

Integrating social skills in task-oriented 3D IVA^{*}

Fran Grimaldo¹, Miguel Lozano², Fernando Barber², and Juan M. Orduña²

¹ Institute of Robotics, University of Valencia,
Pol. de la Coma, s/n (Paterna) Valencia, Spain
`fran.grimaldo@robotica.uv.es`

² Computer Science Department, University of Valencia,
Dr. Moliner 50, (Burjassot) Valencia, Spain
{`miguel.lozano`, `fernando.barber`, `juan.orduna`}@uv.es

Abstract. This paper presents a set of mechanisms oriented to incorporate social information into the decision taking of task-oriented 3DIVA. The aim of this approach is to integrate collaborative skills in different character's roles (seller/buyer, worker, pedestrian, etc.) in order to enhance its behavioral animation. The collective intelligence expected in this kind of multi-character domains (e.g. storytelling, urban simulation, interactive games, etc.) requires agents able to dialogue/interact with other characters, to autonomously group/ungroup (according to their goals), or to distribute tasks and coordinate their execution for solving possible conflicts. The social model implemented follows the definitions for collaborative agents, since agents use communicative acts to cooperate. In this context, collaboration derives mainly from two points: team formation (grouping for 3DIVA) and task coordination (reducing dependences between agent activities). Finally, we show the results obtained in 3D multi-character simulations (resource competition), created to verify the social behavior introduced.

1 Introduction and related work

Artificial worlds inhabited by 3D Intelligent Virtual Agents (3DIVA) can be considered as Multi-Agent Systems [10] where a number of actors deal with a finite number of shared resources. They are applications where collective intelligence arises as the result of the interaction established between characters that cannot avoid the social aspects of the behavioral animation problem.

3DIVA spectrum comprises a huge range of actors with different features. We basically identify two trends. On one side, we find reactive agents; from the classical *boids* introduced by Reynolds [18] up to more sophisticated crowds [20]. They can animate crowd behavior, normally reduced to movement generation and reactive actions, since individuals are so simple that their operation is quite limited. On the other side, deliberative agents, such as Jack [19] or Steve [9], can execute complicated tasks but they are not really designed to autonomously collaborate with their peers. Somewhere in between, interactive storytelling and

^{*} Supported by the Spanish MCYT under TIC-2003-08154-C06-04

other group simulation scenarios tend to reproduce a global approach in which interactions and communications are normally script driven [17] or predefined by the “author” [6]. Therefore, virtual worlds lack characters intelligent enough to autonomously animate conversations (e.g. manage non expected meetings between two characters).

The simulation of a number of task-oriented agents (devoted to accomplish some tasks/goals in the environment) easily falls in conflictive domains even though the goals are compatible [10]. Obstruction situations then appear when characters compete for the use of shared resources (3D objects). Bearing this in mind, we identify the need to integrate social mechanisms in order to enrich the agent-centered decision making.

Three main problems appear when dealing with inhabited Intelligent Virtual Environments (IVE): communication, coordination and cooperation. Regarding to the first one, two capital Agent Communication Languages (ACL) have been developed; KQML and FIPA ACL. Both define communication between agents as an asynchronous exchange of messages, known as speech-acts. For instance, 3D BDI actors in [11] use FIPA ACL for the internal control of the agent. Unfortunately, interaction does not take place between 3D actors inside the virtual world but between the internal modules that constitute an individual. On the other side, Benford has a system that allows communication inside a crowd [2]. However, the process is controlled by an specific *Third Party Object* and the basic scenario is then formed by three objects.

Different formalisms can be used in order to achieve multi-agent coordination: tuple centers (which follow a blackboard model that do not reflect the peer-to-peer communication we are interested in), interaction protocols and ACL semantics. Nevertheless, all of them have their weaknesses when dealing with complex situations [3]. For example, Multiagent Planning Language (MAPL) [4] uses speech-acts to synchronize planned tasks. As the control over each resource is assigned to a unique agent, the model supports communication and coordination but it does not cooperate nor compete for common resources (as 3DIVA requires). Another example of coordination mechanisms intended for assisting in scheduling activities is Generalized Partial Global Planning (GPGP) [8]. In this approach, static tree structures describing the operational procedure are shared between the characters which merge the meta-plans and figure out the better action order to maximize global utility. As stated in [10], this solution has a limitation in the number of agents trying to make a coherent global plan and it is not very good at facing not planned situations which are quite common in dynamic and unpredictable environments (e.g. unexpected meetings between characters).

Collaborative planning has been deployed in several different domains. In SharedPlans [13], agents must have previously agreed about a common high-level team model and also about certain procedures (e.g. to assign agents and subgroups to subactions) to complete partial actions and plans, thus performing the group decision making. In STEAM architecture [21], oppositely, team formation and conflict resolution are governed by a leader. Besides, once role

allocation is done, team members are supposed to perform their tasks independently. Nevertheless, this ideal situation is not commonly found in shared worlds where actors interfere while executing their actions. Lastly, virtual platoons in the RETSINA system [12] support multi-agent collaboration, thanks to the use of a planner that has all communicative acts predefined inside Hierarchical Task Networks (similar to storytelling approaches). Communication is then used to synchronize tasks and punctually manage conflicts but there is no conversational animation designed to get information and create an internal representation of the other agent that can be considered for future decisions.

mVITAL [1] and SimHuman [23] appear as two close systems in the literature of virtual humans but they are not focused on collaborative 3DIVA. The system presented in this paper, though, follows the Co-X definitions for agent interactions [22] and it exhibits Collaboration. That is, it uses both Conversation (direct peer-peer communication) and Cooperation (joint intent on the part of the individual agents) to face topics such as organizational techniques (grouping), task distribution, coordination, conflict resolution, etc.

The next section reviews the general requirements needed by task-oriented agents in order to extend their planning modules with social skills. We focus on two collaborative mechanisms: teamwork and task coordination. The solutions implemented for these two issues are the subject matter of section 3. Section 4 shows the first results obtained in 3D multi-character environments where resource competition is used to verify the social mechanisms introduced. Finally, we state our conclusions and future work.

2 Requirements for social 3DIVA

Complex IVE (e.g. urban domains) need the instantiation of autonomous 3D actors performing different *client/server* roles; for instance: virtual waiters in a crowded bar, assistants in a virtual museum, and obviously the corresponding customers. This kind of characters face situations that require the animation of dialogues as well as the ability to develop relationships and use them in a proactive way.

We emphasize three different interaction situations while simulating this type of social roles: *resource competition*, *grouping between actors* (i.e. create and destroy groups to fulfill certain objectives) and *joint task execution* (e.g. carry heavy objects by two actors). To resolve them, agents need to embed a planning system able to recognize the lack of information and to manage communication in an autonomous way. This article is focused on the team formation protocol that the characters use to create groups as well as the task coordination mechanism implemented to reduce conflicts when competing for shared objects. *Joint task execution* is out of the scope of this paper.

2.1 Grouping between actors

The Joint Intentions Theory [7] states that a team is created when each agent commits to a goal and receives notification about the commitment of the remain-

ing individuals. However, the theory does not consider important aspects such as how actors arrive at this point, that is, how a team objective is acquired. Systems doing team work have generally avoided this problem by settling the teams beforehand and giving the agents the awareness of being part of the group [13, 12]. This persistent definition of teams, though, is unsuitable in some 3D simulations, where characters can change their roles and make temporary associations to better fulfill parts of their global goal (e.g. ephemeral associations in [5]). In these scenarios, groups can be dynamically formed and disintegrated (i.e. new agents join the group while some others decide to leave).

For example, a 3D construction domain could be inhabited by a number of foremen and laborers that work together to build a virtual space. Sociability can appear when several virtual laborers are assigned the same objective (e.g. make concrete) by different foremen. Initially, they might be unaware of that common assignment, therefore, there would not be a formed team and agents will act independently. But, as soon as they realize their common goal, they will jointly commit to it and resource competition between the partners of the group will consequently be relaxed. That is, one agent's planning process must consider the other members of the group.

To animate this behavior, independent characters first need to detect when their operation is being affected or interrupted by the actions of other individuals³. Once this communication need is identified (i.e. a precondition has been violated), the agents can manage the situation by: a) initiate a conversation, b) exchange their individual goals, c) detect compatibility between them, and d), decide whether to be part of a group or to leave it. In the next section we will present the team formation protocol defined to implement these aspects.

2.2 Task coordination

As mentioned before, resource competition is managed through the use of a task coordination mechanism. Despite the fact that some scenarios tend to reproduce task-independent actors, this constitutes a strong simplification which will directly affect the quality of the resulting animation. Actions being performed by task-oriented agents can be dependent, in these situations, their execution will affect the operation of other 3D actors and will generate coordination problems. Hence, coordination can be defined as *managing dependences between activities* [16].

Basically, coordination mechanisms are based on an information gathering stage, followed by the processing of the data previously exchanged. To minimize the normal dependency problems in multi-activity simulations with shared resources, the participants should complete the perceived state with the activities already planned by other characters. In this manner, the embedded planning formalism will be more informed to select a proper task that reduces interference (e.g. one barman will attend other customers while the desired bottle is

³ In STRIPS-based planners, this can be achieved by regularly checking the preconditions of the current task

being used by another barman). Task coordination within a group will require an internal publication of the ongoing tasks, so that, members of a team can avoid interfering or disturbing their mates (see sections 3.3 and 4).

We have designed another way to avoid the dependences between the activities of two agents: to partition the goal set, so agents try to accomplish different independent subgoals. Unfortunately, identifying the independence of subgoals may be as difficult as the planning process itself and it is not generally affordable for a 3D agent. Therefore, we have defined an estimator to help task coordination attending to the objects involved in the set of facts that define the final goal. This heuristic obtains a good partitioning although it is not guaranteed that the subgoals are independent.

Goal partitioning: We say two facts f_1 and f_2 are related ($f_1 \sim f_2$) if both refer to a common object (see equation 1⁴). Therefore, the partition of a set of facts (S) will be given by the quotient set of this relation: S/\sim .

$$f_1 \sim f_2 \text{ iff } \exists o : (f_1(o, o') \vee f_1(o', o)) \wedge ((f_2(o, o'') \vee f_2(o'', o)) \quad (1)$$

Using this definition each agent can divide his goal in a set of “near independent” subgoals. The aim of this approach is to relax the dependencies between activities being performed by the actors. According to this, when two agents compete for the use of some shared objects, they use goal partitioning and try to select independent subgoals (see section 4).

3 Social model

Agents presented in [15] suffer from autonomous communication modules to coordinate their actuation or exchange information with their partners. Thus, the behavioral consistency of the multi-agent animations can be poor. According to this, we have extended the previous agent centered architecture in order to allow social characters to work together in a common scenario.

Our multi-agent animation system follows a distributed architecture. This modular structure separates the graphics engine and its semantic database (Unreal Tournament) from the deliberative agents. These are in charge of controlling the actuation of the 3D actor inside the virtual world thanks to the use of an heuristic planner. The collaborative features developed for our characters and presented in this paper are based on a communication model [14] in which the environment acts as a transmission channel (similarly to the air in the real world). Therefore, communication can be considered as an action carried out throughout the world; that allows us to design message filters depending on distance to other agents, presence of walls, etc. Opposite to blackboard systems, the environment directly delivers the ACL structured messages between the agents, which animate the conversation.

⁴ This definition is for facts of arity 2, but it can be easily extended to any fact

The social model embedded is composed by three mechanisms: a Conversational Task Controller, which permits agents to handle conversations with their peers (point 3.1); a team formation protocol, that assess the convenience of being part of a group (point 3.2); and a task coordination mechanism, in charge of reducing conflicts while competing for shared resources (point 3.3).

3.1 Conversational Task Controller

The Task Controller governs the agent activity at anytime and it decides what to do depending on the agent and world states. This Finite State Machine (FSM) incorporates several states to control conversations between characters, hence, it is able to animate the behavior of 3D actors with social skills. As shown in figure 1, after having reached a comfortable position to speak (REACH AGENT brings the conversers near), agents will alternatively TALK and LISTEN. These two states allow the agents to generate the typical query/response interchange to gather information. This FSM can only manage one conversation at the same time, thus, agents already part of a dialogue will deny another proposal of interaction.

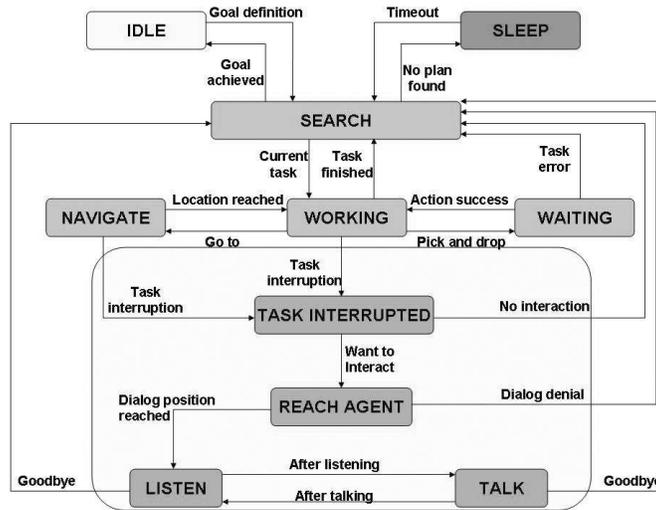


Fig. 1. Conversational Task Controller

One important aspect is when to start a conversation. The need of communication is generated when an action fails due to the interference of another agent (e.g. an actor takes an object that another character wanted to pick). This situation is resolved by conversing with the character who interfered in the success of the execution. Within this dialogue, the agents will communicate their current occupation (i.e. send their current STRIPS task, for example, *move Chair_1*

from *Bedroom to Kitchen*) so that task coordination can be applied further on, as we will explain in point 3.3.

3.2 Team formation

The aim of grouping is to reduce interagent interference and to enhance the quality of the behavioral animation. Whereas non interfering agents do not normally need to agree to reach their goals, others interrupting their operation will need to consider the possibility of temporarily creating a group to better fulfill their tasks and achieve their goals. This evaluation will be done through a team formation protocol which is based on a goal checking between agents. Say agent A and B have goals composed by a set of facts ($G_A = \{f_i\}, G_B = \{f_j\}$), then, three different types of situations are distinguished:

- A couple of agents have *fully related* goals when they share the same set of facts ($G_A = G_B$) or when the facts of one of them are a subset of the facts of the other one ($G_A \subset G_B$ or $G_A \supset G_B$).
- The goals are *partially related* when the intersection is not complete and there are still some facts solely ascribed to one agent ($G_A \cap G_B \neq \phi$).
- The goals are *non related* when there is no intersection between the goals ($G_A \cap G_B = \phi$).

Fully related goals can be problematic, as the close relationship of the objectives could produce many dependences and conflicts during the simultaneous execution (e.g. all the agents want the same objects at the same time). According to this, candidates to form a group will use goal matching in their team formation protocol and will create a group when their goals are fully related.

For instance, one possible extension of the *funny dinner-date* problem [6] can consist on some 3D actors cleaning up a flat. More precisely, the owner could want to clean the whole flat, while two more friends would only help in some tasks (see figure 2). In this scenario, agent A can join agent B to dust the kitchen and agent C to polish the hall. However, transitivity cannot be applied when forming groups. Even though G_A and G_B are fully related, G_B and G_C are non related, as a consequence, B joins A but not C . Thus, team formation protocol is limited to decide if two 3D actors cooperate, without affecting previous commitments. Nevertheless, multi-agent teams can be reproduced due to the fact that characters can separately create as many couples as needed.

The ability to coordinate groups is a key point to produce consistent and lifelike simulations. We consider cooperation as an internal intentional posture of the agents, therefore, while being part of a team, they will continuously communicate their intentions⁵ to their mates in order to facilitate task coordination. Knowing at all times the current actions of the teammates will prevent constant obstruction between characters.

Finally, characters should also be able to leave their teams, that is, they have to communicate their departure of the group when necessary. This occurs when

⁵ Their current task is used to consider the actual intention



Fig. 2. Example of the goals of three agents cleaning up a flat. Intersection relationships between their goals impede to apply transitivity in the team formation protocol.

their goal is no longer fully related with the objectives of the community (e.g. a foreman orders a laborer to change his personal goal).

3.3 Coordinated Task-Oriented agents

Task-oriented agents can coordinate their operation if they know the intentions of their surrounding mates. The information about the ongoing actions that other characters intend to complete can be used to manage the constraints imposed by them. In order to represent the operation of external agents, 3D actors need an extended memory model that, aside from the perceived state of the environment, holds a set of communicative beliefs (*c_beliefs*) acquired through the conversations established with other cohabitants of the virtual world (see figure 3).

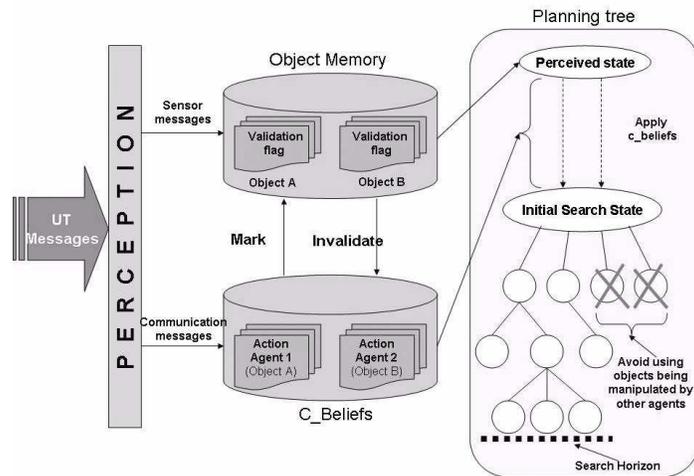


Fig. 3. Coordination model

A *c.belief* corresponds to a task being executed at this moment by another character. Two aspects appear when working with *c.beliefs*:

- Whether to trust another character about his current task is an important issue, because it could lead the actor to an incongruent mental state. Due to the reactive nature of our planning formalism, an actor should not think, for example, that one partner will successfully open a door while a second one is going to close the same door; since he does not know their timing. In this way, all the operators being communicated by external agents will not become a *c.belief* in the memory. Instead, after receiving a communication message, the agent will check the compatibility of the new information with the *c.beliefs* already stored. This checking is based on the resources, which will be locked when an agent has a *c.belief* above them (see *Mark* signal in figure 3). Therefore, when an agent tries to match a new operator, if it uses a locked resource, it will be considered incompatible and it will be discarded. Hence, the actor trusts the first agent who notified his intents and avoids conflicts that might happen with later *c.beliefs*.
- When to remove a *c.belief* from the memory is another problem. However, as characters are continuously perceiving the state of the world, they can contrast the sensorized information against the preconditions of the current *c.beliefs* and delete them when they are not true (*Invalidate* signal in figure 3).

This new information, stored inside the memory, can be used by the planning formalisms of the agent to generate coordinated plans. As previously exposed, characters are benevolent and their intention is to let the others finish their previous commitments. Following this premise, the miniMin-HSP planner used by our 3DIVA [15] will now start the search from a future virtual state resulted from applying the *c.beliefs* over the current perceived state (this mental execution uses the *add* and *delete* lists of the STRIPS definition for the the external ongoing actions). In this manner, it constructs a prospective situation that skips actual dependences. Additionally, each *c.belief* will lock the objects being used by other agents thereby forbidding to jeopardize the success of the tasks previously initiated (see figure 3).

4 Results

In order to verify the techniques previously explained, we have executed several simulations in a structured 3D environment where agents perceive local information (in a room domain) from their synthetic vision sensors [14]. This simulation framework can reproduce social worlds where 3DIVA deal with interactive situations (e.g. urban, building or home-based scenarios).

As a motivating example, we present a problem of multi-character resource allocation (inspired by the *funny dinner-date* problem), where agents compete for the use of shared objects. Objects can be moved using the *move* operator, which is composed by two tasks: *pick object* and *drop object*. These operators

are similar to the ones defined in the classical blocks world, however, in this case we have limited resources. For example, a table can only have one object on it and not infinite objects (as happens in the classical problem), and also an agent may have an object *occupied* (i.e. picked) so no one else can use it.

In the problem presented in figure 4, the characters have to organize a flat composed of four rooms. Snapshot 4a shows a situation with three agents in where: *Agent_1* wants to put all the books in room number 4; *Agent_2* desires to move each plant to room 3; and finally, *Agent_3* has the two previous objectives as well as stacking the boxes together (this goal is partitioned into three independent subgoals). At first, agents are unaware of the others' goals but they will gather data during the simulation thanks to the dialogues established between them when an interruption occurs. This is the case of 4b, where *Agent_1* wanted to move away the plant that was on top of the books but *Agent_2* takes it first. In this situation, since goals are clearly independent they reject to form a group and continue their actuation but knowing the new information about the current task that has been exchanged. Nevertheless, agents with fully related goals will create groups, this is the result of the competition for the book between *Agent_1* and *Agent_3* in 4c. While they are a group, task coordination is continuously applied. According to this, as *Agent_1* informs his partner about his current task when he moves the book to room 4 in snapshot 4d, *Agent_3* can change to another subgoal and avoid interferences (e.g. pile boxes up instead of trying to pick the book as well).

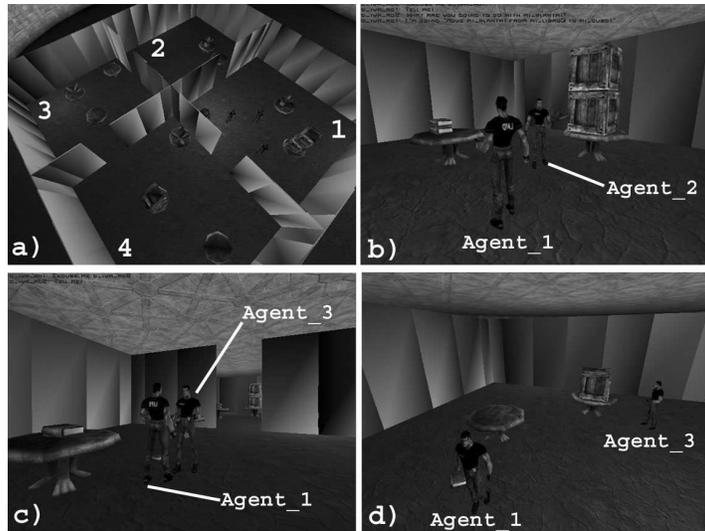


Fig. 4. Snapshots of Team formation and Task Coordination within a 3D IVE.

As stated in [10], cooperation can be measured by three indicators: survival, performance improvement and conflict resolution. According to this, we have

executed some simulations over a 3D blocks world scenario where four agents try to perform a common goal. The survival indicator has no sense in this problem as all agents achieve their goals. We have estimated performance improvement by the amount of executed tasks (TE) and the number of planner invocations (PC). Finally, conflict resolution is estimated by the number of plan interruptions (PI). Table 1 resumes the results obtained.

Agent	PC		PI		TE	
	Simple	Collaborative	Simple	Collaborative	Simple	Collaborative
A	14	13	4	5	10	8
B	9	5	3	2	6	3
C	15	9	4	2	11	7
D	15	6	7	2	8	4

PC = Planner Calls, PI = Plans Interrupted, TE = Task executed

Table 1. 3D blocks world problem results with and without collaboration

Although one single character successfully allocates all the resources in 14 steps, non-communicative individuals need the execution of 35 tasks. The execution of many more tasks than the optimal number needed to complete a goal implies that the simulation will be less realistic. With communicative actors, multi-agent performance can be enhanced as they lower this number down to 22 non-conflictive tasks. Despite the fact that the total number of tasks is not optimal, the final goal is reached faster due to the simultaneous execution. On the other hand, social agents need to invoke their planner less times than the simple ones (see column PC), whose plans are interrupted more frequently (see column PI). To sum up, the new way of planning reduces the number of interferences and produces a coordinated set of tasks whose execution can be overlapped.

5 Conclusions and future work

In this paper we have described a system able to introduce social skills in multi-agent environments. These abilities appear derive from the use of collaborative techniques based on message passing (communication) and the implementation of an extended model of memory that stores the operation of external agents (cooperation). The social model presented is composed by the team formation protocol and the task coordination mechanism that allows 3DIVA to manage dependences, resolve conflicts and enhance the behavioral performance when competing for resources. The first results show the efficiency obtained thanks to the use of goal partitioning and the application of communicative beliefs (*c.beliefs*) to generate coordinated plans. Besides, action interferences are clearly reduced when characters can create groups to achieve a common set of goals.

There is still work in progress in order to evaluate our agent model in more complex scenarios and roles. For example, *joint task execution* (i.e. operators

carried out by more than one agent) needs plenty of communication between the individuals that perform the actions. Currently, dialogs are fully dependent on the planning in the sense that 3D actors interact solely with the objective of gathering information to better fulfill their goals. However, a greater number of dialogues have to be implemented over the Conversational Task Controller so that characters can animate different conversations.

Acknowledgments

We would like to thank all the support and help received from Norman Badler and Jan Allbeck during the stay in the Center for Human Modeling and Simulation in the University of Pennsylvania.

References

1. G. Anastassakis, T. Panayiotopoulos and T. Ritchings. Virtual Agent Societies with the mVITAL Intelligent Agent System. In *Proc. of the 3rd International Workshop on Intelligent Virtual Agents*, 2001.
2. S. Benford, C. Greenhalgh and D. Lloyd. Crowded collaborative virtual environments. In *Proc. of the SIGCHI conference on human factors in computing systems*, 1997.
3. F. Bergenty and A. Ricci. Three approaches to the coordination of multi-agent systems. In *Proc. of ACM Symposim on Applied computing*, pages 367-372, 2002.
4. M. Brenner. A multiagent planning language. In *Proc. of ICAPS'03 Workshop on PDDL*, 2003.
5. T. Bouron and A. Collinot. SAM: a model to design computational social agents. In *Proc. of the 10th European Conference on AI (ECAI'92)*, pages 239-243, 1992.
6. M. Cavazza, F. Charles and S.J. Mead. Planning characters' behaviour in interactive storytelling. *The Journal of Visualization and Computer Animation*, pages 121-131, 2002.
7. P. Cohen and H. Levesque. TeamWork. On *Nous, Special Issue on Cognitive Science and AI*, 1991.
8. K. S. Decker and V. R. Lesser. Designing a Family of Coordination Algorithms. On *Readings in Agents*, Huhns and Singh editors, 1997.
9. C. Elliot, J. Rickel and J. Lester. Lifelike pedagogical agents and affective computing: An exploratory synthesis. *Artificial Intelligence Today* Springer, 1999.
10. J. Ferber. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison Wesley Longman, 1999.
11. C. Geiger and M. Latzel. Prototyping of complex plan based behaviour for 3D actors. In *Proc. of the 4th International Conference on Autonomous Agents*, 2000.
12. J. A. Giampapa and K. Sycara. Team-Oriented Agent Coordination in the RETSINA Multi-Agent System. On *Tech. Report CMU-RI-TR-02-34*, Robotics Institute-Carnegie Mellon University, December 2002.
13. B. Grosz and S. Kraus. Planning and Acting Together. On *AI Magazine*, 1999.
14. M. Lozano, R. Lucia, F. Barber, F. Grimaldo, A. Lucas and A.F. Bisquerra. An Efficient Synthetic Vision System for 3D Multi-Character Systems. In *Proc. of the 4th International Workshop of Intelligent Agents (IVA03)*, September 2003.

15. M. Lozano, F. Grimaldo and F. Barber. Integrating miniMin-HSP agents in a dynamic simulation framework. In *Proc. of the 3rd Hellenic Conference on Artificial Intelligence*, May 2004.
16. T. W. Malone and Kevin Crowston. The interdisciplinary Study of Coordination. *ACM Computing Surveys* , 26(1): 87-119, 1994.
17. C. O'Sullivan et al. Crowd and group simulation with levels of detail for geometry. In *Proc. of the 3rd Irish Workshop on Computer Graphics*, Eurographics Ireland, 2002.
18. C. Reynolds. Steering behaviors for autonomous characters. In *Games developer conference*, 1999.
19. J. Shi, T.J. Smith, J.P. Granieri and N.I. Badler. Smart Avatars in JackMOO. *VR'99: Proceedings of the IEEE Virtual Reality*, 1999.
20. M. Sung, M. Gleicher and S. Chenney. Scalable behavior for crowd simulation. *Computer Graphics Forum*, 23(3), Eurographics 2004.
21. M. Tambe. Towards Flexible TeamWork. On *Journal of Artificial Intelligence Research*, 7: 83-124, 1997.
22. H. Van Dyke Parunak, M. Fleishcher, S. Brueckner and J. Odell. Co-x: Defining what agents do together. In *Workshop on Teamwork and Coalition Formation*, AAMAS, 2002.
23. S. Vosinakis and T. Panayiotopoulos. SimHuman: A platform for real time Virtual Agents planning capabilities. In *Proc. of the 3rd International Workshop on Intelligent Virtual Agents*, 2001.