Vowels Co-ordination Model

Yves Demazeau, Joao-Luis Tavares

Les cahiers
du laboratoire
Leibniz

Vowels Co-ordination Model

Joao Luis T. da Silva and Yves Demazeau

Vowels Co-ordination Model

Joao Luis T. da Silva and Yves Demazeau
Laboratoire LEIBNIZ-IMAG
46, Avenue Felix Viallet
38031 Grenoble, France
Joao-Luis.Tavares,Yves.Demazeau@imag.fr

Abstract

In Multi-Agent Systems, co-ordination plays an important role and addresses a hard problem when taken at distributed and dynamic way. Up to now, several approaches embark solving mechanisms upon agent, interaction, organisation or environment levels indistinctly. However, in a Multi-Agent Oriented Programming perspective, we need to consider the dynamics at every MAS component. In this article we address this MAOP view through a co-ordination model based on the Vowels Paradigm, which offers a componential view of a MAS and a development platform. So, our model takes this decomposition method through a cognitive approach by using multi-agent plan relations and social aspects by means of social dependence. Differently of classical component focused models, we orient our approach to every component of the MAS by addressing some co-ordination approaches into an *integrated co-ordination model*. This model is illustrated through the RoboCup application in order to evaluate these co-ordination ideas.

1 Introduction

During these last years Multi-Agent Systems (MAS) have become more and more popular, as a framework for building large, complex and robust distributed systems. Roughly, this approach is intended to build flexible systems with complex and sophisticated behaviour by combining modular components. The research focuses on more and more dynamic, vague, non-determinist, real-time environments and more realistic situations. Recently, several works have focused on dynamic and distributed scenarios with a high level of interaction (co-operation, co-ordination, and competition) among agents becoming more and more autonomous.

Distributed Artificial Intelligence aims to design co-ordination mechanisms for groups of autonomous, distributed, rational agents. Co-ordination has been defined as the management of dependencies between activities [12]. This management aims to guarantee coherence of orchestrated actions through anticipa-

tion from other's actions [10], while trying to provide global resolution coherently [11]. Until now, technologies and paradigms to co-ordinate multi-agents are centred on one or some basic components of MAS. At agent level, by using architectures based on the reasoning and the control of the agent. This approach meets technologies of planning, decision and control based on the mental states of the agent (BDI approaches) and its capacity of perception and world modelling (interpretation) [9, 16]. At the organisation level, through social structures, dynamic coalition formation and mechanisms of social dependence [18, 1]. Even works on Teamwork [3, 21] are either centred on the agent or on the interaction, because they encapsulate, at the agent level, a subjective interpretation of the global task, kept by Joint Intention rules that guarantee coherence and persistence of global goals. A brief overview about the co-ordination centred on MAS components is provided in the next section.

In this article we are mainly concerned with the problem of co-ordination through multi-agent oriented programming by using the Vowels Paradigm [7]. In other words, we take the multi-agent co-ordination at the standpoint of the dynamics in Agent, Environment, Interaction and Organisation components, instead of a one purely centred to only one of these components. Our aim is to propose a co-ordination model at the multi-agent level, based on a rather cognitive approach through relations between multi-agent plans and social aspects by means of social dependence notions. We present this co-ordination model based on Vowels paradigm in the Section 3 and illustrate the application of the model through a RoboCup application example in the Section 4.

2   Co-ordination through MAS components

Several approaches have addressed the problem of co-ordination. However, the question is always how to supply the agent architecture with functions and interaction patterns to support construction of the system and to allow it to co-ordinate efficiently and robustly. The same approach is used to build MAS centred on Interaction, Environment or Organisation views.

2.1   Agent Oriented View

Most technologies are centred and encapsulated on only one component, usually the agent. At abstract level, the architecture of the agent reflects the dynamic characteristic of the society through a layer that describes the opportunities of co-ordination, communication, co-operation keeping the agent sufficiently autonomous. Also, a social model adds an organisational degree of knowledge that reflects the dynamics of the MAS in terms of the social structure, the hierarchical organisation of roles, resources and goals.

Describing dynamics under agent standpoint, at the specification level, means to include a module or a co-ordination layer (interaction, negotiation or planning) into the agent, by taking into account multi-level reasoning. That is, the level of the co-ordination control lays on the agent architecture by using

2

negotiation-based approaches or task/resource reallocation. On the other hand, the notion of communication protocol enriches an interaction layer that codes messages, intentions and actions of the agent through state graphs or Petri's networks.

2.2 Interaction Oriented View

Usually, most of the work about multi-agent co-ordination has been taken through interaction approaches, by representing dynamic relationships among the agents according to their actions, goals and skills. The nature of their goals defines whether they will cooperate or compete among them and which type of communication protocol should be instantiated. Under co-ordination point of view, the interaction component provides solving action and resource conflicts with negotiation and information exchange.

Mechanisms of interaction should handle the information exchange and the communication management to increase the system knowledge, such as representation of the others and the meta-information about the communication models. These mechanisms are supposed to support the coherence at the representational level of the interactions among agent-environment, through perception and action, and among agent-organisation by the structure of roles/tasks and the joint commitments. For example, the model of Joint Intentions [2] inspires most of the surrounding areas of co-ordination that use Teamwork [3, 21].

The Interaction-Oriented Programming (IOP) framework [20] specifies the interaction among agents under three layers: co-ordination, commitments and collaboration. Here, we are mainly interested in the co-ordination layer, which represents the activities of communicative interaction described by means of protocols, dialogues, arguments and negotiation. The co-ordination layer takes into account the problem of creating specifications for the "conversational" behaviour and the interaction among agents. By considering that the agents act among themselves in a communicative way, and physically with the environment, Singh [19, 20] suggests modelling the conversational aspects through instances of the communicative activity, that is by using speech acts and physical acts, reinforcing the interaction agent-agent and agent-environment, respectively. In this way, the co-ordination under IOP aims to detect conversational relations by means of a bipartite structure of messages to meet co-ordination requirements.

2.3 Environment Oriented View

From an environment standpoint, co-ordination is primarily a social phenomenon characterised in terms of the relations between an entire system and its environment. The Artificial Life and Environment-Oriented Programming (EOP) approaches [8, 14, 15] have stated that close relations between the agent activities being performed in the same environment create dependencies between activities and uncertainty from the perception process. As the agent has to act based on those perceptions and the characteristics of the environment and its

3

dependencies, a co-ordination behaviour is primarily determined by its environment. According to [15], the environment plays an active role in reactive models of co-ordination. supplying information fusion about different agents passing over same locations, distributing information from nearby locations and ignoring too old information not refreshed further.

2.4 Organisation Oriented View

By considering co-ordination through organisation, we find several approaches that take into account social structures, task and role reallocation or dynamic coalition formation. In several cases, the description of the dynamics at organisational level uses structures of knowledge still in the agent's model. As an example, Sichman [18] represents organisational (social) information at the agent level through external descriptions and dependencies between these descriptions through Dependence Networks (DEPNET). Decker [6] removes task/action descriptions from the agent's architecture, simply by describing a structure of tasks with the possible relationships in a higher level.

2.5 Discussion

In the perspective of Agent Oriented Programming, the description of several components of a system becomes dependent of the domain and of certain basic constituents like the agent model, mainly. Although these definitions are well established in the MAS community, they concern only one component and embark co-ordination mechanisms at the level of this component. We are looking for precisely a *decentralisation* of the co-ordination within the basic components (Agents, Environments, Interactions and Organisations) through the Vowels paradigm [7], to find more general design patterns of MAS. The problem is to find a framework that integrates all these technologies so as to fit dynamics in every component. Hence, MAS design could be really componential, in the sense that even the dynamics of the system is foreseen at abstract level, towards Multi-Agent Oriented Programming (MAOP).

3 Vowels Co-ordination Model

The co-ordination problem under the Vowels perspective presents a level of representation focusing basically on co-ordination requirements about classes of dependencies between activities. These classes of dependencies concern relation between plans, resources and actions; conflicting or incompatible goal; task time completion; task/sub-task relationships and so on.

We present a co-ordination model oriented to MAS based on the Vowels approach that focuses on plan relations and social dependence. As we orient our approach to every component of the MAS, we address some co-ordination approaches into an integrated co-ordination model. At the organisational level, we take the Sichman's social reasoning mechanism [18] and we extend it to

use a planning approach also at the Interaction level. The plan relations are borrowed from Martial's work [13] about co-ordination actions among planning agents. In this case, we utilise potential plan relations at agent and organisation levels through the extension of the DEPendence NETwork (DEPNET) from [18]. At Environment level, we associate a description of the activities, resources and world representation to Decker's TÆMS[1] [6]. This level describes possible relations at task level in a TÆMS-like structure, always based on the DEPNET, but this time at global level, that is, at task objective level. In the Follow up, we describe the model under each MAS component (called "brick" at Vowels Paradigm) and the possible co-ordination dependencies that are involved.

3.1 (A)gent-(O)rganisation Co-ordination Level

The main dependence at this level of description is that of the Agent component with regard to another one. Thereupon, co-ordination problems are directly related to dependencies between activities, taken by means of agents' actions, goals and plans.

At this level, we describe a DEPNET-based approach that can be applied either to the organisation level or to the agent level to capture aspects and dependencies for co-ordination. At the agent level, we propose to include external descriptions for each agent as in Sichman's work [18], including some plan details. A DEPNET represents a relation of complementarity between actions from several agents. *A priori*, it represents dependencies between two or more agents for an action and a resource in order to determine a better coalition towards a group of complementary co-operating agents. In Sichman's work, a DEPNET represents social dependencies among agents for actions, resources and goals. This structure is built from a description of actions, resources, goals and plans that each agent possesses about the others throughout an External Description (*ED*), exchanged by all the agents in the society.

Figure 1 illustrates the specification for the DEPNET at agent level. The diagram illustrates goal, action and resource representation to the Sichman's-based External Description, plus our planning level representation. An agent has goals to achieve. A goal is reached by associated plans. Plans are composed by an ordered set of completed instantiated actions. Each action description contains its identification, the necessary set of pre- and post-conditions, required resources and evaluating information such as estimated time and quality, in order to check by time conflicts and handling facilitate or enable action relationships. A co-ordination dependence is related either to a relation between plans, resources and/or actions, or to a conflicting or incompatible goal.

At organisation level, we extend this approach to comply also for roles instead of agents only. In this way, the main idea is to describe each role by means of goals and plans, outlining a kind of pre-established behaviour to the role and supplying it with a description of the necessary tasks to achieve each goal through plans. By describing some predictable information at this level

---

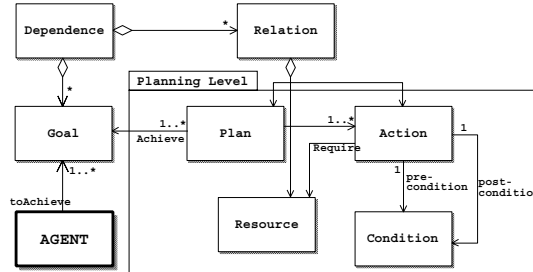[1] Task Analysis, Environment Modelling and Simulation

Figure 1: DEPNET-like specification at agent level.

we can aggregate some relationships between tasks as a means of describing possible co-ordination dependencies. This approach combine some aspects of the non-local effects[2] of Decker's TÆMS, but in our case, we constrain the range of relationships describing some task time completion or task/sub-task relationships in the global structure.

In this approach, the diagram in Figure 2 illustrates goal, action, resource and plan representations to External Descriptions, but related to roles, rather than agents. We take a role as a kind of abstract agent which will calculate its DEPNET with regard to some specifications done by the system designer, to facilitate the description. In this way, a role (abstract agent) has goals to achieve and the description of role structure it the same as for agents. Additionally, at this level we include some possible relations among tasks captured by dependence descriptions such as enabling and facilitating tasks, task/sub-task relationships and resource management.
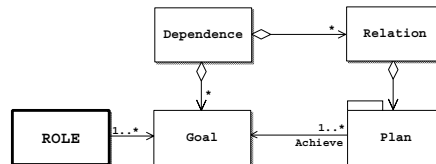
Figure 2: DEPNET-like specification at organisational level.

Each agent has its own local DEPNET that relates its plans with those of the other agents. Hence, those local DEPNETs subsume the organisational level DEPNET described by the roles. The general idea is that before any communication of information gathering by the agents, an intersection between their local and global DEPNETs might detect some relationships or, at least, update old information at agent level. This operation aims to find out an empty

---

[2] As ENABLES, FACILITATES and so on, which describe relationships between two tasks [5].

intersection among both structures, otherwise some services of co-ordination are triggered to solve dependencies.

3.2   (I)nteraction Co-ordination Level

Together with a DEPNET approach at the organisational level, we describe a set of plan relations based on Martial's taxonomy [13]. Martial's work presents a co-ordination framework based on multi-agent plan relations covering potentially harmful (negative) and favourable (positive) relationships among individual plans in a group of autonomous agents. Relations account for resource sharing, temporal conflicts, redundant actions, favourable actions and subsumption. The main idea is that a co-ordinating agent takes two conflicting plans and proposes co-ordination actions, resulting in co-ordinated individual plans. This approach is agent-centred, in the sense that the co-ordinator proposes a reordering of these actions based on spatial and temporal resources. The notion of negative and positive relations is formally presented in order to form the basic concepts of the multi-agent co-ordination framework. By extension, a plan relation is detected if exists such relation among the refinements of the actions in those plans. A positive relation between two plans is defined according to the situations of equality, subsumption and favour. In our model, this case is applied at Interaction level to meet distributed planning requirements by using plan relations such as RESOURCE/TIME CONFLICTS, FAVOUR, REDUNDANCY, EQUALITY relations.

The class "Relation" illustrated in the Figures 1 and 2, describes this level. Hence, if some plan, action or resource relationship is detected, either by means of intersection operation with DEPNETs or throughout explicit marks of relations described by the designer, respective relations will be integrated in the plan description. With these dependencies in hand, we apply some co-ordination actions, such as REDUCE INTERVAL/RESOURCE, REMOVE, ADD and REFINE ACTION.

Otherwise, when relations cannot be solved or evaluated by planning dependence requirements, the model is flexible enough to integrate negotiation and classical contracting or communication approaches. This means, at MAS design and development level, that we addressed the needs a co-ordination problem requires to be solved through the dependencies among the agents. To a modular standpoint, we supply some co-ordination services that can be made by the developer and integrated to the system. For instance, the services can be related to the co-ordination actions, as aforesaid, or by using communication facilities. In this last case, the problem is reduced to the classical approaches for co-ordination at Interaction level and the communication costs must be taken into account.

3.3   (E)nvironment Co-ordination Level

In this level, dependencies between activities are realised by means of a close relation between the agent activities being performed in the same environment.

7

Indeed, co-ordination can be realised as a social phenomenon in terms of the relations between the entire system and its environment [14, 15]. Here, we differentiate "conversational" interaction among agents through performatives from the interaction of the kind A-E-A (Agent-Environment and conversely) that is made in a directly way on the world, by modifying the status of passive objects. Additionally, this cycle is a kind of instance of planning-execution loop, meaning the agent needs a sequence of actions already chose with regard to the previous state of the world to act, certainly decided in a cycle of control that stacks actions in a structure of intentions. Also, resources are related to the Environment component that might be capable to control the information about every resource in the system. This can be achieved through a management of the controlled resource or a state flow representing the availability of resources.

We use some co-ordination mechanisms from Decker's GPGP approach [4] to solve certain dependencies. In order to deal with interdependencies between tasks in a problem-solving situation, Decker presents a framework to model complex task environments through TÆMS. Co-ordination in GPGP is achieved through commitments to certain tasks executed by some time and with some quality. These commitments are represented in the agent's TÆMS task structure and are reached by using a set of co-ordination mechanisms: updating non-local viewpoints; communicating results; handling simple redundancy and handling hard and soft co-ordination relationships. Thus, GPGP generates and adapts plans according to the dependencies represented by TÆMS in order to achieve a certain expected quality or duration measure. So, methods to execute on the planning level (Figure 1) take into account some relations like Enables, Facilitates, Cancel and so on, that are represented into the task structure (local or global DEPNET).

3.4 Discussion

This model covers a great part of the co-ordination requirements that we have considered along this description, such as relation between plans, resources and actions, conflicting or incompatible goal, etc. However, is difficult to realise each co-ordination component separately. Thus, Sichman's work is not enough to address co-ordination problems and we have extended it for planning situation in order to take benefit of plan relation ideas. The advantage is that Martial's approach, integrated to external description, enables some relationship description at organisational and agent levels. In (I)nteraction and (E)nvironment bricks, we associate plan relations to interaction from agents to the each other at communication level, and to the environment through planning approach.

Nevertheless, we not are intended to propose a complete co-ordination model, but one that covers with some constraints what we consider important. Thus, we have analysed co-ordination under some aspects such as MAS oriented and through planning as solving method. By integrating these aspects we meet our constraints of a co-ordination model oriented MAS.

We illustrate the application of the model through a highly dynamic domain, with a strongly uncertain and distributed environment, the RoboCup.

8

We are particularly interested in multi-agent co-ordination by distributed planning, thus we describe a specific situation to check the model under two co-ordination requirements: complementarity of actions/resources and relation between plans/actions. The former is illustrated by an opportunistic pass situation, while the latter involves choosing an action for a common goal.

4   The RoboCup Application

RoboCup is an initiative aiming to supply Distributed Artificial Intelligence and Robotics research with a testbed based on soccer game robots to be used in Distributed Problems Solving methodologies and Multi-Agent Systems [9, 11]. A SoccerServer is used in the graphic simulation of the multiple mobile robots in a bi-dimensional space. Simulation league is constituted of two teams with 11 players each and the management of movements, actions and communication is reached through the server. The server supplies yet another virtual field of games where elements take place such as players, the ball and flags, all of them calculated and positioned with regard to actions passed on for the clients. The connection of the players with the server is made through the protocol UDP/IP, which confers a certain degree of realism and allows simulating incomplete and incorrect information (as noise in the sensors, for example).

4.1   Co-ordination in the RoboCup

We illustrate some ideas of co-ordination and relations between plans by some examples. The motion can be executed in parallel with other actions, for example, positioning on the field while observing the teammate who holds the ball. In this case, the player maintains a certain global confidence about the ball holder, when every player executes the same joint action. If the player which possesses the ball decides to pass it to someone else, it can communicate this decision and the remainder of the team gets ready for a change of the plan. In that case, the ball confidence in every player would be updated and new actions will be selected to carry out (or to solve) these belief change about the ball holder. During the game, many conflict situations can appear and the module of co-ordination would try to solve them. For example, if several players go all together on the ball, these redundant actions are detected by the co-ordination model and solved through communication or perception of the respective position of each other player with regard to the ball.

4.2   Development through the Vowels Approach

We have developed the application through the Vowels Paradigm by using the Volcano platform [17], which allows the development of MAS. By means of this platform and the Vowels approach, we have defined the models and architectures for each brick at design level. The simulation league utilises a soccer server that

provides operators, results and visual display for the clients (players). Thereupon, we have centred this application on the Environment (E) brick as point of start of the application. This brick captures the most of the specificity of the Robocup domain in order to offer more generic models for the other bricks. The communication in Robocup Server is imposed through the server by using the soccer field as interaction medium, so we have decided to not use a conventional (I)nteraction brick and then all communication is realised through the server (E brick). Thus, every message that the agents exchange pass from the Agent (A) brick to the (E) brick directly. The perception is a module that implements a simple parser of the sensor information and updates the model of the world. The module of perception, such as the model of the world, is realised in the (E) brick, so that, part of perception, action and simulation are in fact carried out by the SoccerServer. The model of the world is represented by a memory that keeps sensory and visual information.

For the (A) brick we take an approach of cognitive planning based on a library of pre-defined plans. In this way, Planning is accomplished in every decision that the agent has to take with regard to the change of the world. In the case of Robocup, we consider strategies as agent's individual plans. In this context, plans are a sequence of actions (basic operations) which have to follow a schedule (the interval of the cycle of perception, for example) and execute the sequence of actions to reach a goal, as shown in the Figure 3. For example, if the goal is to follow a dribble, the plan is represented for a sequence of actions by alternating basic operations (`turn`, `dash` and `kick`). But the parameters of the functions are set taking into account the environment and the current situation of the game. Several plans can be pre-established from the knowledge of the domain and we represent these plans through a library of strategies.

A module of action aims to execute basic operations supplied for the module of planning, taking into account the role of every agent, plays through three different levels:

1. *TacticalBehaviour* is a sort of cognitive plan which contains pre-established strategies but which can have relations with the plans of the other players;

2. *IndividualSkills* correspond to actions of higher level and is represented as a set of the basic operators;

3. *BasicOperators* represents available actions in the server to simulate the behaviour of the players on the field.

The (O)rganisation brick is implemented through a generic group/role structure, which handle organisational information about the teams and the players. Every player is autonomous and acts in a surface of game appropriate for every type of function. These functions are represented by roles in an organisational structure where each agent possesses different skills with regards to its role and the situation of the game.
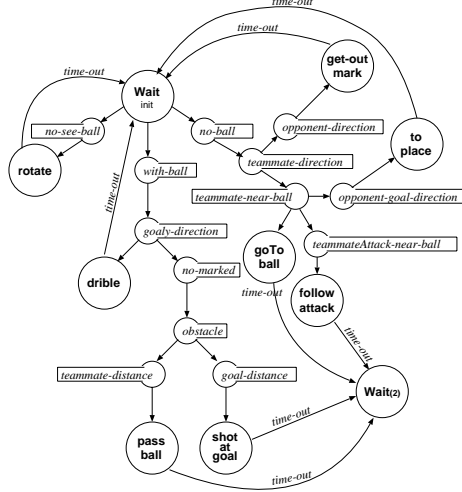
Figure 3: A possible plan of attack that depends on the situation (player with or without ball).

4.3 Co-ordination application through complementarity of actions and resources

In this section we describe the first situation to check for complementarity of actions/resources requirement. Initially, agents build their individual external descriptions, which contains information for social reasoning within agents. In this way, the model allows to deduce dependencies among the agents based on the complementarity of actions, goals and plans. Mental states in the agents are changed according to current situations during the game. Thus, we have to consider not only the complementarity of actions or resources, but also situations during plan execution.

Roles have an external description (ED) such as:

```
Role( Fwd := Goals :{fwdDirAttack(),makeGoal()},
Plans :{fwdDirAttack(),makeGoal(),goHomePosition()},
Actions :{chaseBall(), dribleToGoal(),
          kick(power, goalDirection),
          lookAtHomePos(),turn(homeDir),dash(power)},
Resources :{ball, playerDir, goalDir} ).
```

According to (A-O) co-ordination level (Section 3.1), each agent has instantiated its own individual ED from this role ED in order to trigger its behaviour. At some moment, the player calculates its DEPNET with regard to co-ordination requirements by checking the current ED with the higher level

role *ED*, which represents the game strategy. At this level, a co-ordination dependence is represented by a relation description of some pre- or post-conditions that affect another role's action. For example, a `chaseBall` action depends on the ball resource (through some confidence value). Also, a `followAttack()` plan in the midfield role, depends on the `fwdDirAttack()` plan in the forward role, to an efficient `passBall` action.

By applying the co-ordination model through complementarity of actions and resources, we consider the (O) level and we built a DEPNET concerning player roles, related to external descriptions of agents. In this situation, there is an explicit representation of the role behaviours, such as *Forwards, Defenders*. So, each role has a set of goals, plans and actions in a structure such as it was illustrated in Figure 2 (Section 3.1). The plan is described to each role through the (O) brick and they are assigned to each agent that plays that role. From this moment, each player changes its external description according to environment evolution. Thus, when a collaborative action is required or a co-ordination requirement is detected, the player can check its current plan with the role pattern and decide when to communicate or to act.

We represent only a moment of the game, an opportunistic pass situation: a situation of pass close to the goal of the opponent team as illustrated in the Figure 4. In this situation, the goal of the agent Ag_6 is to shoot towards the opponent goal, but opponent players block it. There is another player Ag_9 in a favourable position, but other opponent blocks him for a direct pass. However, the player Ag_8 knows, according to its external description, that the player Ag_8 is free and it can pass the ball towards the goal or towards the player Ag_9.
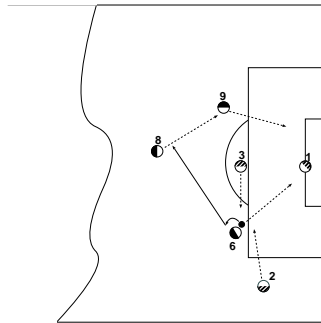
Figure 4: Example of a pass situation.

Regarding dependence of some resources, the player Ag_6 calculates his DEPNET concerning its teammates, Ag_8 and Ag_9. The player Ag_6 wants to move forward but it is surrounded by opponent players; according to the external description it notices whether the agent Ag_8 and Ag_9 is quite close and free. Both players Ag_6 and Ag_9 note that there is an opponent among them, then the simple pass is not possible. The best option with regard to this scenario is to

make the pass for the agent Ag_8, because they know that Ag_8 is free and it can hand on the ball to the agent Ag_9 and from him to the goal. Favourable position events, information about the ball and agent individual skills (`ReceiveBall, DribleToGoal, MakeGoal`) are taken into account for the selection of this strategy. Ag_6's plan is to make the pass for Ag_8 and to run towards the goal. The external description of the agent Ag_6 is shown in Table 1.

Table 1: External description of the agent Ag_6

Agent	Goals	Actions	Resources
Ag_6	PassBall ReceiveBall Drible MakeGoal	PassBall LookatBall LookAtPlayer MakeGoal DribleToGoal	Ball ¬GoalDir ¬PlayerDir

Agent	Plans		
Ag_6	PassBall()	:=	LookAtPlayer(), Kick(power,player).
	MakeGoal()	:=	Kick(power,goal).
	Drible()	:=	DribleToGoal().
	ReceiveBall()	:=	LookAtPlayer(), LookAtBall(), Turn(ballDir), Dash(power).

When there is an opponent in front of the goal, the capacity to mark a goal does not exist any more for the agent Ag_6. Then, dependence towards Ag_9 causes a change in Ag_6's plans.

Ag_8's plan is `ReceiveBall`, to make the pass for Ag_9 and to run towards the goal slightly behind of Ag_6. The external description of the agent Ag_8 is represented in the Table 2[3].

Table 2: External description of the agent Ag_8

Agent	Goals	Actions	Resources
Ag_8	PassBall ReceiveBall Drible MakeGoal	PassBall LookatBall LookAtPlayer MakeGoal DribleToGoal	¬Ball ¬GoalDir PlayerDir

Ag_9's plan is `ReceiveBall`, to make the pass for Ag_6 (`PassBall`(Ag_6)) or to mark a goal (`MakeGoal`). The external description of the agent Ag_9 is represented in the Table 3. Finally, Ag_6's new plan is `GotoGoal(Dash())`, `ReceiveBall` and `MakeGoal`. The new external description of the agent Ag_6 is represented in the Table 4.

---

[3] At this moment, all agents have the same set of plans and, by lack of space, we have omitted redundant plans in next tables.

Table 3: External description of the agent Ag_9

Agent	Goals	Actions	Resources
Ag_9	PassBall ReceiveBall Drible MakeGoal	PassBall LookatBall LookAtPlayer MakeGoal DribleToGoal	¬Ball GoalDir

Table 4: External description of the agent Ag_6

Agent	Goals	Actions	Resources
Ag_6	ReceiveBall Drible MakeGoal	PassBall LookatBall LookAtPlayer MakeGoal DribleToGoal	¬Ball ¬PlayerDir

Agent	Plans		
Ag_6	MakeGoal()	:=	Kick(power,goal).
	ReceiveBall()	:=	LookAtPlayer(),
			LookAtBall(),
			Turn(ballDir),
			Dash(power).

4.4   Co-ordination by relations between plans

The initial state for plan co-ordination requires, for every action of the plan, an identification of the agent which possesses this action in its plan, time interval, duration of the action and quantity of resources required. Each agent keeps a state of co-ordination that groups all information about plans and their relations. So, a state of co-ordination is formed by the local plans of the agent and the relations with other agents' plans (for instance Equal, Subsume, Favour). This information is within the external description in each agent and instantiated from the DEPNET of roles, as defined at section 3. This situation shows co-ordination in the (E) brick, by using plan relations to decide the best action according the world state (section 3.2).

In our example, we shall rather consider the roles of the agents in context, to simplify specification. Then, during a game, the agents are instantiated through a set of pre-established roles (*Forward, Defence, Midfield, Goalie*) in a set of the players, for instance for a 4-3-3 strategy:

Defence = $\{Def_1, \ Def_2, \ Def_3, \ Def_4\}$;
Midfield= $\{Mid_1, \ Mid_2, \ Mid_3\}$;
Forward = $\{Fwd_1, \ Fwd_2, \ Fwd_3\}$;
Goalie = $\{Goa_1\}$.

The local plans of every agent can be built with regard to the set of actions and sequence of pre-established plans. So, an example of local plans can be

distributed in this way:

```
Fwd = {FwdDirAttack(),MakeGoal(),GoHomePosition()}
Def = {PosDefense(),MarkOpponent(),HoldHomePos()}
```

The refinement of plans can be represented by the sequence of actions of a plan hierarchy, for example:

```
FwdDirAttack()={ChaseBall(),DribleToGoal()}
ChaseBall()={LookAtBall(),Turn(ballDir),Dash(power)}
DribleToGoal()={LookAtGoal(),Drible()}
MakeGoal()={Kick(power,goalDirection)}
GoHomePosition()={LookAtHomePos(),Turn(homeDir),
                  Dash(power)}
Drible()={Turn(),Kick(power,dir),Dash(power)}
```

Our example uses a group of forward players, where an instance of a group of attack is represented as:

```
Fwd₁={FwdDirAttack(),MakeGoal(),GoHomePosition()}
Fwd₂={FwdIndirAttack(),MakeGoal(),GoHomePosition()}
```

In a certain moment of the game, if Fwd_1 and Fwd_2 activate actions **FwdDirAttack()** and **FwdIndirAttack()** respectively, the relation Equality is detected, because the action **ChaseBall()** is redundant in the plan refinement and it can be realised by one of the agents. In this case, one of criteria of choice will be the closest player to the ball and, so, Fwd_1 or Fwd_2 remove action **ChaseBall()** while actions of the players are executed by them, simultaneously.

This condition is captured by the model through co-ordination requirements at the environment level. We describe actions, plans and refinements to include some relations at (O) role level. This structure allows to check intersection between external descriptions in the agent and in the role descriptions. This intersection detects the possible relationship and determine a co-ordination service (in this case, remove redundant actions) as a solving alternative to the game conflict.

4.5 Discussion

The model has been illustrated through the RoboCup application. Our example is a twofold one. We have illustrated a decomposition of the RoboCup application through the Vowels Paradigm from analyse level until development and execution phases, using Volcano platform as implementation tool. Also, we have described our co-ordination requirements for this example and shown two situations that were handled by co-ordination through (A-O) and (A-E) bricks. In our work, we have addressed a strongly cognitive model, because we utilise a distributed planning approach to cover with the co-ordination problem. By using an extremely dynamic and uncertain domain, we have addressed the dynamics of MAS at design and development levels. These aspects are captured through dependencies between bricks as regarding to co-ordination requirements. These

dependencies and requirements are handled by relation and dependence classes described in the section 3.1. In the RoboCup application, this class describes some relations between roles and the necessary mechanisms (or services) intended to handle with them, such as strategic formation and related goals like `chaseBall` and `lookAtBall`.

5   Conclusions and Further Work

In this work, we have provided a principled co-ordination model oriented to MAS based on the Vowels approach, by focusing on plan relations and social dependence. Hence, our approach addresses a *decentralisation* of the co-ordination among Agents, Environments, Interactions and Organisations through the VOWELS framework which integrates some of these technologies at design and development level.

In our model, Sichman's DEPNET addresses a complementary view of the tasks at organisational level, while explicit relation descriptions about plans allow the model to handle with completion task and common goals. These aspects are limited in the Sichman's approach, so we have extended it to cover with planning situations at Organisational and Agent levels. Here we claim that a global DEPNET may be described through role features to represent a higher decision level. At Interaction component, our approach introduces some co-ordination requirements in order to check for possible dependencies at DEPNET level. So, the idea of plan relations based on Martial's work addresses the dynamics with the environment through planning level. Traditional scheduling and negotiation can also be aggregated by the developer according to co-ordination services required at task level. For the Environment brick, we have associated some co-ordination mechanisms from GPGP approach with our dependence structure (local and global DEPNET). In this way, some methods executing on the planning level use relations between tasks described in the global DEPNET to cover with some co-ordination requirements at (E) component.

This approach shows that the dynamics of MAS can be handled at each MAS component, rather than only within the agent, as claimed by AOP. Another approaches of co-ordination under IOP, OOP and AOP, have already shown that is possible to take dynamic aspects at each MAS component. In a methodological view, our model covers with the MAS dynamics through its components and addresses some important ideas about a really componential methodology of the multi-agent design and development. Besides, our approach does not adopt some meta-level management in order to avoid loose the decentralized aspects of the distributed coordination handling, by using a centralized control or layer. This approach intends to keep away a decisional bottleneck of a centralized co-ordinator or control and also to centralize amounts of knowledge and control from dependency requirements among components.

We intend to apply this model to another open, distributed and dynamic application, a Distributed Agenda Management System. This application will allow us to explore a more cognitive domain with more planning details. The

16

model has to be refined to cover with all aspects of MAS design and development to provide efficient applicability and general reusability.

References

[1] C. Castelfranchi. Commitments: from individual intentions to groups and organizations. In V. Lesser, editor, *Proceedings of the First International Conference on Multi–Agent Systems (ICMAS'95)*, pages 41–48, San Francisco, CA, 1995. MIT Press.

[2] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–261, Mar. 1990.

[3] P. R. Cohen and H. J. Levesque. Teamwork. *Nous, Special Issue on Cognitive Science and Artificial Intelligence*, 25(4):487–512, 1991.

[4] K. Decker and V. Lesser. Generalized partial global planning. *Journal on Intelligent Cooperative Information Systems*, 1(2):319–346, June 1992.

[5] K. Decker and V. Lesser. Designing a family of coordination algorithms. In V. Lesser, editor, *Proceedings of the First International Conference on Multi–Agent Systems (ICMAS)*, San Francisco, USA, 12–14 June 1995. AAAI Press/MIT Press.

[6] K. S. Decker. TÆMS: A framework for environment centred analysis and design of coordination mechanisms. In G. O'Hare and N. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, chapter 16. John Wiley and Sons, 1996.

[7] Y. Demazeau. Steps towards multi-agent oriented programming. In *1st International Workshop on Multi-Agent Systems (IWMAS'97)*, Boston, 1997.

[8] J. Ferber. *Les Systèmes Multiagents: Vers une Intelligence Collective*. InterEditions, Paris, 1995.

[9] A. Haddadi and K. Sundermeyer. Belief desire intention agent architectures. In G. O'Hare and N. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, chapter 5. John Wiley and Sons, 1996.

[10] N. R. Jennings. Coordination techniques for DAI. In G. O'Hare and N. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, chapter 6. John Wiley and Sons, 1996.

[11] V. R. Lesser. Reflections on the Nature of Multi-Agent Coordination and Its Implications for an Agent Architecture. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):89–111, 1998.

[12] T. W. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, Mar. 1994.

[13] F. Martial. *Coordinating plans of autonomous agents*, volume 610 of *Lecture Notes in Artificial Intelligence and Lecture Notes in Computer Science*. Springer-Verlag Inc., New York, NY, USA, 1992.

[14] S. Ossowski. *Co-ordination in Artificial Agent Societies*, volume 1535 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1999.

[15] H. V. D. Parunak, S. Brueckner, J. Sauter, and R. S. Matthews. Distinguishing environmental and agent dynamics: A case study in abstraction and alternate modeling technologies. In A. Omicini et al., editors, *Engineering Societies in the Agents World*. Springer-Verlag: Heidelberg, Germany, Dec. 2000.

[16] A. S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. *Lecture Notes in Computer Science*, 1038:42–55, 1996.

[17] P.-M. Ricordel and Y. Demazeau. Volcano, a vowels-oriented multi-agent platform. In *Proceedings of the International Conference of Central Eastern Europe on Multi-Agent Systems (CEEMAS 2001)*, Krakow, Poland, 2001.

[18] J. S. Sichman. *Du Raisonnement social chez les agents. Une approche fondée sur la théorie de la dépendance*. PhD thesis, INP Grenoble, Laboratoire LEIBNIZ, 1995.

[19] M. P. Singh. A customizable coordination service for autonomous agents. *Lecture Notes in Computer Science*, 1365:93–106, 1998.

[20] M. P. Singh. Synthesizing coordination requirements for heterogeneous autonomous agents. *Autonomous Agents and Multi-Agent Systems*, 3(2):107–132, 2000.

[21] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

Le laboratoire Leibniz est fortement pluridisciplinaire. Son activité scientifique couvre un large domaine qui comprend aussi bien des thèmes fondamentaux que des thèmes très liés aux applications, aussi bien en mathématiques qu'en informatique.

Les recherches sur les Environnements Informatiques d'Apprentissage Humain et la didactique des mathématiques ouvrent cette pluridisciplinarité sur les sciences humaines, elles jouent un rôle particulier en favorisant les coopérations entre différentes composantes du laboratoire.

* mathématiques discrètes et recherche opérationnelle
* logique et mathématique pour l'informatique
* informatique de la connaissance
* EIAH et didactique des mathématiques

Les cahiers du laboratoire Leibniz ont pour vocation la diffusion des rapports de recherche, des séminaires ou des projets de publication réalisés par des membres du laboratoire. Au-delà, Les cahiers peuvent accueillir des textes de chercheurs qui ne sont pas membres du laboratoire Leibniz mais qui travaillent sur des thèmes proches et ne disposent pas de tels supports de publication. Dans ce dernier cas, les textes proposés sont l'objet d'une évaluation par deux membres du Comité de Rédaction.

Comité de rédaction

* mathématiques discrètes et recherche opérationnelle
  Gerd Finke, András Sebõ
* logique et mathématique pour l'informatique
  Ricardo Caferra, Rachid Echahed
* informatique de la connaissance
  Yves Demazeau, Daniel Memmi,
* EIAH et didactique des mathématiques
  Nicolas Balacheff, Jean-Luc Dorier, Denise Grenier