CS344M: Project Final Report

The Use of Theoretical Models of Team work in RoboCup.

David Terei, School of Computer Science, University of Texas at Austin,

Austin TX, 78712, USA. dterei@cs.utexas.edu

Abstract

RoboCup soccer is an ongoing research domain in which teams of agents, so called multi-agent systems, compete against each other in the name of science and glory for the humans behind them. This paper investigates the use of a theoretical team work model in the RoboCup domain to assist in the execution of team strategies. The formal team work model used for this purpose is the Joint Action (Cohen & Levesque, Confirmation and joint action, 1991a) (Cohen & Levesque, Teamwork, 1991b) and Joint Intention framework (Cohen, Levesque, & Smith, 1998). The model was implemented for a team of agents and used to separate higher level strategies and implement an offside trap strategy for the defensive agents in the team. The results gathered from competition of this team did not show an improvement as desired, instead reducing the effectiveness of the team. The author of this paper does not believe this shows a lack of usefulness of formal team models in the RoboCup domain but instead shows the difficulty in effectively making use of it due to its high overhead in such a dynamic domain and also the further requirement for reducing this overhead effectively through techniques such as agent modeling.

1. Introduction

The RoboCup research domain is an active and interesting area which has greatly helped to advance the state of the art in multi-agent systems. While great progress has been made it is the belief of the author that the field could benefit from the use of formal agent modeling, specifically in the areas of team work and agent tracking.

Teamwork: cooperative effort by the members of a team to achieve a common goal. – American Heritage Dictionary. (Tambe, 1997) [1]

It is likely that many have overlooked this area since the team is already known in the RoboCup soccer domain and given the highly dynamic nature there is little room for the overhead that such formal team work models usually entail. Tambe has shown the effectiveness of such formal team work models through his STEAM (Tambe, 1997) architecture, which was a great inspiration and help in the work done for this paper. Formal teamwork models can be useful to ensure that the team functions as a unit and stays in sync with each other, eliminating a large amount of the errors which can occur when team work is performed based on the assumption that the agents largely have the same knowledge of the environment and so will all behave in the same manner. This approach is analogous to the standard example of the difference between team work and coordinated individual behavior (Cohen & Levesque, Teamwork, 1991b). In usual driving situation it is obvious that the individual cars do not form a team but they do behave in a coordinated manner similar to a team due to shared knowledge and rules (e.g. a red light will cause them to stop). Should one agent be unaware of the red light or break down then this is of no concern to the other agents. In a convoy however, which functions as a team, the agents not only act in a coordinated manner but also are dependent on the success of the other agents and are aware of this fact. Such a formal model of team work as opposed to coordinated action as the majority of the teams in RoboCup soccer currently use has the advantages of being able to handle complex errors, form and disband teams dynamically and provide a mechanism for agents to be dependent on the actions of other agents. These advantages allow for far more complex team strategies to be used and with greater ease and resiliency.

[1] The definition and use of it here is inspired by Tambe's paper "Towards Flexible Teamwork" (Tambe, 1997).

The author of this paper has implemented such a formal team work model, based of joint intention theory and STEAM but also takes inspiration from cognitive science, using ideas from Hayhoe's visual routines theory. (Hayhoe, 2000) This theory postulates that the architecture of the human brain use specialized procedures which are responsible for the complete achievement of the goal being attempted, of the input output and processing. A complete abstract representation of the visual world is not parsed as one step in the brain but instead each visual routine when running has control over all functioning of the brain and only extracts from the visual field the information that is currently needed. A slight comparison can be made between it and Brook's famous Subsumption architecture (Brooks, 1991) in that these routines operate 'horizontally', responsible for the complete function of the human body as opposed to splitting it up into separate section and stages such as vision parsing. However unlike the Subsumption architecture there is no fixed hierarchy between visual routines and they can be selected for use when appropriate with it believed that only close to 6 routines would be running in parallel at any point.

Using these sources and architecture was constructed which makes use of joint intentions theory to model team work and ratify an agents own beliefs and goals as well as those of other agents in the team. The speech act theory from joint intentions was also used to facilitate the exchanging and maintain of each agents belief among all the agents of the team. The architecture also draws on STEAM for details on the implementation of these ideas and the selection process for choosing among various joint actions which are available. Finally the theory of visual routines was used in the design of the relationship of joint actions to one and other and the possibility of running numerous joint actions in parallel. While all of these ideas are present in the current implementation many of them need much more work, particularly the aspects of STEAM which allows for a hierarchical arrangement and dependency among joint intentions, as well as the majority of visual routines theory.

2. Implementation of Preceding Ideas

The preceding ideas were implemented on top of the source code released by the UvA Trilearn team which successfully competed and won the 2003 RoboCup soccer competition (Kok, Vlassis, & Groen, 2003). They subsequently released the source code of the winning team but with all of the higher level strategy removed and a well constructed base architecture remaining.[2] The preceding ideas were implemented using this code base, resulting in a team that was founded during an unimaginative night, inspired by various television commercials. The team is known as "***CheapHomeLoans***".

Brief Overview of the Code

While there are of course changes scattered throughout the entire code base, the majority of the changes occur in these existing files:

- ***Player.cpp/h***
- ***PlayerTeams.cpp/h***

And the bulk of the code changes occurred with the addition of the following new source files:

- ***JoinIntention.cpp/h***
- ***DefaultJointIntention.cpp/h***
- ***DefaultCommunicationJointIntention.cpp/h***
- ***OffsideJointIntention.cpp/h***

[2] The Author of this paper would like to offer a special thanks to the UvA Trilearn 2003 team for the releasing of their code and hopes that future teams of any ranking will also release their teams and perhaps once a few years have passed, release the source code in its entirety.

JointIntention.cpp/h

This file outlines the new fundamental class which implements the version of the Joint Intentions theory outlined in this paper, as well as representing control units for the agent similar to a visual routine as expressed in the visual routines theory. All high level control of the agent takes place as part of a joint intention, even the default strategy, since one of the properties of joint intentions is that they also model an individual agent's goals and beliefs in addition to their use in assisting team work and modeling other agents goals and beliefs.

Player.cpp/h and PlayerTeams.cpp/h

These two classes have been modified to exclusively use Joint Intentions for higher level control of the agents with the *PlayerTeams* class consisting of a strategy of choosing the most appropriate joint intention to execute in the current cycle. It is important to note that each joint intention is constructed upon an agent's initialization and has a life equal to that of the agent. Even when it is no longer being executed it maintains its state. More than one joint intention can be executed in one cycle provided their commands do not conflict with each other, this is useful since team joint intentions require a set-up and shut-down phase during their execution and during these phases other joint intentions can execute. They will also not always need to communicate so other joint intentions can still communicate to establish joint beliefs and share information among team members. Future work could also enable joint intentions to concurrently execute but much work would need to be done to resolve conflicts between their desires

DefaultJointIntention.cpp/h and DefaultCommunicationJointIntention.cpp/h

These two files contain the original ***UvA Trilearn*** default strategy which was released as a place holder for a high level strategy. The strategy has been split into its two components of an action execution strategy and communication strategy. They are run if either the Offside trap joint intention is currently not able to run or it does not need to communicate at present. Here we can see elements of the visual routines theory with the ability to execute more than one Joint Intention concurrently as long as their commands do not conflict. The relation between the joint intentions is flexible but in the current code the relationship between the various joint intentions is a simple priority list.

OffsideJointIntention.cpp/h

OffsideJointInention.cpp/h contains a large amount of the new code implemented for *CheapHomeLoans*. In this class the offside trap strategy is implemented for the use by the defensive agents. It also currently implements a large amount of possibly generic joint intention code related to the formation and deformation of teams; this is largely an implementation of the speech act theory outlined by Cohen, Levesque and Smith (Cohen, Levesque, & Smith, 1998). Future work which could be done would involve moving this generic code to the JointIntention.cpp/h files. Thus the majority of the joint intention theory related code is contained within these files and not in JointIntention.cpp/h.

The other significant change to ***CheapHomeLoans*** is scattered through *WorldModel.cpp/h, WorldModelUpdate.cpp* and *SenserHandler.cpp/h* which involves code to keep track of all the joint intentions and pass on received messages to the appropriate one.

Detailed Overview of the Significant Code and Algorithm's used in Cheap Home Loans

A core component of the joint intentions theory is for each agent to model the belief and desired goal of every other agent in the team. Each agent must keep track of the current state the team is in as well as the desired state for each individual member of the team so that eventually the team state can change once enough team members desire a different state. Cohen, Levesque and Smith outline the concept of weak persistent achievement goal (PWAG) in the theory of joint intentions (Cohen, Levesque, & Smith, 1998). This consist of a goal for each agent that should they come to believe a joint intention to be unachievable, they will be left with a weak achievement goal (WAG) to communicate this belief to the rest of the team. This is a crucial concept in joint intention theory and is implemented in CheapHomeLoans as a persistent goal for an agent to maintain the accurate knowledge of their desire in the other agents of the team. Thus the team collectively has a goal to maintain accurate information in each agent who is a member of the team the desired goal of every other team member. While a goal could potentially be a complex intention in CheapHomeLoans it is the goal to make use of a joint intention or to not use it. Every join intention is its own individual entity and maintains this information about each team members desire to run the joint intention.

```
/* The state of the joint intentin */

enum JIModeT {

  JIMODE_INACTIVE,  /* !< Not active at all */

  JIMODE_STARTUP,  /* !< Can run, trying to establish
common belief    among all team members through
communication */

  JIMODE_ACTIVE,   /* !< Running the joint intention */

  JIMODE_SHUTDOWN,  /* !< Joint intention over (sucess,
failure, not relevant anymore), establishing common belief
among team */

  JIMODE_UNKNOWN

};
```

Code Section 1

The core of the joint intentions theory is captured in a c++ structure used in the code for *CheapHomeLoans*, it can be seen in above in *code section 1*. This structure is used by each agent in a team to model the goal and belief of other agents for the joint intention the structure is part of. Firstly, *pnumber* and *fnumber* are simple identifiers for the agent whose goals and beliefs this structure is encoding. The variable *state* is used to keep track of the desired goal of the agent, the possible state can be seen below in *code section 2*, however JIMODE_*STARTUP* and

JIMODE_*SHUTDOWN* are never stored in this variable as they are not valid desires and not used much internally in the code as there is no formal startup or shutdown mode since the agents are continually maintain accurate information about their desire in other agents. The variable *myState* is used to model what an agent believes a fellow agent believes about its desired state. It is this variable which an agent has a persistent goal to maintain in other agents so as it is accurate. The two floats, *stateConf* and *myStateConf* reflect an agent's confidence in the information it currently has on an agent and these variables are used to determine when communication should take place and if information is accurate enough for decisions to be based upon.

```
/* Model of the team to capture other team members beliefs Construction of
this is handled by subclasses */

struct Team {

  int pnumber; /* Assume player and formation number the same */

  int fnumber; /* future use incase pnumber != fnumber */

  JIModeT state; /* state agent desires to be in, may be actual, may not */

  char MSG[ MAX_SAY_MSG ]; /* Input buffer for each agent to send a msg */

  bool newMsg;  /* Has a new msg arrived? */

  JIModeT myState; /* What state does the agent think i desire */

  float stateConf; /* Confidence of information */

  float myStateConf;

};
```

Code Section 2

The decision to change state, to begin executing the joint intention or to stop executing it is decided upon individually be each joint intention implementation. In the implemented example of the offside trap which involves a team of 4 defenders (although the team and its size is determined dynamically based on agents formation position and can change during the game) to enable the offside trap all team members are required to agree, that is each team members *state* must be equal to *JIMODE_ACTIVE.* The agent must also be sure that all other agents have accurate information about it; the *myState* variable for each agent must also be equal to *JIMODE_ACTIVE*. This information must be accurate to a certain threshold with the accuracy being maintained as mentioned before in the *stateConf* and *myStateConf* variables. For disbandment of the team the requirements are less stringent though as the risk of break away goals out ways the possible benefits of maintaining the trap in the face of disagreement. So for agents to disband from the team and stop the execution of the trap, only half the team, usually two agents, need to desire this. The value of half was chosen after testing various values, it appears to work well as it balances quick disbandment of the team with maintaining it long enough that accurate information for all members is still assured.

3. Performance of CheapHomeLoans

To evaluate the performance of the changes to the base *UvA Trilearn* code, two experiments were conducted[3]. First, *CheapHomeLoans* played 20 matches against the *UvA Trilearn* code with joint intentions enabled; that is the speech act theory enabled. Secondly the previous experiment was repeated with *CheapHomeLoans* playing another 20 matches against *UvA Trilearn* but this time with joint intentions disabled. Disabling joint intentions was achieved by disabling the communication between agents and having them always assume that the rest of the team knew their desire and shared it so that an individual agent would activate or deactivate a joint intention based only on their own beliefs.

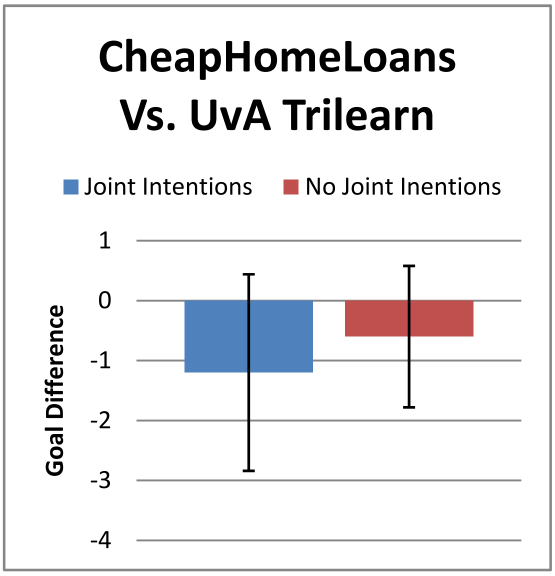

Figure 1. Average score of CheapHomeLoans in a game Vs. UvA Trilearn. Score is the relative score, so a negative score equals a loss of that many goals.

[3] All experimental data can be found in the Appendix.

As we can see from Figure 1, *CheapHomeLoans* with joint intention enabled performed quite badly against the *UvA Trilearn* team, loosing on average by 1.2 ± 1.64 each game. A t-test performed on the average of the difference of the scores for each game with a null hypothesis that the teams performed equal to each other resulted in a p-value of 0.0025. There is little doubt that the changes to *CheapHomeLoans* did not improve the performance of the team but actually degraded it.

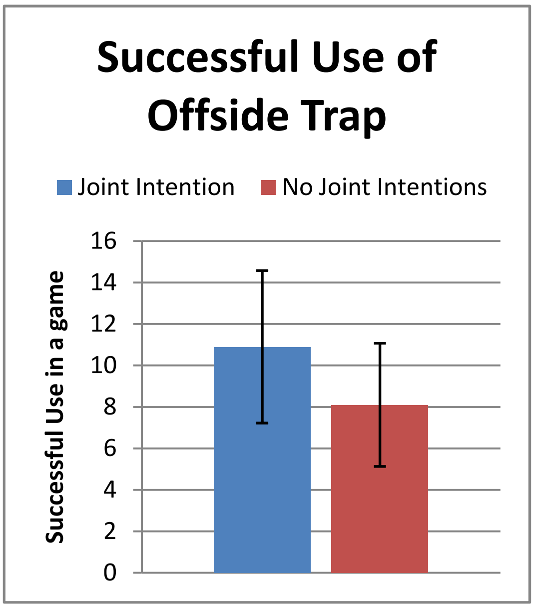

Figure 2. Number of times CheapHomeLoan was able to cause an offside penalty in their favour using the offside trap on average in a game against UvA Trilearn.

When joint intentions were disabled, the team actually performed better. This result is not very surprising and offers some interesting insights into *CheapHomeLoans* which are discussed below. On average, with joint intentions disabled, *CheapHomeLoans* lost to *UvA Trilearn* by 0.6 ± 1.19 goals. The team still performed worse than the *UvA Trilearn* team, a t-test performed in the same manner as above yielded a p-value of 0.025.

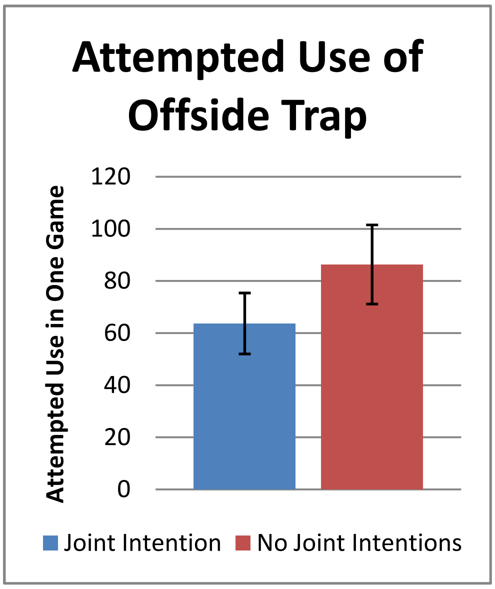

Figure 3. With Joint Intentions enabled, CheapHomeLoans attempted to use the offside trap an average of 63.7 ± 11.7 times a game while without Joint Intentions the team attempted to use the offside trap an average of 86.35 ± 15.19 times a game.

When we compare the performance of *CheapHomeLoans* with joint intentions enabled and joint intentions disabled using a two sample unpaired t-test, we obtain a p value of 0.097. While this is above the usual significance value of 0.05, it does seem to suggest that the degraded performance of *CheapHomeLoans* with joint intentions enabled is significant. We can also see from Figure 2 that with joint intentions enabled *CheapHomeLoans* was able to successfully execute the offside trap on more occasions, 10.9 ±3.68 compared to 8.1 ± 2.97. A two sample unpaired t-test on this result yields a p-value of 0.012, demonstrating it to be statistically significant. Through the use of joint intentions *CheapHomeLoans* is able to successfully execute the offside trap even though it was attempted on fewer occasions. As we can see from Figure 3, *CheapHomeLoans* attempted to use the offside trap significantly less with joint intentions than without. There is an issue with this measurement as it indicates the number of occasions the team engaged the offside trap but not the amount of time which is arguably a better measure. Unfortunately this statistic wasn't gathered during the tests.

4. Discussion

The results of the testing are disappointing in that they clearly show that the changes done to *CheapHomeLoans* failed to improve it compared to its starting point of the *UvA Trilearn* team. However they still demonstrate the usefulness of Joint Intentions and achieve the aim of this paper.

Firstly we must evaluate the performance of the offside trap strategy implemented by looking at the performance of *CheapHomeLoans* with joint intentions disabled. In this case we can see that the performance of the team was still worse than UvA Trilearn showing that the currently implemented offside trap strategy is flawed and needs much future work. The reason for this are multiple, firstly the game of soccer itself makes the offside trap a risky strategy at times as it increases the chance of a breakaway situation ending with a 1 on 1 between a opponent player with the ball and *CheapHomeLoans'* goalie. A successful offside trap strategy is difficult even for real human players and must be adapted to the playing style of the team, sometimes being largely useless with certain teams. The implemented offside trap for *CheapHomeLoans* also isn't of a particularly high quality as its main purpose was to demonstrate the use of joint intentions and because of this it behaves quite aggressively, increasing the risk of breakaways even more. Furthermore a successful offside trap strategy is just one component of a successful defense and it can be argued also relies on a more advance midfield than is present in *CheapHomeLoans*, which was left at the basic strategy from *UvA Trilearn*. An offside trap is often used as a holding strategy, a form of a 'team shepherding' to not only gain an offside penalty but to delay the advance of the forwards to allow the midfield to move into position.

The author of this paper believes that the results demonstrate the usefulness of joint intentions contrary to the score because of the ineffectiveness of the offside trap itself and the larger number of successful executions of the trap with joint intentions enabled. We saw from the results that with joint intentions enabled *CheapHomeLoans* was able to 'successfully' (success being simply defined here by forcing an offside penalty despite the limitations of this approach mentioned above) execute the offside trap strategy on more occasions then without. Also since the offside trap degraded the performance of the team it could be argued that since joint intentions facilitated the use of the offside trap the worse performance is expected and even demonstrates the usefulness of joint intentions. This is an interesting notion but there is little evidence to prove or disprove it and it requires future investigation to determine.

While the benefits of joint intentions have been outlined, particularly in the introduction there are obviously some disadvantages as well, particularly with the approach taken in implementing them for *CheapHomeLoans*. The main problem is the overhead that formal team work models bring in the form of communication. Communication related only to the offside trap for *CheapHomeLoans* during one game averaged sending 486 ± 54 messages and receiving 1022 ± 90. This is a significant problem but it can also be solved as demonstrated by Tambe through the careful use of agent modeling to reduce the need for communication (Tambe, 1997). In Tambe's use of STEAM he was able to reduce the amount of communication necessary to usually less than 5%-10% of what was originally required without agent modeling. Such techniques would need to be implemented in *CheapHomeLoans* for its use of joint intentions to become a viable benefit. The offside strategy itself needs much more work as outlined previously and the efficiency of the current joint intentions framework can also be improved by itself, without the agent modeling. Future work would also entail working on the visual routines theory more and implementing more its ideas in the code and improving the use of it with more strategies implemented on top of the joint intention framework and interacting with each other. The use of just the offside trap or the default strategy supplied by the *UvA Trilearn* team is currently a major flaw in the defensive strategy of *CheapHomeLoans*.

Acknowledgments

This work was conducted under and with the great help of Peter Stone (Associate Professor, The University of Texas at Austin, pstone@cs.utexas.edu) and Doran Chakraborty (The University of Texas at Austin, chakrado@cs.utexas.edu).

Appendix

Below follows the experimental data obtained from running 40 games of *CheapHomeLoans* Vs. *UvA Trilearn*. Of these games, 20 were run without joint intentions and 20 were run with joint intention enabled.

With Joint Intentions

Game	Score		Diff		Offside	Receive Speach	Spend Speach	Team Changed	Team Formed
	CHL	UvA							
1	1	3	-2		19	1180	563	12	82
2	2	3	-1		11	988	463	14	44
3	0	2	-2		7	1073	528	8	70
4	2	5	-3		6	1075	519	18	65
5	2	2	0		8	1130	509	12	68
6	3	2	1		9	1105	488	12	61
7	1	2	-1		10	1024	475	10	48
8	1	3	-2		13	1041	512	12	68
9	2	2	0		15	1052	504	10	71
10	2	3	-1		6	989	499	12	68
11	3	3	0		9	991	512	16	63
12	3	2	1		10	728	329	11	50
13	0	4	-4		12	1051	514	12	78
14	1	1	0		8	975	422	8	57
15	1	2	-1		17	979	443	10	49
16	1	4	-3		7	981	401	14	55
17	3	2	1		15	1070	524	14	83
18	0	1	-1		14	966	488	6	60
18	1	6	-5		10	1009	473	16	54
20	1	2	-1		12	1036	545	10	80
Su	30	54	-24	**Averag**	10.9	1022.15	485.55	11.85	63.7

m				e				
				STDEV	3.684	89.50699767	54.05014728	2.942877217	11.70290112

Without Joint Intentions

Game	Score		Diff	Offside	Receive Speach	Spend Speach	Team Changed	Team Formed
	CHL	UvA						
1	1	1	0	9	0	0	8	109
2	4	5	-1	13	0	0	22	98
3	1	3	-2	8	0	0	12	102
4	2	3	-1	8	0	0	14	85
5	2	3	-1	9	0	0	16	77
6	1	2	-1	12	0	0	10	71
7	0	2	-2	6	0	0	12	69
8	1	2	-1	2	0	0	10	73
9	1	2	-1	14	0	0	6	101
10	3	2	1	6	0	0	14	86
11	2	2	0	7	0	0	8	70
12	2	1	1	10	0	0	12	94
13	1	3	-2	8	0	0	12	103
14	0	1	-1	5	0	0	16	105
15	2	1	1	9	0	0	16	65
16	2	4	-2	6	0	0	12	111

17	2	2	0		11	0	0	10	85
18	0	2	-2		7	0	0	8	74
18	0	0	0		4	0	0	6	71
20	3	1	2		8	0	0	8	78
Sum	30	42	-12	**Average**	8.1	0	0	11.6	86.35
				STDEV	2.972	0	0	3.978891673	15.1944415

Works Cited

Brooks, R. (1991). Intelligence Without Representation. *Artificial Intelligence , 47*, 139-159.

Cohen, P. R., & Levesque, H. J. (1991a). Confirmation and joint action. *Proceedings of the International Joint Conference on Artificial Intelligence* .

Cohen, P. R., & Levesque, H. J. (1991b). Teamwork. *Nous* (35).

Cohen, P. R., Levesque, H. J., & Smith, I. (1998). On Team Formation. *Contemporary Action Theory* .

Hayhoe, M. (2000). Vision Using Routines: A Functional Account of Vision. *Visual Cognition* (7), 43-64.

Kok, J. R., Vlassis, N., & Groen, F. (2003). UvA Trilearn 2003 Team Description. *Proceedings CD RoboCup 2003 Symposium* .

Nair, R., Tambe, M., & Marsella, S. (n.d.). Team Formation for Reformation in Multiagent Domains like RoboCupRescue.

Tambe, M. (1997). Towards Flexible Teamwork. *Journal of Artificial Intelligence Research* , 83 - 124.