28

A GENERIC STRATEGIC LAYER FOR COLLABORATIVE NETWORKS

João Certo[1,2,3], Nuno Lau[1,4], Luís Paulo Reis[2,3]

joao.certo@fe.up.pt, lau@det.ua.pt, lpreis@fe.up.pt

[1] IEETA – Institute of Electronics and Telematics Engineering of Aveiro

[2] LIACC-NIAD&R– Artificial Intelligence and Computer Science Lab, University of Porto,

[3] FEUP – Faculty of Engineering of the University of Porto
Rua Dr. Roberto Frias, 4200-465 Porto, PORTUGAL

[4]DETI –Informatics, Electronics and Telecommunications Dep., University of Aveiro,
Campus de Santiago, 3810-193 Aveiro, PORTUGAL

This paper presents a formal model for a multi-purpose, strategical coordination layer. Based on previous work developed for the RoboCup Soccer simulation, small-size, middle-size, legged leagues and RoboCup Rescue simulation league, a generic coordination model was built that allows the management of collaborative networks of heterogeneous agents. The model uses a multi-level hierarchical approach with the following concepts: strategy, tactics, formations, sub-tactics and roles, from high to low level. Hybrid methods are used to switch formations and tactics. In order to test the model, two strategy instances, for RoboCup Rescue Simulation and RoboCup Soccer, were developed. Strategies are designed with the help of a graphical tool. Results achieved by the team in RoboCup Rescue and Soccer Simulation competitions demonstrate the usefulness of this approach.

1 INTRODUCTION

RoboCup was created as an international research and education initiative, aiming to foster Artificial Intelligence (AI) and Robotics research, by providing standard problems. RoboCup has two main league types: simulation and robotics. Simulation leagues enable research on AI and multi-agent coordination while waiting for the availability of hardware to enable the same type of research.

In RoboCup Soccer leagues two opposing teams play a soccer match, thus creating a dynamic environment. A soccer match provides important scientific challenges, both at an individual level (perception, moving, dribbling, shooting) and at a collective level (strategy, collective play, formations, passing, etc.).

Proposed by Kitano (Kitano, et al., 1999), RoboCup Rescue simulated environment consists of a virtual city, immediately after a big catastrophe, in which heterogeneous, intelligent agents, acting in a dynamic environment, coordinate efforts to save people and property. The agents are of six different types: Fire Brigades, Police Forces, Ambulance Teams and the three respective center agents. Fire Brigades are responsible for extinguishing fires, Police Forces open up blocked routes and Ambulance Teams unbury Civilians. In order to obtain a good score, all these agents work together communicating through supervising center agents.

FC Portugal's research focus is on the development of new coordination methodologies. After successfully developing such methodologies for soccer simulation leagues[1,] the team is working on adapting these methodologies to the Rescue Simulation League already with some success[2].

This paper describes the specification and application of a multi-purpose, multi-domain, adaptable, strategical layer on Multi-Agent Systems (MAS). This layer allows the management of homogeneous and heterogeneous agents, the centralized or decentralized management of the strategy. The paper also presents a graphical strategy building tool, compliant with the defined model.

The layer and tools, although generic, were developed with the RoboCup leagues in mind. However, application domains include businesses composed of several parties. Following such a model, Collaborative Networks (Camarinha-Matos and Afsarmanesh, 2005) as well as MAS in general, may be interesting tools for the analysis and congregation of skills that are required to achieve the goal.

The rest of this paper is organized as follows. The next section presents related work. In section 3 the strategic layer is described. Section 4 presents the graphical tool, showing a strategy for a soccer team and a partial strategy for a rescue team. Section 5 concludes this paper and points out to future work.

2 RELATED WORK

Stone et al. (Stone, 2000; Stone and Veloso, 1999) previously defined periodic team synchronization (PST) domains as domains with the following characteristics: "There is a team of autonomous agents that collaborate towards the achievement of a joint long-term goal". The task at hand was decomposed into multiple rigid roles, assigning one agent to each role. Thus each task's component was accomplished with no conflicts among agents in terms of how they should accomplish the team goal. As it was defined, a role consisted of a specification of an agent's internal and external behaviors. The conditions and arguments of any behavior could depend on the agent's current role, which was a function of its internal state.

Due to inflexibility to short-term changes (e.g. one robot is non-operational), inflexibility to long-term changes (e.g. a route is blocked), and a lack of facility for reassigning roles, a formation was introduced as a teamwork structure within the team member agent architecture. A formation decomposes the task space defining a set of roles with associated behaviors. In a general scenario with heterogeneous agents, subsets of homogeneous agents could flexibly switch roles within formations, and agents could change formations dynamically. Formations included as many roles as agents in the team, so that each role is filled by one agent.

The rest of this section explains the concepts and mechanisms developed for soccer simulation leagues usable within the strategical layer.

FC Portugal's team strategy definition extends the concepts introduced by Stone (Stone, 2000; Stone and Veloso, 1999) and is based on a set of player types (roles) and a set of tactics that include several formations for different game situations (defense, attack, etc) (Reis and Lau, 2001). Formations assign each player a positioning (that determines the strategic behavior) and to each positioning corresponds a player type (that determines the active behavior).

[1] FC Portugal won several World and European championships in different RoboCup soccer leagues in the past seven years.

[2] FC Portugal rescue simulation team achieved very good results in RoboCup, including winning a rescue European champion using these coordination methodologies.

When Stone defined a situation, the concept was bound to set-plays. A situation was a set of world state conditions that triggered a series of predefined behaviors within the roles. This concept was expanded (Reis, et al., 2001) and situations were redefined as a group of easily identifiable logic conditions set for high-level world state parameters. As such, situations would not suffer a considerable, temporal, variation and could be then associated with formations. However not every situation had to have its own formation, a set of replacement situations could be used.

The Situation Based Strategic Positioning(SBSP) mechanism (Lau and Reis, 2002; Reis, et al., 2001) is used for strategic situations (in which the agent believes that it is not going to enter in active behavior soon). In this system, the agent calculates its base strategic position in the field in that formation, adjusting it according to the ball position and velocity. This enables the team to cover the ball while remaining distributed along the field, just like in real soccer.

The DPRE , Dynamic Positioning and Role Exchange (and Dynamic Covering) (Reis and Lau, 2001), was based on previous work from Peter Stone et al. (Stone, 2000; Stone and Veloso, 1999) which suggested the use of flexible agent roles with protocols for switching among them. The concept was extended and players may exchange their positionings and player types in the current formation if the utility of that exchange is positive for the team.

In the case of communication in single channel, low bandwidth, and unreliable domains the challenge is deciding what and when to communicate. In ADVCOM (Reis and Lau, 2001) (Intelligent Communication Mechanism), agents use communication in order to maintain world states updated by sharing individual world states, and to increase team coordination by communicating useful events (e.g. a positioning swap). The main innovation of this communication strategy is that agents communicate when the utility of their communication is higher than those of their teammates, using mutual modeling to estimate these utilities.

3 MODEL FOR THE STRATEGIC LAYER

The model here depicted provides a structured method of representing, building and managing a strategy in a scenario where a team of agents is used. The terms *scenario* and *agent* should be considered as broader terms. *Scenario* can be a simulation, a game, a virtual organization (Foster, et al., 2001) or any other kind of set where there is an environment, with *agents* who have one or more objectives. Likewise *agents*, besides being software computational entities, can be any kind of independent units like robots, persons or participants of a virtual enterprise (Cardoso and Oliveira, 2005).

This model handles static, dynamic, reactive or nonreactive environments and is designed to manage team strategy and cooperation. A team is an aggregation of *agents* with common goals. When *agents* in a team work together cooperatively they do *teamwork* (Cohen and Levesque, 1991; Tambe, 1997). In this model, homogeneous and heterogeneous *agents* can be used. In heterogeneous environments the term *agent type* is used for differentiation.

3.1 Structure

In order to better explain the model, a top-down approach will be followed. Figure 1 represents the proposed model and depicts the interconnections between the concepts presented in this model, expanding one branch for each concept.

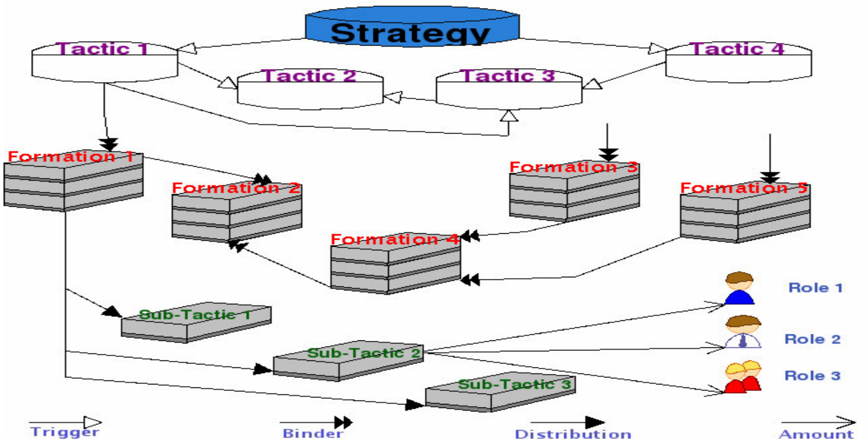

Figure 1 - Schematic of strategic concepts.

3.1.1 Strategy

Informally, a strategy is the combining and employment of means in large-scale, long-range planning and the act of directing operations for obtaining a specific goal or result. In this scope, a strategy can be seen as a set of different configurations of collaborative networks or virtual organizations resulting in different approaches to reach the same ultimate goal.

Formally, a *strategy* is a combination of *tactics* used to face the scenario and the *triggers* to change between tactics, where a *strategy* can have several *tactics:*

Strategy={Tactics, Triggers}; Tactics={Tactic 1, Tactic 2,...,Tactic t}, t □ *N* (**1**)
Triggers = {Trigger 1, Trigger 2, ... , Trigger tg } , tg □ *N*

3.1.2 Tactic

A tactic is an approach to face the scenario in order to achieve a goal. Tactics deal with the use and deployment of agents in the scenario for different, identifiable situations. Tactics specify which resources of the collaborative network should be used to reach the different, partial, goals.

Formally, a *tactic* defines *agents*' *formations* as the arrangement of *agents, situations* as the combination of scenario conditions that can be seen as a more particular problems and *binder* as the association between a *formation* and a *situation* or between several *situations* and a *formation*. *Tactics* can optionally also set *tactical parameters,* the default thresholds on which *agents* base their decisions.

A *tactic* should be self-sufficient, i.e., it does not need other tactics to function through all the simulation. There can be only one *tactic* active at one given time.

Tactic = {Formations, Situations, Binders, [Tactical Parameters]} ; (**2**)
Formations = {Formation1, Formation2, ... , Formation f } , f □ N ;
Situations = {Situation 1, Situation 2, ... , Situation s} , s □ N ;
Binders = {Binder 1, Binder 2, ... , Binder b} , b □ N ;
Tactical Parameters = {Tactical Paramtr1,..., Tactical Paramtr tp} , tp □ N

In a *situation*, the conditions that make it unique are defined. The *binder* sets the *situations* that lead to a *formation*. Optionally, a *binder* can set the connection between several origin *formations* and a terminus *formation* through *situations*.

Situation = {Condition 1, Condition 2, ... , Condition cd } , cd □ N (**3**)
Binder = {[Origin Formations], Situations, Terminus Formation}.,
[Origin Formations] , Terminus Formation □ Formations

3.1.3 Formation

A formation is a high-level structure that aggregates all the agents with the intent of assigning them to specific sub-tactics. The aggregation is either wrought by using agents that belong to the same type, have the same more immediate goals, or both. As such, a formation can be seen as a collaborative network for a pressing goal.

Formally, a *formation* is a specific association of *sub-tactics* with a defined *distribution* that may specify an *agent type*. Only one *formation* can be active at any given time. As such, the *formation* must include *sub-tactics* for all *agents*. The same *sub-tactic* can be used more than once in a *formation*. This allows an implicit definition of *Group*. Let *sub-tactics* be a multiset (Stanley, 1997) where m(SubTactic st) defines the multiplicity of a *sub-tactic*.

Formation = { Distribution, Sub-Tactics, [Agent Types] } , (**4**)
Sub-Tactics = {(SubTactic1, m(SubTactic1)), (SubTactic2, m(SubTactic2), …
, (SubTactic st, m(SubTactic st))} , st □ N

For each element in *sub-tactics* there is a correspondent value in a *distribution*:

Distribution = {Value1, Value2, ... , Value v} , v = $\sum m(\mathit{Sub\text{-}Tactics\ st})$ (**5**)

A distribution specifies absolute or percentage distribution values for each sub-tactic in the formation. These *values* always refer to *agent types* when applicable. In this manner, the total of *values* can surpass 100%, but not for a specific *agent type.*

The association with *agent type* is implicit when a *sub-tactic* can only be applied to one *agent type*. Otherwise, when more than one *agent type* can be used (see section 3.1.5), an *agent type* must be specified for that *sub-tactic*:

[Agent Types] = {Type1, Type2, ..., Type ty } , ty □ N (**6**)

3.1.4 Sub-Tactic

A sub-tactic reflects the approach to face the scenario of a limited set of agents either partially for a number of situations or during the whole scenario. A sub-tactic can be seen as a limited association of resources, a virtual enterprise or a network.

Formally, a *sub-tactic* is an association of *roles* with one default *amount* of *agents* assigned to those *roles*. Additionally a *sub-tactic* may also have *sub-tactical parameters* to reflect specific thresholds, *agent* parameters, coordination options or other values that are needed to configure the *roles* used on the *sub-tactic*.

$$\begin{gathered}
\text{Sub-Tactic} = \{\text{Amounts, Roles, [Sub-Tactical Parameters]}\} \qquad \textbf{(7)}\\
\text{Roles} = \{\text{Role 1, Role 2, ... , Role r}\}\ ,\ \mathrm{r} \ \square\ N\\
\text{Amounts} = \{\ \text{Amount 1, Amount 2, ... , Amount a}\}\ ,\ \mathrm{a} = \sum \mathit{role}\ \mathrm{r}
\end{gathered}$$

A *sub-tactic* can have one or more *roles* and for each *role* in *sub-tactic* there is an *amount* in *amounts*. Like in a distribution, an *amount* specifies either absolute or percentage values for each *role* in the sub-*tactic*. Percentage *amounts* in a given *sub-tactic* must total 100%.

Sub-tactics can be divided into *Typed Sub-Tactic*s and *Generic Sub-Tactics*. In a *typed sub-tactic* at least one of the *roles* is associated with an *agent type*, which becomes the *sub-tactic's* type. In order to ease the handling of different *agent types*, it is not possible to use *roles* of different *agent types* in the same *sub-tactic.* As such, *typed sub-tactic* can only use *roles* for one *agent type* together with *generic roles*. As a consequence, to build a *formation* with different *agent types,* there should be at least one *sub-tactic* for each *agent type*.

A *generic sub-tactic* is a particular kind of *sub-tactic* without any association with an *agent type*. Thus, in a generic *sub-tactic,* only *generic roles* can be used. As it was previously stated, if a *generic sub-tactic* is used in a *formation* that contains *sub-tactics* for more than one *agent type*, an *agent type* must be specified. This type is specified together with a *distribution value* when *agents* are assigned to a *generic sub-tactic*.

3.1.5 Role

A role is a normal or customary activity of an agent in a particular environment. In collaborative networks, the role can be seen as the activity pattern of each participant in a virtual enterprise.

Formally, a *role* is a set of *algorithms* in a defined sequence that describes an *agent*'s behavior. The behavior description is expected to include, when relevant, the specification on how the *agent,* should coordinate with *agents* in the same *role* or in other *roles*. The coordination can be of three different kinds: all *agents* with the same *role* can either act individually, form one *group* or form several smaller *groups* (with a rule specified inside the *role*).

The *role* also defines partial objectives accordingly to the coordination method used. Although *roles* can describe the behavior for an entire scenario, they can also describe the behavior for only a given time frame or *situation*. *Teams* form their *roles* by combining different motion and action mechanisms with partial objectives. The *role* level is the lowest in the proposed model.

As in *sub-tactics*, *roles* can be divided into *Typed Role* or *Generic Role*. A *typed role* is a particular kind of *role* that can only be assumed by one *agent type*. Using heterogeneous *agents* does not necessarily means that *typed roles* or *agent types* will be used in the *strategy*. *Typed roles* are used when, in heterogeneous *agents*, there is a need to employ the different *agent*'s properties or capabilities. A *generic role* is a kind of *role* that can be assumed by any of the *agent types* used in a *tactic*.

3.1.6 Decision, Supervising and Communication

The decision maker depends on the *agents*' organization and types set by the scenario. In teams where there is only a supervisor and all the *agents* are “dummy”, the strategical layer will obviously only be applied to the supervisor.

In MAS, the first rule is that all *agents* have full knowledge of the strategical layer being used. Then if all *agents* have a good, shared, world state knowledge, the layer can be used with no extra communication. This is accomplished because all the *agents* switch their *tactics*, *situations* and *formations* based on the same conditions and at almost the same time. When a team already uses a mechanism like ADVCOM (section 2) the strategical layer can also be applied to scenarios where communications is limited and unreliable without extra communication overload.

If *agents* have more limited computational resources but still have good world state knowledge synchronization, the layer can be computed only by a supervising *agent*. This *agent* would only have to communicate a new *formation* whenever declared by the strategical layer. The supervising *agent* is chosen taking into account the *agent* who normally has more computational resources. Some scenarios specifically have supervising *agents*.

3.2 Agent Assignment

The strategical layer defines either absolute or percentage forms for distribution values and role amounts. This possibility is given so that strategies can be built independently from the agent number used in the scenario.

Another possibility of the model is to use both absolute and percentage forms simultaneously. In this model, for both distribution values and role amounts, absolute forms for values take priority over percentage value forms. This means that agents are assigned first to roles in a sub-tactic specified with absolute distribution values and with absolute role amounts in the referred role. Next agents are assigned to sub-tactics with only absolute forms of distribution values. The succeeding priority is assigning agents to roles specified by absolute role amounts in a sub-tactic with a percentage distribution values.

Finally, for the remainder agents that use percentage forms in the mixed method, or when the percentage form is the only assignment method used, the assignment priorities are: first to sub-tactics and then to roles. When converting to absolute numbers, the values are truncated.

If agent types are in use, the previously defined assignment method is applied separately to each agent type. As it is easily concluded the mixed method allows the definition of priority roles in environment where the total agent number is unknown.

The agent assignment methods defined what roles needed to be used, particularly for environments where the total agent number is unknown. In order to assign a specific role to a specific agent, in its simpler form, agents can orderly chose a role based on the environment identification (sequential agent id). Optimal role assignment depends on scenario conditions like proximity to objectives, relative agents' positions, etc.. Based on this fact, the model does not specify a method. In fact, a method like DPRE (section 2.) that uses dynamic role exchanges is strongly advisable. To be noted that the strategical layer is still compatible with dynamic, situation based positioning like SBSP (section 2.). This is accomplished because the positioning systems are specified inside the role.

4 GRAPHICAL TOOL FOR BUILDING STRATEGIES

The graphical tool provides a visual interface for building strategies. By using graphical representations of the strategic layer components, it is possible to interconnect them. The tool exports the edited strategy to an XML file which can be used to implement the layer in *agents*. It also features a C++ code generator still in its early stages. The tool's GUI is provided by Kivio, a flowcharting and diagramming application for the KOffice[3] application suite. A customized, installable, stencil set with the layer objects was also built (Figure 3).

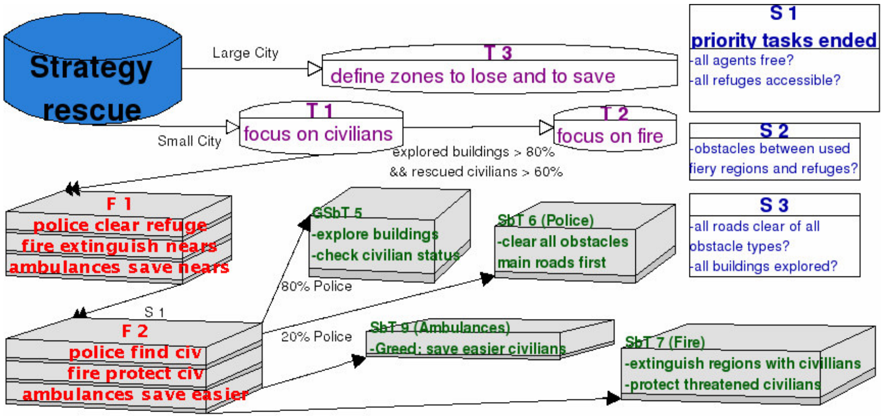

Figure 2 - Partial rescue strategy

Figure 2 depicts a simplified rescue strategy expanded only in one tactic and one formation. In order to fully perceive the strategy some additional knowledge of the rescue simulation league is advisable. On situation S1 formation F1 switches to F2. Note that the *generic sub-tactic* GSbt 5 uses *generic roles* to find civilians. This role can be performed by Fire Brigades, Police Forces or Ambulance Teams.

Using soccer as an example, for a simple strategy, the same sheet can be used to represent the entire layer as seen in Figure 3.

[3] KOffice is an office suite for the K Desktop Environment released under free software/open source licenses. Available at http://koffice.org/.

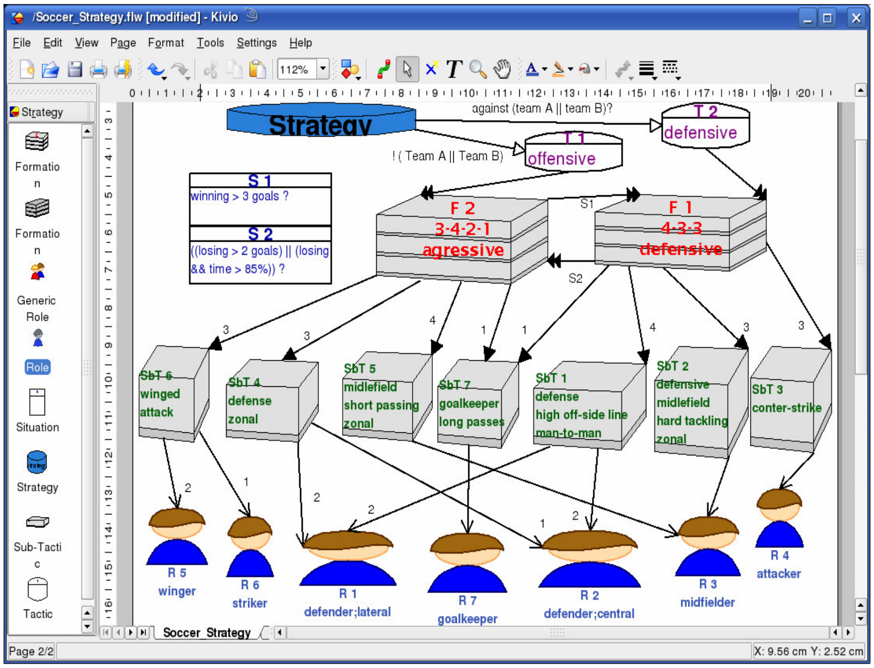

Figure 3 - Screenshot of the graphical tool featuring a soccer strategy

As shown, a trigger can originate in a *strategy* thus defining the initial *tactic*. Likewise a *binder* can originate in a tactic thus defining the initial *formation*. For a more complex *strategy* a multi-sheet is recommended separating *strategy*, each *tactic*, *formations*, *situations* and *sub-tactics*. In fact, when defining several *tactics* which use at least one *binder* with no precedence (*origin formation)* a separate sheet for each tactic is mandatory. Figure 2 although only expanding on one strategy branch (not possible under the layer) uses formation F3 and F5 with no precedences.

5 CONCLUSION AND FUTURE WORK

The proposed strategical layer is now fully integrated with our soccer and rescue teams and is successfully being used in our rescue team. The layer also maintains full compatibility with all our RoboCup soccer teams, as the soccer model is a particular case of the specified generic layer. Results in international competitions for both the domains tested, proved the success of the layer.

The model flexibility expands from using it in an environment where a single program manages all homogeneous “dummy” robots, to its collective use in heterogeneous, multi-agent systems. When domains have similar nature like in soccer simulation and soccer robotic leagues, the strategies defined in one, can easily be adapted to the others. This is achieved by only modifying the roles in the existing sub-tactics.

Using the model in areas outside the implemented domains should be straightforward. The layer can be used in areas where there is a need of management of distributed activities like in collaborative environments.

Strategies are easily built through the use of a very user-friendly graphical tool. By using a frequently improved, open source, editor as its base, the developed graphical tool can take advantages of its innovations.

In the future, further development of the graphical tool's source code generator is expected, both in efficiency and language portability. These developments will enable a more generalized use of the strategic layer in the context of RoboCup and in other cooperative domains. Thus, we plan to use the strategical layer, with different instantiations, built using the graphical tool, in all our teams (simulation 2D, simulation 3D, small-size, middle-size, legged, simulation rescue and physical visualization) participating in European and world RoboCup competitions in 2007.

6 REFERENCES

1. Camarinha-Matos LM, Afsarmanesh H. "Collaborative networks: a new scientific discipline". In Journal of Intelligent Manufacturing, Springer Netherlands, 2005; 16(4-5): 439-452.
2. Cardoso HL, Oliveira E. "Virtual Enterprise Normative Framework Within Electronic Institutions". In Engineering Societies in the Agents World V, Springer Berlin/Heidelberg, 2005; 3451: 14-32.
3. Cohen PR, Levesque HJ. "Teamwork". In Noûs, Special Issue on Cognitive Science and Artificial Intelligence, 1991; 25(4): 487-512.
4. Foster I, Kesselman C, Tuecke S. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations". In International Journal Of High Performance Computing Applications, Sage Science Press, 2001; 15: 200-222.
5. Kitano H, Tadokoro S, Noda I, Matsubara H, Takahashi T, Shinjou A, Shimada S. "RoboCup Rescue: search and rescue in large-scale disasters as a domain for autonomous agents research". In Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, 1999; 739-743.
6. Lau N, Reis LP. "FC Portugal 2001 Team Description: Configurable Strategy and Flexible Teamwork". In RoboCup 2001: Robot Soccer World Cup V, Birk, A., Coradeschi, S., and Tadokoro, S., eds. Springer Berlin / Heidelberg, 2002; 2377: 1-10.
7. Reis LP, Lau N. "FC Portugal Team Description: RoboCup 2000 Simulation League Champion". In RoboCup-2000: Robot Soccer World Cup IV, Stone, P., Balch, T., and Kraetzschmar, G., eds. Berlin: LNAI Springer-Verlag, 2001; 2019: 29-40.
8. Reis LP, Lau N, Oliveira E. "Situation Based Strategic Positioning for Coordinating a Simulated RoboSoccer Team". In Balancing Reactive and Social Deliberation in MAS, Hannebauer, M., Wendler, J., and Pagello, E., eds. Berlin: LNAI Springer, 2001; 2103: 175-197.
9. Stanley RP. Enumerative Combinatorics, Cambridge University Press., 1997; ISBN: 0-521-55309-1.
10. Stone P. Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer, MIT Press, 2000; ISBN: 0262194384.
11. Stone P, Veloso M. "Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork". In Artificial Intelligence, 1999; 110(2): 241–273.
12. Tambe M. "Towards Flexible Teamwork". In Journal of Artificial Intelligence Research, 1997; 7: 83-124.