IMPLEMENTING HIERARCHICAL AGENT-HUMAN TEAMWORKS VIA CONSTRAINT-BASED MODELS

Clauirton Siebra and Austin Tate
Artificial Intelligence Applications Institute
Centre for Intelligent Systems and their Applications
School of Informatics, The University of Edinburgh
Appleton Tower, Crichton Street, EH9 9LE, Edinburgh, UK
{c.siebra,a.tate}@ed.ac.uk

Abstract

The teamwork research defines several important concepts regarding the design of agents so that they are in fact able to play as members of a collaborative group. Currently such research has been focused on multiagent domains, where agents have full autonomy to take decisions. However this model is not common in real applications where humans may wish to be in control for critical decisions. This paper discusses the use of a constraint-based ontology to create models of mixed-initiative interaction for teamworks. Furthermore we show a practical application on how hierarchical users can use this approach during operations in critical domains.

Key Words
Teamwork, mixed-initiative planning, ontology.

1. Introduction

An important feature of the agents' development is the level of autonomy that we must provide to agents. In general agents are designed as black boxes that proactively generate outcomes based on intelligent processes (planning, learning, etc.). Therefore, human users are not able to participate during the deliberative processes of agents, to restrict the agents' options or to customise the solution according to their desires.

This black box approach is not suitable for real applications where humans may wish to be in control for critical decisions. Military, search and rescue and long-term space missions, for example, deal with human lives and some decisions can create situations of risk for them.

Considering this fact, some projects have been investigating ways of enabling human control on the agents' processes. The O-Plan Project [1] provides a mixed-initiative style of planning that supports interactions on the part of users, such as ordering goals for expansion, selecting operators to apply and choosing instantiations for planning variables. The TRAINS System [2] treats the human-agent interaction as a spoken dialogue in which humans set the short-term objectives and agents deal with the details. The Lookup System [3] uses a cost-benefit (Bayesian) approach to decide if it needs to take autonomous actions or interact with users during the development of tasks.

Mixed-initiative interaction refers to a flexible interaction strategy in which each participant of the interaction contributes what it is best suited to do at the most appropriate time. In fact the mixed-initiative interaction is not only used by users to keep the control of the agents, but as a technique to combine the abilities of humans and agents. We can say that while users have the ability to take decisions based on their past-experiences (case-base reasoning), agents are able to generate and compare a significant number of options, showing both positive and negative points of such options.

This work discusses the use of such ideas in a teamwork context. Teamwork [4] has become the most widely accepted metaphor for describing the nature of multiagent cooperation, however this gives rise to additional challenges to human-agent interaction [5]. In our application we are designing a hierarchical coalition system that supports the joint planning and execution activities of several human users during search and rescue operations. The mixed-initiative models use a constraint-based ontology that defines ways of restricting the deliberative processes of agents while providing information to assist the users' decisions. We show that the use of constraints is a natural approach to fill the gap between the different ways that the agents and humans solve problems.

The remainder of this document is structured as follows: section 2 summarises the principal ideas of teamwork and shows how such ideas can be extended/adapted to an agent-human teamwork scenario. Section 3 presents the implementation of these ideas via constraint-based models. Section 4 describes a practical application where joint humans are involved in search and rescue operations. Finally section 5 discusses some conclusion and future works.

2. Teamwork: from Multiagent to Human-Agent Interaction

The teamwork theory provides a set of formal definitions that lead the design of collaborative systems. The principal idea is that the team's joint activities do not consist merely of coordinated individual actions, but each participant needs, for example, to make commitments on reporting status of their ongoing activities (failure, completion and progress) and supporting the activities of others participants.

Several works have proposed both frameworks and implementations using the teamwork concepts. SharedPlans [6] argues that each collaborative agent needs to have mutual beliefs about the goals and actions to be performed, and about the capabilities, intentions and commitments of others agents. The Collage system [7] uses this theory to specify discourses between collaborative agents in a simple air travel application. STEAM [8] is an implemented model of teamwork, based on Joint Intention Theory [9], where agents deliberate upon communication necessities during the establishment of joint commitments and coordination of responsibilities.

Although these and others works [10,11,12] have different approaches to deal with different technical problems, they agree that agents involved in collaborative environments need to make commitments on joint activities, reach consensus on plans and also make commitments to the constituent activities of such plans.

While such early research on teamwork was mainly focused on agent-agent interaction, there is a growing interest in various dimensions of human-agent interaction [13]. This new way of thinking on teamwork theories/applications will require additional requirements for their development. Relevant requirements[1] are:

• During agent-human interactions, agent inaction while waiting for a human response can lead to potential miscoordination with others coalition members. Agents need to be specified to deal with human delays, but avoiding decisions that can lead to erroneous actions;

• Local decisions taken by a coalition member can seem appropriate for her/him, but may be unacceptable to the team. Thus agents also play the role of restricting the options of users in accord with the global coalition decisions;

• Human users have an additional need of understanding what and why something is happening or will be carried out by the coalition. This is particularly significant when a teammate responds in a specific way to some previous request;

---
[1] Adapted and extended from [5,14,15].

• To operate properly, team members must understand their role inside the coalition and how to play this role in a collaborative way (having, for example, the knowledge about what information is required by their teammates);

• Furthermore agents need to control the direct interaction between human participants, trying to interpret and summarize the knowledge exchanged during the operations and turn clear the strengths and weaknesses of their teammates.

Two initial projects in human-agent teamwork are considering some of these requirements using the same approach of adjustable autonomy, but from different perspectives. In [5] (referred to here as P1) agents are implemented through Markov decision processes to reason about costs and uncertainty of individual and team actions. Using such technique, agents are able to dynamically vary their degree of autonomy, deciding when they need to interact with human users. We can note that the agents still have a considerable control of the interaction. Thus this approach is not very suitable to critical applications.

Differently, the work in [14] (referred to here as P2) explores a human-centred perspective, where humans are seen as the crucial elements in the system and agents are fitted to serve human needs. The models of interaction are represented in the form of policies. As long as the agent operates within the policy, it is otherwise free to act with complete autonomy. Human users can impose and remove the policies at any time, adjusting in this way the level of autonomy of agents. Consequently the main threat is to know how to specify good polices in accordance with the current scenario.

These projects also present a slight difference in terms of the way that agents are deployed in support of the coalition performance. According to [16] there are three deployment options. The first is to support the individual team members during the performance of their own tasks. The second is to allocate to the agent its own subtask as if we were introducing another member into the team. This approach is exemplified in P2 where agents (robots) act as members of the team during space missions, and an important function of human users is to set policies to restrict their behaviour. The last option is to support the team as a whole. P1 explores this option via its *Electric Elves* multiagent system, which accounts for assisting research groups in rescheduling meetings, choosing presenters, tracking people's locations and so on.

3. The Constraint-Based Approach

This section discusses how we are using a constraint-based ontology to implement models of hierarchical human-agent teamwork. In this way, section 3.1 presents the hierarchical structure of our framework and how it influences some aspects of the models. Section 3.2

summarises the <I-N-C-A> constraint-based ontology that we are using to produce the models. Section 3.3 shows how the teamwork concepts can be expressed via this ontology, while section 3.4 expands such ideas to a human-agent context.

3.1 Hierarchical Organisations

In our research to develop a framework for coalition support systems [17], we are considering hierarchical organisations composed by three levels of decision-making (Fig. 1): strategic, operational and tactical. In this structure, joint users perform different planning activities assisted by customised agents.

As better detailed in [17], users at the strategic level account for building plans at a high level of granularity (analysis and directions). Operational users, in general, account for refining the plans produced in the strategic level, deciding who will carry out the tasks (synthesis and control). Tactical users are the components that, in fact, accomplish the tasks (reaction and execution).
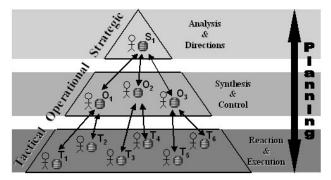

Fig.1: Abstract idea of a three-levels hierarchical coalition, where each joint user is assisted by an agent

Differently to the projects discussed in section 2 (P1 and P2), the agents are mainly concerned in supporting the individual team members of each level during the performance of their own planning activities. In order the coalition effectiveness will indirectly emerge as consequence of local improvements.

The use of a hierarchical organisation has direct influences on the human-agent teamwork models. Such influences are mainly related to the fragmentation of the coalition into subteams. If a hierarchical coalition has n participants, each of them will establish a set of relationships with from 1 to $n-1$ others participants. Thus, in general, a participant does not need to consider the whole coalition during its processes of decision-making or information sharing for example, but only its subteam (set of relationships).

Based on this idea we can trace some simplifications to the human-agent teamwork models. Each member m has no more that one superior and consequently one source of task delegation. This restriction avoids processes of

negotiation between two or more superiors to allocate the same resource m. Negotiation is a complex and time-consumer kind of interaction. Therefore in domains that are time-critical, such as disaster relief operations, it could be appropriate to avoid such interaction.

In the same way, commitments on activity reporting (progress, completion or failure) need to be made only with the unique superior. This superior accounts for locally solving the problems of its subteam. If this is not possible, it will report failure to the upper level. We can note that there is a trend of enclosing the activity problems in local subteams, instead of spreading the problem for whole coalition.

3.2 <I-N-C-A>: A Constraint-Based Ontology

<I-N-C-A> (Issues – Nodes – Constraints – Annotations) [18] is a general-purpose ontology that can be used to represent a plan (Fig. 2) as a set of constraints on the space of all possible behaviours in the application domain. Each plan is considered to be made up of a set of "Issues" and "Nodes". Issues represent potential requirements that need to be considered at sometime. Nodes represent activities in the planning process that may have sub-nodes (sub-activities) making up a hierarchical description of plans. Nodes are related by a set of detailed "Constraints" of diverse kinds such as temporal, resource, spatial and so on. "Annotations" add complementary human-centric and rationale information to the plan, and can be seen as notes on their components.

```
plan = element plan
{
    element plan-variable-declarations
        { element list { plan-variable-declaration* } }?
    & element plan-issues
        { element list { plan-issue* } }?
    & element plan-issue-refinements
        { element list { plan-issue-refinement* } }?
    & element plan-nodes
        { element list { plan-node* } }?
    & element plan-refinements
        { element list { plan-refinement* } }?
    & element constraints
        { element list { constrainer* } }?
    & element annotations
        { map }?
}
```

Fig.2: Part of the <I-N-C-A> schema to plan specifications

By having a clear description of the different components within a synthesised plan, <I-N-C-A> allows such plans (or part of them) be manipulated and used separately from the environment in which they are generated. This feature enables that agents to individually work on different parts of the plan, but without losing the awareness of collaboration.

3.3 Modelling the Teamwork Ideas

In this work, in particular, we are using the Joint Intention Theory [9] to lead our implementation of the teamwork ideas. Basically a team Θ jointly intends to perform a plan p if all its members are jointly committed to complete p. In this way joint intentions are developed on two principal definitions: weak achievement goals (WAG) and joint persistent goals (JPG).

WAG specifies the conditions under which a member holds a goal, and the actions it must take if the goal is satisfied or impossible. WAG(μ,Θ,p,e) implies that one of the following conditions holds:

• Member μ believes that p is not true and desires p to be true (performed) at some future time;

• Having μ privately discovered p to be true, impossible or irrelevant (due to extra condition e), μ has committed to having this private state become public to Θ.

JPG specifies a joint commitment of the team in achieving a goal. A JPG(Θ,p,e) holds if all the following conditions are satisfied:

• Θ mutually believes that p is currently false;

• Θ have p as their mutual goal;

• Each member of Θ holds p as a WAG until Θ mutually believes that p is true, impossible or irrelevant.

We are implementing such concepts in the following way. Considering that the strategic level agent S_1 (Fig.1) has a plan p and the agents O_1, O_2, and O_3 will receive sets of activities a_1, a_2 and a_3 that compose p, then $O_{[1,2,3]}$ need to make a joint commitment on the performance of $a_{[1,2,3]}$ so that p can be true (if possible). As <I-N-C-A> has a clear description of the different components within a synthesised plan (section 3.1), it is possible to define messages that enable the transit of components among the agents. Together with components, such messages also enclose elements that can support the establishment of a JPG. <I-N-C-A> messages for activities are defined as:

```
acitivity ::= < activity
   status = [blank|complete|executing|possible|impossible|n/a]
   priority = [lowest|low|normal|high|highest]
   sender-id = "name"
   ref = "name"
   report-back = [yes|no] >
         <pattern>pattern-element</pattern>
</activity>
```

The operational level agents $O_{[1,2,3]}$ consider the status attribute (of the received messages) as their believes about the current situation of $a_{[1,2,3]}$. While each activity ($a_{[1,2,3]}$) is not complete, $O_{[1,2,3]}$ belief the p is still false. Thus $O_{[1,2,3]}$ indirectly have p as their mutual goal.

The role of $O_{[1,2,3]}$ is to try to perform $a_{[1,2,3]}$ until they individually believe that the performance of such activities is impossible. There are two ways of the activities reach this status: the performer agent individually concludes that it is not able to execute its activity, or S_1 has cancelled p and consequently all its related activities. In both cases the agents use report messages to turn partially (only to the necessary agents) public this information. For example, if O_1 believes that a_1, is impossible, it sends a report to S_1 with this information. Then S_1 will decide if p must be cancelled. If p is cancelled, S_1 also sends reports to $O_{[2,3]}$ so that they abandon $a_{[2,3]}$. The possible report type messages are: success, failure, progress, information and event.

The extra-condition e in the formulation indicates the relevance of p (or their activities) at the current time. Agents can temporarily abandon the performance of an activity if there are others more important activities. In our project this idea is represented via priorities. Components as activities and issues have a qualitative priority attribute, which indicates the importance of their completion. According to the priorities of others activities, for example, agents can decide which one has to be considered during the operation.

Plan activities are sent to agents via delegations. The notion of delegation is very important as a way of setting responsibilities to agents [19]. Generally the delegation of activities does not only involve the sending of an activity itself, but also the sending of a set of constraints associated with it. We are exploring that idea to expand the joint intention implementation so that it avoids conflicts between individual activities and supports the idea of a member assisting their partners.

Conflict between activities can appear because agents are each building their own plans [17] to perform an activity. So, they can produce effects that disrupt the activities of others agents. We are dealing with this threat by adding to delegations, constraints of others activities that need to be respected during a specific interval of time. In this way, activity delegations can contain a set of constraints of several types, which are defined as:

```
constraint ::= < constraint type = "name" relation = "name" >
   <parameters>
      <list>
         <pattern-assignment>
            <pattern>pattern element... </pattern>
            <value>pattern element...</value>
         </pattern-assignment>
      </list>
   </parameters>
</constraint>
```

Using this model, each agent also knows which conditions (constraints) needs to be true so that the activities of their partners can be done. Thus it can try to turn true such conditions if it notices that they are false during the performance of its own activities. As last case

the agent can inform a specific partner that its constraint is not holding as expected. Finally this model also improves the information sharing because the sending of information is guided by the "constraint-based knowledge" that each agent has of the activities of its partners.

3.4 Models of Human-Agent Interaction

According to the <I-N-C-A> approach, each plan is considered to include a set of issues and nodes (activities). During the planning process the general idea is to build a net of activities that resolve as many issues as possible, considering the priority of such issues. This process can be seen in two cycles: first the agent tries to add the nodes (activities) that handle an issue, **respecting** the constraints associated with such issue. After that, if the agent is able to create the nodes, the new constraints are **propagated** so that they restrict the addition of future new nodes. The human interference in this process chooses the issues that are going to be considered and, mainly, leads to the creation of nodes.

Research in mixed-initiative interaction has explored the dialogue/discourse technique [2,7] as the most natural way of leading the work of agents. However such techniques involve some threats regarding the natural process language (NPL) principles. If we analyse a human-agent dialogue, we will note that dialogues are only a high-level way of setting constraints on the deliberative process of agents. Fig. 3 shows an example.

---

1. **Human**: ... go from Atlanta up to Syracuse.
2. **Agent**: Please tell me a route to use to get from Atlanta to Syracuse.
3. **H**: Go via Charleston and Pittsburgh.
4. **A**: Ok. But traffic through Cincinnati is being delayed due to construction.
5. **H**: Ok, let's go via Charleston and Buffalo instead.
6. **A**: No problem.

Fig.3: Example of part of a dialogue (from [2])

In this example, "go from Atlanta to Syracuse" (step.1) is an goal and "go via Charleston and Pittsburgh" (step.3) are constraints set by the human user. However the agent appoints problems in one of the constraints (step.4) and the user decides to relax it (step.5).

In our approach we are using direct mechanisms to add, change or delete constraints, which guide the addition of nodes to deal with an issue. Actually there are two classes of constraints: internal constraints which users can manipulate and external constraints which users can not change (because they are assigned by others users). In this way we can note that there is a mutual process of restriction. While users can set constraints to restrict agents, agents have constraints to restrict the options/behaviour of human users.

The concept of external constraints is very important to adapt the mixed-initiative interaction to a teamwork scenario. While a direct representation of issues and activities ensures that users understand their role inside the coalition, external constraints permit a team (mainly superior members of subteams) to avoid individual users taking unacceptable decisions.

4. A Scenario Example

This example illustrates some concepts discussed in the previous section. To that end, we are using an indoor disaster scenario where a subteam (4 firemen and 1 coordinator) are looking for victims inside a building on fire and in process of collapse. We must highlight the importance of decisions in systems like that because the system may lead firemen forward to unsafe situations.

The members are assisted by IP^2 agents [20], which currently implement some of the ideas discussed in this paper. IP^2 provides an additional 3D viewer to its users that shows paths and locations of objects inside a building (Fig. 4). The coordinator accounts for creating the operational plan in accordance with the capabilities of their firemen. For example, a fireman equipped with a gas mask can go to floors which are full of smoke.

Each fireman has information on his panel on what he needs to do (issues and activities), what he cannot do (external constraints), directions of how to do things (superior constraints) and restrictions on the performance time. The communication protocol identifies when an agent receives an activity delegation and, if its user does not reply during a time pre-specified, the unique decision that the agent takes is to report a failure by timeout.

A brief conclusion based on this domain is that the software agents do not take any final decision. In fact their functions are to provide options to users (based on several kinds of constraints) and keep the actions on time, reporting possible failures to superior agents.
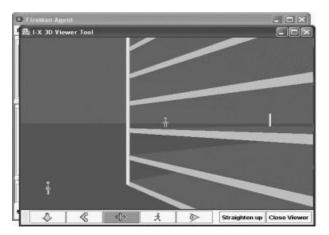

Fig.4: View of a 3D scenario. Objects are modelled via VRML and the IP^2 imports them using the Java 3D API.

5. Conclusion and Directions

Unlike previous teamwork systems designed primarily to take humans out of the planning processes, many efforts are motivated by the need of keeping humans in control of the agents' deliberation and supporting a mutual assistance between agents and humans. The modelling of plans and activities information via a constraint-based ontology enables users to restrict the creation of plans in a natural and simple way, while fills the gap between the different reasoning methods of humans and agents.

Our approach aims to support coalitions in critical domains. Shortly, we intend to integrate our initial prototype with a disaster simulator (Robocup Rescue Simulator), which will provide a dynamic and unpredicted environment for analysis and tests.

Acknowledgement

References

[1] A. Tate, Mixed Initiative Interaction in O-Plan, *Proc. AAAI Spring Symposium on computational Models for Mixed Initiative Interaction*, Stanford, California, USA, 1997.

[2] G. Ferguson, J. Allen and B. Miller, TRAINS-95: Towards a Mixed-Initiative Planning Assistant, *Proc. 3rd Conf. AI Planning Systems*, AAAI Press, Menlo Park, Calif., 1996, 70-77.

[3] E. Horvitz, Uncertainty, action, and inter-action: in pursuit of mixed-initiative computing, *IEEE Intelligent Systems, 14*(5), 1999, 17-20.

[4] P. Cohen and H. Levesque, Teamwork, *Special Issue on Cognitive Science and Artificial Intelligence*, 25, 1991, 487-512.

[5] P. Scerri, D. Pynadath and M. Tambe, Adjustable Autonomy in Real-world Multi-Agent Environments, *Proc. 5th International Conference on Autonomous Agents*, Montreal, Canada, 2001, 300-307.

[6] B. Grosz, L. Hunsberger and S. Kraus, Planning and Acting Together, *AI Magazine, 20*(4), 1999, 23-34.

[7] C. Rich and C. Sidner, COLLAGEN: When Agents Collaborate with People, *Proc. 1st International Conference on Autonomous Agents*, 1997, 284-291.

[8] M. Tambe, Towards Flexible Teamwork, *Journal of Artificial Intelligence Research, 7*, 1997, 83-124.

[9] J. Levesque, P. Cohen and J. Nunes, On Acting Together, *Proc. AAAI National Conference on Artificial Intelligence*, Menlo Park, CA, 1990.

[10] D. Kinny, M. Ljungberg, A. Rao, G. Tidhar and E. Werner, Planned Team Activity, *Proc. 4th European Workshop on Modelling Autonomous Agents in a Mulit-Agent World*, Rome, Italy, 1992.

[11] N. Jennings, Commitments and conventions: the foundation of coordination in multiagent systems, *The Knowledge Engineering Review*, 8, 1994.

[12] N. Jennings, Controlling cooperative problem solving in industrial multiagent systems using joint intentions, *Artificial Intelligence*, 75, 1995.

[13] J. Bradshaw, G. Boy, E. Durfee, M. Gruninger, H. Hexmoor, N. Suri, M. Tambe, M. Uschold and J. Vitek, Software Agents for the Warfighter, *ITAC consortium Report*, Cambridge, MA: AAAI Press/The MIT Press, 2002.

[14] M. Sierhuis, J. Bradshaw, A. Acquisti, R. Hoof, R. Jeffers and A. Uszok, Human-Agent Teamwork and Adjustable Autonomy in Practice, *Proc. 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, NARA, Japan, 2003.

[15] T. Lenox, T. Payne, S. Hahn, M. Lewis and K. Sycara, MokSAF: How should we support teamwork in human-agent teams?, *CMU-R1-TR-99-31*, Carnegie Mellon University, 1999.

[16] T. Lenox, M. Lewis, E. Roth, R. Shern, L. Roberts, T. Rafalski and J. Jacobson, Support of Teamwork in Human-Agent Teams, *IEEE Conference on Systems, Man, and Cybernetics*, 1998, 1341-1346.

[17] C. Siebra and A. Tate, I-Rescue: A Coalition Based System to Support Disaster Relief Operations, *Proc. 3rd International Conference on Artificial Intelligence and Applications*, Benalmadena, Spain, 2003.

[18] A. Tate, <I-N-C-A>: an Ontology for Mixed-Initiative Synthesis Tasks, *Proc. IJCAI Workshop on Mixed-Initiative Intelligent Systems*, Acapulco, Mexico, 2003.

[19] K. Myers and D. Morley, Directing Agent Communities: An Initial Framework, *Proc. IJCAI Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents*, Seattle, WA, 2001.

[20] A. Tate, J. Dalton and J. Stader, IP[2] – Intelligent Process Panels to Support Coalition Operations, *Proc. 2nd International Conference on Knowledge Systems for Coalition Operations*, Toulouse, France, 2002, 184-190.