing coordinators (see Figure 1).

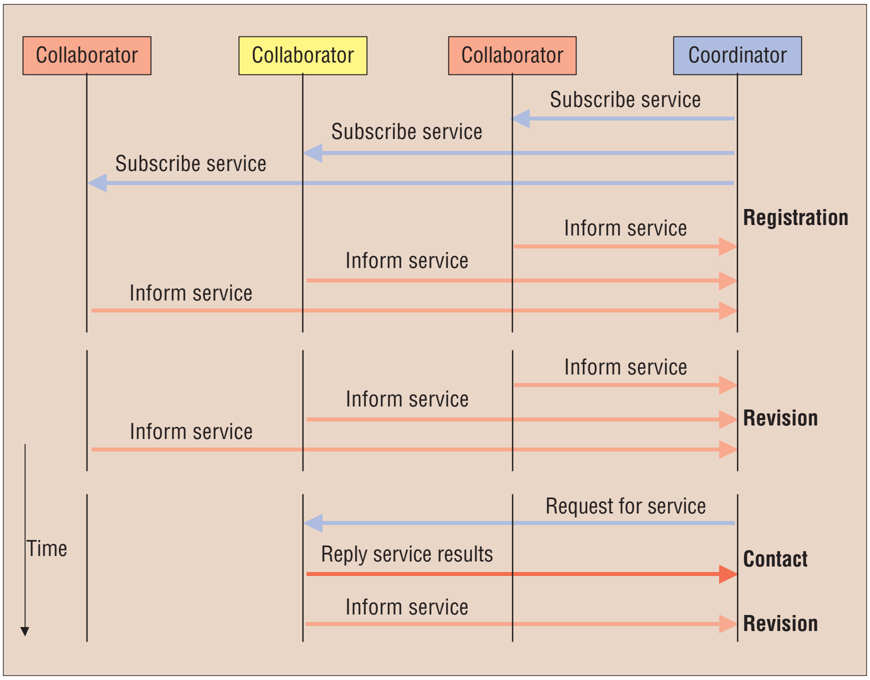


Figure 1. The collaboration coordination process. The coordinator solicits collaborators, who update the coordinator if their capabilities change.

On receiving collaboration proposals, the coordinator selects the best possible collaborators using the contract net protocol (see Figure 2) The coordinator optimizes the coalition's quality by seeking the coalitions that can contribute the most and in the shortest possible time. As we describe in the "Knowledge disclosure" section, agents prefer to form coalitions within their own alliance because going outside it entails knowledge loss. However, if an agent fails to find an alliance that can achieve the goal within its alliance, it extends negotiations across neighborhoods. Such a multistaged process requires substantial computational resources and typically fails in complex communities.

Team action planning

In this phase, a team of collaborative agents creates a team action plan that defines exactly how each team member will contribute to achieving the goal, accounting for such things as resources, deadlines, and so on. To begin, the coordinator decomposes the goal into subtasks and then searches for the most efficient subtask allocation within the coalition.

Collaborators advertise their services in an informative but efficient form. Currently, we use linear approximation of the discrete function that maps the delivery amount into due dates. The coordinator's acquaintance model thus stores imprecise but compact

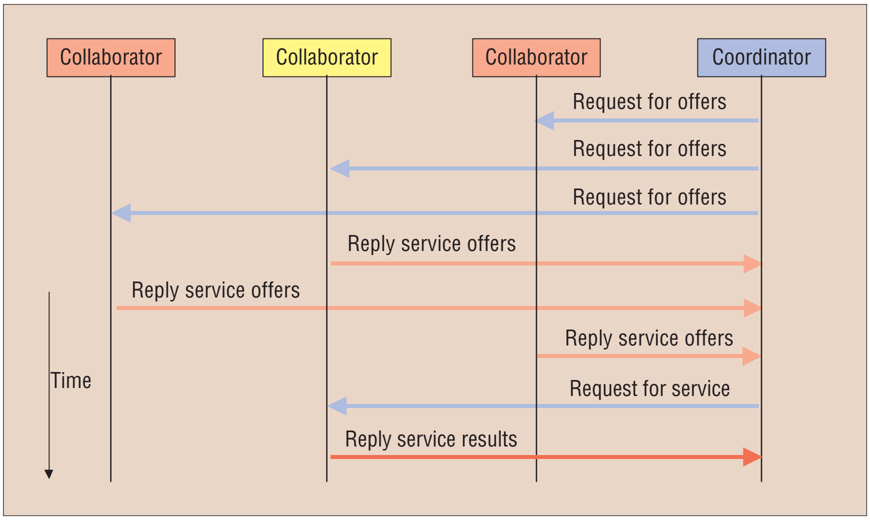


Figure 2. Single-stage contract net protocol process. The coordinating agent requests services from other agents, who then send collaboration proposals. The coordinator then selects the best possible collaborators.

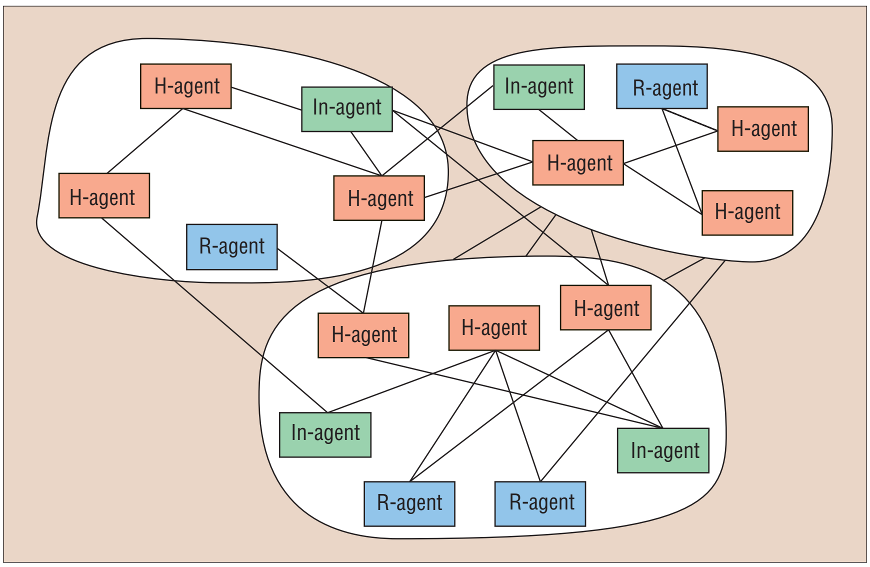


Figure 3. CPlanT architecture features three interacting agent classes: resource agents (R-agents), In-need agents (In-agents), and humanitarian agents (H-agents).

social knowledge that it can efficiently parse. According to this social knowledge, the coordinator suggests the most optimal request decomposition and resource allocation in a contract proposal.

There might be several achievable action plan options; the coordinator seeks either the cheapest or the fastest possible plan; coalition members then work together with the coordinator to achieve the precise plan, often through several rounds of negotiation. This process is iterated until there is no conflict in the expected capacity of the collaborators and the proposed delivery.

CPlanT architecture

As Figure 3 shows, the CPlanT architecture has three specific agent classes:

- *Resource agents* (R-agents) represent the in-place resources (such as roads and airports) required to deliver humanitarian aid. R-agents are passive and do not initiate any kind of humanitarian effort.
- *In-need agents* (In-agents) represent the conflict centers that need help, such as villages or cities.
- *Humanitarian agents* (H-agents) represent the participating aid agencies. Because they contribute to aid missions, H-agents are technically a subclass of R-agents. However, H-agents are active and can initiate the coalition-formation process.

Each H-agent can participate in one alliance of "friendly" agents and simultaneously be actively involved in several agent coalitions, cooperating to fulfill specific shared tasks. The computational and communication complexity of forming such a coalition depends on the amount of previously prepared information the agents have about each other and how sophisticated they are at reasoning about the other agents' resources, plans, and intentions. Therefore, each agent reasons about other agents within its reasoning scopes—their "neighborhoods"—and vice versa. An agent (here, agent A) typically has several types of neighborhoods:

- The *total* neighborhood, $\alpha(A)$, is the set of all agents that agent A is aware of (that is, it knows about their existence and can communicate with them).
- The *social* neighborhood, $\mu(A)$, is a set of agents that A has specific information about, such as the services they provide and their status and load. This neighborhood consists of the agents set that A reasons about—$\mu^+(A)$—and the agents set that reasons about A–$\mu^-(A)$. Therefore, $\forall B \in \mu^-(A): A \in \mu^+(B)$.
- The *cooperation* neighborhood—$\varepsilon(A)$—is a set of agents jointly collaborating (or committed to collaboration) in achieving one or more shared goals.

Knowledge sharing

To reason about each other, agents must share some of their knowledge. Let's suppose that the operator, $(\mathsf{Bel}\,A\varphi)$, expresses agent A's awareness of the formula φ being true.[4] We say that agent A_0 intentionally shares its knowledge $\mathsf{K}(A_0)$ with a set of agents $\delta(A_0) \subseteq \Theta$, provided that

$$\mathsf{K}(A_0) = \{\varphi\}: \forall\varphi \in \mathsf{K}(A_0): \forall A_i \in \delta(A_0): (\mathsf{Bel}\,A_i\varphi) \wedge \forall B_i \notin \{\delta(A_0) \cup \{A_0\}\}: (\mathsf{Bel}\,A_0 \neg (\mathsf{Bel}\,B_i\varphi)).$$

It thus follows that if an agent B knows some of the shared information, but agent A_0 is unaware of this fact, agent B is not regarded as a member of the $\delta(A_0)$ set of agents, representing A_0's neighborhood. This classification suggests three levels of an H-agent's knowledge sharing: public, semi-private, and private knowledge.

Public knowledge is shared within the entire multiagent community. If we assume that all the agents know about each other—that is, $\forall A, A \in \Theta$: $\alpha(A) = \Theta$—we define the public knowledge $\mathsf{K}_\mathsf{P}(A_0)$ of agent A_0 as:

$$\mathsf{K}_\mathsf{P}(A_0) = \mathsf{K}(A_0) \qquad \text{where } \delta(A_0) = \alpha(A_0).$$

Public knowledge is freely accessible within the community. We define public knowledge specifically as the agent's name, the organization type it represents, the general objectives of its activity, the country its registered in, its human-to-human contact (telephone, fax number, and email), its human-to-agent contact (http address), its agent-to-agent contact (the IP address, incoming port, and agent communication language), and, finally, its available services.

Semi-private knowledge is shared within an agent's social neighborhood. We define semi-private knowledge, $\mathsf{K}_\mathsf{s}(A_0)$ of agent A_0, as

$$\mathsf{K}_\mathsf{s}(A_0) = \mathsf{K}(A_0) \qquad \text{where } \delta(A_0) = \mu(A_0).$$

Given that this is the OOTW domain, we cannot assume that agents will share knowledge within overlapping alliances, thus the social neighborhood must have the following property: $\forall A \in \Theta$: $\mu^{-}(A) = \mu(A)$. That is, members of a social neighborhood must share information about their resources.

Agents own and administer their *private knowledge.* We define private knowledge, $\mathsf{K}_\mathsf{P}(A_0)$ of agent A_0, as

$$\mathsf{K}_\mathsf{pr}(A_0) = \mathsf{K}(A_0) \qquad \text{where } \delta(A_0) = \{\,\}.$$

Key types of private knowledge include an agent's collaboration preferences, alliance restrictions, coalition-leader restrictions, as well as an agent's planning and scheduling algorithms.

Defining alliances, coalition, and team action plans.

Alliances and coalitions have different formal properties, as we now describe.

Alliances. Alliances are a long-term cooperation agreement among agents. Alliance members constitute each other's social neighborhoods. Assuming that each agent also belongs to its own social neighborhood—$\forall A \in \Theta$: $A \in \mu(A)$—we define an alliance as a set of agents κ, so that

$$\forall A \in \Theta: \exists \kappa: A \in {}^\wedge \forall A_i \in \kappa: \kappa = \mu(A_i).$$

We regard a singleton agent as an alliance with just one member. Given the requirements for reciprocal knowledge sharing within an alliance, it follows that

$$\forall A \in \kappa: \kappa = \mu(A).$$

Therefore, an alliance cannot overlap with another alliance:

$$\forall \kappa_{1,} \kappa_2 \subseteq \Theta: (\exists A: A \in \kappa_1 {}^\wedge A \in \kappa_2) \Rightarrow \kappa_1 \equiv \kappa_2.$$

Coalitions. A coalition is a set of agents that agreed to fulfill a single, well-specified goal. Coalition members are committed to collaborate on this goal. Assuming $\forall A \in \Theta$: $A \in \varepsilon(A)$, we define coalition as a set of agents χ, so that

$$\forall \chi(\tau) \subseteq \Theta: \forall \mathrm{A} \in \chi(\tau) \subseteq \varepsilon(\mathrm{A}).$$

Given an agent collaboration, $\varepsilon(A, \tau)$, with respect to the goal τ,

$$\varepsilon(A) = \bigcup_{\tau} \varepsilon(A, \tau), \text{ and } \forall \chi(\tau): \chi(\tau) = \varepsilon(A, \tau).$$

Unlike an alliance, we regard a coalition as a short-term agreement between collaborative agents. Although coalitions are ideally a subset of an alliance, they can consist of agents from different alliances.

Team action plan. Agents must agree on how they will achieve the goal τ. The team action plan is thus a decomposition of a goal τ into a set of tasks $\{\tau_i\}$ that are delegated to coalition members. We denote each task by its responsible agent, due time, start time, and price. Assuming that agent A_j is responsible for implementing task τ_i (where $\tau = \{\tau_i\}$) on time $\mathsf{due}(\tau_i)$, starting at start(τ_i) for the price(τ_i), we define the team action plan $\pi(\tau)$ as a set

$$\pi(\tau) = \{\langle \tau_i, A_j, \mathsf{start}(\tau_i), \mathsf{due}(\tau_i), \mathsf{price}(\tau_i)\rangle\}.$$

The team action plan $\pi(\tau)$ is correct if all collaborators can implement the task in the given time at the given price. The plan is accepted if all agents commit to implementing the task in the given time at the given price. Similarly, goal τ is achievable if $\pi(\tau)$ is correct and planned if $\pi(\tau)$ is accepted. Obviously, there is an important relation between the team action plan and the coalition: A coalition $\chi(\tau)$ achieves a goal τ by implementing a team action plan $\pi(\tau)$, if and only if $\chi(\tau) = \{A_j\}$ and $\pi(\tau)$ is correct.

Knowledge disclosure

Measuring the loss of information that agents want to keep private is difficult, because information has different value to agents with different metareasoning capabilities.[5] However, to roughly categorize information loss, we distinguish between two types of information leaks:

- *Strong* information disclosure occurs when an agent loses private or semi-private knowledge as a side effect of some proactive step (such, when it sends a request).
- *Weak* information disclosure occurs when an agent deliberately discloses private knowledge to other agents (such as, when it sends an **inform**-type message).

In our system, each agent experiences weak knowledge loss when forming an alliance. An agent experiences strong knowledge loss if it communicates with an agent outside of its alliance. For example, once agent A_1 from alliance κ_1 sends a service request for τ to agent A_2 from alliance κ_2, agent A_1 reveals information about both its intent (A_1 is doing something that requires τ) and its capabilities (A_1 cannot do τ). At the same time, a collaboration proposal from A_2 reveals agent A_2's capabilities (A_2 can implement the service in time t_1, for example). However, this type of knowledge disclosure is less problematic, because agent A_2 acts on behalf of its entire alliance. Therefore, if A_2 offers services that are not ultimately used, it loses information about the alliance's capabilities but not specifically its own.

Social knowledge

Most agents use two types of knowledge: problem-solving and social. Problem-solving knowledge guides an agent's autonomous local decision-making processes, including coalition formation and team action planning, and is stored in the agent's computational core. Social knowledge expresses other behavioral patterns related to conversations or negotiation scenarios,[2] including the agent's capabilities, load, experiences, resources, and commitments. This knowledge is stored in an agent's acquaintance model.

Researchers have investigated several acquaintance models; we based ours on the tri-base acquaintance model. [6] As Figure 4 shows, we organize social knowledge in four separate knowledge structures:

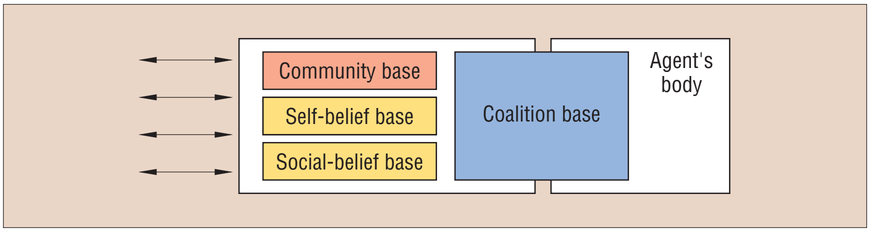


Figure 4. The CPlanT acquaintance model. The model is divided into four sections, each containing a particular type of knowledge. The arrows on the left side represent communication links between the agent and its neighborhood.

- The *community base* (Com-BB), which contains community members' public knowledge: $\mathsf{Com\text{-}BB}(A_0) = \{\mathsf{K_p}(A_i)\}$ for $\forall A_i \in \alpha(A_0)$.
- The *self-belief base* (Self-BB), which contains the agent's reflective knowledge about itself and its public, semi-private, and private knowledge: $\mathsf{Self\text{-}BB}(A_0) = \{\{\mathsf{K_p}(A_0)\}, \{\mathsf{K_S}(A_0)\}, \{\mathsf{K_{Pr}}(A_0)\}\}$.
- The *social-belief base* (Soc-BB), which contains the agent's semi-private knowledge about peer alliance members: $\mathsf{Soc\text{-}BB}(A_0) = \{\mathsf{K_S}(A_i)\}$ for $\forall A_i \in \mu(A_0)$.
- The *coalition-base* (Coal-BB), which is a dynamic collection of knowledge about peer coalition members, past and possible future coalitions, and of permanent coalition-formation rules. Because it contains both problem-solving knowledge (the coalition-formation rules) and social knowledge (information about coalition members, and past and future coalitions), the coalition base belongs to both the acquaintance model and the agent's computational core.

Table 1 shows knowledge examples for each acquaintance model category.

In principle, social knowledge substantially reduces the set of agents (ideally to one) that the coordinating agent in the CNP process will request.[1] However, social models have high maintenance costs. Such maintenance might be requestor driven (driven by the acquaintance model's owner, the coordinator) or—as we chose for CPlanT—provider driven (driven by those the model represents,

Table 1. Instance of an H-agent's acquaintance model.

Self-Belief Base		
Public knowledge	**Semi-private knowledge**	**Private knowledge**
Port: 1500 **ip_address**: "147.32.86.167" **Country:** suffer terra **City:** north port **Type:** Religious **Ontologies:** fipa-am, cplant-ontology	Food: 3000 Nurses: 50 Trucks: 13	Alliance restrictions: ("country","Suffer Terra") Leader restrictions: ("type","Military") City restrictions: ("muslim",50) Cooperates with: ("type","government")
Social belief base		
Agent: ST Police	Armed-people:30	
Agent: Christian STHO	Food: 3500 Nurses: 22	Clothing: 280 Medical-people: 12
Community belief base		
Agent: Suffer Terra Government	Suffer Terra Government@iiop://147.32.84.131:2188/Suffer Terra Government Type: Government Services: Food, Civil-material, Medical-material, Clothing Ontologies: FIPA-AGENT-MANAGEMENT, MAP-ONTOLOGY, PORT-ONTOLOGY, CPLANT, ALLIANCE Languages: SL1, KIF, State: ACTIVE Country: Suffer Terra, City: Suffer Town	
Agent: Christian STHO	Christian Suffer Terra Humanitarian Organization@iiop://147.32.84.131:2210/Chr ST Humanitarian Organization Type: Religious Services: Food, Clothing, Medical-people, Nurses, Medical-material Ontologies: FIPA-AGENT-MANAGEMENT, MAP-ONTOLOGY, PORT-ONTOLOGY, CPLANT, ALLIANCE Languages: SL1, KIF, State: ACTIVE Country: Suffer Terra, City: North Port	
Coalition Base		
Rules	(VOLCANIC-AVERAGE-SMALL-TOWN —> Time: 220 (Requirements: Medical-material 60, Food 1500, Civil-material 30000, Medical-people 16, Civil-people 27, Nurses 19) ...	
Coalitions	(coalition (Members: Suffer Terra Government, Suffer Terra Police, Christian Suffer Terra Humanitarian Organization)	
	(Services: Food, Civil-material, Medical-material, Clothing, Military-people, Food, Clothing, Medical-people, Nurses) (Price-for-coordination: 5)) (planned-coalition (Task name: Suffer-Town-24-1-2002/17-49-53.1 (Coalition members: Suffer Terra Government, Suffer Terra Police, Christian Suffer Terra Humanitarian Organization) (Coalition leader: Christian Suffer Terra Humanitarian Organization (Disaster: Volcanic, Degree: 1, (Allocations: Civil-material, 80000, Allocation Time: 350 Food, 80000, Allocation Time: 350 Medical-material, 80000, Allocation Time: 350))	

the collaborators providing the service). An example of a requestor-driven strategy is periodical revisions,[7] in which the knowledge owner periodically checks model consistency with the potential collaborators. Some systems use a cooperation trader[8] agent, which maintains the agents' social knowledge.

Exploiting the acquaintance model's social knowledge lets us

- Minimize required communication traffic and thus increase problem-solving efficiency.
- Maintain the quality of coalition operations (measured by the delivery time for humanitarian relief and the percentage of request coverage).
- Minimize how much semi-private information agents lose when negotiating the mission (by revealing minimal information about the agent services, status, and intention).

It also lets us minimize the shared information that potential coalition leaders have about other agents and use to plan optimal missions

Implementation and testing

Testing CPlanT's correctness required a well-defined, formal, yet realistic scenario in which we could represent, model, and initiate all aspects of agents' nontrivial behavior. We thus developed a scenario for humanitarian relief operations, inspired by several real-world projects.[3,9,10]

Scenario: Sufferterra

Sufferterra is a hypothetical scenario of a suffering island and several imaginary countries prepared to help (see Figure 5). We encoded scenario knowledge in XML and implemented the computational model in Allegro Common Lisp. Figure 6 shows a code sample for a scenario object.

In Sufferterra, the R-agents specify the physical arrangements of the geographical objects and the resources they provide. The problem specification does not distinguish the level of modeling granularity; that is, we can implement either individual objects or groups of objects as a single R-agent. For testing purposes, we implemented the entire area map as a single R-agent.

The H-agents subscribe to the R-agent for specific information and are thus informed about any change in physical arrangements on the island. We implemented a simple In-agent as a part of the CPlanT community.

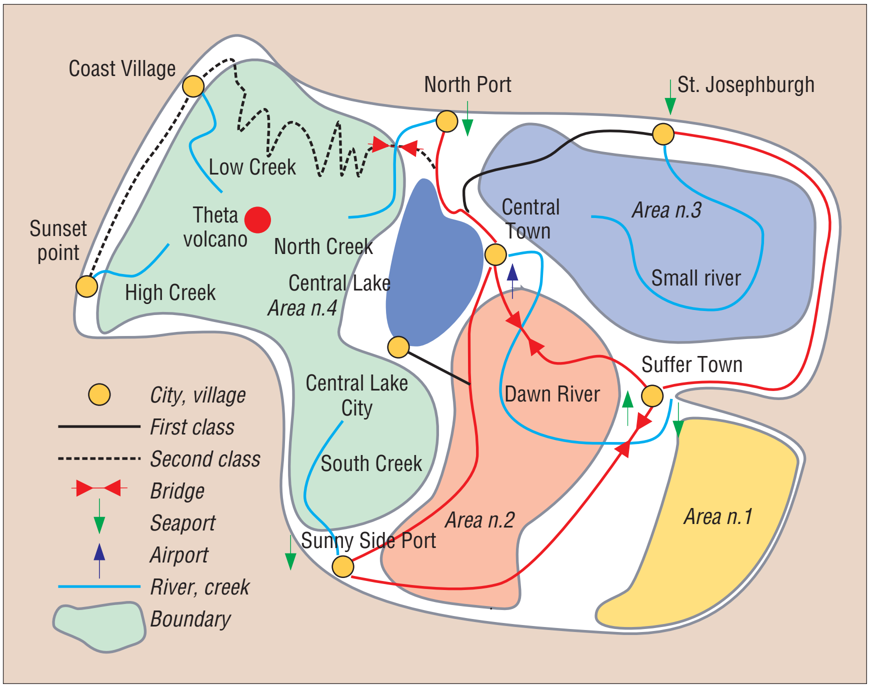


Figure 5. A map of Sufferterra. The island-based scenario was inspired by several real-world humanitarian relief projects.

```
<city>
    <name> "Suffer Town" </name>
    <national-composition> "((christian 67) (muslim 18) (native 13) (other 2))"
    </national-composition>
    <population> "50000" </population>
    <seaport>
        <ID> "1" </ID>
        <capacity> "25" </capacity>
        <material-hour> "200000" </material-hour>
    </seaport>
    <airport>
        <ID> "1" </ID>
        <capacity> "30" </capacity>
        <material-hour> "100000" </material-hour>
        <runway> "3000" </runway>
    </airport>
</city>
```

Figure 5. Example of XML encoding of the Suffer Town object, which represents a city on Sufferterra island.

Through the In-agent's running instance, we composed a "call-for-help" request and executed the coalition planning process. The request included the disaster type (volcanic, hurricane, flood, or earthquake) the disaster's degree (on a 1 to 9 scale), the location, and the targeted H-agent.

To visualize Sufferterra, we implemented a meta-agent in Java that lets us view the system's logical structure, including alliances, coalitions, team action plans, and other community properties. We also implemented a separate visualization for communication traffic monitoring, which is not an agent itself, but rather is a part of the multiagent platform that we mainly used for debugging. We viewed the community and sent requests using a Web server and common Internet browsers and WAP phone interfaces.

CPlanT experiments and results

The objective of our experiments was to evaluate communication and computation requirements, solution quality, and private and semi-private knowledge disclosure. Our tests involved a total of 20 agents.

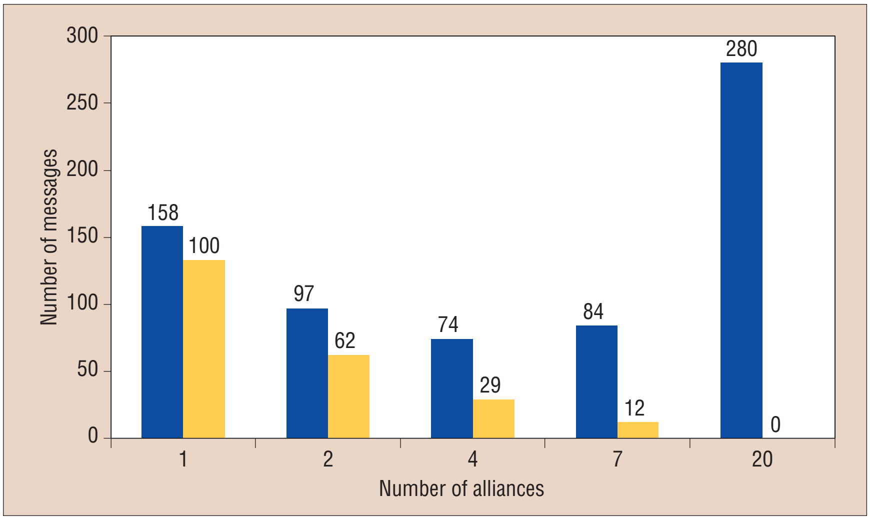


Figure 7. Communication traffic in communities with different numbers of alliances. The light bar depicts the maintenance messages, and the dark bar illustrates the overall system communication.

Communication traffic. In our experiments, we observed the communication traffic both in different architecture arrangements of the community, such as different alliances, and through different task complexities, such as the number of contracts.

We tested several different agent configurations, including all 20 agents in one alliance and agents clustered in 2, 4, 7, and 20 alliances. We conducted all tests using a set of 19 measurements for each community arrangement. Given the systems operation explained above, it follows that the latter case of 20 alliances ($\forall A$: $\mu(A) = \varnothing$) does not exploit any advantages of the acquaintance model contract, because the community interacts without using any social knowledge.

An important part of communication traffic occurs when the coordinator requests a plan from the system. By exploiting prepared social knowledge, our aim was to minimize communication traffic at this critical moment. The cost we paid, however, was in increased communication traffic during idle times. During such "idle" times, agents are busy maintaining the social knowledge stored in their acquaintance models. Communication traffic grows as the number of alliances increase and each member stores a larger acquaintance model that it must parse to find a coalition.

As Figure 7 shows, when we increase the number of alliances (and decrease the average number of alliance members), we reduce the communication requirements for model maintenance. The difference between the dark and light bars shows communication at the critical plan-request time; as the chart shows, we save most when there is one huge alliance.

We identified an optimal community arrangement of four alliances. However, we could not define an optimal system structure, because the agents cannot predict future tasks or the number of agents required for implementing them. Clearly, for tasks requiring few agents, we prefer small alliances; a task requiring many agents is best served by a larger alliance. Thus, the target task determines the optimal coalition size.

Coalition quality. In the Sufferterra scenario, two key attributes influenced coalition value: success rate, determined by how many requested resources the coalition provided; and delivery time, determined by when the coalition delivered the resources to the requestor. Our experiments showed no evidence of dependence between a coalition's success rate and its communication mechanism. However, as the number of alliances increase, the overall delivery time increases due to the additional costs of coordination among the alliances.

Knowledge disclosure. Minimizing private and semi-private knowledge disclosure is a key challenge. In our experiments, we tried to measure both types of leaks. As expected, the highest rate of strong disclosure appears when there are 20 alliances, as such an arrangement has the highest CNP-based communication traffic among alliances (see Figure 8a and b). With 20 alliances, agents are totally independent, and there is no weak disclosure. Unlike

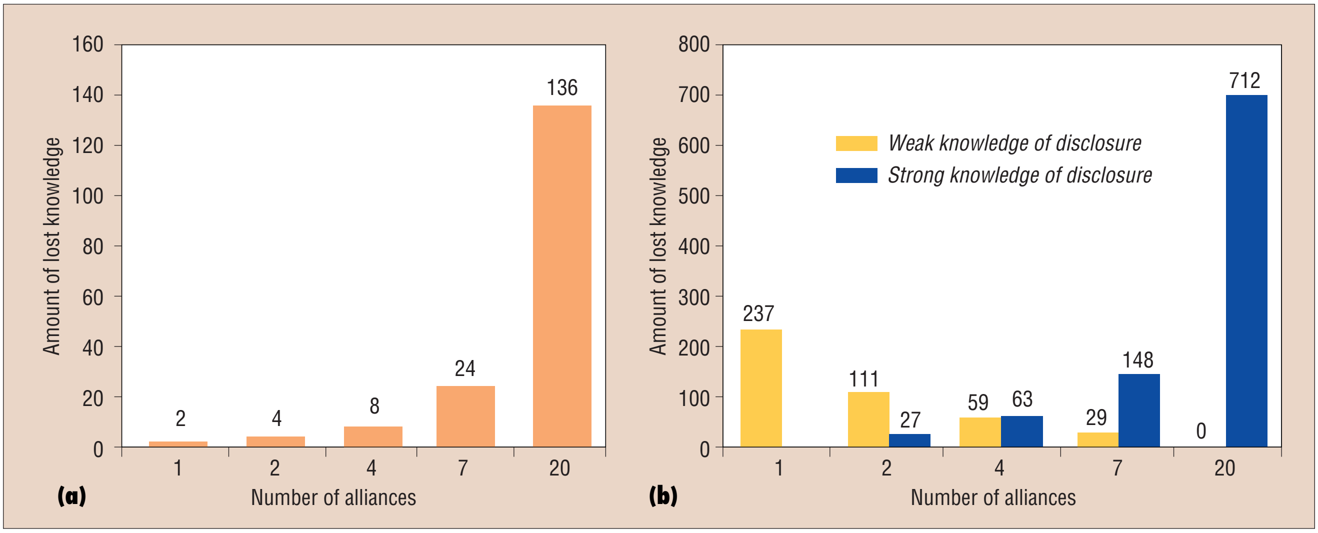


Figure 8. Information disclosure. (a) The relation between private information disclosure and number of alliances. (b) Disclosure of the semi-private knowledge.

Related Work: Coalition Planning Research

Coalition formation and coalition planning is a well-investigated area. Researchers have shown that the problem of finding an optimal coalition is NP complete.[1] Typically, researchers suggest different negotiation strategies and analyze the complexities of the coalition-formation process.[2]

When coalition quality is the optimization target, agents usually act collaboratively, and researchers have published many centralized planning mechanisms for coalition formation.[3] Unlike collaborative agents, self-interested agents maximize their own profit when participating in a coalition, no matter how well they will perform as a group. Among the researched properties of self-interested agent communities are their stability, worst-case profit, or payoff division among the agents.[4]

The OOTW domain combines cooperative and self-interested behaviors. Agents providing humanitarian aid tend to cooperate in crisis but operate competitively and with self-interest over the long term. It was this factor that led us to investigate the alliances concept, in which agents agree to collaborate (and potentially form a coalition).

In addition to agent collaboration or competition, profit is a key optimization criterion in the coalition-formation process. Along with coalition quality, we must account for at least two other important aspects of the OOTW domain. The first is response time. Although optimal coalition formation is a deeply complex problem, agents also have limited resources, so a reasonably good answer, quickly provided, is often much better than an optimal coalition that takes a long time to find.[1,5]

The second factor is the potential for communication traffic overload, which often results when implementing a multiagent system with numerous agents that interact heavily.[6] In our research, we've attempted to decompose the reasoning process and distribute it among the agents. We've kept the agents' deliberation process simple and focused on minimizing agents' communication interaction to structure communities within a reasonable timeframe. Because OOTW agents are in part self-interested, they're motivated to stay hidden from some agents and advertise their collaborative capabilities to others. For this reason, we must consider information disclosure, which motivated our study of information leaks in coalition formation.

Some researchers have investigated teamwork in the similar domain of evacuation scenarios.[7] They suggested integrating existing software applications in Teamcore wrapper agents. Unlike our acquaintance model, which contains only social knowledge, Teamcore wrapper agents also maintain domain-specific team plans and a goal hierarchy. In this system, agent teams share a team-oriented program, which is a joint knowledge structure that coordinates their activities. In CPlanT, the acquaintance models do not contain an explicit team action plan. Rather, the coalition structure and team action plan results from the inter-agent negotiation process. However, combining these approaches, so that agent behavior is coordinated by a team action plan that results from agent negotiation, might be an interesting topic for further research.

Investigators approaching the problem from the perspective of game theory solve a more complex problem. Whereas in our case, each task is sent to a hierarchical community and is coordinated by a single agent, in game-theory research, all agents are equal.[8] The agents autonomously analyze their own value and, through negotiations, try to discern which coalition is the most profitable for them to join. This problem (and the agent community) is inherently more complex and causes communication problems. Because optimal cooperation between two agents is not always reciprocal, several negotiation stages are required. We did not face a problem of that complexity. However, in CPlanT, we must optimize not only which coalition an agent should join, but also which services agents should provide to the coalition (such as team action planning). We could use the game-theory approach in the CPlanT's alliance-formation phase. However, because agents are continuously joining the system, the overall optimality of alliance distribution would be rather difficult to maintain.

In addition to the contract net protocol, classical auctioning mechanisms offer other possible negotiation strategies. In combinatorial actions, an agent is typically motivated to make the biggest profit or contribute to a coalition in the best way. However, in our case, all auctioneers and bidding agents collaborate. A bidding agent tries to provide the auctioneer with a bid that best approximates the resources it can provide, and the auctioneer suggests the best possible resource allocation. Also, CPlanT agents do not speculate about whom to work with. Because we optimize the private information loss, collaboration within an alliance is always preferred. However, we do see potential for optimizing multiple auctioning mechanisms for team action planning within several overlapping coalitions.[9]

References

1. T. Sandholm and V. Lesser, "Coalitions among Computationally Bounded Agents," *Artificial Intelligence*, Special Issue on Economic Principles of Multi-Agent Systems, vol. 94, no. 1, 1997, pp. 99–137.
2. O. Shehory and S. Kraus, "Coalition Formation among Autonomous Agents: Strategies and Complexity," *From Reaction to Cognition*, C. Castelfranchi and J. Muller, eds., Springer Verlag, Heidelberg, 1995, pp. 57–72.
3. T. Sandholm et al., "Coalition Structure Generation with Worst Case Guarantees," *Artificial Intelligence*, vol. 111, nos. 1–2, 1999, pp. 209–238.
4. C. Li and K. Sycara, *Algorithms for Combinatorial Coalition Formation and Payoff Division in an Electronic Marketplace*, tech. report CMU-RI-TR-01-33, Carnegie Mellon Univ., Pittsburgh, 2001.
5. J. Steinmetz et al., "An Efficient Anytime Algorithm For Multiple-Component Bid Selection in Automated Contracting," *Agent Mediated Electronic Trading*, Springer Verlag, Heidelberg, 1998, pp. 105–125.
6. G.A. Kaminka, D.V. Pynadath, and M. Tambe, "Monitoring Deployed Agent Teams," *Proc. 5th Int'l Conf. Autonomous Agents* (*Agents-2001*), ACM Press, New York, 2001.
7. M. Tambe, "Towards Flexible Teamwork," *J. Artificial Intelligence Research*, vol. 7, 1997, pp. 83–124.
8. M. Klusch et al., "Coalition Formation in a Power Transmission Planning Environment," *Proc. 2nd Int'l Conf. Practical Applications of Multi-Agent Systems (PAAM-97)*, 1997; available at www.uclm.es/area/gsee/JavierC/papers/paper_DAI.ps.
9. P. Anthony et al., "Autonomous Agents for Participating in Multiple On-line Auctions," *Proc. IJCAI Workshop, E-Business and the Intelligent Web,* Advanced Knowledge Technologies, 2001; available at www.csd.abdn.ac.uk/ebiweb/papers/anthony.pdf.
10. S. Ketchpel, "Coalition Formation Among Autonomous Agents," *Proc. 5th European Workshop on Modeling Autonomous Agents in a Multi-Agent World*, Springer Verlag, Berlin, 1993, pp. 73-88.

independent agents, agents within a single alliance have no strong semi-private information disclosure. Given their different natures, we saw no value in comparing strong and weak disclosure.

One interesting finding was that neither a single alliance nor 20 alliances is the best arrangement for concealing agents' private and semi-private knowledge. With one alliance, the semi-private knowledge becomes public; with no alliance, each CNP will reveal information about the contractors' intentions. It's rather difficult here to find a good compromise in number of alliances. What matters is the probability that a request will be unfulfilled by an alliance, and thus force the coalition leader to subcontract other agents. Alliance numbers and structures in our domain emerge naturally according to the agents' private knowledge and other collaboration restrictions. Therefore, it makes no sense to suggest an optimal number of alliances for a given community.

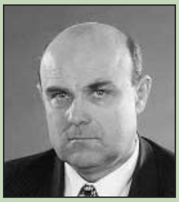
Research shows that the complexity of negotiations involved in coalition formation are exponentially explosive in nature.[11,12] And, as the "Related Work: Coalition Planning Research" sidebar describes, in the absence of constraints, finding an optimal coalition is an NP-complete problem. Nonetheless, we were able to significantly reduce this negotiation complexity for three key reasons.

- We organized agents into alliances; because most coalitions are created within an alliance, it reduces the negotiation space.
- The coalition leader within an alliance is randomly selected. Because each coalition member has the same coordination capacity and can manage negotiations, they don't compete for the role.
- Within an alliance, negotiation is based on the acquaintance model's social knowledge and the CNP technique; pure CNP negotiations are used only for interalliance negotiations, which are less common. Although CNP is rather inefficient, it's important in the latter case because it keeps agents from different alliances independent, and thus they avoid disclosing semi-private knowledge across alliances.

In our approach, we did not prioritize the requirement for global coalition optimality, because this is not a main challenge in the OOTW planning. We are currently investigating the concept of meta-reasoning related to agent's ability to speculate about each other's intentions, goals, resources, private knowledge, and so on. We are also trying to formalize more precise characteristics of agent-collaboration efficiency so we can identify a causality between the amount of shared knowledge and coalition quality. Finally, we're investigating agents' social knowledge to see if they might be capable of reasoning about inaccessible agents—a very practical problem in planning real-time humanitarian relief operations.

The Author

Michal Pěchouček is an assistant professor of artificial intelligence in the Department of Cybernetics, Czech Technical University, and leads the Agent Technology Group at the Gerstner Laboratory. He also works as a senior consultant with CertiCon software and has coordinated several international research projects. His research focuses on agent technologies, multiagent systems, coalition formation, and production planning. He received an engineering degree in technical cybernetics from Czech Technical University and an MSc in IT/knowledge-based systems from the University of Edinburgh. Contact him at pechoucek@labe.felk.cvut.cz.



Vladimír Mařík is a full professor in the Department of Cybernetics at Czech Technical University. He is also managing director of the Rockwell Automation Research Center, Prague. His research interests are in artificial intelligence, multiagent systems, soft-computing, production planning, and computer-integrated manufacturing applications. He received his MSc and PhD degrees in artificial intelligence at the Czech Technical University, Prague. Contact him at marik@labe.felk.cvut.cz.

Jaroslav Bárta is a PhD student in the Department of Cybernetics at the Czech Technical University, and a researcher in the Centre of Applied Cybernetics. His research interests are in technical cybernetics, artificial intelligence, and biocybernetics. He received an engineering degree from the Czech Technical University. Contact him at barta@labe.felk.cvut.cz.

Acknowledgments

AFOSR/European Office of Aerospace Research and Development contract number F61775-00-WE043 and the Ministry of Education, Youth and Sports of the Czech Republic, Grant No. LN00B096, supported our research.

References

1. R. Smith, "The Contract Net Protocol: High-Level Communication and Control in Distributed Problem Solver," *IEEE Trans. Computers*, vol. 29, no. 12, 1980, pp. 1104–1113.
2. V. Marík, M. Pechoucek, and O. _tepánková, "Social Knowledge in Multi-Agent Systems," *Multi-Agent Systems and Applications*, M. Luck et al., eds., Springer Verlag, Heidelberg, 2001, pp. 211–245.
3. E.C.T. Walker, "Panel Report: Coalition Planning for Operations Other Than War," *Knowledge Systems for Coalition Operations 1999 Workshop (KSCO) Workshop*, Artificial Intelligence Applications Institute, University of Edinburgh, 1999.
4. M. Wooldridge, *Reasoning about Rational Agents*, The MIT Press, Cambridge, Mass., 2000.
5. M. Pechoucek and D. Norrie, *Knowledge Structures for Reflective Multi-Agent Systems: On Reasoning about other Agents*, tech. report 538, Dept. Mechanical and Manufacturing Eng., Univ. Calgary, Alberta, Canada, 2000.
6. M. Pechoucek V. Marík, and O. _tepánková, "Towards Reducing Communication Traffic In Multi-Agent Systems," *J. Applied System Studies*, special issue on Virtual Organizations and E-Commerce Applications, vol. 2, no. 1, Spring, 2001.
7. V. Marík et al., "ProPlanT: Multi-Agent System for Production Planning." *Int'l J. Applied Artificial Intelligence*, vol. 14, no. 7, 2000, p. 727–760.
8. W. Cao, C.G. Bian, and G. Hartvigsen, "Achieving Efficient Cooperation in a Multi-Agent System: The Twin-Base Modelling," *Cooperative Information Agents,* P. Kandzia and M. Klusch, eds., Springer Verlag, Heidelberg, 1997, pp. 210–221.
9. R.A. Rathmell, *A Coalition Force Scenario 'Binni—Gateway to the Golden Bowl of Africa,'* tech. report, Defense Evaluation Research Agency, UK Ministry of Defence, 1999.
10. G.A. Reece and A. Tate, *The Pacifica NEO Scenario*, tech. report. ARPA-RL/O-Plan2/TR/3, Artificial Intelligence Applications Inst., Univ. Edinburgh, Scotland, 1993.
11. O. Shehory and S. Kraus, "Coalition Formation among Autonomous Agents: Strategies and Complexity," *From Reaction to Cognition*, C. Castelfranchi and J. Muller, eds., Springer Verlag, Heidelberg, 1995, pp. 57–72.
12. T. Sandholm et al., "Coalition Structure Generation with Worst Case Guarantees," *Artificial Intelligence,* vol. 111, nos. 1–2, 1999, pp. 209–238.