Animating groups of Socially Intelligent Agents

Francisco Grimaldo, Miguel Lozano, Fernando Barber, and Guillermo Vigueras
Computer Science Department - University of Valencia
Dr. Moliner 50, 46100 Burjassot (Valencia), Spain
{francisco.grimaldo, miguel.lozano, fernando.barber, guillermo.vigueras}@uv.es

Abstract

This paper presents a multi-agent framework oriented to animate groups of synthetic humans that properly balance task-oriented and social behaviors. We mainly focus on the social model designed for BDI-agents to display socially acceptable decisions. This model is based on an auction mechanism used to coordinate the group activities derived from the character's roles. The model also introduces reciprocity relations between the members of a group and allows the agents to include social tasks to produce realistic behavioral animations. Furthermore, a conversational library provides the set of plans to manage social interactions and to animate from simple chats to more complex negotiations. The framework has been successfully tested in a 3D dynamic environment while simulation a virtual university bar, where groups of waiters and customers can interact and finally display complex social behaviors (e.g. task passing, reciprocity, planned meetings...).

1. Introduction

Socially intelligent agents are autonomous problem solvers that have to achieve their goals by interacting with other similarly autonomous entities [13]. Bearing this in mind, multi-agent systems are normally referred to as societies of agents, and provide an elegant and formal framework to design social behaviors for 3D autonomous characters. A major goal in behavioral animation is the construction of an intelligent system able to integrate the different techniques required for the realistic simulation of the behavior of virtual humans. Among them, we can include perception, motion control, goal selection, action execution, communication between agents, their interaction with the environment, etc [14]. Many agent models have been introduced in 3D virtual worlds to increase the behavioral complexity of the actors involved. When designing such agents, the main concern has normally been with the decision-making mechanism, as it is the responsible for the actions that will be finally animated. Traditionally, designers have sought to make their agents rational so that they can act efficiently (i.e. the shorter plan the better). Therefore, social simulations have incorporated group coordination, as self interested agents (i.e. agents devoted to accomplish a set of goals) easily come into conflicts even though their goals are compatible, producing low quality animations. However, virtual humans in 3D scenarios normally appear representing roles (e.g. a virtual guide, a waiter, a customer, etc.) and social aspects should also be considered.

In this paper we introduce a market-based social model that follows the Multi-Agent Resource Allocation approach presented in [13], where agents express their preferences using utility functions. This model coordinates the activities of groups of virtual humans and include social actions in the agent decision-making. In accordance with the main parameter of the model, that is $Sociability$, the agents can autonomously balance their task-oriented behaviors (e.g. waiters serve products to customers) and their social skills (e.g. negotiation to obtain these products, assumption of external actions/favours or animation of simple chats). The dynamics of social interactions is inspired by the theory of Piaget [18], over which we have implemented reciprocal task exchanges between agents.

The structure of the paper is as follows. In section 2 we describe briefly some previous literature on the field. In section 3 we present the social multi-agent framework and sections 4 and 5 review the main components of the behavioral multi-agent system implemented. Section 6 describes an illustrative example modelled to test our framework and also comments some of the plots extracted from the simulated characters. Finally, section 7 summarizes the theoretical results extracted from the previous experiments and analyzes them.

2. Related work

Many research has been done in behavioral animation of virtual agents during the last few years [22, 16, 2, 10, 15, 17, 8]. A good introduction to the field can be found in

[1]. The pioneer work of Dimitri Terzopoulos [25] showed how to design a framework to animate natural ecosystems with minimal input from the animator. He simulated *Artificial fishes* in virtual underwater worlds. However, human behavior is clearly different and more complex to emulate. Possibly, the more relevant works in this field came from Thalmann's group [7, 21, 24]. The goal of these previous works was to design agents with a high degree of autonomy without loosing control. Their agents are an extension of the BDI architecture described in [20], and they include internal states as emotions, reliability, trust and others.

Behavioral animation has also been tackled from the field of coordinated multi-agent systems. For example in Generalized Partial Global Planning (GPGP) [6], agents merge the meta-plans describing their operational procedures and figure out the better action in order to maximize the global utility. Another example is Multi-agent Planning Language (MAPL) [5], which assigns the control over each resource to a unique agent and uses speech-acts to synchronize planned tasks. Collaboration is supported in the RETSINA system [9] thanks to the use of communicative acts that synchronize tasks and occasionally manage conflicts. Team formation and task coordination for heuristic search planning characters is presented in [12] to adapt better to the dynamism of shared environments. MAS-SOC [3] aims at creating a platform for multi-agent based social simulations, which is similar to our purposes. In this context, work is ongoing in order to incorporate social-reasoning mechanisms based on exchange values [23].

Although the results obtained by the previous approaches show realistic simulations of many task-oriented behaviors, autonomous characters should also display social behaviors (e.g. interchanging information with their partners or grouping and chatting with their friends). This kind of socially intelligent animation agents is required in many complex simulation environments: military/civil simulations, social pedestrians in virtual cities, games, and probably very soon in large scale distributed environments such as Second Life. The multi-agent framework presented aims to easily design groups of socially intelligent animation agents. These agents manage autonomously their rationality and sociability skills at interactive frame rates, and they are designed to display high quality behavioral animations in 3D dynamic worlds.

3. Social multi-agent framework

The social multi-agent framework presented in figure 1 has been developed over Jason [4], which allows the definition of BDI agents using an extended version of AgentSpeak(L) [19]. The animation system (virtual characters, motion tables, etc) is located at the 3D engine, which can run separately. The agent decision-making is defined in the

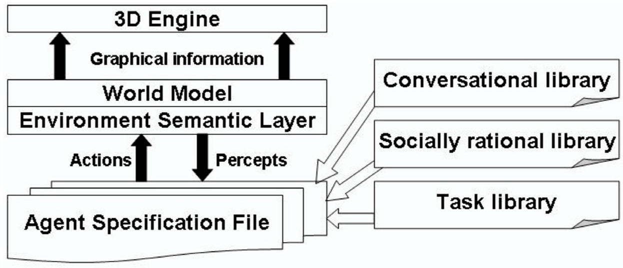

Figure 1. Multi-agent animation framework.

Agent Specification File. This file contains the initial beliefs as well as the set of plans that make up the agent's finite state machine. The *Task Library* contains the set of plans that sequence the actions needed to animate a task. For instance, a virtual waiter serving a coffee will go to the coffee machine to get the coffee and will give it to the customer afterwards. Here, modularity is guaranteed since the *Task library* can be changed depending on the environment and the roles being simulated.

As stated above, only rational behaviors are not enough to simulate agent societies, therefore, we have included a *Social library* to manage different types of social situations. This library is based on an auction model that uses the social welfare concepts to avoid conflicts and allow the agents to behave in a coordinated way. The *Social library* also incorporates a reciprocity model in order to promote social interactions among the members of a group. The *Conversational library* contains the set of plans that handle the animation of the interactions between agents (e.g. ask someone a favor, planned meetings, chats between friends...). Finally, the environment is handled by a *Semantic Layer* [11] which acts as an interface between the agent and the world. It is in charge of perceiving the state of the world and executing the actions requested by the agents, while ensuring the consistency of the *World Model*.

4. Social library

The simulation of worlds inhabited by interactive virtual actors normally involves facing a set of problems related to the use of shared limited resources and the need to animate pure social behaviors. Both types of problems are managed by the Social library by using a Multi-agent Resource Allocation approach [13]. This library allows any agent to auction tasks in order to reallocate them so that the global social welfare can be increased. Tasks are exchanged between agents using a first-price sealed-bid (FPSB) auction model where the agents express their preferences using performance and social utility functions.

The performance utility function $U^i_{perf}(\langle i \leftarrow t \rangle)$ of a bidder agent i reflects the efficiency achieved when the task t is allocated to the agent i ($\langle i \leftarrow t\rangle$). There can be many reasons for an agent to be more efficient: it may perform the task faster than others because of his know-how or it may be using a resource that allows several tasks to be performed simultaneously – e.g. a coffee machine in a virtual bar can be used by a waiter to make more than one coffee at the same time. The utility function has to favor the performance of the agents, but high performances can also be unrealistic for the animation of artificial human societies. For example, if all agents work as much as they can, they will display unethical or robotic behaviors. Furthermore, agents should also show pure social behaviors to animate the normal relations between the members of a society.

Whereas the performance utility function modelled the interest of an agent to exchange a task from an efficiency point of view, we introduce two additional social utilities to represent the social interest in exchanging a task. The aim of social utilities is to promote task allocations that lead the agents to perform social interactions with other agents – e.g. planned meetings with their friends. Therefore, these functions take into account the social relations established between the agents and defined in the ontology to compute the value that expresses their social preferences. Negotiation of long sequences of actions is not very interesting for interactive characters, as plans are likely to be thwarted due to the dynamism of the environment and to other unpredictable events. Thus, we define the following social utility functions:

- Internal social utility ($U^i_{int}(\langle i \leftarrow t, j \leftarrow t_{next}\rangle)$): is the utility that a bidder agent i assigns to a situation where i commits to do the auctioned task t so that the auctioneer agent j can execute his next task t_{next}.
- External social utility ($U^i_{ext}(\langle j \leftarrow t\rangle)$): is the utility that a bidder agent i assigns to a situation where the auctioneer agent j executes the auctioned task t while i continues with his current action.

The winner determination problem has two possible candidates coming from performance and sociability. In equation 1 the welfare of a society is related to performance, hence, the winner of an auction will be the agent that bid the maximum performance utility. On the other hand, equation 2 defines the social winner based on the maximum social utility received to pass the task to a bidder ($U^*_{int}(t)$) and the maximum social utility given by all bidders to the situation where the task is not exchanged but performed by the auctioneer j ($U^*_{ext}(t)$).

$$winner_{perf}(t) = \left\{ k \epsilon Agents | U^i_{perf}(t) = \max_{i \epsilon Agents}\{U^i_{perf}(\langle i \leftarrow t\rangle)\} \right\} \tag{1}$$

$$winner_{soc}(t) = \begin{cases} j & U^*_{ext}(t) >= U^*_{int}(t) \\ i & U^*_{ext}(t) < U^*_{int}(t) \wedge U^i_{int}(t) = U^*_{int}(t) \end{cases} \tag{2}$$

To balance task exchange, social utilities are weighted with a reciprocity matrix (see equations 3 and 4). We define the reciprocity factor w_{ij} for two agents i and j, as the ratio between the number of favors – i.e.tasks – that j has made to i (see equation 5).

$$U^*_{int}(t) = \max_{i \epsilon Agents}\{U^i_{int}(\langle i \leftarrow t, j \leftarrow t_{next}\rangle) * w_{ji}\} \tag{3}$$

$$U^*_{ext}(t) = \max_{i \epsilon Agents}\{U^i_{ext}(\langle j \leftarrow t\rangle) * w_{ij}\} \tag{4}$$

$$w_{ij} = Favours_{ji}/Favours_{ij} \tag{5}$$

At this point, agents can decide whether to adopt this kind of social allocations or to be only rational as explained previously. They choose between them in accordance with their $Sociability$ factor, which is the probability to select the social winner instead of the rational winner. $Sociability$ can be adjusted in the range [0,1] to model intermediate behaviors between efficiency and total reciprocity. This can provide great flexibility when animating characters, since Sociability can be dynamically changed thus producing different behaviors depending on the world state.

5. Conversational library

The auction-based model presented above represents a useful technique to obtain group coordination through social commitments. Apart from the auction carried out internally, it is clear that the agreements should be animated according to the situation being simulated. To manage the animation of social commitments, we have developed a set of plans that uses several conversations to display the agreement reached.

Table 1 summarizes the actor plans depending on the winner of an auction. On the upper half of the table, when a task (t) is auctioned, a bidder (i) will be the performance winner only if he can do it faster than the others. When an agent i wins an auction, the auctioneer (j) can approach i and animate the agreement starting a dialogue. For instance, he can shout at the winner: *”Please, make a cup of coffee for me!”*; where a positive answer is necessary to be consistent with the agreement previously reached. When the auctioneer is also the winner no social agreement is normally produced. However, these situations can be also useful to animate some failures in social commitments, which will add more variability to the character’s behavior (e.g. *”Sorry I can not do it now because I have a lot of work to do”*).

The lower half of the table shows the actions performed when the winner is a social winner, that is, an agent that

Performance winner			
W	Precondition	Action	Response
j	None	None	None
	near(i,Res) not(near(i,j))	shout(j,i,make(t))	shout(i,j,no)
	near(i,Res) near(i,j)	tell(j,i,make(t))	tell(i,j,no)
i	noise(high) noise(high)	approach(i) tell(j,i,make(t))	tell(i,j,yes)
	noise(low)	shout(j,i,make(t))	shout(i,j,yes)
Social winner			
W	Precondition	Action	Response
j	None	plan_meeting(j,i) chat(j,i)	chat(i,j)
i	noise(high)	approach(i) tell(j,i,make(t)) plan_meeting(j,i) chat(j,i)	tell(i,j,yes) chat(i,j)
	noise(low)	shout(j,i,make(t)) plan_meeting(j,i) chat(j,i)	shout(i,j,yes) chat(i,j)

Table 1. Conversations to animate performance and social agreements.

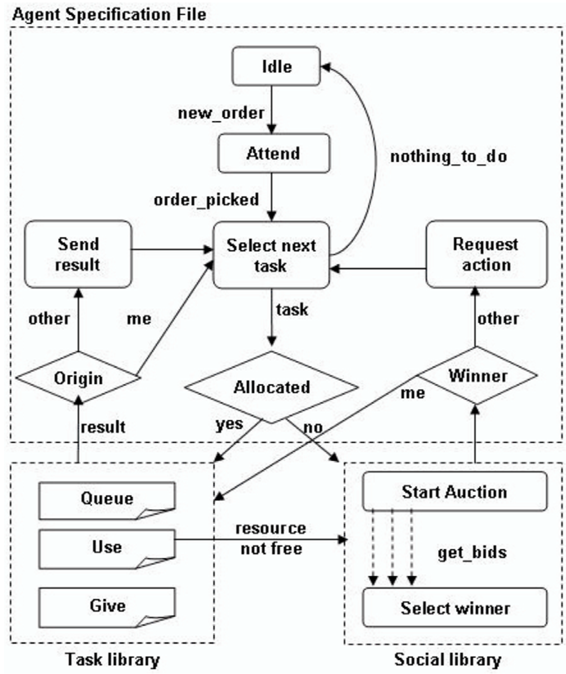

Figure 2. Waiter specification file.

obtains the best social reward in accordance with the utility values received from its friends. In this case, a conversation always occurs and the auctioneer can *approach and tell* or *shout at* its partner the commitment made (e.g. *"Go to the counter, please, I want to chat with you!"*). Planning social meetings is a mechanism oriented to animate short chats between two actors, therefore, they will start chatting when they are close enough and a meeting was previously planned between them. In both cases, the animated actions have preconditions, such as the level of noise in the environment or the distance between actors or resources (*near(x,y)*). The noise level can be easily derived from the whole number of actors in the bar and those who are near the winner.

6. Application example

In order to test the presented social multi-agent framework, we have created a virtual university bar where waiters take orders placed by customers. The typical objects in a bar (e.g. a juice machine) behave like resources that have an associated time of use to supply their products (e.g. 2 minutes to obtain an orange juice) and they can only be occupied by one agent at a time. Waiters are governed by the finite state machine shown in figure 2, where orders are served basically in two steps: first, using the corresponding resource (e.g. the grill to produce a sandwich) and second, giving the product to the customer. Tasks are always auctioned before their execution in order to find good social allocations.

Equations 6 and 7 define the utility values returned by the performance utility function for these tasks. This function aims at maximizing the number of parallel tasks being performed and represents the waiters' willingness to serve orders as fast as possible. Social behaviors defined for a waiter are oriented to animate chats between his partners. Therefore, waiters implement the internal and external social utility functions detailed in equations 8 and 9, where $Near$ computes the distance between the agents while they are executing a pair of tasks. These functions evaluate social interest as the chance to meet a partner in the near future (i.e. a planned meeting).

$$U^i_{perf}(\langle i \leftarrow \text{'Use'}\rangle) = \begin{cases} 1 & \text{if } [(i = Auctioneer) \wedge IsFree(Resource)] \vee \\ & [IsUsing(i, Resource) \wedge \\ & not(IsComplete(Resource))] \\ 0 & \text{Otherwise} \end{cases} \tag{6}$$

$$U^i_{perf}(\langle i \leftarrow \text{'Give'}\rangle) = \begin{cases} 1 & \text{if } [(i = Auctioneer) \wedge nextAction = NULL] \vee \\ & [currentTask = \text{'Give'} \wedge not(handsBusy < 2)] \\ 0 & \text{Otherwise} \end{cases} \tag{7}$$

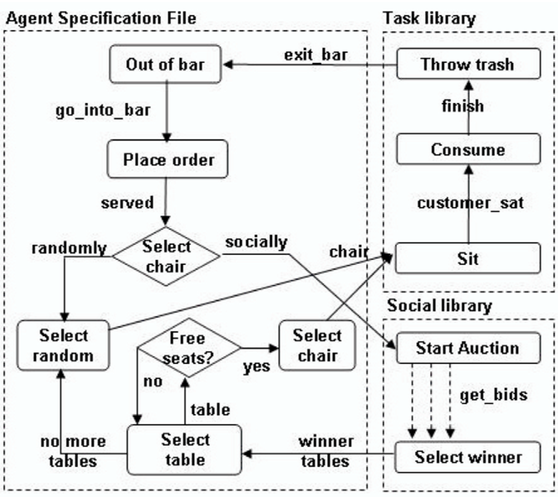

Figure 3. Customer specification file.

$$U^i_{int}(\langle i \leftarrow t, j \leftarrow t_{next}\rangle) = \begin{cases} 1 & \text{if } IsFriend(i,j) \wedge Near(t, t_{next}) \wedge \\ & ExecTime(t_{next}) > RemainTime(currentTask) \\ 0 & \text{Otherwise} \end{cases} \tag{8}$$

$$U^i_{ext}(\langle j \leftarrow t\rangle) = \begin{cases} 1 & \text{if } IsFriend(i,j) \wedge Near(currentTask, t) \\ 0 & \text{Otherwise} \end{cases} \tag{9}$$

On the other hand, customers place orders and consume them when served. Now, we are not interested in improving customer performance but in animating interactions between the members of a social group. Thus, we have implemented three classes of customers that use auctions to solve the problem of where to sit. The finite state machine in figure 3 governs the actuation of customers. Depending on his or her sociability factor, a customer randomly selects a chair or starts an auction to decide where to sit and consume. This auction is received by all customers in the bar, which use the external social utility function defined in equation 10 to promote meetings with others of the same class. We define the performance and the internal social utility functions as 0 since task passing is not possible in this case (i.e. no-one can sit instead of another customer). Finally, when a social meeting emerges, both waiters and customers use the plans in the Conversational Library to sequence the speech-acts needed to animate commitments, greetings or simple conversations.

$$U^i_{ext}(\langle j \leftarrow \text{'Sit'}\rangle) = \begin{cases} 1 & \text{if } IsSameClass(i,j) \wedge \\ & IsConsuming(i, auctionedTable) \\ 0 & \text{Otherwise} \end{cases} \tag{10}$$

The $Sociability$ factor of both customers and waiters

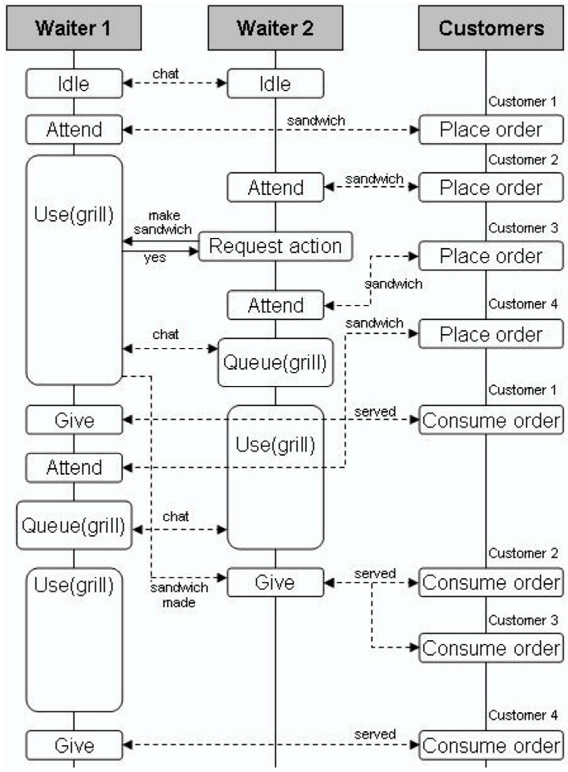

Figure 4. Waiter specification file.

can be adjusted in order to balance performance and sociability. For example, the plot depicted in figure 4 corresponds to a simulation in the virtual bar where 2 waiters with $Sociability = 0.6$ take the orders placed by 4 customers. In this animation, all customers want to have a sandwich but the grill, that is needed to prepare them, is shared by the waiters. Although the grill cannot be used by more than one agent at a time, we have provided this object with the capability of making 4 sandwiches simultaneously.

Waiters in figure 4 can chat while no customer is waiting to be served. Then, waiters serve customers in order of appearance using a standard dialogue as seen in snapshot 5a. When a resource that is needed to perform a task (e.g. the grill to make a sandwich) is already in use, a waiter has two possibilities: (a) try to pass the task and continue serving; or (b) animate social chats (i.e. casual conversations) while waiting for the resource to be free. The decision is made probabilistically depending on the $Sociability$ factor of the agent. For example, when *Waiter 2* takes the order from *Customer 2*, he realizes that *Waiter 1* is also preparing a sandwich, hence, he asks him to make another sandwich (see snapshot 5b). However, this situation is solved in a different way when the same waiter attends *Customer 3*. In this case, he decides to wait for the shared resource (*Queue(grill)*) to be able to chat with his partner. Once

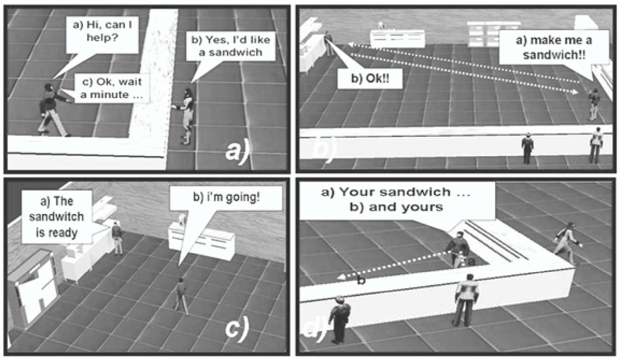

Figure 5. Animating interaction situations: (a) attend situation, (b) action request situation, (c) inform result situation and (d) serving situation.

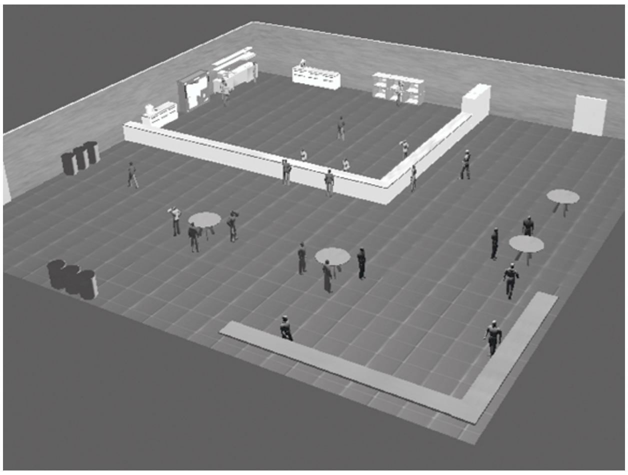

Figure 6. 3D virtual university bar example.

a reallocated task has been completed, the waiter that has performed it informs the applicant agent about its result. For instance, in snapshot 5c, *Waiter1* tells *Waiter 2* that the sandwich is ready to be served. Similarly to using a resource for solving several tasks simultaneously, the waiters can use their two hands to carry more than one product at the same time. In snapshot 5d, *Waiter 2* carries two sandwiches that are given to *Customer 2* and *Customer 3*.

Finally, the effects of sociability over customers are shown in figure 6. Here, 7 waiters serve 16 customers belonging to three social groups (represented as different avatars in the scene). We have defined customer $Sociability$ as 1, thus, when served, they try to sit with other customers of their same group. See how avatars of the same class sit at the same table whenever it is possible.

7. Results

To illustrate the effects of the social techniques previously applied we have animated the virtual university bar example with 10 waiters serving 100 customers both with different sociability factors [1]. We measure the efficiency of a group of waiters as the ratio between the optimal simulation time and the real simulation time (see equation 11). $Throughput$ is an indicator in the range [0,1] that estimates how close a simulation is to the ideal situation in which the workload can be distributed among the agents and no collisions arise.

$$Throughput = \frac{T^*_{sim}}{T_{sim}} = \frac{N_{tasks} * \overline{T_{task}} / N_{agents}}{T_{sim}} \quad (11)$$

Figure 7 shows the $Throughput$ obtained by different types of waiters versus self-interested agents (i.e. agents with no social mechanisms included). Self-interested agents collide as they compete for the use of the shared resources and these collisions produce high waiting times as the number of agents grows. We can enhance this low performance with elitist agents ($Sociability = 0$) which exchange tasks with others that can carry them out in parallel thus reducing the waiting times for resources. Nevertheless, they produce unrealistic outcomes since they are continuously working if they have the chance, leaving aside their social relationships (e.g. chats between friends). The $Sociability$ factor can be used to balance rationality and sociability. Therefore, the $Throughput$ for the sort of animations we are pursuing should be placed somewhere in between elitist and fully reciprocal social agents ($Sociability = 1$).

$Throughput$ is an estimator for the behavioral performance but, despite being a basic requirement when simulating groups of virtual characters, it is not the only criterion to evaluate when we try to create high quality simulations. Therefore, we have defined another estimator that takes into account the amount of time that the designer of the simulation wants the agents to spend on their social interactions. According to this, we define the following simulation estimator:

$$Animation = \frac{T^*_{sim} + T_{social}}{T_{sim}} \quad (12)$$

, where T_{social} represents the time devoted to chatting and to animating social agreements among friends. In our virtual bar we have chosen T_{social} as the 35% of T^*_{sim}. Figure 8 shows the animation values for 10 reciprocal social waiters with 4 degrees of friendship: all friends, 75% of the agents are friends, half of the agents are friends and only

[1]Visit http://www.uv.es/ agentes to download some 3D animation videos of the virtual university bar.

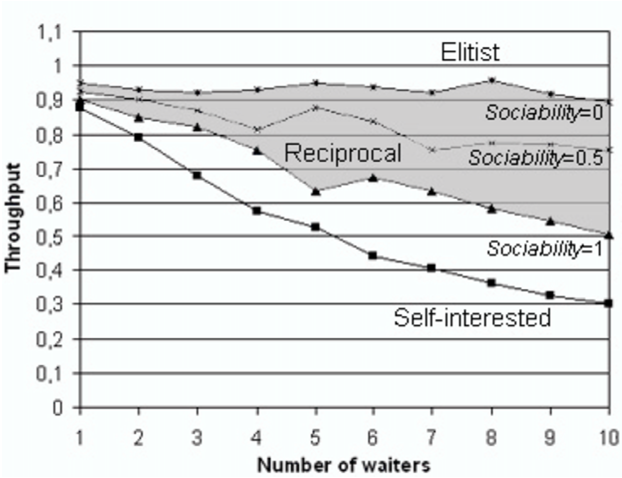

Figure 7. Throughput results for waiters.

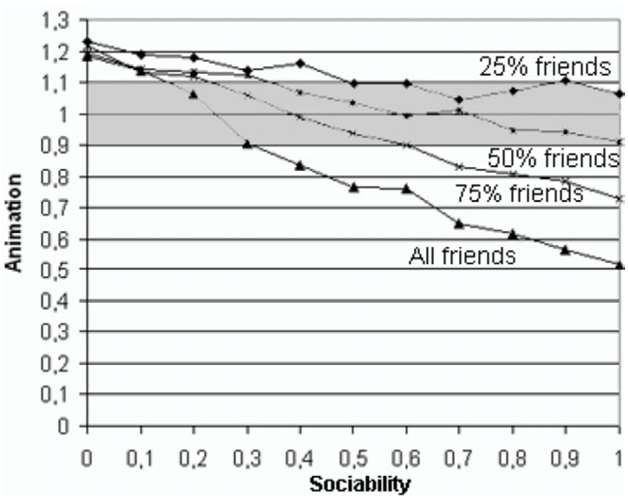

Figure 8. Animation results for waiters.

25% of the agents are friends. As we have already mentioned, low values of $Sociability$ produce low quality simulations since the values obtained for the animation function are greater than the reference value ($Animation = 1$). On the other hand, high values of $Sociability$ also lead to low quality simulations, especially when the degree of friendship is high. In these cases, the number of social conversations being animated is too high to be realistic and animation is far from the reference value. The animation function can be used to extract the adequate range of values for the $Sociability$ factor, depending on the situation being simulated. For example, in our virtual bar we consider as good quality animations those which fall inside $\pm 10\%$ of the reference value (see shared zone in figure 8). Hence, when all the waiters are friends, good animations emerge when $Sociability \in [0.1, 0.3]$.

Agent	T_{wait}	T_{use}	T_{give}	$Balance$
1	0	32	19	-6
2	3	4	26	-3
3	14	52	1	28
4	3	16	28	-3
5	0	7	30	-16
6	3	37	17	-1
7	0	67	4	21
8	0	45	17	1
9	7	5	23	-11
10	1	6	41	-10

Table 2. Time distribution for 10 waiters with $Sociability = 0$ (time values are in seconds).

Finally, tables 2 and 3 show the amount of time devoted to execute each type of task in executions with 10 elitist waiters ($Sociability = 0$) and 10 fully reciprocal social waiters ($Sociability = 1$). The irregular values in the columns T_{use} and T_{give} in table 2 demonstrate how some agents have specialized in certain tasks. For instance, agents 2, 5, 9 and 10 spend most of their time giving products to the customers while agents 3 and 7 are mainly devoted to using the resources of the bar (e.g. coffee machine, etc). Although specialization is a desirable outcome in many multi-agent systems, egalitarian human societies need also to balance the workload assigned to each agent. In table 3, fully reciprocal social waiters achieve equilibrium between the time they are giving products and the time they are using the resources (see columns T_{use} and T_{give}). Furthermore, the reciprocity factor balances the number of favors exchanged among the agents (see $Balance$ column). The payoff is the increase in the waiting times, since social agents will sometimes prefer to meet friends in a resource than to reallocate the task (compare columns T_{wait} in tables 2 and 3).

8. Conclusions

The animation of groups of intelligent characters is a current research topic with a great number of behavioral problems to be tackled. We aim at incorporating human style social reasoning in character animation. Therefore, this paper presents a technique to properly balance social with task-oriented plans in order to produce realistic social animations. The multi-agent animation framework presented allows for the definition of different types of social agents: from elitist agents (that only use their interactions to increase the global performance of the group) to fully reciprocal agents. These latter agents extend the theory of social welfare with a reciprocity model that allows the agents to control the emergence of unethical behaviors and promote

Agent	T_{wait}	T_{use}	T_{give}	$Balance$
1	16	69	34	-2
2	18	58	24	-2
3	41	45	16	0
4	48	60	27	3
5	34	58	12	-1
6	48	64	14	-2
7	18	48	24	1
8	33	45	24	4
9	46	36	21	0
10	27	56	20	-1

Table 3. Time distribution for 10 waiters with $Sociability = 1$ (time values are in seconds).

social interactions among the members of a group. Work is ongoing to provide the agents with mechanisms to self-regulate their $Sociability$ factor depending on their social relations and on their previous intervention. Thus, agents will be able to dynamically adjust to the situation in order to stay within the boundaries of good quality animations.

Acknowledgements

This work has been jointly supported by the Spanish MEC and European Commission FEDER funds under grants Consolider Ingenio-2010 CSD2006-00046 and TIN2006-15516-C04-04..

References

[1] N. Badler, C. Philips, and B. Webber. *Simulating Humans: Computer Graphics, Animation and Control.* Oxford University Press, 1993.

[2] B. Blumberg and T. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In *Proc. of SIGGRAPH'95*, pages 47–54. ACM, 1995.

[3] R. H. Bordini, A. C. da Rocha, J. F. Hübner, A. F. Moreira, F. Y. Okuyama, and R. Vieira. MAS-SOC: a social simulation platform based on agent-oriented programming. *Journal of Artificial Societies and Social Simulation*, 8(3), 2005.

[4] R. H. Bordini and J. F. Hübner. Jason. Available at http://jason.sourceforge.net/, March 2007.

[5] M. Brenner. A multi-agent planning language. In *Proc. of ICAPS'03 Workshop on PDDL*, 2003.

[6] K. S. Decker and V. R. Lesser. *Readings in Agents*, chapter Designing a family of coordination algorithms. 1997.

[7] N. Farenc, R. Boulic, and D. Thalmann. An informed environment dedicated to the simulation of virtual humans in urban context. In P. Brunet and R. Scopigno, editors, *Proc. of EUROGRAPHICS"99*, volume 18(3), pages 309–318. The Eurographics Association and Blackwell Publishers, 1999.

[8] J. Funge, X. Tu, and D. Terzopoulos. Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In *Proc. of SIGGRAPH'99*, pages 29–38. ACM, 1999.

[9] J. A. Giampapa and K. Sycara. Team-oriented agent coordination in the RETSINA multi-agent system. Tech. report CMU-RI-TR-02-34, Robotics Institute-Carnegie Mellon University, 2002.

[10] J. Granieri, W. Becket, B. Reich, J. Crabtree, and N. Badler. Behavioral control for real-time simulated human agents. In *Symposium on Interactive 3D graphics*, pages 173–180. ACM, 1995.

[11] F. Grimaldo, F. Barber, and M. Lozano. An ontology-based approach for IVE+VA. In *Proc. of IVEVA'06: Intelligent Virtual Environments and Virtual Agents*, 2006.

[12] F. Grimaldo, M. Lozano, F. Barber., and J. Orduña. Integrating social skills in task-oriented 3D IVA. In S.-V. LNAI, editor, *Proc. of IVA'05: The 5th International Working Conference on Intelligent Virtual Agents*, 2005.

[13] L. M. Hogg and N. Jennings. Socially intelligent reasoning for autonomous agents. *IEEE Transactions on System Man and Cybernetics*, 31(5):381–393, 2001.

[14] A. Iglesias and F. Luengo. Intelligent agents in virtual worlds. In *Proc. of CW04: International Conference on Cyberworlds*, pages 62–69. IEEE Computer Society, 2004.

[15] P. Maes, T. Darrell, B. Blumberg, and A. Pentland. The ALIVE system: full-body interaction with autonomous agents. In *Proc. of Computer Animation'95*, pages 11–18. IEEE Computer Society Press, 1995.

[16] M. McKenna, S. Pieper, and D. Zelter. Control of a virtual agent: the roach. In *Proc. of SIGGRAPH'90*, volume 24 (2), pages 165–174. Computer Graphics, 1990.

[17] K. Perlin and A. Goldberg. IMPROV: a system for scripting interactive agents in virtual worlds. In *Proc. of SIGGRAPH'96*, pages 205–216. ACM, 1996.

[18] J. Piaget. *Sociological studies*. Routledge, 1995.

[19] A. S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In S. Verlag, editor, *Proc. of MAAMAW'96*, number 1038 in LNAI, pages 42–55, 1996.

[20] A. S. Rao and M. P. Georgeff. Modeling rational agents within a bdi-architecture. In *Proc. of KR'91: The 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 473–484. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1991.

[21] S. Raupp and D. Thalmann. Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):152–164, 2001.

[22] C. Reynolds. Flocks, herds and schools: a distributed behavioral model. *Computer Graphics*, 21 (4):25–34, 1987.

[23] M. Ribeiro, A. C. da Rocha, and R. H. Bordini. A system of exchange values to support social interactions in artificial societies. In ACM, editor, *Proc. of AAMAS'03: Autonomous Agents and Multi-agent Systems*, 2003.

[24] D. Thalmann and J. Monzani. Behavioural animation of virtual humans: What kind of law and rules? In *Proceedings of Computer Animation*, pages 154–163. IEEE Computer Society Press, 2002.

[25] X. Tu and D. Terzopoulos. Artificial fishes: physics, locomotion, perception, behavior. In *Proc. of SIGGRAPH'94*, pages 43–50. ACM, 1994.