A Co-operative Framework for Strategic Planning

Nikolaos Kalyviotis and Huosheng Hu

Department of Computer Science, University of Essex,
Wivenhoe Park, Colchester CO4 3SQ, UK
Email: nkalyv@essex.ac.uk, hhu@essex.ac.uk

Abstract

In a Multi-Agent System (MAS) co-operation is the key to success. However, in order to achieve a satisfied degree of co-operation between the agents, planning has a key role to play. This paper introduces the framework and design issues regarding *strategic planning* and how this can lead to co-operation in a MAS. In order to explain these issues a number of examples from the RoboCup domain are used. The strategic planning framework was used in order to implement the strategic plans needed by "Essex Wizards 2000" simulation team in the RoboCup 2000 competition in Melbourne.

Key words: Multi-Agent Systems, Multi-Agent Planning, Co-operation, RoboCup.

1 Introduction

RoboCup (Robot soccer) is an initiative that started four years ago. The purpose of this initiative is to promote Artificial Intelligence (AI) and robotics research [4], providing a common test-bed (soccer), where various agent theories and algorithms can be tested. The technologies that are involved in RoboCup cover a wide spectrum of research such as design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning and planning, intelligent robotics, sensor-fusion, and so forth [3, 5, 6]. RoboCup is divided in four leagues, *namely Software agent league, Small size robots league, Middle size robots league*, and *Sony legged robot league*.

1.1 The RoboCup Software Agent League

The way that software agent league differs from the other three leagues, is that it is not constrained by any hardware based problems. The agents are purely synthetic and have great flexibility and many design and implementation possibilities. In addition the domain that RoboCup provides is very interesting for MAS research, because it comprises of a unique combination of characteristics [7]:

- It is a real-time domain where success depends on acting in response to a dynamically changing environment.
- Noise existence affects the ability of the agents to accurately perceive and affect the world.
- The domain is collaborative, in the sense that agents share a common goal.
- Finally the domain is also adversarial. This means that also another team of agents exist which has competing goals.

The fundamental issue for researchers who wish to build a team for RoboCup is to design a multi-agent system that behaves in real time, performing reasonable goal-directed behaviours. As the goals and the situations of the game change quickly, and also as the state-space is huge, agents should be able to play the game strategically [6]. Three challenges have to be accomplished: learning challenge, teamwork challenge, and opponent modelling challenge

1.2 Planning

In MAS planning is an important element that can result in efficient co-operative behaviour. "Distributed

Proc. TIMR 01 - Towards Intelligent Mobile Robots, Manchester 2001.
Technical Report Series, Department of Computer Science, Manchester University, ISSN 1361 - 6161.
Report Number UMCS-01-4-1, http://www.cs.man.ac.uk/cstechrep/titles01.html

Planning" is usually the term that is used when referring to planning in MAS. Distributed problem solving is confused with distributed planning mainly because sometimes the problem the agents are solving is to construct a plan. For example, how the agents plan to work together – decompose problems into sub-problems, allocate these sub-problems, exchange sub-problem solutions, and synthesise overall solutions – is itself a problem that agents need to solve [1]. In fact distributed planning can thought as a specialisation of distributed problem solving. The term "distributed" can refer to either the process or the product, or even both of them. As a result of that, the meaning of distributed planning can be threefold and more specifically it can mean any of the following categories of planning:

- Centralised Planning for Distributed Plans: where the plan is constructed by one agent and the execution is done by many agents.
- Distributed Planning for Centralised Plans: where the problem is constructed by many agents and the execution is done by one agent.
- Distributed Planning for Distributed Plans: where both the construction and the execution of the plan are done by many agents.

In general, planning concerns the process by which people select a course of action, deciding what they want, formulating and revising plans, dealing with problems and adversity, making choices, and eventually performing some actions [9]. As planning involves reasoning about actions, it has been approached by an AI point of view. This approach that begun in late sixties early seventies gave birth to the "classical AI planning problem"[10]. Sometimes however, planning systems do not have to create a plan. They just have to retrieve a plan from their memory that is applicable to the problem they are trying to solve. These plans are called "canned plans" by Wilensky [9].

1.3 Overview of Strategic Planning

Strategic Planning involves a mapping of "plans" to "situations". The mapping is one-to-one which means that a particular plan is mapped to a particular situation. For example if we consider a set of situations *S:{s1,s2,s3,s4}* and a set of plans *P:{p1, p2, p3, p4}* then the following mapping can emerge, as shown in figure 1.

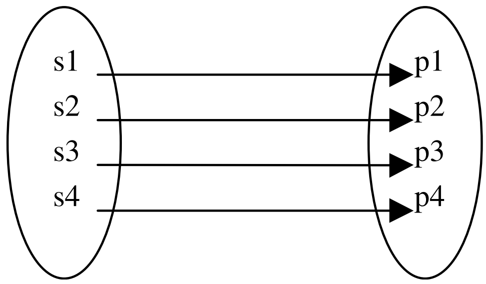

Figure 1 Mapping plans to situations

A plan is mapped (or assigned) to a situation only if that situation is important. For instance, the importance of a situation can be determined by the frequency of its occurrence. An important situation is called "Strategic Situation" and hence the plan mapped to that is called "Strategic Plan".

In RoboCup, like in any other MAS, there are certain situations that occur repeatedly. Therefore a great advantage can be gained if plans are devised for such situation ahead of time. These plans are what we call strategic plans. In the next section of this paper the framework, the definition and design issues of strategic plans is presented. The way that co-operation can emerge by using strategic plans is described in section 3. In section 4 the implementation of strategic plans in Essex Wizards and the results obtained in the RoboCup Competition in Melbourne are discussed. Finally conclusions and future work are presented in section 5.

2 The Framework

Before any attempt of implementing strategic plans is made, a framework based on which the strategic plans will be implemented should be set. The design of the framework that was used for the implementation of the plans required by our software team, is generic. This means that theoretically the same design can be used in other MAS for the implementation of strategic plans. However so far this design has been used only in the RoboCup domain.

2.1 Important concepts

Three concepts are very important in the context of strategic Planning, namely *Triggers, Actions*, and *Abort* Conditions. Therefore before anything else is mentioned about "Strategic Planning" those three concepts will be explained.

2.1.1 Triggers

A Multi-Agent environment can be very complex and dynamic. There is a large number of states that the environment can possibly have. For instance in the RoboCup domain we can have 10^{27} possible states [7]. Every state has a number of characteristics, which make it unique. For example in the RoboCup domain some of the characteristics that a state can have are:

- the X and Y co-ordinates of the players and the ball
- the velocity of the players and the ball
- the body and neck facing direction of the players
- the playmode, and so on

These kinds of characteristics are what we call *primal characteristics*. Let the set P of all primal characteristics of a state E be:

$$P : \{c_1, c_2, c_3, \ldots, c_{n-2}, c_{n-1}, c_n\} \qquad (1)$$

Then we can define a new set of characteristics S that comprises of subsets of P:

$$S : \{P_1, P_2, P_3, \ldots, P_{n-2}, P_{n-1}, P_n\} \qquad (2)$$

This new set of characteristics is called *superior* and every element in this set is called *superior characteristic*. Therefore a state E comprises of both *primal* and *superior* characteristics. Using the RoboCup domain again as an example we can see how we can build superior characteristics of primal ones. Say that two of the primal characteristics of the environment are:

- a team-mate is at the co-ordinates (30, 18) (**c_1**)
- the ball is at the co-ordinates (30.6, 17.7) (**c_2**)

Combining the primal characteristics c_1 and c_2 we can produce a superior characteristic P_1 which is the following:

- Our team has possession of the ball (**P_1**)

Every superior characteristic can be a trigger of a strategic plan. Primitive characteristics can also be triggers but this is not advisable at least in the RoboCup domain. So when we talk about triggers in strategic plans we talk about primitive and superior characteristics of the environment. Trying to define triggers in the context of strategic planning now, we can say that:

"*Triggers are either primitive characteristics of the environment or superior characteristics that emerge form the primitive ones.*"

In figure 2 we can see a graphical representation of the above definition.

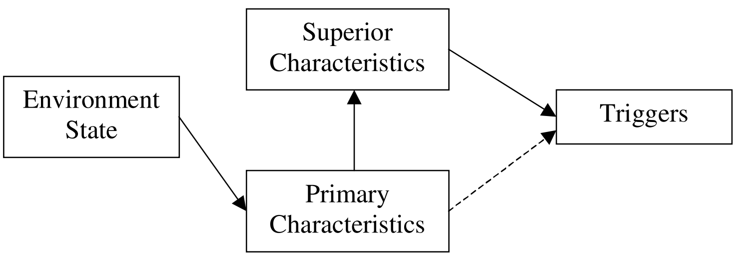

Figure 2 Graphical definition of triggers in strategic planning

2.1.2 The Actions

The agents have to react to environment changes in order to achieve their goal. The capabilities of an agent define the number of actions that it can take. For example in RoboCup an agent can has the capability of performing one of the following actions: *Dash, Turn, Kick, Say, Turn Neck, Catch Ball, Change View*, and *Move*, which are what we call *low-level* actions.

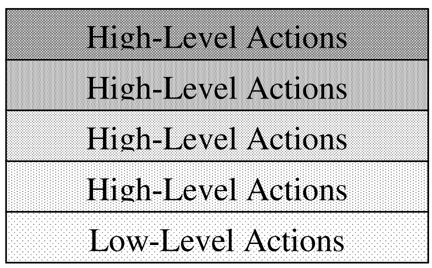

Figure 3 Levels of actions

As the tasks that the agents have to perform are complex, another level of actions is essential. This level is called high-level and the actions that exist in this level are called *high-level* actions. *High-level* actions process the information of the environment and use a combination of *low-level* actions in order to perform a complex behaviour. In addition to that high-level actions can be build on top of other high-level actions. This can result in a stuck of action levels where an action on level X can use any action than exist in any lower level. For example an action on level 3 can use any high-level actions on levels 2 and 1 and any low-level actions as shown in figure 3.

Having defined low-level and high-level actions, we can now define the term action in the context of strategic planning:

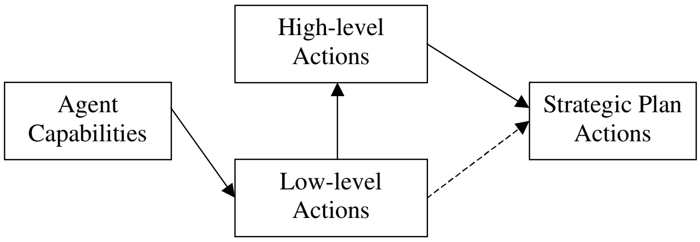

Figure 4 Graphical definition of actions in strategic planning

> "Any kind of behaviour that emerges from either low-level or high-level actions is what we call an *action* in the strategic planning context".

As the definition above states, low-level actions can be used as strategic plan actions. However this is something that is not advisable since strategic plans usually require complex behaviours that can only be achieved on a higher level of actions.

2.1.3 The Abort Conditions

The *abort conditions* are exactly the same as the *triggers*, only the are used for different purposes, as we will see later on. The definition that we have given to the abort conditions is also similar to that of the trigger:

> "*Abort conditions are based on either primitive characteristics of the environment or superior characteristics that emerge form the primitive ones*."

Again here, it is not a good idea to construct abort conditions based on primitive characteristics.

2.2 Defining Strategic Plans

As it was mentioned earlier the main components of a strategic plan are: the triggers, the actions, and the abort conditions. Although those three components were defined in the previous sections nothing has been mentioned about how they are used in strategic plans. Here is a brief description of their use:

- ***Triggers****:* The triggers are used as signals, allowing or forbidding the action that is associated with them to be executed.
- ***Actions****:* The actions are a combination of low-level actions that change partially the environment every time they are executed.
- ***Abort Conditions****:* As the actions that an agent can perform change partially the environment, a safeguard (the abort conditions) must be in place to ensure that the conditions of the environment are still suitable for the strategic plan to keep being executed.

Having defined the main components of a strategic plan we can now give a definition of what we consider as a strategic plan.

> "*Strategic Plan is a list of <u>actions</u> that an agent should execute. Each action comes with a set of <u>triggers</u> and <u>abort conditions</u>. When all the triggers of an action are true then the agent can start executing that action. While doing that if any of the abort conditions becomes true then the agent should abort the execution of the strategic plan. The execution of a strategic plan starts only when the triggers of the first action are true.*"

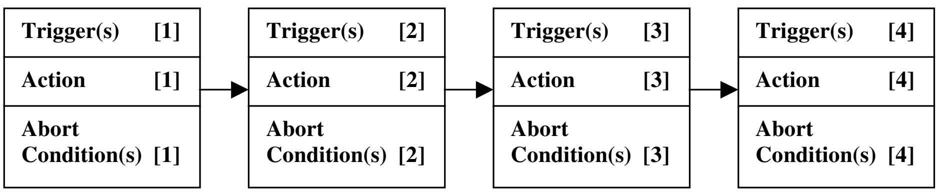

Figure 5 A graphical representation of a strategic plan that has 4 actions

In the example depicted in figure 5 we can see a strategic plan that has 4 actions. This strategic plan can be executes as soon as all the triggers of the first stage become true. If this is the case then the agent can start executing the action of the first stage. As it was mentioned before that action is usually a high-level action which means that a series of low-level actions have to be executed. Therefore it might take many low-level actions for the high-level action to be accomplished. During that time if any of the abort conditions becomes true the execution of the strategic plan should abort. If there is no abortion and the high-level action is accomplished successfully then we can move to the next stage where the new set of triggers should be examined. Then the whole process repeats until the action of the fourth stage is accomplished or until the strategic plan aborts. Generally a strategic plan can abort because of two reasons:

- One of the abort conditions in the current stage is true, or
- We are moving to a stage where not all the triggers are true

In either case we say that the strategic plan has *terminated unsuccessfully*. Otherwise if all the actions of the strategic plan have executed successfully, we say that the strategic plan has *terminated successfully*.

2.3 Designing Strategic Plans

The process of designing strategic plans in any domain has a number of stages that have to be done in a particular order (fig. 6). These stages are presented below in the order that they should happen.

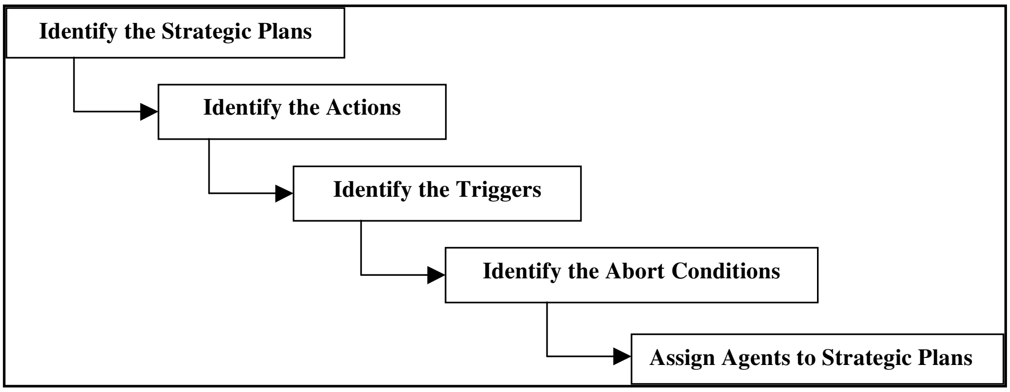

Figure 6 The Strategic Plan design process

- ***Identify the Strategic Plans*** - First of all the strategic plans that are needed should be identified. In order to do so any situations that occur repeatedly should be identified. For every such case we can have a strategic plan in place.

- ***Identify the Actions*** -- After the situations where a strategic plan would be helpful have been identified, then a solution to each one of those situations must be proposed. Then each solution must be translated to some high-level actions that an agent should take in order to achieve its aim.

- ***Identify the Triggers*** --The triggers can be seen as the preconditions of every action that is involved in the strategic plan. In other words the triggers should make sure that the action that is coupled with them can start being executed. Therefore every precondition needed for each action should be identified and translated to triggers.

- ***Identify the Abort Conditions*** **--** While an agent is executing a high-level action, many low-level actions are being executed. As a result of that the environment can partially change. Therefore we must identify a set of post-conditions that must be true every time the environment changes, for each high-level action that exists in the strategic plan.

- ***Assign Agents to Strategic Plans*** **--** A very important issue in the design of strategic plans is how we assign agents to strategic plans. The initial triggers of a strategic plan could be true for a group of agents. However we want only one of them to execute the strategic plan. An easy way of solving this problem is to assign specific agents to specific strategic plans. Although this is valid it is not always efficient. The way that we propose to solve this problem is to assign agents to strategic plans dynamically. For instance let us consider that in the RoboCup domain one team wants to have a strategic plan for taking the kick ins. The initial trigger in this case could be "*our team has won a kick in*" and the action associated with that could be "*go to the ball*". The problem that arises here is that the initial trigger is true for every agent in the same team. Say that we assign a specific agent – a defender – to that strategic plan. This will work well for most of the kick in situations in our half, but not for the ones in the opponent's half. The defender can still make a run and take the kick in but this is inefficient since it will take a lot of time and he will probably get tired by the time he comes back to defence. In such case it make more sense for an agent whose role is forward to execute the strategic plan and take the kick in. If we assume that every agent has a unique role then we can assign agents to strategic plans dynamically based on their role and the actions of the strategic plans. Say that A is the set of actions that are included in strategic plan B, and R is the role of agent G. Then we can build a function F that returns true if agent G can be assigned to strategic plan B, and false otherwise.

$$\boldsymbol{F (A, R) \rightarrow \{ true, false\}} \tag{3}$$

The implementation of such a function assumes that the roles of all the agents at time t_i are known.

3 Building Co-operative behaviours using Strategic Plans

As we can see from figure 7 co-operation is subclass of co-ordination [2]. So when we talk about co-ordination of a group of agents this does not necessarily mean that those agents have common goals. In case that the agents have common goals then co-operation must be achieved. Otherwise, in case that the agents have conflicting goals, then we have competition.

Like in any MAS, co-operation is also extremely important in the RoboCup domain. A group of players (agents) that use co-operation can obviously outperform a group of players that does not use co-operation. After all football is a team game and every team has to use co-operation in order to be effective. However co-operation is an open concept that covers many topics. For example in the RoboCup domain passing and receiving a pass is a form of co-operation between two agents. Team formations and switching of roles are also two other kinds of co-operation that we can find in RoboCup. Another form of co-operation that can be applied in RoboCup and in any MAS is any kind of "*multi-step, multi-agent plans that can be executed in specific situations*".

RoboCup is a complex and dynamic multi-agent domain. Unfortunately, uncertainties in this complex and dynamic domain make co-operation a difficult task. This happens because agents usually have different, incomplete, and possibly inconsistent views of their environment. Furthermore, agents may unexpectedly fail in fulfilling responsibilities related to their roles. According to [8], the solution to those problems that appear in complex domains such as RoboCup, is flexible co-ordination and communication:

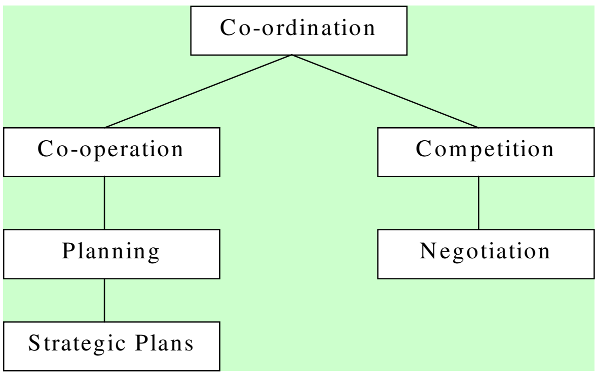

Figure 7 Cooperation versus coordination

"*Highly flexible coordination and communication is key in addressing such uncertainties. Simply fitting individual agents with precomputed coordination plans will not do, so for their inflexibility can severe failures in teamwork, and their domain-specificity hinders reusability*"

According to the above quotation of Milind Tambe, one can argue that strategic plans are inflexible since they involve the execution of the same actions in the same order. However this is not true. Although strategic plans comprises of a pre-set high level action that are executed in order, they are flexible in *commitment* and *execution*:

- ❑ ***Flexibility in commitment***: When an agent in engaged in a strategic plan it does not necessarily mean that all the actions will be executed. As it was mentioned earlier every action is coupled by a set of triggers and abort conditions. Therefore an agent can abort the execution of a strategic plan if the conditions of the environment are not appropriate.
- ❑ ***Flexibility in execution***: The agents in strategic plans execute high level actions. Those actions can be highly flexible and dynamic. For example if the action is to dibble from point A to point B, the execution of the action will vary (or even it can abort resulting in aborting the strategic plan) depending on the number of opponents between those to points (A and B).

So far we have been discussing how we can design strategic plans that can involve only one agent at a time. However the framework that we have described so far is sufficient in achieving strategic planning on team level, resulting in various kinds of co-operative behaviours. Say that there is certain situation where a (team) strategic plan that needs two agents to co-operate with each other is essential. The way that this can be done is to have two strategic plans, one for each participating agent. In figure 8, we can see an example of how two strategic plans can result in a co-operative behaviour.

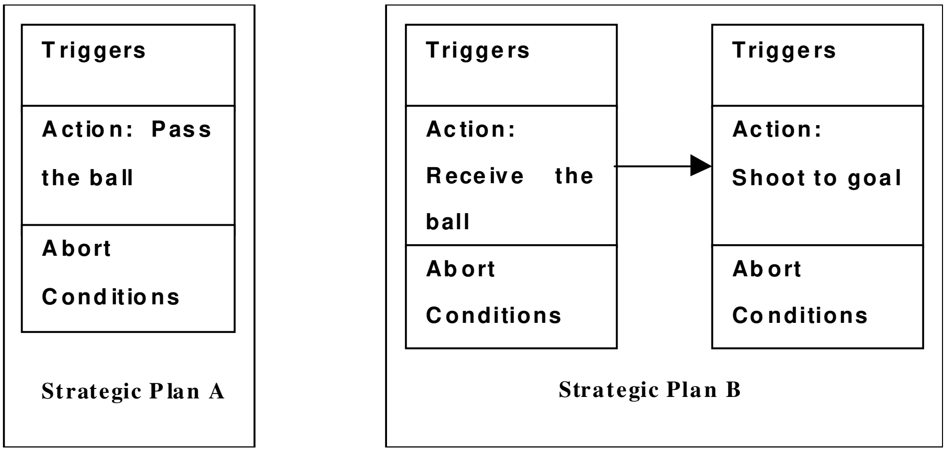

Figure 8 On the left there is strategic plan A where an agent passes the ball to another agent when all the triggers are true. On the right we have strategic plan B where an agent receives the ball and then he shoot to goal. Given the right initial triggers for those two strategic plans we could have a co-operative behaviour where an agent X passes to an agent Y, and then agent Y shoots to goal.

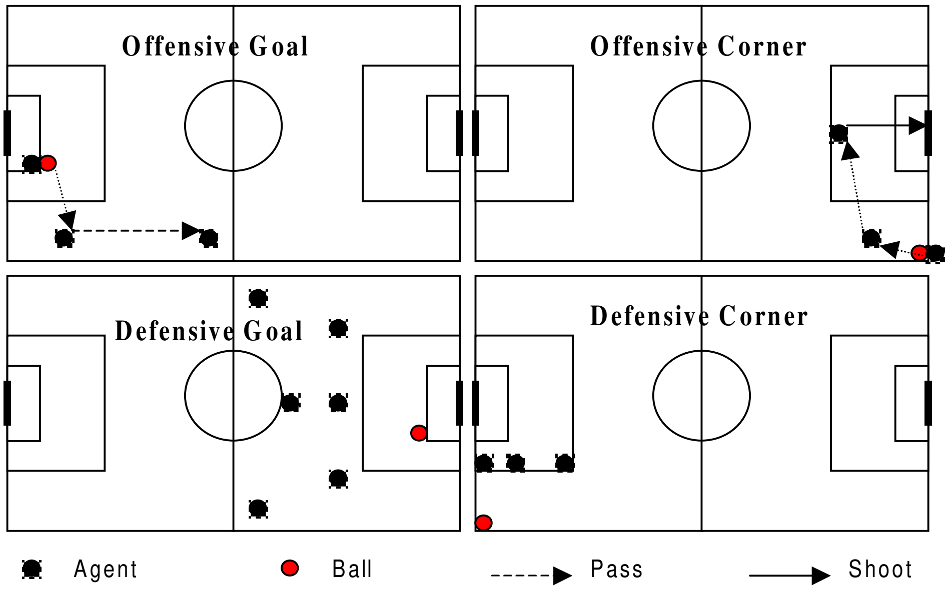

Figure 9 Four team strategic plans implemented in Essex Wizards

4 Implementation and Results

The strategic plans in Essex Wizards were implemented using the framework described in section 2.1 and 2.2, the design issues in sections 2.3, and the co-operation techniques in section 3 as guidelines. Each one of the strategic plans that we implemented is a part of a team strategic plan. Four of the team strategic plans that were implemented are shown in figure 9. The team strategic plans are divided into two categories: i) active and, ii) passive. Passive team strategic plans require a group of agents to co-operate in order to position themselves according to the plan. Active team strategic plans involve co-operation using positioning, passing and shooting. For example *Defence Corner Kick* and *Defence Goal Kick* are passive, while *Offence Corner Kick* and *Offence Goal Kick* are active , as shown in figure 9.

4.1 Results

In figure 10 we can see how co-operation can emerge using *the Strategic Planning* framework that was presented in this paper. This is a goal scoring team Strategic Plan that involves three players (number 11, 10, and 9) and it is called "*Crossing*". For better understanding of the situation, the trails of those three players and the trail of the ball have been added in the above picture. Let us now see what happens in each one of the six stages that synthesise the team Strategic Plan "*Crossing*":

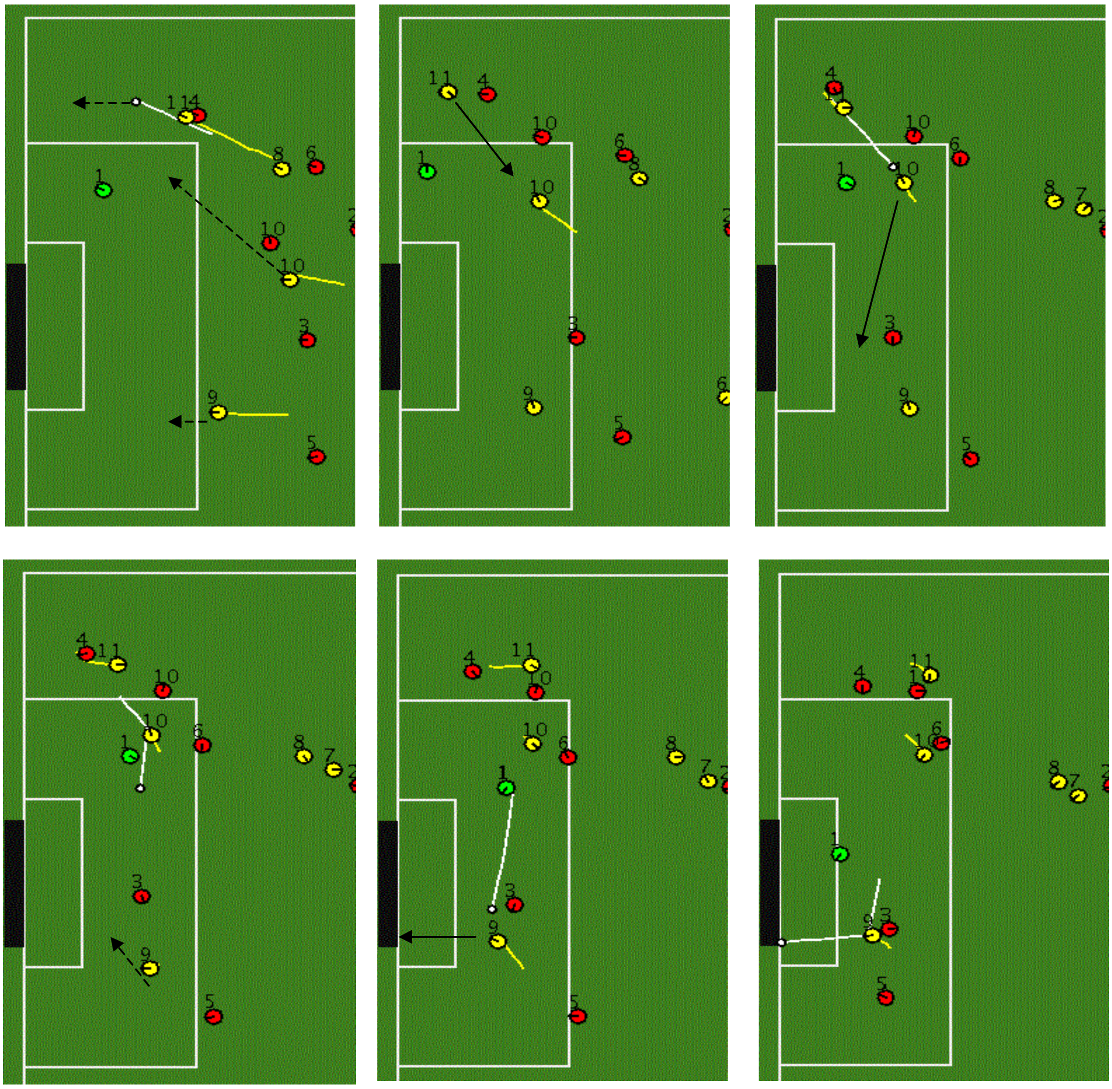

Figure 10 Result of co-operation based on Strategic Planning

- ***Stage 1***: Player 11 engages himself to a Strategic Plan. The first action of this plan is to dribble towards position *A*. This behaviour of player 11 triggers the Strategic Plans of players 10 and 9. Therefore players 10 and 9 start executing the first action of their Strategic Plans which is to move to positions *B* and *C* respectively.
- ***Stage 2***: All players are in their positions: Player 11 is in position *A* with the ball, player 10 in position *B*, and player 9 in position *C*. The fact that all the players are in their positions triggers the next action of the Strategic Plan of player 11 which is to pass to player 10.
- ***Stage 3***: Player 11 executes the last action of his Strategic Plan, which is to pass the ball to player 10. From that point on player 11 is no more engaged to his Strategic Plan. When player 10 receives the ball he makes a pass to open space *D* which is right in front of player 9.
- ***Stage 4***: Player 10 receives the ball makes the pass to open space *D*. That was the end of his Strategic Plan. Meanwhile player 9 starts moving towards position *D*.
- ***Stage 5***: Player 9 is about to receive the ball and execute the last action of his Strategic Plan which to shoot to the goal.
- ***Stage 6***: Player 9 receives the ball and then he shoots to the goal and scores! This was the last action of his Strategic Plan.

5 Conclusions and Future Work

Strategic planning is a relatively simple but very useful method. Although it has been used successfully only in the RoboCup environment so far, there is no reason why they can not be used in the other domains. Given that adequate knowledge of the domain is acquired and careful design and implementation of the strategic plans has been done, then the performance of a MAS that uses those strategic plans can be increased significantly.

Having a strategic plan for a situation that happens frequently is a good idea. Having more than one strategic plan for the same situation is even better. The only problem is that the agent should choose which one to execute. For this reason some kind of machine learning must be used as a decision mechanism. Neural networks can be used in order to accomplish that, since they are effective in noisy environments. For instance, a neural network with three layers can be designed for this purpose, as shown in figure 11 below. The bottom layer can be used as an input layer, containing information about the environment, while the top layer will output the strategic plan that should be executed. We are currently investigating how to implement this framework in order to make our strategic planning be more flexible and adaptive.

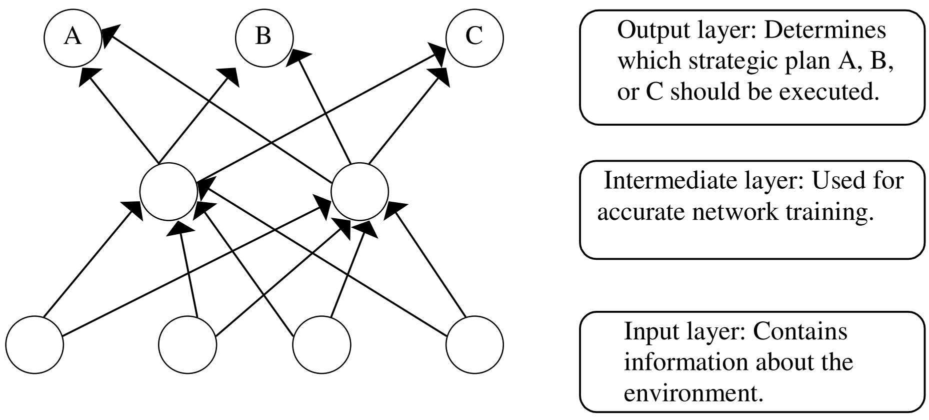

Figure 11 An Example of how neural networks can help in strategic plan decision

Acknowledgements

We would like to thank Kostas Kostiadis and Matthew Hunter who have made contribution to the project. Our thanks also go to the University of Essex for the financial support to the project by providing the Research Promotion Fund DDPB40.

References

[1] E. Durfee: "*Distributed Problem Solving and Planning*", appeared in Multiagent Systems edited by Gehard Weiss, The MIT Press, 1999

[2] M. N. Humns, L. M. Stephens: "*Multiagent Systems and Societies of Agents*", appeared in Multiagent Systems edited by Gehard Weiss, The MIT Press, 1999

[3] M. Hunter, K. Kostiadis and H. Hu: "*A Behaviour-based Approach to Position Selection for Simulated Soccer Agents*", Proc. of the 1st European RoboCup Workshop, 28 May-2 June 2000

[4] H. Kitano, "*RoboCup: The Robot World Cup Initiative* ", Proceedings of The First International Conference on Autonomous Agent (Agents-97)), Marina del Ray, The ACM Press, 1997

[5] H. Kitano, M. Tambe, P. Stone, M. Veloso, S. Coradeschi, E. Osawa, H. Matsubara, I. Noda, and M. Asada :"*The RoboCup Synthetic Agent Challenge, 97* ", International Joint Conference on Artificial Intelligence (IJCAI97), 1997

[6] K. Kostiadis and H. Hu: "*Reinforcement Learning and Co-operation in a Simulated Multi-agent System*", Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 990-995, Kyungiu, Korea, 17-21 October 1999

[7] P. Stone: "*Layered Learning in Multi-Agent Systems* " PhD Thesis, Carnegie Mellon University, December 1998

[8] M. Tambe: "*Towards Flexible Teamwork*", Journal of Artificial Intelligence Research 7, 1997

[9] R. Wilensky: "*Planning and Understanding*", Addison-Wesley Publishing Company, Massachusetts, 1983

[10] D. Wilkins: "*Practical Planning: Extending the Classical AI Planning Paradigm*", Morgan Kaufmann Publishers, California, 1988