

En **MIGUEL LOZANO IBÁÑEZ**, professor contractat doctor en l'àrea de Llenguatges i Sistemes Informàtics de la Universitat de València,

CERTIFICA que la subsegüent memòria

“Integració d’habilitats socials en l’animació comportamental d’actors sintètics”

ha sigut realitzada sota la seua direcció, al Departament d’Informàtica de la Universitat de València, per **Francisco Grimaldo Moreno** i constitueix la seua tesi per a optar al grau de Doctor en Enginyeria Informàtica.

Perquè així conste, en compliment de la legislació vigent, presenta aquesta declaració a l’Escola Tècnica Superior d’Enginyeria (ETSE) de la Universitat de València, a 3 de març de 2008.

EL DIRECTOR

Miguel Lozano Ibáñez

Aquesta tesi ha estat finançada de forma conjunta pel Ministeri d'Educació i Ciència de l'Estat Espanyol i pels fons FEDER de la Comissió Europea sota els projectes TIN2006-15516-C04-04 i Consolider Ingenio-2010 CSD2006-00046, respectivament.





UNIVERSITAT DE VALÈNCIA

Departament d'Informàtica

TESI DOCTORAL

**INTEGRACIÓ D'HABILITATS
SOCIALS EN L'ANIMACIÓ
COMPORTAMENTAL D'ACTORS
SINTÈTICS**

Presentat per:

D. Francisco Grimaldo Moreno

València, maig de 2008

Treball dirigit per

Dr. D. Miguel Lozano Ibáñez

A la meua família

*...sols En nós dos, aMor se manifesta,
e nós vivents, no Li fallirà casA...*

Ausiàs March

Agraïments

Al llarg del procés kafkià que suposa realitzar una tesi doctoral, l'autor necessita la col·laboració de tot un seguit de persones per poder arribar a l'ansiejat final. Aquestes breus paraules queden lluny de l'agraïment que realment mereixen totes aquelles persones, sense l'ajuda de les quals, aquest treball hagués estat directament impossible.

En primer lloc, vull agrair profusament el Dr. Miguel Lozano per l'esforç que m'ha abocat al llarg de tots aquests anys. Del seu consell, sempre d'amic, no només floreix aquesta tesi sinó també una forma honorable d'entendre la investigació i la vida que, per a mi, és tot un exemple a seguir. Regracie així mateix la seua família, Loles i el nouvingut Lucas, per obrir-me les portes de sa casa i per permetre'm furar tant de temps del dia a dia de Miguel. En segon lloc, desitge agrair l'ajut incondicional de Fernando Barber com a company fidel de *viatge*. Les seues bones maneres esquitxen cadascuna de les aportacions d'aquesta tesi, la qual sense elles, mancaria del formalisme i funcionalitat escaient. Vull reconèixer especialment la confiança dipositada pel Dr. Juan Manuel Orduña vers la recerca desenvolupada en aquesta tesi i en els membres que la duen a terme. Gràcies pel recolzament econòmic i per convidar-nos a formar part d'un grup d'investigació amb tantes ambicions. Agraïsc l'afany de servei de Vicente Cavero, ja que he tingut el privilegi de comptar amb ell davant qualsevol problema tècnic. De Guillermo Vígueras són les últimes animacions 3D i la meua gratitud pel seu bon treball.

Envie un afectuós agraïment al Dr. Joan Pelechano i al Dr. Jesús Albert per tota la dedicació que desinteressadament han tingut amb mi des de sempre. També a la resta de companys del Departament d'Informàtica i de l'Institut de Robòtica de la Universitat de València per la companyonia (especialment a Rafa Lucía pel seu suport en els moments inicials). Vull expressar la meua gratitud cap a la Dra. Núria Pelechano i el Dr. Norman I. Badler per totes les atencions rebudes durant l'estada realitzada al *Centre for Human Modeling and Simulation* (University of Pennsylvania - EUA).

Tocant a la vessant lingüística d'aquesta tesi, cal agrair l'altruisme de dos grans amics: Óscar Díaz i Ivan Carbonell. Sense el primer, cap publicació en llengua anglesa haguera assolit el nivell necessari. Del mestre Carbonell agraïsc les seues correccions i la seua devoció dominical per mostrar-me un camí que m'està donant tant.

Finalment, però no menys important, agraïsc tot el temps que la meua família ha sacrificat per mi; sense ells, de ben segur que no hagués arribat a fer aquesta tesi. D'Emi vénen tots els ànims que he necessitat per concloure aquest treball i el desig de no despertar mai del nostre somni, que a poc a poc, es va fent realitat...

Resum

A la primeria del segle XXI, el món físic perd distància en la cursa per la realitat que l'enfronta als mons virtuals. Dia rere dia, els mons simulats arrabassen un pam de terreny i inclouen nous camps d'aplicació que no sols complementen els serveis que es troben al món real, sinó que poden arribar a substituir-los en un futur possiblement no molt llunyà.

La simulació de mons virtuals habitats per personatges 3D és un problema complex que necessita la integració d'àrees diverses com ara els gràfics per computador i la intel·ligència artificial. De la primera, hom pot obtenir la credibilitat gràfica de l'escena, de la segona, l'autonomia i la interactivitat pròpia de l'animació comportamental dels així coneguts com 3DIVA (*3D Intelligent Virtual Agents*). Normalment, la interactivitat dels 3DIVA s'adreça a l'entorn (p. ex. la capacitat d'executar tasques sobre els objectes 3D) o a l'usuari (p. ex. la capacitat de comunicació amb un usuari humà, la reproducció d'emocions o actituds expressives, etc.). Amb tot, cal fer una passa endavant quan es treballa amb societats artificials com les que apareixen, per exemple, dins dels mons virtuals dels jocs d'entreteniment, dels simuladors per a l'aprenentatge de tasques civils o militars, o de ciberespais com el famós SecondLife. Malgrat el realisme gràfic assolit per les tècniques d'animació, és necessari aprofundir en la interacció entre aquests actors sintètics autònoms per tal de fer-los evolucionar cap a una intel·ligència social que estiga més a prop de la sociabilitat humana.

L'objectiu principal d'aquesta tesi és la integració d'habilitats socials en l'animació comportamental d'humanoides autònoms situats dins de mons virtuals 3D. Per tant, en primer lloc fem una revisió dels mons virtuals habitats, així com dels diferents mecanismes d'animació comportamental emprats normalment en actors sintètics. En segon lloc, repassem els models de sociabilitat més comuns que hom pot trobar dins l'estat de l'art dels agents socialment intel·ligents.

La presa de decisions que han de dur a terme els actors sintètics amb capacitats socials per interaccionar amb la resta d'agents comporta processos cognitius complexos i requereix un coneixement abstracte dels elements de l'entorn. Darrerament, hom ha proposat la **inclusió d'informació semàntica dins dels entorns virtuals**, coneguts sota el terme anglosaxó de *Semantic Virtual Environments* (SVE). D'acord amb açò, en aquest treball proposem l'ús d'**ontologies** per aconseguir tres objectius: millorar la sensorització

d'escenes complexes, definir operatives generals que els agents puguen reutilitzar en situacions diverses i definir les relacions socials establides entre els membres d'una societat artificial.

La natura proactiva dels actors sintètics ha fet que les tècniques d'animació comportamental més comunes, basades en la selecció dinàmica d'accions, es divideixen fonamentalment en dos grups: la planificació de tasques (p. ex. les basades en *STRIPS*) i els sistemes basats en regles (p. ex. els models *Belief-Desire-Intention*). Les aportacions principals d'aquesta tesi es troben en la inclusió d'habilitats socials en ambdós paradigmes, ja que els considerem aproximacions vàlides extensament emprades i moltes vegades complementàries. Primerament, presentem com aconseguir **comportaments col·laboratius en grups d'agents basats en planificadors heurístics**. Fem servir la comunicació d'accions per a la coordinació entre les activitats dels humanoides i la comunicació d'objectius per a la cooperació entre els personatges 3D. Llavors, hem desenvolupat un mecanisme de sospesament heurístic que permet assignar un pes a cada objectiu, de manera que reflectisca la seua importància social. No obstant això, l'eficiència no és l'única manera d'expressar sociabilitat. Pel contrari, la presa de decisions humana sovint té en compte més d'un punt de vista i no tots persegueixen la consecució ràpida de les tasques encarregades. Per tal causa, presentem **MADeM (Multi-modal Agent Decision Making)**, un procés de presa de decisions de tipus social que pot ser emprat per agents BDI per tal de prendre decisions socialment acceptables. MADeM és capaç d'avaluar diferents solucions a un problema mitjançant tècniques socials basades en l'economia de mercats les quals fan servir les subhastes com a mecanisme perquè els agents expressen les seues preferències.

Finalment, exposem un conjunt d'experiments que s'han realitzat per a comprovar l'impacte de les distintes tècniques socials desenvolupades. Per tal d'avaluar objectivament els resultats, no sols s'analitzen les traces de comportament dels personatges 3D sinó que es mostren diverses mètriques de rendiment i d'animació que mesuren aspectes com: la coordinació, l'especialització, l'altruisme, la reciprocitat, etc. Comptat i debatut, exposem les conclusions més significatives d'aquest treball així com les línies d'investigació futures.

Paraules clau: Animació comportamental, entorns virtuals semàntics, simulació social.

ÍNDIX

I	MARC DE TREBALL	1	
	CAPÍTOL 1	Introducció	3
	1.1	Introducció i Motivacions	3
	1.2	Objectius	6
	1.3	Aportacions	7
	1.4	Organització del treball	8
II	L'ESTAT DE L'ART	11	
	CAPÍTOL 2	Entorns virtuals 3D habitats	13
	2.1	Poblament dels entorns virtuals 3D	13
	2.2	Informació semàntica en entorns virtuals	15
	2.2.1	Semàntica del món: Representació de l'entorn	17
	2.2.2	Semàntica de les accions: Representació de les tasques	18
	2.2.3	Semàntica de les societats: Representació de les relacions	20
	2.3	Animació comportamental en 3DIVA	21
	2.3.1	Agents físics	23
	2.3.2	Agents racionals	25
	2.4	Mecanismes de selecció dinàmica d'accions	27
	2.4.1	Planificació de tasques	28
	2.4.2	Sistemes basats en regles	32
	2.5	Conclusions	35

CAPÍTOL 3	Societats artificials	37
3.1	Comportament social en personatges 3D	37
3.2	Sistemes multiagent	41
3.3	Interacció en sistemes multiagent	43
3.3.1	Comunicació	44
3.3.2	Coordinació	45
3.3.3	Col·laboració	48
3.4	Xarxes socials	50
3.4.1	Xarxes de dependència	51
3.4.2	Xarxes de confiança	52
3.4.3	Xarxes de preferència	53
3.5	Conclusions	55
III	APORTACIONS	57
CAPÍTOL 4	Marc de Simulació	59
4.1	Introducció	59
4.2	Marc de simulació multiagent	60
4.3	Entorn Virtual Semàntic	63
4.3.1	SVE Core Ontology i Domain Specific Ontologies	65
4.3.2	La capa semàntica	69
4.4	Arquitectura d'agents socialment intel·ligents	72
4.4.1	Esquemes d'acció	76
4.4.2	Mòdul de comunicació	78
4.5	Conclusions	80
CAPÍTOL 5	Presa de decisions socials	81
5.1	Introducció	81
5.2	Requeriments socials dels 3DIVA	83
5.2.1	Detecció de dependències	84
5.2.2	Presa de decisions coordinada	86
5.3	Col·laboració amb planificadors heurístics	87
5.3.1	Comunicació directa i agrupament entre actors	88
5.3.2	Coordinació de tasques amb recursos compartits	93
5.3.3	Cooperació mitjançant sospesament heurístic	96
5.4	MADeM: Multi-modal Agent Decision Making	101

5.4.1	Elements i notació	102
5.4.2	Tipus de recursos	103
5.4.3	Expressió de preferències	105
5.4.4	Procediment de presa de decisions	106
5.5	Conclusions	112
CAPÍTOL 6 Resultats		113
6.1	Descripció dels experiments plantejats	113
6.2	Sospesament heurístic en el món de blocs	115
6.3	Món de blocs 3D multipersonatge	118
6.3.1	Coordinació de tasques	119
6.3.2	Cooperació amb objectius externs	122
6.3.3	Conclusions a la col·laboració amb planificadors heurístics	126
6.4	Bar universitari virtual	128
6.4.1	Definició de l'entorn	129
6.4.2	Definició dels actors sintètics	130
6.4.3	Expressió de preferències	133
6.4.4	Animació d'acords socials	135
6.4.5	Resultats socials	137
6.4.6	Coordinació vs. Sociabilitat	140
6.4.7	Conclusions MADeM	146
CAPÍTOL 7 Conclusions, aportacions i treballs futurs		149
7.1	Conclusions generals i aportacions	149
7.2	Treballs futurs	151
7.3	Publicacions derivades d'aquesta tesi	152
BIBLIOGRAFIA		155
IV ANNEXOS		171
ANNEX A Ontologies de l'entorn virtual semàntic		173
A.1	SVE Core Ontology	173
A.2	Domain Specific Ontology: Bar virtual	183
A.3	Instanciació del món: Bar virtual	184

ÍNDIX DE TAULES

3.1	Categories de comunicació.	44
3.2	Diferents benestars socials expressats amb CUFs.	55
5.1	Tipus d'interacció en els sistemes multiagent [Ferber, 1999].	84
6.1	Resultats de coordinació per a dos agents en el món de blocs 3D.	119
6.2	Resultats de coordinació per a quatre agents en el món de blocs 3D.	122
6.3	Diferents graus de cooperació amb objectius externs.	126
6.4	Tasques subhastades pels agents en el bar virtual.	131
6.5	Exemples de converses per animar acords socials.	136
6.6	Tasques realitzades per diversos tipus de cambriers.	138
6.7	Favors intercanviats amb diferents models d'actitud personal.	139
6.8	Distribució temporal (en segons) de les tasques executades per 10 cambriers.	146

ÍNDEX DE FIGURES

1.1	Exemples de mons virtuals 3D: (a) ciutat habitada de Stormwind en el joc Warcarft, (b) simulador militar del Departament de Defensa dels E.U.A. i (c) ciutat d'Amsterdam a SecondLife.	4
2.1	Ontologia per representar entorns urbans [Costa de Paiva et al., 2005]. . .	18
2.2	Ontologia per a la representació de les tasques [Chang et al., 2005]. . . .	19
2.3	Representació ontològica d'una xicoteta societat feudal [Kao et al., 2005].	20
2.4	(a) Visió piramidal clàssica de l'animació comportamental [Funge et al., 1999] (b) Espectre actual d'actors 3D.	22
2.5	Exemples de presentadors 3D: (a) Cléo i (b) Jack.	24
2.6	Diferents gests i expressions facials reproduïts per <i>Greta</i> [Pelachaud, 2005].	24
2.7	Màquina d'estats finits que defineix el pla d'atac d'un NPC en Quake II. .	26
2.8	Boids: (a) Model bàsic [Reynolds, 1987] i (b) aplicació a un banc de peixos virtuals.	27
2.9	Planificació jeràrquica emprada en l'entorn SodaJack [Geib et al., 1994]. .	29
2.10	Descomposició de les tasques que descriuen completament l'actuació d'un actor mitjançant HTN [Cavazza et al., 2002].	30
2.11	Modelatge cognitiu: (a) esquema general del model i (b) arbre de planificació [Funge et al., 1999].	31
2.12	Tècnica de planificació heurística: (a) Accions possibles per a un personatge 3D, (b) Arbre de decisió HSP [Lozano, 2005].	32
2.13	Exemples que empen SOAR: (a) Esquema d'operadors utilitzat per SO-ARBot i (b) Esteve [Elliott et al., 1999].	33
2.14	Model BDI emprat en [Caicedo and Thalmann, 2000] per a l'animació d'un client en un bar virtual.	34

3.1	Personatges 3D que col·laboren (a) amb l'usuari humà [Prada and Paiva, 2005] i (b) amb altre caràcter autònom [Ciger, 2005].	40
3.2	Sistemes multiagent: (a) visió general i (b) àrees d'aplicació [Ferber, 1999].	41
3.3	Protocol d'interacció <i>Request when</i> en FIPA.	46
3.4	Exemple d'ordenació de les tasques de tres agents en GPGP.	48
3.5	Descomposició d'una missió d'atac militar [Tambe, 1997]. Els operadors marcats amb * corresponen a operadors d'equip.	49
3.6	Exemples de xarxes de dependència: a) treball conjunt en equip i b) entorn de mercat.	52
3.7	Estratègia de delegació de tasques amb diversos criteris de confiança definida en [Falcone et al., 2004].	53
4.1	Marc de simulació multiagent proposat.	61
4.2	Taxonomia de classes per a la creació d'un bar virtual.	65
4.3	Propietats dels classes definides en l'ontologia.	67
4.4	Relacions socials definides en l'ontologia.	68
4.5	Arbre de dependència sensorial.	70
4.6	Arquitectura dels nostres agents socialment intel·ligents.	73
4.7	Exemple d'operador genèric expressat mitjançant la parametrizació de: (a) un pla BDI o (b) un operador basat en el model STRIPS.	77
4.8	Exemples de converses entre dos agents per: (a) conèixer la tasca actual d'un altre, (b) conèixer els objectius d'un altre.	80
5.1	Espectre de decisions des dels agents racionals als agents socials.	82
5.2	Detecció de dependències en el mòdul controlador dels agents basats en el planificador <i>miniMin-HSP</i>	90
5.3	Estructura de la memòria dels actors sintètics.	92
5.4	Exemple de les relacions entre els objectius de tres agents que estan netejant un apartament.	93
5.5	Model de coordinació de tasques amb recursos compartits.	94
5.6	Procediment de presa de decisions MADeM.	107
6.1	Cooperació en funció del pes heurístic: a) mitjana d'objectius externs aconseguits; i b) mitjana d'accions extres realitzades.	116
6.2	Comparativa dels temps de resolució en funció del pes heurístic.	118
6.3	Problema d'apilar quatre caixes entre dos personatges.	120
6.4	Animació 3D de la construcció de la pila de caixes entre dos actors.	121

6.5	Traça simple de cooperació entre dos agents en el món de blocs 3D. . . .	123
6.6	Cooperació entre tres actors sintètics en un entorn 3D.	124
6.7	Exemple d'adquisició d'objectius externs.	125
6.8	Definició semàntica de l'entorn del bar virtual.	129
6.9	Relacions socials entre els personatges que habiten el bar virtual.	131
6.10	Controladors dels agents: (a) cambrers i (b) clients.	132
6.11	Comparativa entre els clients mandrosos i els clients socials.	140
6.12	Entorn 3D habitat que simula un bar virtual.	141
6.13	Traces senzilles de comportament: (a) coordinació màxima entre cambrers; (b) balanceig entre coordinació i sociabilitat.	142
6.14	Animació de diverses situacions interactives: (a) Atendre un client, (b) Demanar un favor a un company, (c) Informar del resultat d'una acció (d) Servir un client.	143
6.15	Estimadors de coordinació i sociabilitat per a un grup de cambrers: (a) <i>Throughput</i> i (b) <i>Animation</i>	144

PART I

MARC DE TREBALL

INTRODUCCIÓ

L'home, eixe ésser tan feble, ha rebut de la natura dues coses que haurien de fer d'ell el més fort dels animals: la raó i la sociabilitat.

Lucio Anneo Séneca (4 a.C, 65 d.C.).

1.1 INTRODUCCIÓ I MOTIVACIONS

El grau d'avançament assolit per la tecnologia en els últims temps ha permès apropar els mons virtuals al món físic fins a un punt en què podem veure certs aspectes dels ciberespais definits en la ciència-ficció [Gibson, 1984, Stephenson, 1992] com una fita abastable. Amb el suport d'un maquinari cada vegada més potent, les tècniques d'animació gràfica són capaces de reproduir escenaris tridimensionals en temps real amb molta fidelitat. Paral·lelament a l'evolució d'Internet, aquest fet ha provocat l'aparició de **mons simulats** on usuaris d'arreu del Món poden interaccionar amb diferents elements virtuals (com ara objectes 3D o representacions gràfiques d'altres usuaris conegudes com *avatars*). Alguns exemples típics d'aplicacions que fan servir aquests mons 3D són els jocs d'entreteniment, els simuladors per a l'aprenentatge de tasques civils o militars, o ciberespais com el famós SecondLife (veure la figura 1.1).

Dia rere dia, els mons simulats arrabassen un pam de terreny i inclouen nous serveis que no sols complementen aquells que es troben al món real, sinó que àdhuc podrien



Figura 1.1: Exemples de mons virtuals 3D: (a) ciutat habitada de Stormwind en el joc Warcraft, (b) simulador militar del Departament de Defensa dels E.U.A. i (c) ciutat d'Amsterdam a SecondLife.

arribar a substituir-los parcialment en un futur no molt llunyà. Aquesta funcionalitat en exclusiva podria arribar a comprometre l'hegemonia del món tangible i esdevenir, per tant, el sorgiment d'una *interrealitat* [van Kokswijk, 2007] poblada de **societats artificials d'humanoides virtuals potencialment autònoms**.

Un aspecte a què s'enfronten molts d'aquests mons és al repte d'habitar l'entorn amb criatures virtuals autònomes (siguen humanoides o animals) que no sols mostren una bona qualitat gràfica sinó també una mena de **credibilitat de comportament**. L'objectiu final de l'**animació comportamental** és la construcció d'un sistema intel·ligent capaç d'integrar les tècniques necessàries per a la simulació realista del comportament dels éssers virtuals: la percepció, el control motor, la presa de decisions, l'execució d'accions, la comunicació entre els individus, la interacció amb l'entorn, etc. Aquesta necessitat ha promogut la inclusió de tècniques provinents del camp de la intel·ligència artificial per tal d'aconseguir actors virtuals intel·ligents (IVA, *Intelligent Virtual Agents*) amb **propietats**

centrades en l'agent com ara l'autonomia, la reactivitat i la proactivitat. En aquesta línia es troba el treball seminal d'aquesta tesi, realitzat al Grup de Caràcters Intel·ligents (ICG) del Departament d'Informàtica de la Universitat de València, que consistia a implementar un sistema d'animació comportamental de personatges 3D basat en tècniques de planificació heurística [Lozano, 2005].

Nogensmenys, els actors virtuals intel·ligents encara manquen d'una qualitat que els assembla als seus referents originals: **la intel·ligència col·lectiva**. La intel·ligència col·lectiva sorgeix com a resultat de la interacció entre els personatges i aquesta no pot oblidar els trets socials de l'animació comportamental ¹. Aquesta necessitat és encara més escaient quan parlem d'humanoides virtuals o actors sintètics, ja que la sociabilitat que un observador extern espera d'un humà simulat és més complexa que la mostrada per altres animals, amb menys capacitats de raonament.

La recerca duta a terme en l'àrea de la interacció dels humans 3D s'ha centrat majoritàriament en la millora de la comunicació entre un caràcter autònom i un usuari humà. Com a resultat d'aquests treballs hom pot trobar actors sintètics amb els quals mantindre converses més o menys realistes com per exemple presentadors, doctors o guies virtuals (veure el capítol 2). Per a incrementar la seua fidelitat, aquests personatges sovint inclouen mecanismes d'expressivitat gestual i corporal amb els quals exterioritzar emocions o emfasitzar el significat del discurs. Malgrat açò, els **actors socialment intel·ligents** també haurien de ser capaços d'adreçar-se a altres individus autònoms per tal d'aconseguir els seus objectius, sense la intervenció d'un usuari humà. Normalment, els mons artificials estan poblats de diversos caràcters que comparteixen un nombre finit de recursos. Llavors, els conflictes apareixen ràpidament a causa de la voluntat d'usar un mateix recurs de manera simultània. Per exemple, un mateix martell 3D podria ser desitjat per dos fusters virtuals; un per clavar un clau i un altre per encaixar dues peces. Un altre exemple de recurs compartit és una comanda feta per un client en un bar virtual; la qual podria ser atesa per diversos cambrers. Per tant, l'ús de tècniques de **coordinació i cooperació** és necessari perquè els personatges autointeressats prompte cauran en conflicte i produiran simulacions de baixa qualitat. Per exemple, els cambrers començaran a obstruir-se si pretenen atendre la mateixa comanda.

A més a més, els humanoides virtuals habitualment representen un rol en l'escenari (p.

¹Segons la teoria d'agents, les quatre propietats que defineixen un comportament intel·ligent són: l'autonomia, la reactivitat, la proactivitat i la sociabilitat [Wooldridge and Jennings, 1995].

ex. fuster, cambrer, etc.) que estableix relacions diverses entre els membres de la societat artificial. Aquests lligams defineixen una **xarxa social** que hauria de ser considerada per prendre **decisiones socialment acceptables** i fugir així de comportaments robòtics. Els actors intel·ligents necessiten avaluar l'impacte social de les seues accions i decidir la seua actuació d'acord amb la societat simulada. Hom espera, per exemple, que dos actors i amics virtuals intercanvien informació, s'agrupen i conversen entre ells de tant en tant.

En resum, cal aprofundir en la interacció entre els actors sintètics autònoms per tal de fer-los evolucionar cap a una intel·ligència social capaç d'emular la presa de decisions humana, essent açò la **principal motivació** d'aquesta tesi. Aquesta tasca comporta fer una sinergia entre múltiples disciplines com ara els gràfics per computador, la intel·ligència artificial, la psicologia i la sociologia. De manera particular, aquest document estudia l'ús de tècniques provinents de la teoria de la decisió social (*Social Choice Theory*) i de l'economia del benestar (*Welfare Economics*).

1.2 OBJECTIUS

L'objectiu principal d'aquesta tesi és la integració d'habilitats socials en l'animació comportamental d'humanoides autònoms situats dins de mons virtuals 3D.

La generació de comportaments socials animals (p. ex. el moviment d'una ramada d'aus o del bestiar) es pot aconseguir amb l'ús de tècniques de raonament reactives que no necessiten d'una representació de l'entorn gaire sofisticada. Tanmateix, la presa de decisions que han de dur a terme els humans virtuals per interaccionar tant amb els objectes com amb la resta d'agents comporta processos cognitius complexos i requereix un coneixement abstracte dels elements de l'entorn. Darrerament, hom ha proposat la **inclusió d'informació semàntica dins dels entorns virtuals**, coneguts sota el terme anglosaxó de *Semantic Virtual Environments* (SVE). D'acord amb açò, un primer objectiu d'aquesta tesi serà la cerca d'una solució genèrica per a la representació simbòlica dels mons virtuals 3D.

La natura proactiva dels actors sintètics ha fet que les tècniques d'animació comportamental més comunes, basades en la selecció dinàmica d'accions, es divideixen fonamentalment en dos grups: la planificació de tasques (p. ex. les basades en *STRIPS*) i els

sistemes basats en regles (p. ex. els models *Belief-Desire-Intention*). Considerem ambdues aproximacions vàlides, extensament emprades i moltes vegades complementàries. Per exemple, és possible combinar la planificació per a la seqüenciació de tasques en el baix nivell i el model BDI per al manteniment dels objectius de més alt nivell. En conseqüència, l'objectiu principal d'aquesta tesi ha estat la inclusió d'habilitats socials en ambdós paradigmes. Més concretament, hem tractat els següents subobjectius:

- **Col·laboració en planificadors heurístics:** A pesar que les simulacions d'humans virtuals no requereixen normalment l'execució de plans òptims, és clar que la credibilitat d'un conjunt de personatges depèn en gran mesura de la coherència de les seues accions dins del context del grup. Aquesta coherència pot emergir de comportaments pròxims a la situació més eficient com ara el canvi i l'adaptació de les tasques que realitza un caràcter dependent d'aquelles realitzades per la resta de companys. L'alt dinamisme dels entorns 3D fa desaconsellable la creació centralitzada d'un pla comú per a tots els agents, ja que aquest serà invàlid en poc de temps. Ben al contrari, mitjançant la coordinació i la cooperació de plans individuals, els agents poden evitar interferències i traure profit en la consecució dels seus objectius personals.
- **Presa de decisions multimodal:** Les decisions humanes sovint atenen més d'un punt de vista i no tots persegueixen un comportament coordinat. Per exemple, un actor virtual mandrós podria triar descansar en comptes de treballar conjuntament amb un company per aconseguir els objectius més ràpidament. Així mateix, un actor virtual calculador podria decidir ajudar un altre dependent de l'intercanvi de favors prèviament realitzats. L'objectiu final d'aquest treball ha estat el desenvolupament d'un mòdul de raonament general capaç de tindre en compte preferències diferents per als agents i que pugui ser emprat per agents BDI a l'hora de prendre decisions socialment acceptables.

1.3 APORTACIONS

De conformitat amb els objectius definits en el punt anterior, les aportacions principals d'aquesta tesi es poden resumir en els següents punts:

- Proposem un model general d'**entorn virtual semàntic (SVE) basat en ontologies** amb el qual es poden aconseguir tres objectius bàsics per a l'animació d'actors sin-

tètics: la millora de la sensorització d'escenes complexes, la definició d'operatives generals que els personatges puguin reutilitzar en situacions diverses i la definició de les relacions socials i d'organització establides entre els membres d'una societat artificial.

- Proposem una nova **tècnica per aconseguir comportaments col·laboratius en grups d'agents basats en planificadors heurístics**. Fem servir la comunicació d'accions per a reduir les interferències entre les activitats dels humanoides i la comunicació d'objectius per a la cooperació entre els personatges 3D. Llavors, hem desenvolupat un mecanisme de sospesament independent del planificador i de la funció heurística que permet assignar un pes a cada objectiu, de manera que reflectisca la seua importància social. Així doncs, els agents poden adoptar dinàmicament objectius externs (de menys importància que els propis) i aconseguir comportaments coordinats sense augmentar la complexitat de les seues decisions.
- Presentem **MADeM (*Multi-modal Agent Decision Making*)**, un procés de presa de decisions de tipus social que pot ser emprat per agents BDI per tal de prendre decisions socialment acceptables. MADeM és capaç d'avaluar diferents solucions a un problema mitjançant una tècnica social basada en el mercat que fa servir les subhastes com a mecanisme perquè els agents expressen les seues preferències. A més a més, MADeM permet definir diversos ordres socials en una societat (p. ex. elitista, utilitari, etc.) així com actituds personals dels seus membres (p. ex. egoisme, altruisme, etc.).

1.4 ORGANITZACIÓ DEL TREBALL

L'estructura d'aquest document es divideix en els següents capítols:

- El capítol 2 fa una repassada als entorns virtuals habitats per actors sintètics. En aquest capítol s'estudia la inclusió d'informació semàntica als mons simulats. Així mateix, s'analitza l'espectre actual dels actors virtuals intel·ligents 3D (3DIVA) i es descriuen els mecanismes més comuns de selecció dinàmica d'accions en agents racionals.
- En el capítol 3 s'examinen els exemples més representatius d'animació comportamental social en personatges 3D. A continuació, fem una anàlisi dels models

de sociabilitat més comuns que hom pot trobar dins l'estat de l'art dels sistemes multiagent. Llavors, definirem conceptes sovint borrosos com ara: coordinació, cooperació, col·laboració, tipus de xarxes socials, etc.

- El capítol 4 presenta el marc de simulació multiagent dissenyat per a la integració d'actors sintètics amb habilitats socials en entorns virtuals 3D. D'una banda, precognitzem el nostre model d'entorn virtual semàntic (SVE) basat en ontologies OWL (Web Ontology Language) per a la representació semàntica del món. D'una altra, proposem una arquitectura modular per controlar l'actuació d'actors socialment intel·ligents.
- El capítol 5 conté les aportacions principals d'aquesta tesi en matèria de presa de decisions social: a) la tècnica de coordinació de tasques i de sospesament d'objectius externs per aconseguir comportaments col·laboratius en planificadors heurístics; i b) MADeM, el procés de presa de decisions multimodal i social basat en el repartiment de recursos en sistemes multiagent (*Multi-Agent Resource Allocation, MARA*).
- En el capítol 6 examinem els resultats obtinguts en tres exemples de complexitat creixent: el món de blocs clàssic, el món de blocs 3D multipersonatge i l'animació d'un bar universitari virtual. Demostrem l'impacte de les distintes tècniques socials desenvolupades mitjançant l'anàlisi de les traces de comportament i la comparació de diverses mètriques de rendiment i d'animació que mesuren aspectes com: la coordinació, l'especialització, l'altruisme, la reciprocitat, etc.
- Finalment, el capítol 7 arreplega les conclusions generals del treball realitzat a més de les línies de recerca que se'n deriven i que encara resten obertes. D'acord amb açò, s'indicaran els treballs d'investigació futurs així com els ja publicats.

PART II

L'ESTAT DE L'ART

ENTORNS VIRTUALS 3D HABITATS

En aquest capítol analitzem els entorns virtuals 3D habitats per actors sintètics autònoms com a evolució dels escenaris despoblats de la realitat virtual clàssica. Dins d'aquests entorns, el grau de realisme assolit pel comportament dels personatges depén fonamentalment de dos aspectes. El primer fa referència a la **gestió semàntica** de la informació de l'escena; la qual pot afavorir la interacció tant amb els objectes del món com amb la resta d'individus. El segon, apunta al mecanisme de **presa de decisions** del personatge, el qual governa la seua animació comportamental.

2.1 POBLAMENT DELS ENTORNS VIRTUALS 3D

Els entorns virtuals tridimensionals (EV3D) han experimentat una evolució molt significativa a causa de la gran quantitat d'avanços tecnològics que s'han produït des que comencés el seu desenvolupament. Encara que foren concebuts originàriament amb el propòsit de reproduir entorns potencialment perillosos per a l'entrenament militar (p. ex. els simuladors de vol), la popularització de la tecnologia de visualització gràfica en temps real per a l'ús civil significà el sorgiment del concepte més ampli de **realitat virtual (RV)** [Stanney, 2002]. Des del bressol d'aquest terme tan lax anaren apareixent tot un conjunt d'EV3D en **camp d'aplicació** tan diversos com: l'enginyeria, l'ergonomia, la medicina, l'entrenament, l'educació, la visualització científica, l'esbarjo, etc.

L'augment continu de la potència del maquinari i la millora del programari gràfic 3D a l'abast de l'usuari particular creà noves expectatives de realisme en els EV3D. Enfront de la visió clàssica d'immersió presencial perseguida pels dispositius d'entrada i eixida com

els cascs de RV o els guants de dades, els EV3D malden per assolir una **immersió mental de l'usuari**. És a dir, aquella a la qual s'arriba quan l'usuari se sent integrat en una realitat més enllà de la física; com la que sotmet l'espectador d'una representació teatral o el lector d'un llibre. Per tal d'assolir-la, calia substituir les escenes estàtiques i despoblades dels inicis per mons poblats d'objectes i d'altres personatges, el comportament dels quals ressemblés els seus equivalents en el món real.

D'aquesta manera sorgeixen els **entorns virtuals intel·ligents** (*Intelligent Virtual Environments* o IVE) [Luck and Aylett, 2000], combinació de la intel·ligència artificial (IA) i les tècniques de vida artificial amb els entorns virtuals. Aquesta sinergia ha donat els seus fruits en la construcció d'EV3D dinàmics [Aylett and Cavazza, 2001] i en la seua interacció amb l'usuari [Calderón et al., 2003]. No obstant això, l'aspecte en el qual es produeix una major convergència és aquell que versa sobre els personatges 3D que poblen els entorns virtuals. Hom pot concebre un personatge 3D sota el paradigma d'agent autònom provinent de la IA, ja que comparteixen propietats importants. Bàsicament, ambdós necessiten un sistema de control amb capacitats de percepció, de presa de decisions i d'actuació per comportar-se de forma correcta en el seu entorn. Per tant, els actors sintètics hereten el problema clàssic del comportament autònom al qual també s'enfronten, per exemple, els agents *software* o els agents robòtics. El terme científic que ajunta aquest marc amb els requeriments dels entorns i dels agents virtuals és el d'**actors virtuals intel·ligents 3D** (3DIVA).

El desenvolupament de 3DIVA comporta la resolució d'una multitud de problemes complexos (p. ex. la navegació, la planificació, els models de personalitat, etc.) i la seua integració en entorns 3D [Badler et al., 1993]. Aquest fet ha provocat que els grups d'investigació hagen centrat els seus esforços en assumptes concrets. Les aportacions més representatives les trobem en els camps de l'entreteniment, de la pedagogia i de la comunicació i socialització en comunitats virtuals per Internet [Laird and Duchi, 2000, Elliott et al., 1999, Pelachaud, 2005]. Com a resultat d'açò, els **entorns virtuals habitats** es troben esquitxats d'agents 3D que s'especialitzen a aconseguir comportaments versemblants a distints nivells: físic, racional, social... (veure apartat 2.3).

L'animació del comportament d'un o més agents en un entorn 3D proporciona l'experiència visual per poder validar el grau de versemblança o realisme comportamental assolit. En aquest sentit, el terme credibilitat apareix com una norma escassament definida [Badler and Allbeck, 2000] que hauran de complir aquests personatges 3D i que

influirà decisivament en el disseny de les seues arquitectures. Hom ha distingit la **recerca en agents autònoms** de la **recerca en sistemes multiagent** [Luck and Aylett, 2000]. La primera, que tractarem en el punt 2.3 d'aquest capítol, versa normalment sobre la generació d'actors sintètics que es comporten de manera egocèntrica i para especial esment a la coherència de comportament en l'ús dels objectes de l'entorn i en l'expressió gestual i emocional. Tanmateix, el comportament dels 3DIVA autointeressats prompte es veu compromés en contextos poblats per diversos agents (p. ex. amb l'aparició de conflictes en l'accés a recursos compartits). Aleshores, la intel·ligència col·lectiva és una necessitat per a molts actors sintètics i especialment per els humanoides virtuals, ja que la sociabilitat que hom espera d'un humà simulat és més complexa que la mostrada per altres animals. D'acord amb açò, cal incorporar mecanismes d'interacció entre els distints agents autònoms que formen una societat artificial. Aquest aspecte és estudiat per la recerca en sistemes multiagent i serà tractat en el capítol 3 d'aquesta tesi.

La presa de decisions que duen a terme els humans virtuals per interaccionar tant amb els objectes com amb la resta d'agents comporta processos cognitius complexos i requereix un coneixement abstracte dels elements de l'entorn. En conseqüència, abans de repassar els mecanismes d'animació de comportaments que han estat aplicats comunament sobre els 3DIVA, analitzem amb més detall les solucions emprades per a la representació de la informació continguda en els entorns virtuals 3D.

2.2 INFORMACIÓ SEMÀNTICA EN ENTORNS VIRTUALS

A l'hora de generar comportaments complexos en els personatges 3D, la preocupació principal ha estat generalment el procés de presa de decisions, ja que és el responsable de les accions que finalment són animades. Tanmateix, hi ha un tret molt característic dels entorns virtuals que demana una atenció especial de bon antuvi: la **nombrosa informació** que desprenen. Imaginem, per exemple, la quantitat d'informació que contenen les prestatgeries plenes de llibres d'una biblioteca virtual. Si tota aquesta informació hagués de ser tinguda en compte, la presa de decisions seria del tot intractable. Malgrat aquest fet, encara no existeix un formalisme comunament acceptat de gestió de la gran quantitat d'informació associada als entorns virtuals 3D.

Una primera aproximació per a organitzar la informació d'una escena tridimensional

són els sistemes basats en grafs d'escena com per exemple Java3D [JAVA3D, 2008] o OpenSceneGraph [OSG, 2008]. A pesar que poden incloure senzilles regles d'animació i sistemes molt bàsics d'intel·ligència artificial destinats al programari de visualització gràfica [Rodríguez-García, 2004], aquestes estructures de dades s'empren fonamentalment per al retallament automàtic de parts no visibles (*culling*). Llavors, les solucions basades en grafs d'escena ofereixen normalment una interacció amb els objectes de l'entorn nul·la o molt limitada.

Al contrari, els motors de jocs (p. ex. Unreal Tournament [Epic games, 2008]) permeten la creació de sistemes molt interactius [Lewis and Jacobson, 2002]. Els motors de jocs solen fer servir un model d'esdeveniments per notificar els canvis que es produeixen en un entorn dinàmic i un llenguatge de *script* per a la definició de les modificacions que provoca una acció sobre l'estat del món. En aquest context però, el codi font que implementa la interacció roman difícilment reutilitzable. Per exemple, mentre que un bàrman de la vida real té l'habilitat general de servir begudes als clients, un bàrman virtual normalment té definit un operador per a cada tasca que pot dur a terme. Així, podríem trobar-nos amb un bàrman virtual capaç de servir un suc de taronja i una copa de vi però sense cap idea de com preparar una tassa de te. Açò succeirà sempre que els agents virtuals no tinguin un coneixement general sinó particular de les situacions a les quals s'enfronten.

Darrerament, hom ha proposat la inclusió de models de representació semàntics per a gestionar la informació dels entorns virtuals, coneguts aleshores com **Entorns Virtuals Semàntics (SVE)**. La definició d'una base de coneixement semàntic pot beneficiar la producció dels entorns virtuals tridimensionals (secció 2.2.1). De manera especial si estan poblats per personatges intel·ligents, els quals necessitaran interaccionar tant amb els objectes com amb la resta d'agents de l'escena. D'una banda, l'ús de la semàntica pot aprofitar a la interacció agent-objecte, a través d'esquemes d'acció més generals (secció 2.2.2). D'altra banda, la interacció agent-agent pot millorar gràcies a la definició de les relacions que lliguen els agents i que restringeixen les seues possibles actuacions (secció 2.2.3).

Des del punt de vista dels entorns virtuals habitats, podem classificar les aportacions semàntiques en tres categories, que detallarem a continuació: la semàntica del món o representació de l'entorn; la semàntica de les accions o representació de les tasques i la semàntica de la societat o representació de les relacions entre els agents.

2.2.1 Semàntica del món: Representació de l'entorn

La comunitat dels gràfics per ordinador ha començat a incloure en els últims temps l'ús d'informació semàntica en la construcció dels mons virtuals. Per exemple, el sistema VR-Wise [Pellens et al., 2005] facilita el disseny d'entorns simulats als usuaris no especialistes en realitat virtual. Per tal de fer açò, aquest programari empra un model conceptual del món definit per mitjà d'un conjunt d'ontologies expressades en el llenguatge DAML+OIL [Joint US/EU ad hoc Agent Markup Language Committee, 2008]. La idea és, per tant, que el dissenyador trie els conceptes de l'ontologia de definició del món i, posteriorment, que el sistema s'encarregue automàticament de transformar-los en les representacions gràfiques escaients. Els llenguatges emprats en aquesta aproximació han sigut VRML i X3D per al *renderitzat* de l'escena en entorns web.

El concepte abstracte d'entitat virtual ha estat desenvolupat per Mario Gutiérrez en el seu treball de tesi sobre SVE [Gutierrez, 2006]. Aquest concepte inclou la geometria i la interfície d'usuari dels objectes 3D i s'usa principalment per a la visualització i la interacció amb els objectes en diferents contextos o dispositius. Aquest treball també incorpora una versió preliminar d'una ontologia per a humans virtuals (implementada en XML) que té com a objectiu: la modelització del cos humà, la seua animació i la interacció dels humans virtuals amb uns objectes especials anomenats *Smart Objects* [Kallmann and Thalmann, 1999]. Els *Smart Objects* són el resultat d'estendre les dades geomètriques d'un objecte 3D amb informació semàntica per a la seua correcta manipulació (p. ex. les posicions per a agafar-lo). Malauradament, la interacció personatge-*Smart Object* ha estat normalment governada per un conjunt d'animacions predefinides (*scripts*), ja que la idea era reduir la complexitat de les tasques mitjançant la transferència d'informació orientada a l'animació a l'entorn virtual.

Les ciutats virtuals poblades per actors sintètics han fet ús, així mateix, d'informació semàntica per a definir els perfils de la població i les topologies d'entorns urbans complexos [Farenc et al., 1999, Thomas and Donikian, 2000, Costa de Paiva et al., 2005] (veure la figura 2.1). Aquests entorns informats defineixen particions espacials (p. ex. edificis, carrers, cruïlles, etc.) i els objectes que podem trobar dins d'aquestes (p. ex. passos de vianants, parades d'autobús, etc.). Aquesta informació estructurada és usada per agents intel·ligents com per exemple vianants o conductors de cotxes virtuals, bàsicament amb propòsits de navegació.

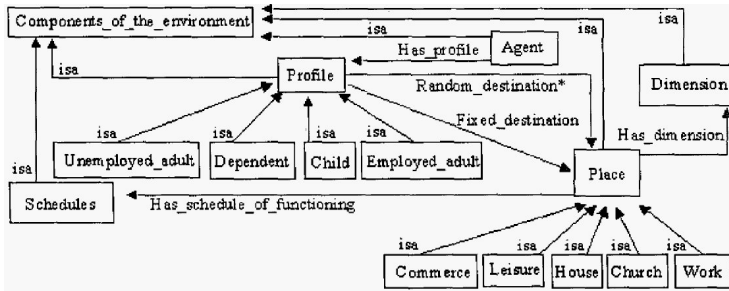


Figura 2.1: Ontologia per representar entorns urbans [Costa de Paiva et al., 2005].

2.2.2 Semàntica de les accions: Representació de les tasques

La interacció agent-objecte pot ser enriquida gràcies a la informació semàntica en esquemes d'acció complexos [Gil, 2005]. Tot seguit mostrem un conjunt d'exemples representatius. Primerament, el raonador específic d'objectes OSR (*Object Specific Reasoner*) [Levison, 1996] era capaç de classificar els objectes en taxonomies i de decidir què es podia fer amb ells depenent d'aquesta informació (p. ex. els contenidors podien ser oberts amb les mans). En segon lloc, la representació parametrizada de les accions [Badler et al., 2000] permetia definir tasques no instanciades (dites UPARs) que no especificaven ni l'actor virtual ni els objectes involucrats en l'acció. Sent així, totes juntes representaven qualsevol acció possible en el món. A l'hora d'executar una acció en un moment particular, les UPARs s'instanciaven amb les entitats virtuals afectades. En tercer lloc tenim el llenguatge d'alt nivell *Task Definition Language* [Vosinakis and Panayotopoulos, 2003], que persegueix l'objectiu de definir tasques independents del context. El principal avantatge del llenguatge proposat és el fet de permetre que les tasques siguin construïdes fàcilment i que puguin ser emprades per agents diferents en entorns diversos. Finalment, la separació d'accions i conseqüències en [Soto and Allongue, 2002] mostra un altre ús interessant de la semàntica per aconseguir la interoperativitat i la reusabilitat de les entitats virtuals. Dissortadament, els autors no defineixen una representació que pugui ser emprada en la construcció i gestió dels mons virtuals.

Una de les primeres solucions per a la representació de la informació consisteix a inserir coneixement en el món mitjançant anotacions (*Knowledge in the World*) [Doyle, 2002]. Aquesta aproximació proposa la codificació dins del món d'aquell coneixement i aquelles habilitats d'un caràcter que són específiques d'un entorn particular; mentre que les

habilitats invariables romanen part de l'agent. D'aquesta manera, s'afavoreix la reusabilitat dels actors en dominis diferents. Recentment però d'una manera similar, el sistema SeVen [Otto, 2005] ha tractat d'aconseguir programari que pugui ser reutilitzat sobre diversos entorns virtuals en un context web. Aquest sistema admet la definició de SVEs fent ús d'RDF (*W3C Resource Description Framework*) i categoritza els objectes mitjançant un camp explícit de *tipus d'objecte*. D'acord amb aquest camp, els agents poden extraure informació rellevant per a executar tasques sobre l'objecte en qüestió. Nogensmenys, les anotacions estàtiques no són prou quan es treballa amb entorns dinàmics com els entorns virtuals 3D.

Des de la comunitat d'agents, una aproximació semànticament millorada pretén aconseguir l'anotació de l'entorn de forma dinàmica per mitjà d'una capa conceptual basada en ontologies [Chang et al., 2005]. L'ús d'una capa conceptual pot ajudar els agents intel·ligents a planificar el seu comportament mitjançant la inferència ontològica (veure la figura 2.2). No obstant això, l'esquema d'acció adoptat en aquesta solució té diverses limitacions d'expressivitat; l'efecte d'una acció es restringeix a un únic objecte i sempre comporta el canvi d'aquest a una instància d'un altre concepte qualitatiu diferent. Ben al contrari, de vegades les accions poden afectar múltiples objectes o canviar una propietat d'un objecte sense modificar cap concepte. Un exemple seria l'emplenament d'un got amb una ampolla de vi. Malgrat que la quantitat de líquid variarà, ambdós objectes haurien de romandre instàncies dels mateixos conceptes.

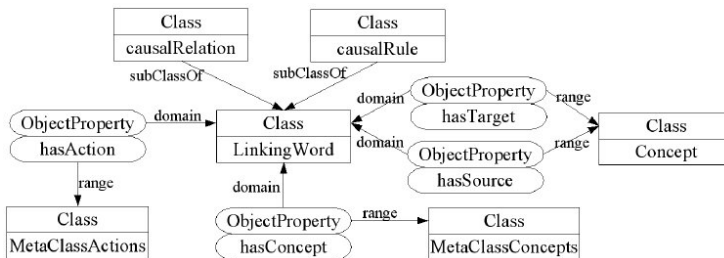


Figura 2.2: Ontologia per a la representació de les tasques [Chang et al., 2005].

La representació dels objectes i de l'operativitat dels agents pot ajuntar-se en una base de coneixement comuna mantinguda per l'entorn. El llenguatge de descripció d'entorns per a simulacions multiagent ELMS (*Environment Description Language for Multi-Agent Simulation*) [Okuyama et al., 2005] permet definir tant les propietats dels objectes com els *costos* dels agents: percepció i actuació. És a dir, el dissenyador pot controlar quines

Els agents cognitius autònoms poden augmentar la seua eficiència i adaptabilitat a les alteracions de l'entorn si són capaços de representar explícitament i d'explorar, a través de mecanismes de raonament adients, les capacitats d'altres agents i de les organitzacions en què estan immersos. Hom pot concebre una organització com el conjunt de restriccions en el comportament dels agents a fi de conduir-los a una finalitat comuna. Per descriure una organització s'han fet servir models funcionals (TÆMS [Decker, 1998]), estructurals (AALAADIN [Ferber and Gutknecht, 1998]) i mixts (TOVE [Fox et al., 1998]). Més contemporani, el model Moise+ [Giménez-Lugo et al., 2005] permet la descripció d'una organització en tres dimensions: estructural, funcional i deontològica. El model estructural defineix els grups i els rols que poden interpretar els agents. També inclourà diverses relacions entre ells com ara relacions d'autoritat, de responsabilitat, etc. El model funcional defineix l'objectiu final del grup, les missions necessàries per aconseguir-lo i les tasques en les quals es descompon. Finalment, el model deontològic associa els rols responsables de cadascuna de les tasques, és a dir, les obligacions dels agents a l'organització.

Un últim exemple d'informació semàntica allotjada a l'entorn virtual i adreçada a l'ús per part dels agents que el poblen són els *coordination artifacts* [Viroli et al., 2006]. Aquests *artefactes* són unes entitats no racionals que ofereixen protocols de coordinació per als agents que col·laboren dins d'una societat. Açò és, en comptes d'enviar i rebre missatges directament, els agents executaran accions sobre els *coordination artifacts*. En conseqüència, aquesta tècnica suposa una alternativa interessant d'interacció estructurada enfront dels llenguatges de comunicació d'agents (ACLs).

En la primera part d'aquest capítol hem analitzat el problema de la representació semàntica en els entorns virtuals habitats. La segona part versarà sobre els mecanismes de presa de decisions emprats més a sovint en l'animació comportamental dels 3DIVA.

2.3 ANIMACIÓ COMPORAMENTAL EN 3DIVA

L'animació comportamental d'agents virtuals 3D té com a objectiu la construcció de sistemes intel·ligents capaços d'integrar diverses tècniques a fi d'aconseguir la simulació versemblant dels seus comportaments. Entre ells podem incloure la percepció, el control motor, la selecció d'objectius, l'execució d'accions, la comunicació entre els agents, etc [Iglesias and Luengo, 2004]. La visió clàssica de l'animació comportamental, de-

fensada per Demetri Terzopoulos des de la *Universitat de Toronto*, mostra en forma de jerarquia piramidal la sinergia entre els gràfics per computador, les tècniques d'animació i la intel·ligència artificial (veure la figura 2.4a). Aquesta visió concep el problema com una evolució des dels models geomètrics fins les tècniques de modelat cognitiu [Tu and Terzopoulos, 1994, Funge et al., 1999].

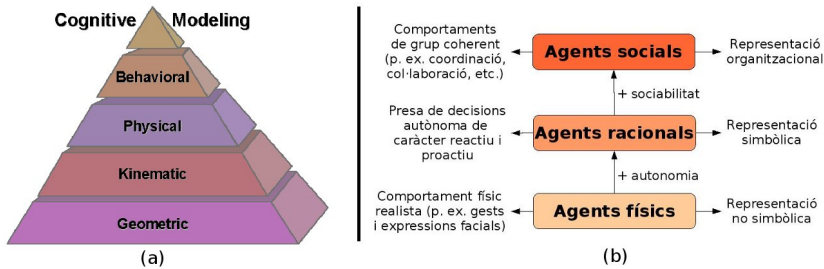


Figura 2.4: (a) Visió piramidal clàssica de l'animació comportamental [Funge et al., 1999] (b) Espectre actual d'actors 3D.

El treball desenvolupat en aquesta tesi s'emplaça en la part alta d'aquesta piràmide. En aquest context, l'estat de l'art dels 3DIVA pot ser analitzat mitjançant un espectre d'agents com el que apareix a la figura 2.4b. Hom distingeix tres categories principals: els agents físics, els agents racionals i els agents socials. D'una banda, els **agents físics** posen l'èmfasi en les habilitats motores per tal d'aconseguir comportaments realistes a nivell físic (p. ex. interacció física amb l'entorn, expressions facials i gestuals, etc.). Aquests agents normalment actuen de manera guiada i perceben l'EV3D mitjançant sensors que treballen a un nivell no simbòlic. D'altra banda, els **agents racionals** treballen el comportament amb una perspectiva operativa, és a dir, com el problema d'animar un conjunt d'accions coherents amb els objectius definits per a l'agent i l'estat de l'entorn. Al contrari dels físics, es caracteritzen per una presa de decisions autònoma, la qual necessitarà sovint informació simbòlica de l'EV3D per poder raonar. Per últim trobem els **agents socials**, els quals tenen com a objectiu enriquir la presa de decisió amb comportaments de grup versemblants (p. ex. comunicació entre els agents autònoms, coordinació de les seues tasques, etc.). Mentre que els agents racionals decideixen en funció de punts de vista personals o egocèntrics, els agents socials tenen en compte també a la resta d'agents en el seu procés de presa de decisions. Per tant, aquests agents necessitaran, a més, d'informació organitzacional sobre les relacions que s'estableixen en la societat artificial.

Idealment, els humanoides 3D haurien de mostrar una interacció física amb la resta

del món totalment realista així com també haurien de posseir habilitats racionals i socials similars a les dels humans. Ara bé, tots aquests aspectes comporten la resolució de problemes complexos, de manera que els investigadors han tendit a posar l'èmfasi en una o altra part de l'espectre. Tot seguit, fem un repassada amb més detall als agents físics i racionals suara esmentats. Com ja hem dit abans, deixem la discussió al voltant dels agents socials per al capítol 3 d'aquest document.

2.3.1 Agents físics

El propòsit dels agents físics és aparenyar **comportaments realistes a nivell físic**. Normalment, aquests personatges 3D manquen de sensors i de mecanismes de presa de decisions, cosa que fa del seu sistema motor el principal encarregat de proporcionar els signes d'expressivitat necessaris. A causa d'aquesta èmfasi en la manifestació física de l'actor en el seu entorn, aquests agents són anomenats habitualment *embodied agents*.

L'aproximació més senzilla a l'hora d'animar un actor d'aquest tipus consisteix a descriure'l completament al llarg de la simulació mitjançant un guió, de manera similar a com un director de cinema dirigeix els seus actors. Els **presentadors 3D** són l'exemple més clar d'**actors 3D guiats**. La figura 2.5a mostra *Cléo*, un dels primers actors sintètics en temps real desenvolupat al MEDIALAB de París per a Canal + França. El control corporal de *Cléo* es feia mitjançant la captura dels moviments realitzats per un actor real mentre que el control gestual feia servir un guant de dades. L'ús de guions programats o *scripts* ha estat una altra manera molt comuna de realitzar el guiatge automàtic d'un actor virtual. Un exemple representatiu és *Jack Presenter* [Noma and Badler, 1997], un presentador virtual que empra l'humanoide de cos sencer *Jack* [Badler, 1997], desenvolupat originàriament a la Universitat de Pennsylvania (veure la figura 2.5b). Les dades d'entrada al presentador virtual inclouen típicament comportaments verbals, els quals són processats per un generador sintètic de la parla, així com comandaments de control referits al llenguatge corporal (p. ex. els gests manuals per dirigir l'atenció del públic cap a una determinada informació existent en la pantalla).

Sovint l'animació d'un actor 3D es veu reduïda a només el cap, raó per la qual hom parla moltes vegades de **caps parlants** o *talking heads*. Aquests personatges han centrat la seua atenció en el moviment sincronitzat dels llavis amb el discurs i en l'expressió facial d'emocions. Aquestes funcionalitats són molt profitoses per al desenvolupament d'agents

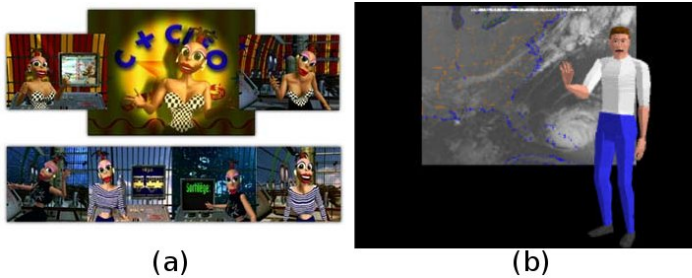


Figura 2.5: Exemples de presentadors 3D: (a) Cléo i (b) Jack.

que actuen com a interfície entre una persona i un ordinador (p. ex. assistents d'un programa d'ordinador, presentadors d'informació per Internet, sistemes de videoconferència 3D, etc.). El realisme expressiu de la cara és un camp de recerca amb una gran tradició en l'animació d'humans 3D [Thalmann et al., 1998] i encara hui genera molta activitat [Pelachaud, 2005, Niewiadomski and Pelachaud, 2007]. Un exemple actual són els treballs realitzats pel grup de Catherine Pelachaud a la Universitat de París per tal de proveir *Greta*, el seu agent conversacional, amb realimentació conversacional i expressió d'emocions (veure figura 2.6).



Figura 2.6: Diferents gests i expressions facials reproduïts per Greta [Pelachaud, 2005].

Les característiques referides adés no tan sols són interessants per als actors completament guiats sinó també per als personatges 3D semiautònoms, com els que representen els usuaris humans en els entorns tridimensionals multiusuari basats en la xarxa (p. ex. SecondLife [Linden Lab, 2008], Active Worlds [Activeworlds Corporation, 2008], etc.). Aquestes comunitats virtuals, conegudes genèricament com entorns virtuals distribuïts (*Distributed Virtual Environments* o DVE)¹ naixen amb la idea d'oferir un espai comú d'esbargiment representat mitjançant un EV3D, on els usuaris estaran representats per ac-

¹O bé com MMORPG (*Massive Multiplayer On-line Role-Playing Games*). Darrerament, també han estat encunyades com MMORLGs (*Massive Multi-player Online Real-Life Games*), ja que l'usuari pot modificar el seu avatar i interpretar un rol més dinàmic o inclús múltiples rols.

tors semiautònoms o *avatars* que els permetran percebre l'entorn, reconèixer els altres usuaris (pel seu avatar), comunicar-se i socialitzar-se amb ells. Nogensmenys, ciberespais tan famosos com SecondLife encara manquen d'humanoides totalment autònoms que puguin poblar el món sense que hi haja un usuari humà controlant-lo per darrere. De fet, aquest és un punt al voltant del qual encara resta oberta una gran discussió moral i legal [van Kokswijk, 2007].

2.3.2 Agents racionals

Al contrari dels agents físics, els agents racionals destaquen per la seua **capacitat de prendre decisions** d'una manera autònoma. Segons la intel·ligència artificial, un agent és racional si fa allò correcte, atenent a la informació de què disposa. Llavors, els comportaments simulats, així com el grau de racionalitat assolit, dependran llavors del model o formalisme d'IA que es faça servir en cada moment. Podem entendre la presa de decisions com el procés de cerca de les accions a realitzar per un personatge. Atenent a la quantitat de cerca realitzada, hom pot veure els agents racionals com una gradació d'agents: des dels que no realitzen cap cerca sinó que només reaccionen a estímuls externs (**reactivitat**) fins els que inclouen algú mecanisme de cerca d'accions futures per aconseguir uns objectius propis (**proactivitat**).

Per definició, un agent és reactiu si percep el seu entorn i reacciona davant els seus canvis. El sistema de presa de decisions de l'actor serà, doncs, l'encarregat de reaccionar als canvis que afecten el seu comportament i de filtrar aquells que li són irrellevants. El formalisme computacional més bàsic, emprat amb extensió en el control autònom d'actors 3D, és el de les **màquines d'estats finits**. Aquests autòmats codifiquen l'actuació de l'agent com un graf d'estats junt amb les condicions que provoquen el canvi d'un estat a un altre. El sistema multiagent IMPROV [Perlin and Goldberg, 1996] és un exemple clàssic de sistema d'animació comportamental de personatges 3D que segueix aquesta filosofia. Així mateix, alguns humans virtuals desenvolupats pel matrimoni Thalmann, als laboratoris *MIRALab* de la *Universitat de Gènova* i *VRLAB* de la *Universitat de Lausane*, han fet servir les màquines d'estats per a controlar el seu comportament reactiu [Molet et al., 1997]. Els videojocs han emprat el mot *Non-Player Character* (NPC) per a referir-se a les entitats que interpreten algun rol en el món fictici del joc però no estan controlades pel jugador. Tradicionalment, el comportament reactiu dels NPCs ha estat preprogramat en màquines d'estats. Per exemple, la figura 2.7 mostra el pla d'atac d'un

bot en *Quake II*. El principal avantatge de les màquines d'estats és la senzillesa de desenvolupament i l'eficiència d'execució, ja que tots els plans d'actuació estan especificats d'antuvi. Tanmateix, el seu ús comporta certes limitacions de flexibilitat i escalabilitat. Per una banda, com que el comportament de l'agent es troba imbricat en els estats de la màquina, la introducció de canvis de comportament produeix, freqüentment, que el graf d'estats haja de ser refet completament. Per una altra, a mesura que creix la complexitat dels comportaments, les màquines d'estats es fan grans i són més difícils de mantenir.

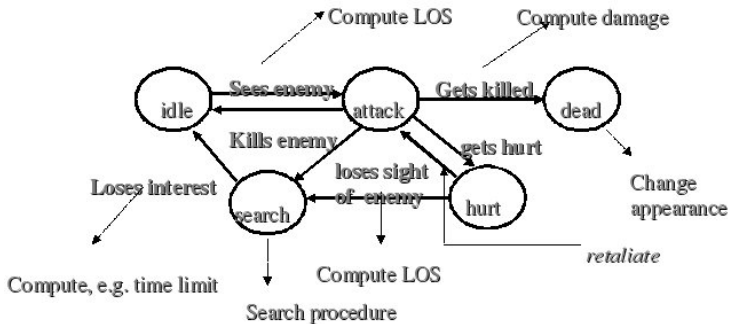


Figura 2.7: Màquina d'estats finits que defineix el pla d'atac d'un NPC en *Quake II*.

Una aproximació molt relacionada, emprada en ocasions per representar autòmats d'estats finits, són els **sistemes basats en regles** (*Rule Based Systems* o RBS). Aquest paradigma de programació defineix el comportament d'un agent a través d'un conjunt de regles. En aquest context, els canvis apercebuts en l'estat del món suposen l'activació d'aquelles regles que són aplicables. D'aquesta manera, l'animació del comportament consisteix a decidir quina regla llançar (resolució de conflictes entre regles) i a gestionar les noves regles que s'activaran com a resultat. Així doncs, els RBS permeten la manipulació flexible d'un gran nombre de regles i, per tant, la definició de comportaments més sofisticats que els aconseguits amb els autòmats d'estats finits. Els agents reactius basats en regles que han estat referència dins dels mons simulats són els *Boids*; introduïts per Reynolds per a la generació de comportaments de navegació propis d'espècies animals com ara els bancs de peixos [Reynolds, 1987, Reynolds, 1999] (veure la figura 2.8b). Aquest sistema es basa en l'ús de regles numèriques per a governar fonamentalment reaccions motores d'atracció, de repulsió i d'alineament amb la resta del grup (veure la figura 2.8a). Això no obstant, mitjançant l'ús de regles simbòliques, els RBS poden mostrar una riquesa de comportament més propera al raonament proactiu dels humans [Laird et al., 1987]. Per a una discussió més pregona al voltant de l'ús dels sistemes basats en regles simbòliques per a l'animació comportamental d'agents 3D proactius cal

veure la secció 2.4.2.

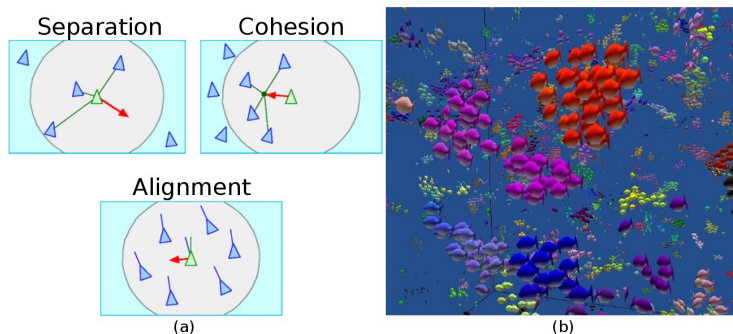


Figura 2.8: Boids: (a) Model bàsic [Reynolds, 1987] i (b) aplicació a un banc de peixos virtuals.

Els mons virtuals 3D habitats són escenaris altament variables en què les propietats de qualsevol objecte poden canviar de forma impredecible, a causa de l'actuació d'altres personatges o per la seua pròpia natura dinàmica. Tot i que els sistemes reactius han estat emprats en la generació de comportaments en entorns dinàmics, no solen incorporar cap mecanisme de cerca d'accions futures. Al contrari, el comportament proactiu d'un personatge autònom és aquell que busca quina és la millor seqüència d'accions per aconseguir uns objectius propis. A causa del dinamisme dels entorns virtuals, la presa de decisions dels agents proactius es realitza mitjançant sistemes de **selecció dinàmica d'accions**. En el següent punt estudiem amb detall els formalismes de selecció dinàmica d'accions emprats amb més freqüència pels 3DIVA proactius.

2.4 MECANISMES DE SELECCIÓ DINÀMICA D'ACCIONS

Les tècniques d'animació comportamental més comunes, basades en la selecció dinàmica d'accions, es divideixen fonamentalment en dos grups: la planificació de tasques i els sistemes basats en regles. D'una banda, la primera tracta la construcció d'un pla o seqüència de tasques que conduïska l'agent des d'un estat inicial cap a un estat objectiu. D'altra banda, la segona té els plans precompilats en un conjunt de regles i es preocupa del seu llançament amb l'objectiu de convertir l'estat del món actual en un estat desitjat. Considerem ambdues aproximacions vàlides, extensament emprades i moltes vegades complementàries. Per exemple, és possible combinar la planificació per a la seqüenciació

de tasques en el baix nivell i el model BDI per al manteniment dels objectius de més alt nivell.

2.4.1 Planificació de tasques

La representació clàssica de la planificació de tasques descriu un problema en termes de predicats, els quals representen assertions sobre l'estat del món ². Les accions possibles es defineixen mitjançant *operadors*, que fan servir les nocions de *precondicions* i *efectes*; que descriuen respectivament quan és factible realitzar-les i quines seran les seues conseqüències sobre l'estat del món. Llavors, l'objectiu de la planificació és trobar una seqüència d'accions que transforme l'estat inicial del sistema en un estat objectiu desitjat. Aquesta definició prové dels anys setanta, quan Fikes i Nilsson descriueren el **sistema de planificació STRIPS** [Fikes and Nilsson, 1971]. Des d'aleshores molts planificadors han estat basats en el model STRIPS, el qual ha sigut estès per tal de permetre quantificadors, efectes condicionals, aritmètica bàsica, incertesa, planificació temporal i moltes altres funcionalitats. Així mateix, diverses famílies d'algorismes han sigut desenvolupades, entre les més notables: Graphplan [Blum and Furst, 1997], FastForward [Hoffmann and Nebel, 2001], *Heuristic Search Planning* [Bonet and Geffner, 2001], etc.

La planificació de tasques ha estat un assumpte molt estudiat des del camp de la intel·ligència artificial i s'ha emprat amb èxit en robòtica, control de processos industrials, control aeronàutic, etc. Això no obstant, la seua aplicació als EV3D és molt escassa. Amb seguretat, una causa d'açò la trobem en la seua **alta complexitat**, que augmenta de forma exponencial amb la **quantitat d'informació** que s'ha de processar durant el càlcul del pla. A fi de reduir la càrrega del planificador, hi ha hagut intents de **simplificació** del problema. Per exemple, és possible eliminar els fets no rellevants per al problema del qual es busca solució en una fase de pre-processament [Nebel et al., 1997]. Per una altra banda, per raó de l'alta **variabilitat de l'entorn**, la creació *off line* d'un pla complet d'actuació per a un actor sintètic és a sovint de poca utilitat; ja que aquest pla es fa inaplicable en poc de temps. En conseqüència, hom ha proposat l'intercalat de les fases de percepció-planificació-acció; model que sol adjectivar-se com *continuous planning*, *situated planning* o *on line planning* [Avradinis et al., 2005].

²Freqüentment, d'acord amb l'assumpció de món tancat, tot allò que no es fa explícit amb un predicat és considerat fals [Nilsson, 2001]

Una bona aproximació per a reduir la complexitat esmentada abans és l'ús de sistemes de **planificació jeràrquica**. Aquest model genera plans amb un grau d'especificació general, els quals són refinats amb detalls d'execució pels agents responsables de les tasques. *SodaJack* [Geib et al., 1994] (veure la figura 2.9) és un exemple d'ús de la planificació jeràrquica en 3DIVA. Fou desenvolupat pel grup dirigit per Norman I. Badler a la Universitat de Pennsylvania i aplicat en un entorn simple en què l'humanoide *Jack* serveix productes als visitants. *Jack* és conduït per un sistema de planificació jeràrquica que utilitza tres planificadors diferents: *ItPlans* com a planificador d'alt nivell, un planificador (*OSR*) especialitzat en la localització d'objectes i un altre planificador específic per a la manipulació dels objectes. *ItPlans* treballa d'una manera iterativa, és a dir, produeix i expandeix els plans depenent dels resultats de les accions prèvies i de l'estat actual de l'entorn. Per exemple, si *Jack* ha de servir un gelat, primer ha d'aconseguir una tarrina. Suposem que les tarrines poden trobar-se en llocs diversos: l'armari o l'aparador. Aleshores, *ItPlans* genera primerament la tasca de cercar una tarrina en l'armari. Si n'hi ha, el pla avança cap a l'objectiu global. Però si no n'hi ha, una nova cerca és generada a l'aparador. L'avantatge és que el sistema pot respondre ràpidament, aspecte vital per a l'animació de comportaments en temps real. Ara bé, el principal desavantatge és que el pla separa l'objectiu final en subobjectius independents i els resol un rere l'altre, cosa que provoca comportaments pobres. Per exemple, l'agent cerca la tarrina a l'armari però després s'adona que també necessita una cullereta. Llavors, torna a començar la cerca seqüencial de nou en comptes d'haver tret els dos objectes alhora.

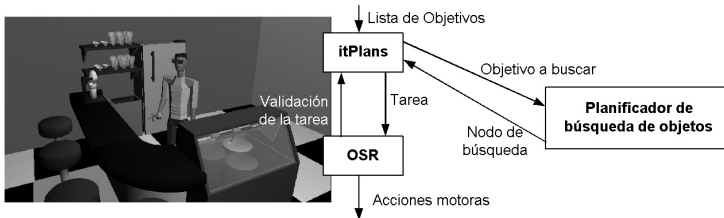


Figura 2.9: Planificació jeràrquica emprada en l'entorn *SodaJack* [Geib et al., 1994].

Un altre tipus de planificació jeràrquica el trobem en els generadors d'històries interactives (*interactive storytelling*). Aquests sistemes estan designats a generar històries dins d'EV3D a partir del comportament de l'usuari (qui interacciona de forma impredecible amb els objectes 3D o passa informació als personatges) i dels personatges autònoms. Diversos projectes han aplicat diferents formalismes de la IA per a millorar la riquesa i variabilitat dels fils narratius, per exemple: *The Oz Project* [Bates, 1992], *Mimesis*

[Young, 2001] i Virtual Storytelling [Cavazza et al., 2002]. Aquest últim ha emprat una tècnica coneguda com **xarxes jeràrquiques de tasques** (*Hierarchical Task Networks* o HTN). Les HTN són un mecanisme per mitjà del qual el creador de la història defineix totes les possibilitats a l'abast dels personatges 3D durant una primera fase coneguda com *authoring*. Com a resultat d'aquesta fase, queda definit el conjunt de plans (parcialment ordenats) que els personatges han de dur a terme (mitjançant arbres jeràrquics de tasques com el de la figura 2.10). Així, en temps d'execució els agents es dedicaran a extraure un dels possibles plans per a monitoritzar-lo posteriorment. Nogensmenys, la simplificació de l'actuació d'un agent en tasques independents amb un ordre parcial predefinit limita la presa de decisions dels personatges, resta variabilitat i flexibilitat als seus comportaments.

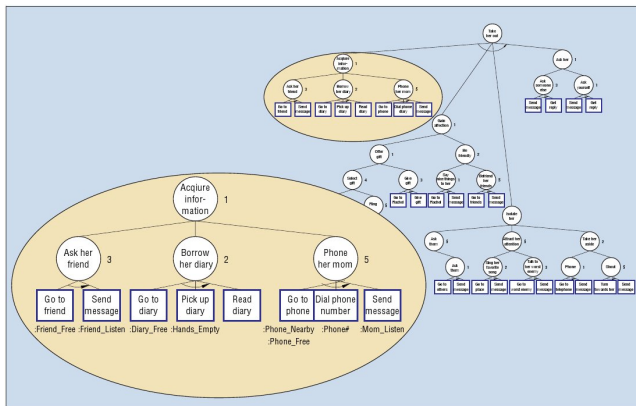


Figura 2.10: Descomposició de les tasques que descriuen completament l'actuació d'un actor mitjançant HTN [Cavazza et al., 2002].

Un exemple similar de simplificació del procés de planificació, basat en el càlcul de situacions, és el proposat pel **llenguatge de modelatge cognitiu** (*Cognitive Modelling Language* o CML) [Funge et al., 1999]. CML ajuda el dissenyador d'una animació a donar instruccions als actors per mitjà de guions predissenyats, a més a més, permet descriure el coneixement que el caràcter té sobre el món mitjançant les accions que pot realitzar (veure la figura 2.11a). Aquest esquema es representa típicament en forma d'arbre de planificació on els nodes representen situacions (veure la figura 2.11b). L'arrel és la situació inicial i cada camí representa una seqüència d'accions. Malgrat que les precondicions de les accions poden fer que certes seqüències no siguin possibles (nodes negres de l'arbre), la cerca d'una seqüència per arribar a una situació o node objectiu encara serà exponencial. Per agilitzar el procés, CML no empra algorismes sofisticats de cerca com A* sinó que fa una poda d'aquest arbre mitjançant *accions complexes*. En aquest context, les

accions complexes són simplement regles sobre execucions d'accions no permeses (nodes marrons de l'arbre). Aquesta tècnica fou usada en l'obtenció de comportament cognitiu dins dels món submarins de Tu i Terzopoulos [Tu and Terzopoulos, 1994], que ja havien integrat amb èxit comportaments físics i reactius.

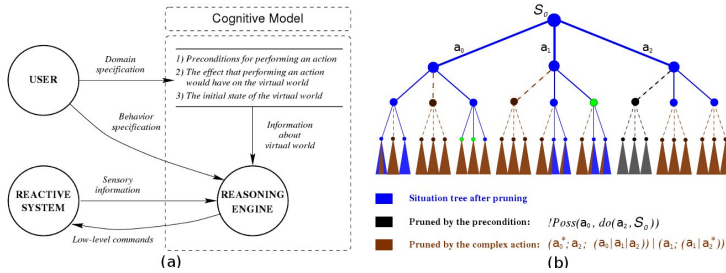


Figura 2.11: Modelatge cognitiu: (a) esquema general del model i (b) arbre de planificació [Funge et al., 1999].

En els últims anys, les **tècniques de planificació heurística** (*Heuristic Search Planning* o HSP) [Bonet and Geffner, 2001] també han estat proposades com a aproximacions molt interessants a l'hora d'animar comportaments de personatges 3D en entorns altament canviants [Lozano, 2005, Ciger, 2005]. Per oposició als sistemes on l'autor defineix el fil argumental de la història (coneguts com *top-down*), en l'animació amb HSP l'actor construeix el seu pla de forma proactiva; cosa per la qual els anomenen sistemes *bottom-up*. Aquest fet, junt amb l'intercalat de les fases de decisió i d'execució d'accions, fa que aquestes solucions s'adaptin millor als canvis que es produeixen en l'entorn. Així mateix, la generació automàtica del pla fa que els actors puguin tractar situacions que, malgrat tindre importància narrativa, no foren contemplades per l'autor en el seu guió original (p. ex. quan dos actors sintètics que són amics es troben de manera inesperada en un indret virtual). D'acord amb açò, donat un conjunt d'operadors que defineixen cadascuna de les possibles accions que un caràcter pot realitzar (figura 2.12a), la presa de decisions consisteix a cercar la seqüència d'accions que permet arribar a un estat objectiu. Per exemple, la figura 2.12b mostra l'arbre amb les possibilitats a l'abast d'un personatge 3D que té el propòsit de preparar una cita: embolicar un regal, netejar la casa, prendre una dutxa, fer el sopar... Les decisions adoptades correspondran amb el pla de longitud mínima, el qual evita la repetició (en certa manera inversemblant) d'accions innecessàries com ara rentar-se les mans cada vegada que realitza una acció. Per guiar l'algorisme de cerca al llarg d'aquest arbre i garantir una solució acceptable en un temps reduït, els HSP fan servir les funcions heurístiques. L'ús d'heurístiques independents del domini proveeix de robustesa

aquests mètodes, els quals poden funcionar correctament en distints escenaris 3D i per a diversos rols (p. ex. un venedor virtual, un cambrer, etc.) [Lozano, 2005].

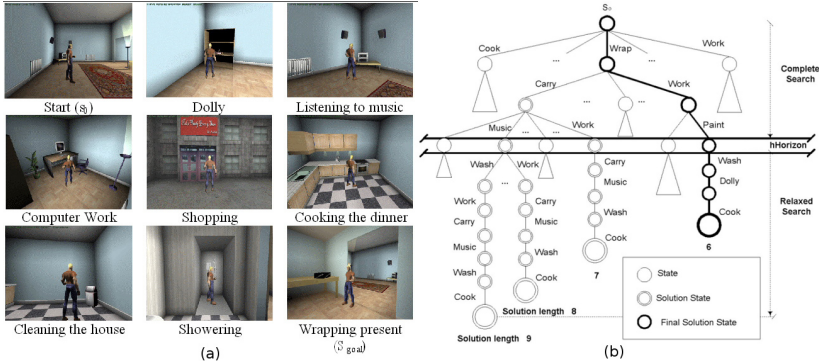


Figura 2.12: Tècnica de planificació heurística: (a) Accions possibles per a un personatge 3D, (b) Arbore de decisió HSP [Lozano, 2005].

En el capítol 3 reprendrem la discussió sobre aquest tipus de mètodes d'animació comportamental per a la presa de decisions en un entorns multiagent. Com a part de les aportacions d'aquesta tesi, en el capítol 5 proposarem una tècnica per aconseguir comportaments col·laboratius en grups d'agents basats en planificadors heurístics.

2.4.2 Sistemes basats en regles

Els sistemes basats en regles, o sistemes de producció, són un paradigma de programació adient per al control de sistemes que requereixen declarativitat, modularitat i tècniques d'exploració; com és el cas dels personatges 3D intel·ligents. Com ja fou esmentat en la secció 2.3.2, aquests sistemes empen les regles com a mecanisme de representació del coneixement. De manera general, una regla de producció té el següent format: **SI** antecedent **ALESHORES** conseqüent. L'antecedent és una condició que s'ha de complir perquè la regla siga aplicable i el conseqüent pot ser interpretat com la conclusió a què s'arriba o l'acció que s'ha de realitzar. En aquest context, el raonament, dut a terme per un motor d'inferència, pot ser de dos tipus: cap endavant (*forward chaining*) o cap endarrere (*backward chaining*). El raonament cap endavant parteix dels fets que defineixen l'estat actual i explora les regles per traure les conclusions, essent un exemple de raonament des de les dades cap als objectius. Pel contrari, el raonament cap endarrere va dels objectius cap a les dades, és a dir, a partir de les conclusions que es busquen, cercar els fets que les

produeixen. Un aspecte que han de resoldre ambdós raonaments són els conflictes entre les regles, produïts bé perquè els conseqüents són contradictoris o perquè hi ha més d'una regla que es pot aplicar. Les solucions van des de l'associació d'un valor numèric a cada regla que expresse la seua preferència o probabilitat fins a la definició de meta-regles que especifiquen l'ordre a aplicar quan hi ha diverses regles plausibles.

Un exemple ben conegut de sistema basat en regles és SOAR [Laird et al., 1987], desenvolupat per Laird a la Universitat de Michigan. SOAR permet l'encadenament bi-direccional de regles (*forward/backward chaining*) i la resolució de conflictes mitjançant la definició de meta-regles. És capaç de raonar sobre les regles (accions a executar) a fi d'identificar la més escaient i és capaç de generar automàticament sub-objectius per a resoldre objectius més grans que no es poden resoldre directament. SOAR ha estat utilitzat en els videojocs per a simular NPC controlats per ordinador. Per exemple, la figura 2.13a mostra l'esquema proposat pel SOARBot [Laird and Duchi, 2000], provat sobre Quake II, Unreal Tournament i Descent III, i que implementa aproximadament 800 regles de comportament dedicades a proporcionar intel·ligència tàctica. L'actor pedagògic Esteve [Elliott et al., 1999] també fa servir SOAR per monitoritzar de manera reactiva de les tasques de l'alumne (veure la figura 2.13b).

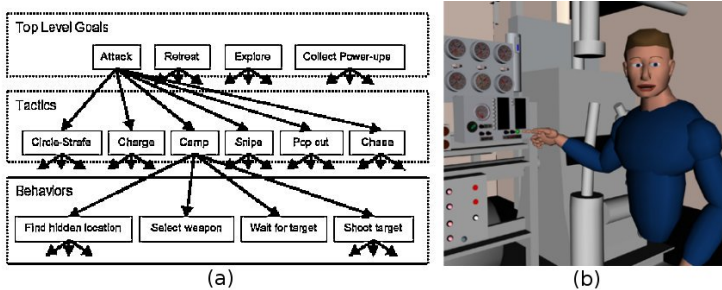


Figura 2.13: Exemples que empren SOAR: (a) Esquema d'operadors utilitzat per SOAR-Bot i (b) Esteve [Elliott et al., 1999].

Un model per a agents intel·ligents que fa servir, típicament, els sistemes basats en regles és el **model BDI**; introduït originàriament per Bratman l'any 1987 [Bratman, 1987]. Aquest paradigma postula que el comportament d'un agent autònom pot ser controlat atenent a tres conceptes: les seues creences (*Beliefs*), els seus desitjos (*Desires*) i les seues intencions (*Intentions*). Les creences d'un agent expressen el seu coneixement sobre el món i sobre el seu estat intern. Les creences poden també incloure regles d'inferència que permeten la creació de noves creences mitjançant raonament cap endavant (p. ex. si

un objecte està pegant-se foc aleshores crema). Els desitjos representen les motivacions de l'agent. Són els objectius d'alt nivell que l'agent persegueix, com ara el desig de no tindre fam o d'anar a una festa. Finalment, les intencions representen l'estat deliberatiu de l'agent, és a dir, allò que l'agent ha decidit realitzar. Açò és, les intencions són desitjos per la consecució dels quals l'agent ha començat a executar un pla. Un pla és una seqüència d'accions que l'agent pot executar a fi d'aconseguir una o més de les seues intencions. Successivament, els plans poden estar formats per altres subplans; per exemple, un pla per a pegar una volta en el cotxe pot incloure un subplan per a buscar les claus del cotxe.

Normalment, el model BDI usa una llibreria de plans creats prèviament per a enfrontar-se a les diverses situacions que li poden aparèixer al llarg de la seua vida. Agents BDI que usen la planificació de tasques en comptes de llibreries de plans precompilats també són possibles [Meneguzzi et al., 2004], però no són comuns. El desavantatge de les tècniques que usen **llibreries de plans preconstruïts** és la seua forta dependència amb el domini per al qual han estat desenvolupades. Tanmateix, la seua senzillesa de desenrotllament i la rapidesa de decisió han fet del model BDI una aproximació que ha estat emprada a l'hora d'implementar 3DIVA. Alguns exemples els trobem en els contingents reactius dels videojocs, com ara l'agent *KGBot* [Kim, 2003] d'*Unreal Tournament* o els NPCs del joc *Black & White* [Molyneux, 2001]. El model BDI ha inspirat des de les arquitectures d'agents individuals racionals [Caicedo and Thalmann, 2000] (veure la figura 2.14) fins el moviment de munions d'agents (*crowds*) [Raupp and Thalmann, 2001].

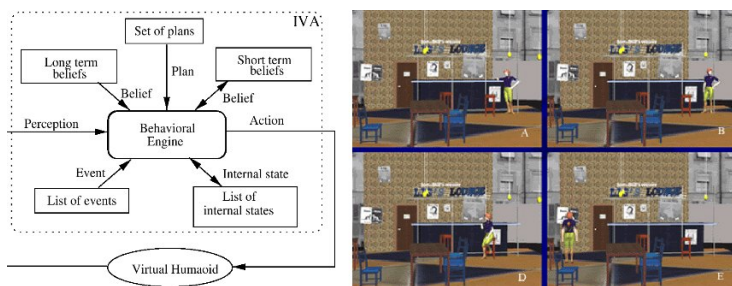


Figura 2.14: Model BDI emprat en [Caicedo and Thalmann, 2000] per a l'animació d'un client en un bar virtual.

El model BDI s'ha emprat extensament com el paradigma base per al disseny de sistemes multiagent de propòsit general (p. ex. Jason [Bordini and Hübner, 2007] o Jadex [Distributed Systems and Information Systems Group - University of Hamburg, 2008]). En el capítol 5 presentarem la nostra aportació social en aquest context: **MADeM (Multi-**

modal Agent Decision Making). MADeM és un procés de presa de decisions de tipus social que pot ser emprat per agents BDI per tal de prendre decisions socialment acceptables en entorns virtuals 3D.

2.5 CONCLUSIONS

El poblament dels mons virtuals 3D amb personatges autònoms és un requisit per assolir la desitjada immersió visual i mental de l'usuari. Tanmateix, aquest objectiu ha esdevingut una tasca força complexa a causa de la quantitat d'aspectes a què s'ha de parar esment. D'una banda, un aspecte que ha demanat una especial atenció ha estat el de la representació de la informació, en uns entorns tan informats com són els EV3D. La presa de decisions dels humans virtuals per interaccionar tant amb els objectes com amb la resta d'agents comporta processos cognitius complexos que requereixen un coneixement abstracte o simbòlic dels elements de l'entorn. D'acord amb açò, l'ús d'informació semàntica ha demostrat ser profitós per a la representació de l'entorn, de les tasques i de les relacions entre els agents. D'una altra banda, l'ús de tècniques provinents de la IA ha permès l'evolució des dels actors amb una bona fidelitat locomotriu als agents que incorporen comportaments racionals. Tal volta un dels aspectes més ansiejats haja sigut la reproducció d'una presa de decisions coherent amb l'estat de l'entorn i als desitjos del personatge. Per raó d'açò, els actors 3D han emprat un ampli ventall de mecanismes per a la reproducció de comportaments reactius i proactius.

En el següent capítol centrem el nostre discurs en els **agents socials**, agents que han estat escassament tractats en la literatura dels 3DIVA. Les dificultats a les quals s'ha enfrontat l'animació comportamental d'un actor individual han fet que els personatges 3D hagen deixat sovint de banda el seu comportament social. Nogensmenys, de la mateixa manera que la IA ha permès evolucionar des dels agents guiats als 3DIVA, convé incorporar el coneixement provinent del camp dels sistemes multiagent i de la sociologia per tal de millorar el comportament de les societats artificials d'actors sintètics.

SOCIETATS ARTIFICIALS

En aquest capítol estudiem el comportament social dels actors sintètics i centrem l'atenció en la **interacció** entre els personatges autònoms. La inclusió d'habilitats socials en la presa de decisions dels caràcters és un aspecte de vital importància als entorns virtuals habitats, ja que aquests conformen una mena de societat artificial. En aquest context, els sistemes multiagent proporcionen un marc de treball elegant i formal pel disseny de comportaments socials per a personatges 3D. En conseqüència, repassem els mecanismes d'interacció multiagent emprats més a sovint per satisfer els objectius d'un individu en contextos poblats per altres entitats autònomes. Analitzem així mateix les xarxes socials, ja que són solucions que van més enllà de la racionalitat individual i inclouen models d'organització interessants per assolir els objectius globals d'una societat.

3.1 COMPORTAMENT SOCIAL EN PERSONATGES 3D

Freqüentment, els actors sintètics no estan a soles en l'entorn, sinó que comparteixen un espai comú d'actuació amb altres personatges amb qui formen una mena de societat artificial. Llavors, els seus comportaments poden traure profit de la **inclusió d'habilitats socials**, cosa que farà que els anomenem **agents socials**. L'afegiment de comportament social no és una tasca senzilla ja que afecta distints nivells d'actuació (p. ex. des de la capacitat de conversar amb realisme a la modificació de l'actuació per atendre ordres externes). A continuació, mostrem un conjunt representatiu de comportaments socials que han estat aplicats sobre personatges 3D. Primerament, introduïm els caràcters proveïts de la capacitat de conversa amb un usuari humà. En segon lloc, parlem de l'establiment de diàlegs i la interacció entre dos caràcters autònoms. En tercer lloc, mostrem algunes solu-

cions proposades per aconseguir comportaments socials en grups de personatges. Finalment, esmentarem com el comportament social ha escalat en la generació de moviments versemblants d'aglomeracions d'actors virtuals.

Una de les expressions socials més característica dels éssers humans és la **capacitat de conversa**. Així doncs, s'han fet molts esforços per proveir els humanoides virtuals amb aquesta habilitat¹. Potser un dels primers exemples fou el treball en caràcters interactius intel·ligents (*Smart Interactive Characters*) que es realitzà a la *Universitat de Stanford* sota la supervisió de Barbara Hayes-Roth [Hayes-Roth, 2008]. Aquest treball s'adreçà a la construcció de *ments* per a caràcters virtuals amb diferents qualitats humanes com ara: identitat i personalitat; afecte i empatia; coneixement i experiència; relacions socials i evolució de les mateixes; etcètera. Els personatges implementats foren principalment assistents i guies virtuals capaços de mantenir una conversa en llenguatge natural amb un usuari humà perquè aquest dirigís el seu comportament. En aquesta línia es troben també els humans virtuals desenvolupats pel laboratori *MIRALab* de la *Universitat de Gènova* [Pandzic et al., 1998]. Aquests humanoides es comuniquen a través d'una interfície de text i un analitzador de missatges. Tot i això, no mostren una interacció autònoma entre diferents actors virtuals, ja que els diàlegs només s'estableixen entre un actor sintètic i l'usuari humà. L'animació de diàlegs entre individus autònoms també ha estat tractada pels agents conversacionals [Cassell et al., 1994, Bickmore and Cassell, 2001]. Emperò, aquests actors paren especial atenció a la simulació gestual, mentre que la conversa és generada automàticament per un planificador simple i normalment incapaç de raonar sobre la informació que està essent comunicada.

El camp d'aplicació del drama interactiu (*interactive drama*) ha emprat accions predefinides per incorporar comportaments socials realistes que sí atenen a la informació intercanviada. La metodologia de W. Scott Neal [Reilly, 1996] per a la creació de comportament social defensa que la personalitat d'un actor influeix de forma tan determinant en el seu comportament que els comportaments universals no són possibles de crear. Per tant, proporciona un conjunt de mecanismes per ajudar l'artista en la creació de comportaments socials particulars per a caràcters específics en entorns concrets. De manera similar, els generadors d'històries interactives (*interactive storytelling*) animen les interaccions entre els personatges de l'escena arran de la trama (*storyline*) global definida per l'autor de la història [Cavazza et al., 2002].

¹Perseguint, així, un objectiu similar al formulat per l'antic test de Turing.

Els projectes suara esmentats únicament inclouen l'animació de pocs individus (normalment, un o dos), tanmateix, el comportament social és encara més escaiant quan hom parla de **grups d'agents**. L'etologia ha inspirat alguns sistemes d'animació de grups d'animals artificials [Tomlinson and Blumberg, 2002, Delgado-Mata and Aylett, 2004]. En aquest context, la transmissió d'emocions mitjançant un model de feromones ha estat un mecanisme provat amb èxit per influenciar el comportament de la resta d'animals del ramat. La recerca en humanoides virtuals també ha desenvolupat models per aconseguir comportaments versemblants a nivell de grup. Per exemple, el sistema *Avatar Arena* [Schmitt and Rist, 2003] defineix un model de dinàmica de grup per a l'establiment de negociacions en grups reduïts. En aquest model, cadascun dels agents té una personalitat individual i unes relacions d'atracció social envers els altres. Aquestes característiques guien les interaccions dels agents i la generació dels diàlegs de negociació entre caràcters.

En els últims temps, els mons virtuals han esdevingut sistemes on múltiples usuaris i personatges autònoms comparteixen un entorn d'actuació comú [Bioware, 2003, Linden Lab, 2008]. Aquests escenaris interactius sovint presenten tasques que han de ser resoltes pels participants de manera col·laborativa. El model SGD (*Synthetic Group Dynamics*) [Prada and Paiva, 2005] és un exemple, basat en teories socio-psicològiques, que ha estat implementat en grups d'actors sintètics que col·laboren amb l'usuari en la resolució de certes tasques dins d'un entorn virtual (veure figura 3.1a). Tocant a la col·laboració entre personatges autònoms, el treball doctoral de Jan Ciger [Ciger, 2005] utilitza un tipus especial d'accions anomenades accions delegades (*delegated actions*). Per mitjà d'aquestes accions, un humà virtual pot passar l'execució d'una acció a un altre agent, sempre que ell no la pugui realitzar. Per exemple, en la figura 3.1b un agent obri la porta a un altre, que està ocupat empenyant una caixa.

El comportament social també ha estat inclòs en l'animació de **munions d'agents virtuals** (*crowds*). La primera aproximació en aquest camp foren els *Boids* desenvolupats per Reynolds [Reynolds, 1987]. Aquests agents perceben la seua rogalia i reaccionaven a un conjunt de regles senzilles. Com a resultat, sorgia un moviment de grup que reproduïa patrons socials versemblants (p. ex. el vol d'un aviram). Benford [Benford et al., 1997] proposà un model que feia servir la comunicació amb un objecte especial anomenat *Third Party Object* i permetia, aleshores, la creació i la separació de grups de personatges virtuals. Darrerament, el model local de forces socials introduït per Helbing [Helbing et al., 2005] ha inspirat sistemes com HiDAC [Pelechano et al., 2007], amb el qual hom pot obtenir comportaments emergents com la formació de fileres d'a-



Figura 3.1: Personatges 3D que col·laboren (a) amb l'usuari humà [Prada and Paiva, 2005] i (b) amb altre caràcter autònom [Ciger, 2005].

gents o embussos en zones amb molta densitat. Nogensmenys, els personatges que trobem generalment a les *crowds* no tenen una concepció social pregona i manquen de les capacitats de construcció de relacions socials. Llavors, són adequats per a la simulació de moviments realistes a nivell macroscòpic però el seu comportament social a nivell de l'individu serà molt limitat.

Com es desprèn dels exemples anteriors, la inclusió de comportaments socials en personatges 3D no és un problema nou. El repte actual, a què ens enfrontem en aquesta tesi, consisteix a ajuntar els models socials amb els models de presa de decisions racional (presentats en 2.4), de manera que es puguin obtenir actors sintètics autònoms amb ambdues capacitats: racionalitat i sociabilitat. Aquest tipus de personatges són requerits pels entorns virtuals habitats com ara els simuladors civils o militars, els jocs per ordinador i els mons virtuals en xarxa d'última generació.

El comportament racional i social ha estat un tema tractat profusament per la comunitat d'agents. Convé analitzar la recerca duta a terme en aquest camp per veure quines solucions poden ser interessants a l'hora de generar **actors sintètics socialment intel·ligents**. De forma general, els agents socialment intel·ligents han estat definits com aquells ens capaços de resoldre problemes de forma autònoma i d'assolir els seus objectius, si escau, per mitjà de la interacció amb unes altres entitats igualment autònomes [Hogg and Jennings, 2001]. Els sistemes multiagent, referits sovint com *societats d'agents*, proporcionen un marc de treball elegant i formal pel disseny de comportaments racionals i socials per a personatges 3D.

3.2 SISTEMES MULTIAGENT

Un agent és una entitat física o virtual capaç d'actuar, de percebre l'entorn i de comunicar-se amb uns altres agents. És autònom, té habilitats per assolir els seus objectius i es troba localitzat a sovint dins d'un sistema multiagent (*Multi-Agent System* o MAS) [Ferber, 1999]. Hom ha definit un MAS com una xarxa feblement acoblada d'agents, els quals interaccionen per resoldre problemes que es troben més enllà de les seues capacitats o del seu coneixement individuals [Sycara, 1998]. D'una manera més formal, podem descriure un MAS amb la *tupla* $\langle E, O, A, R, Op, L \rangle$, on:

- E - és un entorn dimensional.
- O - és un conjunt d'objectes situats en E .
- A - és un conjunt d'agents que compleix que $A \subseteq O$.
- R - és un conjunt de relacions que enllacen els objectes i els agents de O .
- Op - és un conjunt d'operacions que permeten els agents percebre, produir, consumir, transformar i manipular els objectes.
- L - és un conjunt de lleis universals que representen l'efecte de les operacions al món.

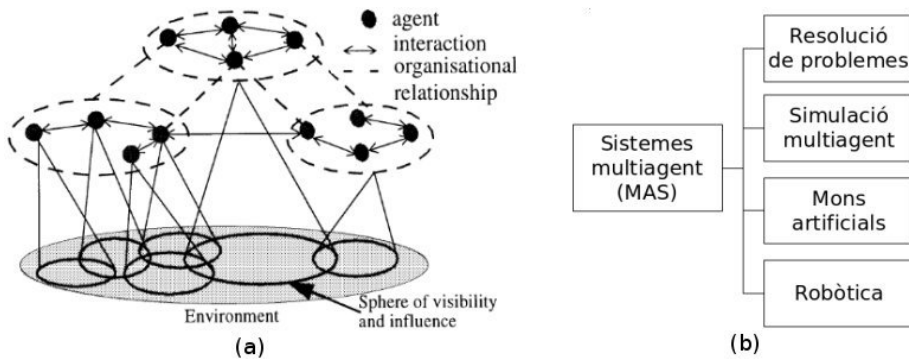


Figura 3.2: Sistemes multiagent: (a) visió general i (b) àrees d'aplicació [Ferber, 1999].

La figura 3.2a mostra una visió general de l'estructura d'un sistema multiagent. En la part inferior trobem l'entorn compartit que els agents ocupen. Cadascun dels agents

té una esfera d'influència en aquest entorn, açò és, una porció que pot controlar total o parcialment. Quan l'entorn es controla de forma conjunta amb altres agents, les esferes se solapen. Aquesta intersecció complica la vida dels agents, ja que per aconseguir l'estat del món desitjat, els agents han de tindre en compte també l'actuació d'altres agents. La part superior de la figura mostra les interaccions que es produeixen entre els agents així com les relacions organitzacionals que els lliguen (p. ex. un agent pot ser company d'un altre o tindre autoritat sobre un altre). Atenent a la figura 3.2b, l'aplicació dels sistemes multiagent abasta quatre àrees principalment:

Resolució de problemes: L'ús d'un MAS pot ser una bona alternativa a la resolució centralitzada de problemes; bé perquè la natura del problema siga distribuïda o bé perquè la distribució entre un conjunt d'agents supose una forma més eficient d'organitzar la resolució. Alguns exemples són la monitorització distribuïda d'una xarxa, les aplicacions de disseny industrial o els problemes basats en restriccions.

Simulació multiagent: Els MAS permeten crear dominis artificials que actuen com xicotets laboratoris en els quals provar teories sobre comportaments locals. La simulació multiagent és utilitzada hui dia per a millorar el coneixement en camps diversos (p. ex. ciències socials, educació, biologia...).

Mons artificials: Els quals poden ser utilitzats per descriure mecanismes d'interacció específics i analitzar el seu impacte en el sistema a nivell global. De manera anàloga al joc de la vida (*Game of Life*) [Gardner, 1970], l'objectiu d'aquesta àrea és la construcció de societats d'agents i l'estudi de la seua evolució.

Robòtica: Un robot pot ser concebut com un MAS en què cada agent s'encarrega d'un objectiu específic. La consecució conjunta de tasques més senzilles i concretes farà doncs que l'objectiu global es complete. Així mateix, els MAS poden ser usats per a la coordinació de diversos robots mòbils que es troben situats dins d'un espai comú.

Els entorns virtuals 3D habitats encaixen, doncs, dins de l'àrea d'actuació dels MAS. En aquests mons simulats, un conjunt d'entitats autònomes hauran de treballar de forma conjunta per poder obtenir com a resultat una mena de comportament intel·ligent a nivell col·lectiu. D'acord amb açò, alguns investigadors han optat per desenvolupar el seu propi sistema multiagent, com ara els sistemes *mVITAL* [Anastassakis et al., 2001] i *SimHuman* [Vosinakis and Panayiotopoulos, 2001]. Darrerament, els MAS de propòsit general (p.

ex. *Jason* [Bordini and Hübner, 2007]) han estat proposats també com a bones aproximacions per a l'animació comportamental d'humans autònoms en entorns tridimensionals [Torres et al., 2003].

La **intel·ligència col·lectiva** que hom desitja obtenir no pot ser generada amb les tècniques tradicionals de la Intel·ligència Artificial. Aquestes centren l'atenció en els processos cognitius de l'agent i els tracten normalment com sistemes independents; ja que segueixen la premissa que la intel·ligència és una propietat de l'agent. Ans al contrari, els MAS són estudiats per la d'Intel·ligència Artificial Distribuïda (*Distributed Artificial Intelligence* o DAI). La DAI considera que, sense deixar de banda les capacitats cognitives individuals, la intel·ligència sorgeix com el resultat de la **interacció** entre els agents; els quals formen part d'un sistema interconnectat a través de processos socials. En conseqüència, la recerca en MAS s'ha centrat en la interacció, com a base per a l'enteniment del comportament i l'evolució d'un sistema multiagent. En el següent punt identifiquem diferents tipus d'interacció i definim els models operacionals a què condueixen.

3.3 INTERACCIÓ EN SISTEMES MULTIAGENT

La investigació al voltant de les interaccions en sistemes multiagent sovint ha encunyat el comportament amb adjectius com *coherent*, *col·laboratiu*, *cooperatiu*, *competitiu* o *coordinat*. Diverses aproximacions han tractat d'organitzar i relacionar tots aquests termes, a l'hora de categoritzar amb precisió com interaccionen els agents. Potser una de les més brillants de totes és la taxonomia que considera tots aquests termes com especialitzacions d'un concepte abstracte anomenat **correlació** [Parunak et al., 2002].

La manera més general de mesurar com un conjunt d'agents fan coses junts és mitjançant la seua informació conjunta, també coneguda com entropia conjunta o de correlació. La correlació és un terme teòric que correspon amb la no-independència estadística entre els comportaments dels agents, sense atendre a l'estructura interna ni a l'organització externa. Fonamentalment, un agent està correlacionat quan les seues accions són estadísticament dependents de les accions dels altres. Aquesta dependència provindrà d'un flux d'informació entre un agent individual i l'exterior; és a dir, l'estat de l'entorn i la resta dels agents. Aquest fet fa que l'anàlisi de la comunicació siga de capital importància en l'estudi de la interacció.

		Relació entre els agents	
		Centralitzada	Descentralitzada
Flux d'informació	Directe	Comandament	Conversa
	Indirecte	Restricció	Competició

Taula 3.1: Categories de comunicació.

3.3.1 Comunicació

El flux d'informació entre un agent i l'entorn que l'envolta pot trametre's de dues maneres: directament i indirecta. Mentre que la primera duu associat un intercanvi de missatges entre els agents, la segona aprofita la capacitat comunicativa del mateix entorn, el qual pot emmagatzemar informació perquè siga percebuda per altres agents posteriorment. Tocant a la relació entre els agents, distingim dos tipus de comunicació: centralitzada i descentralitzada. En la primera categoria trobem les relacions jeràrquiques en què un agent *cap* mana sobre un agent *subordinat*. Inversament, en la comunicació descentralitzada tots els agents es parlen d'igual a igual, en anglès *peer-to-peer*. La taula 3.1 classifica diversos **tipus de correlació** depenent del mode de comunicació de la informació i de les relacions entre els agents.

La correlació en una topologia centralitzada pot ser exercida per un agent supervisor fent servir comandaments envers els subordinats o bé modificant certes propietats de l'entorn per restringir de forma indirecta la seua actuació. Un exemple de **comandament** el trobem en un paleta virtual que dóna ordres als seus manobres en una escena 3D. Pel que fa a la **restricció**, el paleta podria proveir els manobres d'un conjunt de recursos determinats perquè realitzaren les seues tasques. La comunicació directa en topologies descentralitzades, també anomenada **conversa**, ha estat tractada extensament per la recerca en negociació entre agents intel·ligents així com pels agents virtuals conversacionals (veure el punt 3.1). La comunicació indirecta descentralitzada es produeix quan els agents perceben els canvis de l'entorn a causa de les accions dels altres agents. El problema més comú dins d'aquesta categoria és la **competició**, la qual succeeix quan els agents volen accedir a recursos limitats i compartits. La competició, una situació molt corrent als entorns virtuals 3D habitats, demana l'ús de tècniques de coordinació per evitar la interferència entre les accions realitzades pels agents (recordem els exemples del fuster i el cambrer introduïts en el punt 1.1).

3.3.2 Coordinació

Un dels tipus d'interacció emprat més a sovint per la comunitat dels sistemes multiagent és la **coordinació**. Hom pot concebre la coordinació com un cas específic de correlació en què hi ha implicat un procés de **comunicació** entre l'agent i el context que l'envolta. D'acord amb Bergenty i Ricci [Bergenty and Ricci, 2002], la coordinació en MAS ha fet servir diferents formalismes que podem classificar en les següents vessants: centres de tuples, protocols d'interacció, semàntica dels llenguatges de comunicació d'agents. Totes tres tenen els seus avantatges, però també reben certes crítiques.

Els **centres de tuples** segueixen la filosofia dels models de pissarra, en els quals no es requereix contacte ni físic ni temporal entre els agents. El seu ús comporta una solució no acoblada per a la coordinació, ja que la comunicació i el processat es realitza en els centres de tuples; és a dir, fora dels agents. En conseqüència, aquests models són adequats en entorns dinàmics i oberts. Tanmateix, els seus crítics addueixen que són sistemes limitats per raó de la centralització i del fet que freqüentment només atenen el comportament observable dels agents i no realitzen cap suposició sobre les actituds mentals.

Els **protocols d'interacció** són descrits mitjançant una màquina d'estats finits. Els estats d'un protocol identifiquen els estadis en què els agents poden estar al llarg de la comunicació i les transicions representen els missatges intercanviats entre un emissor i un receptor. Diversos llenguatges de comunicació d'agents (*Agent Communication Languages* o ACL) han estat desenvolupats. Els més destacats són KQML [Finin et al., 1994] i FIPA ACL [FIPA (Foundation for Intelligent Physical Agents), 2008] (veure la figura 3.3). Ambdós basen la comunicació en un intercanvi asíncron de missatges. Els missatges s'estructuren seguint la teoria coneguda com actes de la parla (de l'anglès *speech-acts*), la qual fa servir fonamentalment dos camps: el verb corresponent al tipus de missatge (nomenat *performative*) i el contingut intercanviat. Els protocols d'interacció han estat reprovats per la seua escassa descripció formal (p. ex. les màquines d'estats no solen especificar formalment el contingut dels missatges). Altres crítiques han vingut de l'alt nivell d'acoblament d'aquesta solució, que moltes vegades fa que una part útil del codi de l'agent es trobe barrejada amb codi exclusiu per a la comunicació.

La **semàntica dels ACL** aposta per la introducció de les accions comunicatives en el procés de planificació d'un agent, junt amb la resta d'accions possibles. En aquest context, les lleis de coordinació emergeixen de la interacció entre els objectius de l'agent

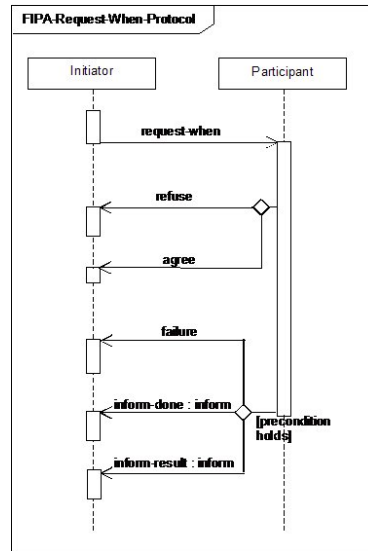


Figura 3.3: Protocol d'interacció Request when en FIPA.

i les restriccions en la consecució dels susdits objectius. Açò és, són el resultat dels efectes i les precondicions dels missatges i de les accions realitzades per l'agent. Llavors, la interacció no està dissenyada d'antuvi sinó que sorgeix com a resultat del procés de planificació, una missió força complexa per un planificador quan el món deixa de ser trivial.

Els actors sintètics que poblen els mons virtuals necessiten emprar la coordinació per evitar el caos i estalviar-se interferències innecessàries que poden sorgir si cada caràcter va només a la seua. Malgrat que els humans virtuals no requereixen en principi l'animació de plans òptims, és clar que la versemblança d'un grup d'individus depén en gran mesura de la coherència de les accions envers els altres. Aquesta coherència pot ser aconseguida amb agents que executen plans coordinats propers a l'òptim i que s'adapten a canvis impredecibles.

Coordinació de plans

El problema de la coordinació de plans en entorns multiagent (*Multiagent Plan Coordination Problem* o MPCP) apareix quan un conjunt d'agents ha de planificar la seua actu-

ació de manera individual però poden traure profit amb la coordinació dels seus plans. Diferents perspectives han estat estudiades dins del MPCP [Durfee, 1999, de Weerd, 2003]. La coordinació amb **agents especialitzats** és un exemple en què els individus són dotats amb la capacitat exclusiva de realitzar certes accions [Kambhampati et al., 1991] o de manipular certs objectes [Brenner, 2003]. Aquesta aproximació redueix la complexitat del problema de coordinació però crea agents predictibles. Nogensmenys, els personatges 3D orientats a realitzar tasques en un món virtual generalment comparteixen les capacitats d'actuació; cosa que fa molt important la consideració del grau de dependència entre les activitats i els objectius dels agents.

Cox i Durfee [Cox and Durfee, 2005] han introduït recentment una tècnica eficient de **post-planificació** per a produir plans òptims i coordinats en agents quasi independents. Hi ha tot un seguit d'investigacions en coordinació que versen sobre com sincronitzar de forma eficient els plans després que aquests ja hagen estat construïts [Georgeff, 1983, Ephrati and Rosenschein, 1994, Yang, 1997]. Un exemple ben representatiu és *Generalized Partial Global Planning* (GPGP) [Decker and Lesser, 1997]. En aquesta aproximació els plans amb les accions que volen realitzar els agents són intercanviats. Posteriorment, els agents barregen aquests plans i ordenen les accions fent servir unes relacions que informen sobre quines poden facilitar l'execució d'altres (veure la figura 3.4). El criteri és maximitzar una funció d'utilitat global que normalment està associada amb la rapidesa en l'execució del pla. Tot i això, l'us de plans complets resultat d'una fase de post-planificació no és recomanable en els entorns virtuals. L'alt dinamisme d'aquests entorns, on els usuaris i els personatges actuen de manera asíncrona, farà que els plans siguin inaplicables en poc de temps i hagen de ser avortats. La coordinació en entorns asíncrons ha estat tractada amb agents que intercalen planificació i acció [Durfee, 1999]. De vegades, la coordinació s'ha aconseguit mitjançant planificadors STRIPS amb noves precondicions; les quals representen restriccions que s'apliquen en temps de planificació [Boutilier and Brafman, 2001]. Unes altres vegades, és una fase de **pre-planificació** la que aplica algun tipus de restricció, com per exemple certes lleis socials [Shoham and Tennenholtz, 1995].

Com ja hem dit, la correlació mostrada per una població d'agents és un terme abstracte que no fa referència ni a l'organització externa de la societat ni a l'estructura interna de l'agent. L'estudi dels processos de comunicació i de les interrelacions entre els agents ens ha dut a l'anàlisi de la coordinació. En el següent punt analitzem la col·laboració com una expressió de la lògica interna d'un agent.

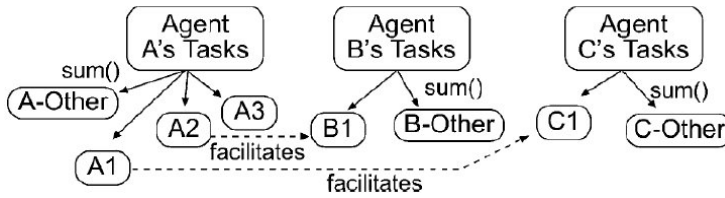


Figura 3.4: Exemple d'ordenació de les tasques de tres agents en GPGP.

3.3.3 Col·laboració

Una societat artificial pot mostrar correlació sense que els seus membres tinguin cap intencionalitat envers els altres. Per exemple, en la competició per un recurs compartit, uns agents podrien coordinar-se mentre romanen lliures de malícia o de bondat. Tanmateix, hom pot identificar dos tipus especials de correlació que apareixen lligats a les intencions internes d'un agent: la **cooperació** i la **contenció**. La cooperació requereix l'existència d'unes intencions de treball conjunt amb la resta d'agents. Ben al contrari, la contenció implica la determinació de frustrar les intencions d'un altre. Llavors, és fàcil comprendre perquè aquests qualificatius només haurien de ser atribuïts als agents cognitius; els quals tracten d'imitar les representacions i els processos cognitius humans. La cooperació no duu associada que la comunicació entre els agents haja de ser directa o indirecta. Un cas que mereix especial atenció és aquell que es produeix quan els agents cooperen i conversen directament, situació reconeguda com **col·laboració**.

Una peça clau en l'estudi del treball en equip (*teamwork*) ha estat la **teoria de les intencions conjuntes** (*joint intention theory*) [Cohen and Levesque, 1991]. Segons aquesta teoria, la formació d'un equip necessita: a) que tots els agents implicats hagen adquirit el compromís d'aconseguir un objectiu comú i b) que reben aquest mateix compromís de la resta de membres de l'equip. Wooldridge i Jennings desenvoluparen un model formal de resolució col·laborativa de problemes basat en l'ús de compromisos conjunts així com de convencions o normes per definir l'admissibilitat d'un compromís [Wooldridge and Jennings, 1994]. Aquest paradigma ha inspirat diverses arquitectures de simulació multiagent. Destacats en aquest àmbit són els treballs realitzats per Milind Tambe sobre el seu sistema STEAM [Tambe, 1997]. STEAM utilitza un líder centralitzat per a la formació dels equips i la resolució de conflictes. Aquest agent divideix els objectius globals de l'equip en subobjectius que hauran de ser aconseguits per rols diferents. Una vegada assignats els rols, se suposa que els membres de l'equip poden completar les

seues tasques de manera independent (veure la figura 3.5). Malauradament, aquesta situació ideal és difícil d'aconseguir en els mons virtuals, on diferents actors poden interferir quan executen les seues accions.

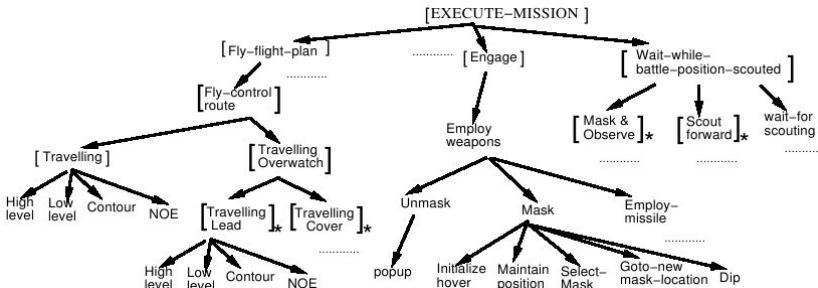


Figura 3.5: Descomposició d'una missió d'atac militar [Tambe, 1997]. Els operadors marcats amb * corresponen a operadors d'equip.

Els **plans compartits** (*SharedPlans*) [Grosz et al., 1999] han tractat la col·laboració com el problema d'elaborar un pla conjunt que tinga en compte l'estat mental i els objectius de múltiples participants. Aquesta aproximació necessita que els agents coneguen de primer moment el model de l'equip, és a dir, totes les actuacions possibles per aconseguir l'objectiu (encara que d'algunes no tinguen detalls específics perquè no les faran ells). Així mateix, hauran d'acordar certs procediments de decisió, com ara l'assignació d'agents a accions, de manera que puguin completar el pla global. *SharedPlans* s'ha aplicat amb èxit en diferents dominis: des del comerç electrònic a interfícies col·laboratives en sistemes d'aprenentatge a distància.

Finalment, la **delegació** de tasques també ha estat emprada per obtenir col·laboració en societats amb un model jeràrquic de responsabilitat [Ioeger and Johnson, 2001]. Dins d'aquestes societats, alguns agents líders poden encarregar l'execució de certes tasques directament a altres agents subordinats. Un sistema influent en aquesta àrea és *Open Agent Architecture* (OAAA) [Martin et al., 1999]. Aquest sistema fa servir un *facilitador* de serveis com ara: la comunicació entre agents distribuïts o la subdivisió de tasques complexes en trossos més menuts que puguin ser executats independentment per múltiples agents. En comptes d'usar un agent facilitador central, el sistema RETSINA [Giampapa and Sycara, 2002] fa servir la tècnica de les pàgines grogues (directoris d'agents que ofereixen certes capacitats). RETSINA intercala planificació amb execució i dóna suport a la col·laboració mitjançant un planificador que modela les tasques a través de xarxes jeràrquiques de tasques (HTN). Per tant, els actes comunicatius es troben

predefinits en l'arbre HTN. Llavors, el planificador en temps d'execució pot sol·licitar informació així com l'execució d'accions a un altre agent i suspendre eixa línia d'actuació fins que arribe el resultat.

La recerca en MAS ha estudiat els diferents tipus de correlació com un mecanisme per desenvolupar sistemes més productius. Nogensmenys, la racionalitat no és l'únic camí per aconseguir la **congruència**, definida com el grau en què les interaccions entre els agents satisfan les expectatives generals del sistema. Per exemple, els personatges 3D sovint operen en entorns compartits amb recursos limitats, cosa que apunta a l'ús de tècniques de coordinació i col·laboració. Ara bé, els actors sintètics normalment representen rols humans, com ara un cambrer i un client en un bar virtual. Aleshores, hom espera que aquestes relacions socials també es manifesten en el seu comportament. El raonament social ha estat estudiat profusament en la comunitat dels MAS per incorporar **elements socials** en els agents cognitius [Conte and Castelfranchi, 1995, Conte, 1999]. Com a resultat, els models d'interacció entre els agents han evolucionat cap a les xarxes socials, les quals tracten d'imitar les estructures socials que hom pot trobar a la realitat [Hexmoor, 2001].

3.4 XARXES SOCIALS

La component social de la interacció ha rebut una atenció considerable en el camp dels sistemes multiagent [Jennings and Campos, 1997, Castelfranchi, 1998]. La recerca d'elements socials que influencien la presa de decisions dels membres d'una societat artificial ha versat sobre temes com ara: els **rols**, les **relacions** de poder o confiança, les **normes**, les **preferències** dels agents, etc. De manera general, podem afirmar que tot agent immers en un grup juga un paper o rol; el qual determina les seues responsabilitats. Així mateix, els rols d'una societat defineixen sovint relacions entre els agents (p. ex. la relació de poder que existeix entre un agent líder i un subordinat). Aquestes relacions poden fer-se més complexes i arribar a construir xarxes que representen l'estructura d'una organització. Llavors, l'actuació dels agents es veu restringida, ja que han d'obeir una sèrie de regles o normes dins de l'organització. En aquest context, els agents poden canviar d'un grup a un altre quan les seues preferències internes no coincideixen amb els requeriments de l'organització. Hi ha un ampli ventall de treballs que han tingut com assumpte les normes i els contractes de compromís per formalit-

zar les restriccions que defineixen els comportaments acceptables dins d'una organització [Shoham and Tennenholtz, 1995, Agotnes et al., 2007]. L'anàlisi exhaustiva d'aquest problema queda fora de l'abast d'aquesta tesi.

Les organitzacions i les **xarxes socials** humanes han estat un tema d'estudi durant molt de temps [Scott, 1991]. L'objectiu de l'anàlisi de les xarxes socials és comprendre els lligams entre els individus (com els individus es relacionen els uns amb els altres) així com extraure certes propietats associades al grup sencer. La forma més comuna de representar les relacions socials d'un grup és mitjançant un **sociograma**. Un sociograma és un graf on els nodes solen representar agents o predicats mentre que els arcs que els enllacen corresponen a les relacions. Identifiquem tres categories principals de xarxes socials atenent al significat d'aquests arcs: a) les xarxes de dependència, b) les xarxes de confiança i c) les xarxes de preferència.

3.4.1 Xarxes de dependència

L'autonomia dels agents es troba de vegades limitada per xarxes d'interdependència, on els agents depenen els uns dels altres per aconseguir els seus propis objectius. La **teoria de la dependència**, proposada per diversos autors [Castelfranchi et al., 1992, Sichman et al., 1994], considera que un agent i és dependent d'un altre agent j pel que fa a un objectiu p quan: i manca almenys d'una de les accions o recursos necessaris per assolir p , mentre que j pot realitzar la susdita acció o està en possessió del recurs adient. Al contrari, hom diu que l'agent j té poder sobre l'agent i . Per exemple, un treball conjunt en equip pot ser representat com una agregació on cada agent depèn de la resta per arribar a obtenir un objectiu compartit (veure la xarxa completa de la figura 3.6a). Els entorns de mercat també poden ser vists com xarxes de dependència, encara que menys cohesionades (veure la figura 3.6b). En aquests entorns, el desig comú perquè el mercat genere beneficis fa que les empreses implicades facen tractes i tinguen socis de negocis [Conte, 1999].

La teoria de la dependència dóna resposta, així, a les dues preguntes fonamentals en una societat d'agents autònoms. Un agent aconsegueix que el seu objectiu personal esdevinga social, açò és, que siga adoptat per uns altres agents, gràcies al seu poder social sobre aquests. De manera anàloga, un agent autònom forma part d'una interacció social per raó de la dependència social que el lliga a un altre agent.

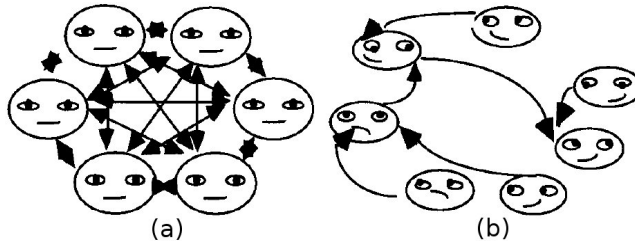


Figura 3.6: Exemples de xarxes de dependència: a) treball conjunt en equip i b) entorn de mercat.

El **raonament social** [Sichman and Demazeau, 2001] es basa en la teoria de la dependència per implementar la formació dinàmica de coalicions. El disseny de l'estructura d'una organització amb una xarxa de dependència és interessant. Nogensmenys, el model de l'organització no pot ser concebut en temps de disseny als mons oberts, on els agents poden unir-se i deixar la societat en qualsevol moment. Com que aquest tipus d'agents normalment no disposa d'una representació correcta i completa dels altres, aquesta aproximació incorpora un procediment de revisió de creences. Finalment, el model distingeix dos tipus d'interaccions socials d'acord amb les coalicions acceptades: la cooperació (diversos agents que treballen de forma conjunta per assolir un objectiu comú) i l'intercanvi social (diversos agents que treballen de forma conjunta per assolir diferents objectius individuals).

3.4.2 Xarxes de confiança

Els agents sovint són considerats entitats autònomes en les quals no és possible exercir control sobre el seu comportament intern. Dins d'un MAS, els agents es veuen involucrats en la realització de tasques col·lectives que requereixen la cooperació i la refiança en altres agents. La possibilitat que un agent no es comporte de manera correcta, intencionadament o no, produeix riscos de fiabilitat i robustesa a nivell del sistema, especialment en el cas de sistemes oberts. Hom ha proposat com a solució l'ús de **models de confiança** (*trust models*). Un dels primers treballs en models de confiança fou el realitzat per Steve Marsh al Regne Unit [Marsh, 1994]. El seu model distingeix tres nivells de confiança: bàsic, general i situacional. La confiança bàsica (*Basic Trust*) és l'actitud general de l'agent per confiar en els altres. La confiança general (*General Trust*) correspon amb la confiança genèrica dipositada sobre un agent concret. Finalment, la confiança situacional (*Situational*

Trust) és l'actitud de confiança envers els altres en circumstàncies específiques.

Els models de confiança proveeixen els agents amb models d'altres agents perquè puguin decidir si confien en ells o no. Habitualment, la font d'informació que s'empra en la presa de decisions confiada és la **reputació** [Castelfranchi and Falcone, 1998]. Paradoxalment, no existeix una definició comunament acceptada per aquest terme. De bell principi, la reputació fou considerada un valor únic, associat a un agent i mantingut pel sistema, que podia fer referència a la diligència. Les xarxes socials han permès que dos agents tinguin opinions diferents sobre la reputació d'un tercer [Sabater and Sierra, 2002]. Darrerament, ha estat proposada una ontologia funcional de reputació per permetre la interoperabilitat entre diferents models de confiança [Vercouter et al., 2007].

Això no obstant, a l'hora de confiar o delegar tasques en una xarxa de confiança, altres fonts d'informació també poden ser tingudes en compte. Per exemple, el treball de Rino Falcone [Falcone et al., 2004] fa servir com a estratègia de delegació, les habilitats per a executar certes tasques (*ability*) o la predisposició per adoptar tasques externes (*willingness*). A més a més, incorpora un mecanisme per alterar aquests valors en funció de l'estat de l'entorn (veure la figura 3.7).

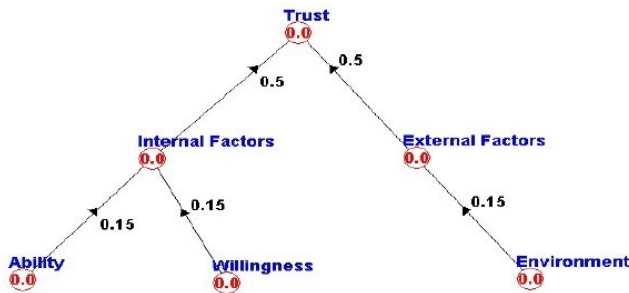


Figura 3.7: Estratègia de delegació de tasques amb diversos criteris de confiança definida en [Falcone et al., 2004].

3.4.3 Xarxes de preferència

Hom pot entendre la presa de decisions com el procés cognitiu que permet seleccionar una alternativa d'entre un conjunt de possibilitats. En entorns col·lectius, com ara els sistemes multiagent, aquestes decisions haurien de tindre en compte també les opinions

externes. D'acord amb açò, les xarxes de preferència permeten que els agents expressen les seues **preferències individuals** mitjançant una relació d'ordre sobre el conjunt d'alternatives. Mentre que aquest conjunt siga reduït, els problemes són computacionalment senzills. Tanmateix, molts problemes presenten un conjunt d'alternatives amb estructura combinatòria. Llavors, l'elecció del llenguatge de preferència ha estat un problema tractat pregonament en el camp de la selecció social (*social choice*) [Lang, 2005]. En la pràctica, els agents normalment expressen les seues preferències fent servir **funcions d'utilitat**. El propòsit d'una funció d'utilitat és assignar un valor numèric que expresse l'opinió de l'agent sobre cadascuna de les solucions a un problema donat. Una antiga qüestió que afecta les funcions d'utilitat fa referència a la dificultat d'expressar una decisió humana en termes numèrics. Aquesta dificultat, apuntada ja per Jacques Ferber [Ferber, 1999], encara resta per resoldre i necessitarà l'aportació d'altres àrees de coneixement com ara la filosofia i la psicologia.

La teoria de la selecció social no determina un mecanisme concret per a l'obtenció d'aquestes preferències. Depenent de la natura del problema, s'han utilitzat diversos **protocols** [Sandholm, 1998] com ara: les tècniques de votació, les subhastes, la negociació bilateral, la negociació distribuïda, etc. Des d'un punt de vista estratègic, la teoria de jocs (*game theory*) [Osborne, 2004] ha estat la branca de les ciències econòmiques encarregada d'estudiar la presa de decisions en aquests contextos en què els agents no han de ser necessàriament veraços.

Una vegada han estat obtingudes les preferències dels agents de la societat, cal computar quina és la solució socialment més adequada. Aquest problema ha estat tractat sota el formalisme de la repartició de recursos en entorns multiagent (*Multi-Agent Resource Allocation* o MARA) [Chevalyre et al., 2006]. A l'hora de prendre aquesta decisió, MARA emprà el concepte de benestar social (*social welfare*) provinent de l'economia del benestar (*Welfare Economics*). Aleshores, les funcions de benestar social s'usen per definir diversos ordres socials per mitjà de l'agregació de les preferències individuals. En el cas habitual en què les preferències s'expressen mitjançant funcions d'utilitat, l'**ordre social** es defineix a través de **funcions d'utilitat col·lectives** (*Collective Utility Functions* o CUF) [Sandholm and Suri, 2001].

Per entendre el funcionament de les xarxes de preferència, imaginem el següent joc mental: "*Oriol i Marta volen eixir una nit i es plantegen dues possibilitats: anar al cinema o anar al teatre. Suposem que Oriol gaudeix més del teatre i que li assigna un valor*

Solucions/CUFs	Utilitària (Σ)	Igualitària (min)	Elitista (max)
Anar al cinema	7	3	4
Anar al teatre	10	2	8

Taula 3.2: Diferents benestars socials expressats amb CUFs.

8, mentre que la utilitat que dona a anar al cinema és 3. Al contrari, les preferències de Marta són contràries i menys vehements, 4 per anar al cinema i 2 per anar al teatre". La taula 3.2 mostra els diferents ordres socials. Una societat utilitària agafarà aquella solució que maximitza la utilitat total del grup (sumatori de les utilitats), mentre que una societat igualitària agafarà aquella solució que beneficia el més feble del grup (mínim de les utilitats). Finalment, una societat elitista agafarà aquella solució que seguirà les preferències de qui més pot guanyar amb la resolució del problema (màxim de les utilitats).

En les xarxes de preferència, la capacitat d'un agent de modificar amb un diferencial d'importància les preferències rebudes per altres agents permet la definició d'**actituds socials** [Brainov and Sandholm, 1999]. Per exemple, un agent egoista reduiria les utilitats rebudes enfortint d'aquesta manera la seua opinió. Al contrari, un humanoide altruista incrementaria les utilitats rebudes en detriment de la seua preferència. El capítol 5 d'aquesta tesi analitza amb més detall l'expressió de preferències socials amb funcions d'utilitat i l'aplicació de diferencials per modelar actituds com ara la indiferència, la reciprocitat, l'egoisme i l'altruisme.

Les aproximacions basades en MARA han estat aplicades en un rang molt ample d'aplicacions reals [Chevaleyre et al., 2006]: gestió de trànsit aeri, logística, aprovisionament industrial, explotació conjunta de satèl·lits d'observació de la Terra, assignació de recursos en arquitectures mallades (*grid computing*), etc. En aquesta tesi proposarem el seu ús per a la presa de decisions socials d'actors sintètics que habiten mons virtuals.

3.5 CONCLUSIONS

Amb freqüència, els actors sintètics no estan a soles en l'entorn, sinó que comparteixen un espai comú d'actuació amb altres personatges; amb qui formen una mena de societat artificial. Tanmateix, la inclusió de comportaments socials en personatges 3D s'ha

limitat normalment a la interacció versemblant amb l'usuari, mentre que les interaccions amb altres caràcters autònoms han sigut escasses. El repte actual consisteix a incorporar el coneixement que prové de l'àrea dels sistemes multiagent, de manera que es puguin obtenir actors sintètics autònoms proveïts d'ambdues capacitats: racionalitat i sociabilitat. D'acord amb açò, en aquest capítol s'han repassat els diferents models d'interacció que hom pot trobar en un conjunt d'agents que treballen de manera conjunta. Cal destacar la col·laboració, ja que correspon a la situació en què els agents cooperen i conversen directament. Pel que fa als elements socials de la interacció, les xarxes socials han estat presentades com a model de reproducció de les relacions humanes reals. Entre elles, les xarxes de preferència han demostrat ser un mecanisme molt flexible a l'hora de prendre decisions en un entorn social.

La tercera part d'aquest treball (capítols 4, 5 i 6) conté les principals aportacions a l'estat de l'art que ara conclou. En primer lloc, definim un model general d'entorn virtual semàntic basat en ontologies per a millorar la sensorització d'escenes complexes, la interacció agent-objecte i la definició de relacions entre els agents d'una societat artificial (capítol 4). En segon termini, descrivim dos mecanismes per a la inclusió d'habilitats socials en dos dels paradigmes més comuns emprats pels agents racionals. Més concretament, en el capítol 5 presentarem: a) una tècnica de sospesament d'objectius per aconseguir comportaments col·laboratius en grups d'agents basats en planificadors heurístics; i b) MADeM (*Multi-modal Agent Decision Making*), un procés de presa de decisions de tipus social social que fa servir les preferències d'altres agents per tal de prendre decisions socialment acceptables. Darrerament, en el capítol 6 mostrarem els resultats obtinguts i el grau de comportament social assolit pels personatges 3D que utilitzen aquest mecanismes socials.

PART III

APORTACIONES

MARC DE SIMULACIÓ

En aquest capítol, presentem les aportacions menors d'aquesta tesi, arrellegades en el **marc de simulació multiagent** dissenyat per a la integració d'actors sintètics amb habilitats socials en entorns virtuals 3D. Fonamentalment, aquest marc es compon de: a) un model d'**entorn virtual semàntic** per a la construcció i manteniment del món virtual; i b) una arquitectura base d'agents en la qual poder integrar tècniques de raonament social. L'entorn virtual semàntic incorpora l'ús d'**ontologies** per a la representació de la base de coneixement del món i de les relacions entre els agents 3D. Aquest coneixement semàntic beneficia la producció, la percepció i la interacció dels personatges en escenaris 3D complexos. Tocant a l'**arquitectura d'agents socialment intel·ligents**, repassem els mòduls principals requerits per controlar el comportament autònom dels personatges 3D. En aquest context, tractem el problema de l'actuació sobre els objectes de l'entorn (interacció agent-objecte) i de la comunicació entre els personatges (interacció agent-agent). La presa de decisions de tipus social, principal aportació d'aquesta tesi, serà tractada en el capítol 5.

4.1 INTRODUCCIÓ

Com es desprèn del capítol 2, l'evolució de l'animació comportamental dels personatges 3D ha anat de la mà del nivell de representació de l'estat del món. En certa manera, la interacció és directament proporcional a la quantitat de coneixement que els agents perceben. Els entorns virtuals 3D clàssics han emprat diversos mecanismes de representació; des dels grafs d'escena a les tautologies d'objectes. En aquests entorns, les capacitats d'interacció dels seus habitants es reduïen normalment a un conjunt d'objectes

o de protocols de comunicació predefinitos entre els caràcters. Tanmateix, la incorporació de tècniques de raonament cada vegada més sofisticades, com és el cas del comportament social, ha promogut la separació entre el **model semàntic**, el model geomètric i el model comportamental del món. Pel que fa al model semàntic, un dels reptes actuals es troba en la cerca d'una solució genèrica per a la representació semàntica dels mons virtuals 3D; de manera que afavorisca la interacció entre els objectes i els agents que el poblen.

D'acord amb açò, en el punt 4.2 descrivim el **marc de simulació multiagent** dissenyat per a la generació d'entorns virtuals 3D habitats per actors sintètics amb habilitats socials. El model general d'**entorn virtual semàntic** (SVE), inclòs en aquest sistema, es tracta en el punt 4.3. Proposem les ontologies com una solució extensible i reutilitzable per a la representació de l'entorn i de les relacions entre els actors (secció 4.3.1). La construcció d'escenaris complexos, amb nombrosos objectes interactius, així com la percepció de la gran quantitat d'informació que aquests entorns produeixen són qüestions que es tracten en la secció 4.3.2.

La base semàntica del món no inclou els esquemes d'acció dels personatges que el poblen, ja que no considerem que aquests siguin necessàriament compartits. Al contrari, considerem els agents com entitats autònomes potencialment heterogènies que han de mantenir la seua pròpia operativa; la qual podrà ser intercanviada, si escau, amb uns altres agents. En el punt 4.4 revisem els mòduls principals de la nostra **arquitectura d'agent socialment intel·ligent**, on integrarem els mecanismes de presa de decisions socials del capítol 5. En la secció 5.4.2 exposem com la semàntica del món definida en l'ontologia pot ser utilitzada internament pels agents per a la definició d'operatives generals. Per últim, la secció 4.4.2 versa sobre el **sistema de comunicació**; un requisit bàsic per a la presa de decisions social, tractada en el capítol 5.

4.2 MARC DE SIMULACIÓ MULTIAGENT

La simulació d'entorns virtuals habitats és una tasca força complexa que s'enfronta a la resolució de problemes provinents del camp de l'animació gràfica i de la intel·ligència artificial. Aquest fet aconsella una descomposició modular del **sistema de simulació** de manera que es puguin atendre per separat el model semàntic, el model comportamental i el model geomètric [Lozano and Calderón, 2004]. D'acord amb açò, el marc de simulació

multiagent, que proposem en la figura 4.1, està format per: a) un **entorn virtual semàntic**, que crea i manté l'estat del món; b) un conjunt d'**agents socialment intel·ligents**, que genera el comportament dels personatges; i c) un **motor gràfic 3D**, encarregat de la visualització.

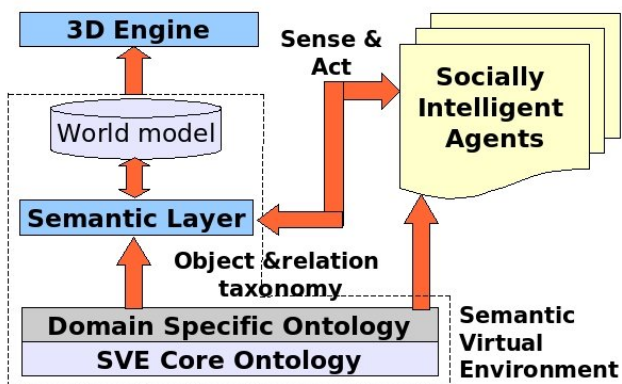


Figura 4.1: Marc de simulació multiagent proposat.

L'entorn virtual semàntic proposat es basa en l'ús d'**ontologies** per a definir la base de coneixement del món així com el conjunt de totes les relacions possibles entre els agents; els quals formen una mena de societat artificial. Distingim dos nivells de representació ontològica (secció 4.3.1). El model base està format per la *SVE Core Ontology*, que és una ontologia única que conté les classes bàsiques per definir qualsevol entorn virtual de manera general, com ara: els objectes contenidors, els objectes mòbils, etc. A l'hora de modelar el coneixement específic associat a certes escenes o situacions, aquesta ontologia base s'estendrà per mitjà de les ontologies específiques del domini escaients (*Domain Specific Ontologies*). Per exemple, la producció d'una cuina virtual necessitarà la creació de diversos tipus d'objectes que hi són habituals: els armaris, les ampolles, els plats, els gots... En aquest context semàntic, la construcció d'un entorn consisteix a crear els objectes 3D de l'escena com instàncies de les classes definides en aquestes ontologies (veure el bar virtual construït en la secció 6.4).

El model del món (*World model*), açò és, la base de dades amb la informació de les entitats que el componen, es gestiona a través de la **capa semàntica** (*Semantic Layer*) (secció 4.3.2). Aquesta capa actua com a interfície entre l'agent i l'estat del món i s'encarrega, fonamentalment, de dos aspectes: la sensorització i l'actuació. D'una banda, la capa semàntica utilitza les relacions de l'ontologia per a reduir el flux d'informació

que provoca la percepció d'objectes rics en informació. Un exemple, a l'anterior cuina virtual, seria un armari que emmagatzemés diferents tipus de plats, gots i coberts. D'altra banda, la capa semàntica es fa càrrec de l'execució de les accions sol·licitades pels agents i del manteniment dels seus efectes sobre el model del món. Per exemple, quan un actor sintètic obri la porta d'un ascensor virtual, aquest roman en l'estat *ocupat* mentre que la porta no siga tancada. L'intercanvi d'informació entre l'entorn virtual semàntic i els agents intel·ligents es fa mitjançant cadenes en XML, les quals segueixen un format de verb i contingut inspirat en els speech-acts de KQML i FIPA ACL [Finin et al., 1994, FIPA (Foundation for Intelligent Physical Agents), 2008].

El model comportamental és aquell que fa referència a la generació del comportament dels personatges autònoms que poblen un entorn virtual. En el nostre marc de simulació, el control principal de cada humanoide es troba situat en un agent diferent. El grau de realisme comportamental assolit dependrà, doncs, de les seues capacitats racionals i socials; cosa per la qual els adjectivem com **agents socialment intel·ligents** (*Socially Intelligent Agents* en la figura 4.1). Aquests agents reben la informació sensorial de la capa semàntica i estan proveïts d'algun formalisme d'IA per a calcular la seqüència apropiada d'accions que els duga a aconseguir els seus objectius. En aquest context, la taxonomia d'objectes definida per l'ontologia s'empra per a definir operadors generals (p. ex. obrir qualsevol tipus de contenidor), els quals incrementaran la reusabilitat dels agents en diferents escenaris (5.4.2). Com veurem en el capítol 5, la inclusió de tècniques socials en aquest procés de presa de decisions abillarà el comportament amb la selecció d'accions socialment acceptables. Així mateix, l'arquitectura d'aquests agents també haurà de contemplar processos de: percepció, control motor, navegació, comunicació, etc. (punt 4.4). La natura distribuïda d'aquesta solució, en què cada agent pot emprar un mecanisme de decisió diferent, així com executar-se en màquines diverses, respon a les fortes necessitats d'heterogeneïtat i escalabilitat dels sistemes de simulació 3D.

Finalment, el model geomètric de l'escenari 3D així com el sistema d'animació (p. ex. taules d'animació, etc.) es troben localitzats en el **motor gràfic 3D** (veure *3D Engine* en la figura 4.1). Aquest mòdul s'encarrega de la visualització gràfica de l'escena. La formalització semàntica del model del món junt amb la separació de la visualització permet intercanviar el mòdul de dibuixat d'una manera senzilla. Com a exemple, el capítol 6 mostra els resultats obtinguts amb dos motors gràfics 3D diferents: el motor del joc *Unreal Tournament* [Epic games, 2008] i la llibreria gràfica *OpenSceneGraph* [OSG, 2008].

D'acord amb el marc de simulació presentat en la primera part d'aquest capítol, en el següent punt descrivim el model proposat d'entorn virtual semàntic. Segonament, analitzem l'arquitectura d'agents intel·ligents que ens serveix de base per a la integració d'habilitats socials.

4.3 ENTORN VIRTUAL SEMÀNTIC

En el context actual dels 3DIVA és comunament acceptat que els personatges 3D, que aspiren a reproduir comportaments intel·ligents, han de manipular informació a distints nivells de representació. Per una banda, cal un nivell baix de representació numèrica per tractar la informació relativa a la posició, velocitat i altres variables geomètriques de les entitats presents a l'entorn. Per una altra banda, un nivell alt de representació simbòlica permetrà als agents conèixer l'estat de l'escenari així com les relacions entre els objectes o agents, i possibilitarà la resolució de situacions diverses esdevingudes en el mateix. El model d'entorn virtual semàntic proposat en aquesta tesi inclou ambdós nivells de representació mitjançant l'ús d'**ontologies**.

El terme ontologia, originari de la filosofia, es refereix a la ciència que descriu els tipus d'entitats presents al món i com es relacionen. A diferència d'un format de missatge (p. ex. XML o XML Schema), una ontologia es caracteritza per ser una representació semàntica del coneixement sobre la qual hom pot raonar. La semàntica formal d'una ontologia especifica com derivar conseqüències lògiques; açò és, fets que no estan literalment presents en l'ontologia però poden ser derivats per la semàntica.

El llenguatge adoptat per a la implementació de les ontologies ha estat **Web Ontology Language (OWL)** [W3C, 2004], recomanació del W3C (*World Wide Web Consortium*) que es preveu que esdevinga l'estàndard *de facto* per a la creació d'ontologies en la Web Semàntica. OWL ha estat desenvolupat com una extensió d'RDF i es deriva del llenguatge ontològic DAML+OIL [Joint US/EU ad hoc Agent Markup Language Committee, 2008]. Una ontologia en OWL inclou, fonamentalment, descripcions de classes, propietats i les seues instàncies. Una classe representa un conjunt d'entitats individuals o instàncies, les quals també són casos particulars de qualsevol superclasse d'aquesta classe. Una propietat especifica la relació que s'estableix entre una instància d'una classe origen i una instància d'una classe destí. En el nostre entorn virtual semàntic, les ontologies defineixen el model

abstracte del món, açò és, la jerarquia de classes d'objectes/agents, les seues propietats i les possibles interrelacions entre ells (veure la figura 4.2).

El llenguatge OWL té tres subllenguatges anomenats, en ordre ascendent d'expressivitat, com: *OWL Lite*, *OWL DL* i *OWL Full*. *OWL Lite* permet la descripció de classificacions jeràrquiques amb restriccions fortes (p. ex. en les restriccions de cardinalitat només permet els valors 0 o 1). Per tant, és l'aproximació més senzilla per a realitzar una migració ràpida des d'un *thesaurus* o altres taxonomies a una representació ontològica. *OWL DL* rep el seu nom de la lògica descriptiva (*Description Logics*), un camp de recerca que estudia aquella part de la lògica de primer ordre que és decidible. *OWL DL* inclou quasi totes les construccions possibles del llenguatge OWL, però imposa algunes restriccions, com ara la separació de tipus (una classe no pot ser tractada a més com un individu o una propietat, i viceversa). D'aquesta manera, *OWL DL* proveeix una **expressivitat** màxima però garanteix que els sistemes de raonament ontològic puguen computar totes les inferències en un temps finit. Finalment, *OWL Full* està orientat a aquells usuaris que volen expressivitat màxima gràcies a una llibertat sintàctica total. Per exemple, una classe en *OWL Full* pot ser tractada simultàniament com una col·lecció d'individus i com un individu de ple dret. Malauradament, no s'espera que cap programari de raonament siga capaç d'admetre totes les característiques d'*OWL Full*. Per tant, el subllenguatge emprat en el nostre marc de simulació és *OWL DL*.

L'elecció d'OWL com a llenguatge ontològic és molt interessant ja que, a causa del seu grau de maduresa, hom disposa d'un conjunt d'eines per a treballar amb ontologies en OWL, com ara: editors que faciliten el desenvolupament (p. ex. Protégé [Stanford Medical Informatics, 2006]), llibreries per a la seua manipulació (p. ex. Jena [Jena, 2008]) i raonadors semàntics (p. ex. RACER [Racer Systems, 2008]). Així mateix, la utilització d'un llenguatge provinent de la Web Semàntica és clau per a l'extensibilitat dels models dels entorns virtuals. La **reutilització** de les ontologies pot dur-se a terme, de manera senzilla, a través d'especificacions OWL ja existents. Aquest fet pot alleugerir la creació de nous mons i la integració de múltiples mons ja existents. A més a més, aquest tipus de solució facilita la **mobilitat** dels humanoides virtuals a través de mons creats per diferents autors. La base de coneixement mantinguda pel SVE desacobla els agents dels entorns, de manera que ambdós poden ser desenvolupats de manera independent.

En la secció següent, analitzem les ontologies que han estat desenvolupades per a la representació semàntica d'entorns virtuals. Aquestes ontologies són a l'abast de la

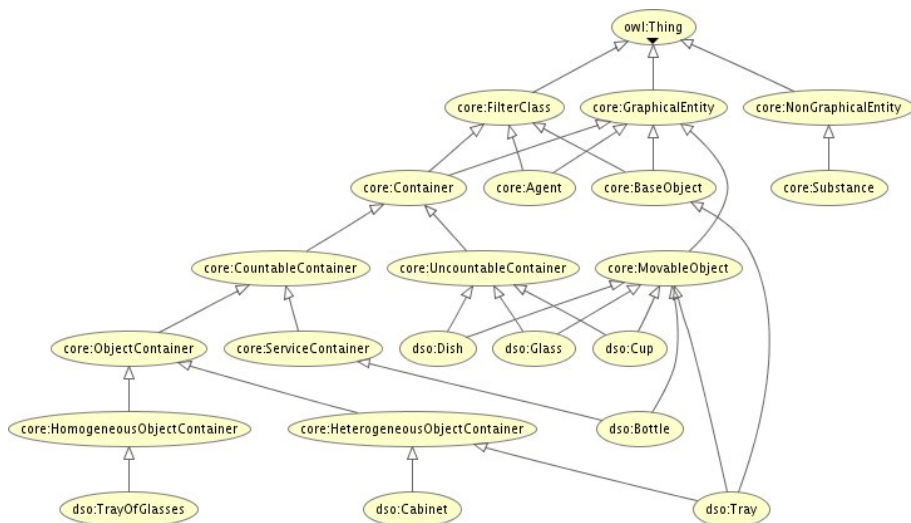


Figura 4.2: Taxonomia de classes per a la creació d'un bar virtual.

comunitat científica des de la *Protege Ontology Library* [Protégé Wiki, 2008], junt amb un recull d'ontologies interessants realitzades en OWL. La secció 4.3.2 versa sobre la capa semàntica, l'altra peça clau del nostre entorn virtual semàntic.

4.3.1 SVE Core Ontology i Domain Specific Ontologies

L'entorn virtual semàntic proposat en aquesta tesi empra les ontologies fent servir dos nivells d'abstracció: l'ontologia base (*SVE Core Ontology*) i les ontologies específiques del domini (*Domain Specific Ontologies*). La *SVE Core Ontology* defineix les **classes bàsiques** que hom necessita per a poder crear qualsevol entorn virtual. D'acord amb açò, la figura 4.2 mostra un fragment de l'ontologia base en la qual, mitjançant l'herència, es defineix una taxonomia d'objectes bàsica (classes amb prefix *core*). L'ontologia completa *SVE Core Ontology* en llenguatge OWL¹ pot ser consultada en l'annex A.1.

La classe *GraphicalEntity* modelitza el conjunt de totes aquelles entitats que disposen d'una representació gràfica pròpia a l'entorn virtual; açò és, tant els agents com els

¹Per al desenvolupament de les ontologies del nostre entorn virtual semàntic hem emprat el programa Protégé [Stanford Medical Informatics, 2006]

diferents objectes 3D de l'escena. Dintre d'aquest conjunt, les classes *MovableObject* i *BaseObject* fan referència, respectivament, a aquells objectes que es poden moure i a aquells en que dipositar altres objectes al damunt. Els diferents tipus d'objectes contenidors que hom pot trobar en un món simulat estan representats per la classe *Container*. En aquest context, distingim dues categories principals: a) els contenidors de substàncies incomptables (*UncountableContainer*), com ara un pot amb sal; i b) els contenidors d'elements comptables (*CountableContainers*), sobre els quals fem una classificació més detallada. D'una banda, la classe *ServiceContainer* representa aquells objectes que proveeixen un cert nombre de serveis associats a entitats no comptables (p. ex. una màquina que serveix cafè). D'altra banda, els contenidors d'objectes (*ObjectContainers*) contindran objectes amb representació gràfica pròpia. Un cas concret de contenidor d'objectes són els *HeterogeneousObjectContainer*, els quals podran mantenir objectes de tipus diferent (p. ex. unes prestatgeries amb llibres, figuretes de porcellana, etc.). Alternativament, els contenidors de tipus *HomogeneousObjectContainer* només contindran objectes indistingibles; com ara una safata plena de gots nets. La classe *NonGraphicalEntity* simbolitza tots aquells objectes físics que no tenen una representació gràfica pròpia al món virtual, sinó que la seua visualització depèn d'uns altres objectes. Per exemple, les substàncies (*Substance*) seran dibuixades pels objectes que les continguem: un got d'aigua, una sucre-ra amb sucre, etc. Per últim, la classe *FilterClass* és una classe especial de filtrat que serà emprada per la capa semàntica a l'hora de reduir la quantitat d'informació apercebuda pels 3DIVA en entorns virtuals amb molts objectes interactius. El seu ús serà analitzat en la secció 4.3.2).

Acabem de veure com la *SVE Core Ontology* proporciona una classificació general d'alt nivell. Tanmateix, a l'hora de crear un entorn virtual particular, hom necessitarà crear objectes proveïts de certes característiques especials. Aleshores, l'ontologia base pot ser estesa amb les **ontologies específiques del domini** escaients. Per exemple, les classes amb prefix *dso* de la figura 4.2 han estat heretades de les classes bàsiques per a crear objectes necessaris en la construcció d'un bar virtual (p. ex. plats, gots, ampolles...). En aquest tipus d'escenari, una safata (*Tray*) serà representada com un contenidor d'objectes heterogenis (*HeterogeneousObjectContainer*) que es pot apilar (*BaseObject*) i moure *MovableObject*. L'ontologia en OWL corresponent a aquestes extensions pot ser consultada en l'annex A.2.

Junt amb la definició de la taxonomia de classes, les ontologies defineixen les **propietats** que les caracteritzen. Per exemple, la figura 4.3 mostra el conjunt de propietats que

The image shows a software interface for an ontology editor. On the left, a vertical sidebar labeled 'S V E Core Ontology' displays a class hierarchy starting with 'owl:Thing'. Under 'GraphicalEntity', there are 'Agent', 'BaseObject', and 'Container'. 'Container' has subclasses 'CountContainer', 'CountlessContainer', and 'MovableObject'. 'CountContainer' has 'ObjectContainer', 'HeterogeneousObjectContainer', and 'HomogeneousObjectContainer'. 'ObjectContainer' has 'ServiceContainer'. 'MovableObject' has 'NonGraphicalEntity'. 'NonGraphicalEntity' has 'Substance' and 'FilterClass'.

The main window shows a 'Property' list for 'rdf:type'. The selected property is 'filterType' with the definition '(multiple owl:oneOf("ALL" "NOTHING" "CLASSES_ONLY"))'. Other properties include 'empty', 'full', 'height', 'objectFree', 'posx', 'posy', 'posz', 'rx', 'ry', and 'yaw', each with its cardinality.

Below the main window, two class-specific property lists are shown. The 'Bottle' class has properties: 'MovableObject', 'ServiceContainer', 'containsSubstance exactly 1 [from ServiceContainer]', 'empty max 1 [from Container]', 'full max 1 [from Container]', 'height exactly 1 [from GraphicalEntity]', 'max exactly 1 [from CountContainer]', and 'num exactly 1 [from ServiceContainer]'. The 'Tray' class has properties: 'HomogeneousObjectContainer', 'MovableObject', 'BaseObject', 'on max 1', 'empty max 1', 'full max 1', 'height exactly 1', 'max exactly 1', and 'num exactly 1'.

Figura 4.3: Propietats dels classes definides en l'ontologia.

comparteixen tots els objectes contenidors (classe *Container*). Podem identificar dos tipus de propietats: les quantitatives i les qualitatives. Les propietats quantitatives representen valors numèrics corresponents sovint amb propietats físiques de l'objecte; per exemple, la posició (*posx, posy, posz*) o les dimensions espacials (*rx, ry*). Pel contrari, les propietats qualitatives normalment representen diferents variables d'estat de l'objecte; com ara si un contenidor està ple (*full*) o buit (*empty*). Les propietats d'una classe també defineixen les **relacions** que es poden establir entre els objectes de la susdita classe i altres objectes de classes diferents. Per exemple, les ampolles (*Bottle*) es relacionen amb les substàncies (*Substance*) que contenen, mitjançant la relació *containsSubstance*. Una altra relació és *on*, que enllaça un objecte mòbil (*MovableObject*) amb l'objecte base (*BaseObject*) on està dipositat.

Les **relacions socials** entre els agents d'una societat artificial també poden ser representades ontològicament en forma de classes i relacions. La figura 4.4 mostra les classes i les relacions definides en l'ontologia base per a la creació de relacions organitzacionals en els entorns virtuals habitats. Distingim tres nivells de relació: el nivell individual, el

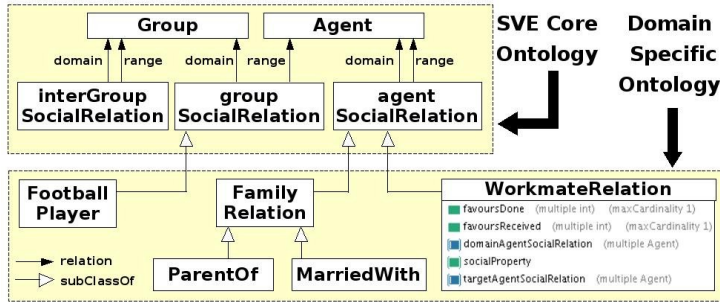


Figura 4.4: Relacions socials definides en l'ontologia.

nivell institucional i el nivell interinstitucional. El **nivell individual** representa les relacions 'un a un' (*one-to-one*) i està modelitzat a través de la classe *agentSocialRelation*. Sempre que un agent es relacione amb un altre agent de forma individual, ambdós estaran enllaçats per una instància d'aquesta classe. Com succeïa abans, diferents dominis d'aplicació poden necessitar relacions específiques. En aixe cas, les ontologies específiques del domini poden heretar de les classes base definides en l'ontologia bàsica i incloure la semàntica requerida per cada situació particular. Per exemple, la relació *WorkmateRelation* de la figura 4.4 duu el compte del nombre de favors intercanviat per dos agents que són companys de treball. Uns altres exemples de relacions individuals són les relacions familiars, com ara la relació de parentiu (*ParentOf*) o de matrimoni (*MarriedWith*).

El **nivell institucional** representa les relacions 'un a molts' (*one-to-many*) i la classe que el simbolitza és *groupSocialRelation*. Quan un agent pertanga a un grup, una instància d'aquesta classe els enllaçarà. La xarxa social creada per aquest tipus de relacions podrà, per exemple, ser explorada en cas que l'agent vulga extraure la resta de membres del grup. La classe *Group* és una abstracció de qualsevol tipus d'agregació. Per tant, aquesta aproximació permet modelitzar des de relacions amb grups físics (p. ex. la classe *FootballPlayer* lligarà els jugadors de futbol al seu equip) fins a agregacions mentals més sofisticades (p. ex. els individus d'una classe social o els seguidors d'una ideologia religiosa). Finalment, el **nivell interinstitucional** representa les relacions 'molts a molts' (*many-to-many*) i servirà per relacionar dos grups, com hem dit, d'índole diversa. Per exemple, dos ciutats virtuals agermanades emprarien aquest tipus de relació social entre grups.

En conclusió, les ontologies permeten definir les classes d'entitats que hom pot trobar

en un món simulat, així com les possibles interrelacions entre elles. En aquest context, la construcció d'un escenari particular consistirà a: a) crear els objectes com a instàncies de les classes definides en l'ontologia; b) donar valors inicials a les propietats com ara la posició, les dimensions o l'estat inicial; i c) relacionar les instàncies adequadament. Per exemple, una lleixa d'un bar virtual amb begudes alcohòliques es pot representar com un objecte de tipus *HeterogeneousObjectContainer* que es relaciona a través de la relació *in* amb les ampolles que conté. Cada ampolla serà un instància de la classe *Bottle* que tindrà un nombre determinat de serveis (valor de la propietat *num*) i es relacionarà amb la beguda que conté com ara whisky o rom (ambdues instàncies de la classe *Substance*). Per a una descripció completa d'un bar virtual habitat per cambrers i clients amb relacions socials veure l'annex A.3, el qual correspon a l'exemple presentat en la secció de resultats 6.4.

Com més estructurada estiga la informació, major capacitat tindran els agents de percepció (secció 4.3.2) i de representació de tasques (secció 5.4.2). Una vegada que hem estudiat la representació ontològica del nostre entorn virtual semàntic, passem a descriure el funcionament de la capa semàntica, la qual fa d'interfície entre el món simulat i els agents que el poblen.

4.3.2 La capa semàntica

La capa semàntica és el mòdul de l'entorn virtual semàntic que s'encarrega de gestionar la base de dades del món al llarg de la simulació. Fonamentalment, aquesta capa realitza tasques de percepció i d'actuació; dos aspectes de vital importància en la interacció entre un actor sintètic i la resta d'entitats de l'escena. Com veurem en aquesta secció, la capa semàntica fa servir el model del món definit en les ontologies per a la consecució satisfactòria de la seua missió.

La **percepció** d'entorns virtuals proveïts de molts objectes amb què interaccionar acostuma a ser una tasca difícil, a causa de la gran quantitat d'informació que contenen. Per exemple, en una llibreria virtual 3D, la percepció completa de tots els llibres suposaria massa informació perquè un agent intel·ligent pogués realitzar una cerca d'accions com les que vérem en la secció 2.4. D'acord amb açò, un dels primers objectius consisteix a reduir la quantitat d'informació dels escenaris 3D a un subconjunt abastable però suficientment expressiu. Per tal d'aconseguir-ho, la capa semàntica construeix un **arbre de dependència sensorial** com el que apareix en la figura 4.5. Aquest arbre organitza els

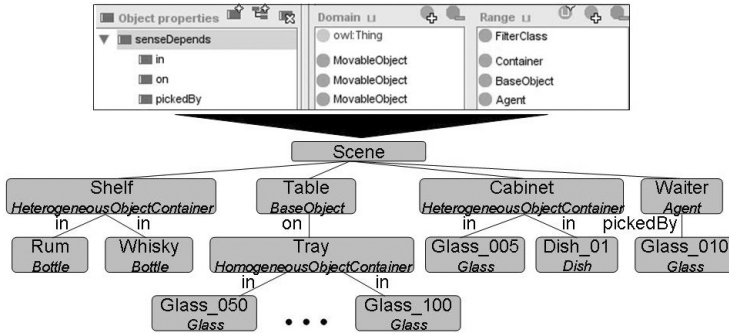


Figura 4.5: Arbre de dependència sensorial.

objectes de l'escena en una disposició jeràrquica, en què uns objectes deleguen la seua percepció en uns altres, atenent a la seua relació de dependència sensorial. La dependència sensorial ha estat modelitzada en l'ontologia base (*SVE Core Ontology*) mitjançant una *super propietat* OWL ² anomenada *senseDepends*. Algunes relacions concretes que estableixen, a banda del seu significat semàntic, una dependència sensorial per a la capa semàntica són: *in*, *on* i *pickedBy*. Per exemple, *in* relaciona els objectes mòbils (*Movable-Object*) amb els contenidors (*Container*), aleshores, els objectes contenidors gestionaran quina informació es pot percebre dels objectes que contenen.

A l'hora de sensoritzar l'estat d'una escena la capa semàntica fa un recorregut de l'arbre de dependència sensorial, de manera que els objectes amb responsabilitat sensorial (nodes pares de l'arbre) puguin filtrar la informació dels objectes subordinats (nodes fills de l'arbre). Com s'aprecia en la definició de *senseDepends*, i per tant de les relacions derivades, els objectes amb responsabilitat sensorial corresponen amb instàncies de la classe *FilterClass*, definida en l'ontologia base. Aquesta classe permet la definició de diferents tipus de **filtrat**, per mitjà d'una propietat anomenada *filterType* (veure la figura 4.3). A hores d'ara, hem implementat tres tipus de filtres referits com: *ALL*, *NOTHING* i *CLASSES-ONLY*. Mentre que el filtre *ALL* deixa passar qualsevol informació, el filtre *NOTHING* fa imperceptible tota la informació dels objectes subordinats. Finalment, el filtre *CLASSES-ONLY* únicament fa públiques les classes dels objectes subordinats. El tipus de filtrat és una propietat amb un valor inicial però pot ser canviada al llarg de la simulació. Aquest fet permet la definició de nivells de detall semàntic en la percepció d'objectes complexos. Per exemple, hom pot modelar una vitrina com un objecte de tipus

²Les propietats OWL utilitzen la mateixa metodologia que els objectes i utilitzen l'herència per classificar els tipus de propietats.

Cabinet que només publica les classes dels objectes continguts mentre està tancada (filtre *CLASSES-ONLY*) i que publica tota la informació quan s'obri (filtre *ALL*).

Un altre aspecte per a reduir la quantitat d'informació dels entorns virtuals amb molts objectes d'interacció és **evitar la multiplicitat**. Hem definit les instàncies de tipus *HomogeneousObjectContainer* com aquells contenidors que enclouen una certa quantitat d'objectes indistingibles. A fi d'evitar informació redundant, la capa semàntica permet un tipus especial de filtrat sobre aquesta classe d'objectes. La propietat *numBaseObjects* (figura 4.3) permet publicar la informació d'un nombre més reduït d'objectes interactius. Aquesta propietat es pot ajustar dinàmicament, com ja succeïa amb la propietat *filterType*. Així, la interacció està garantida en tot moment mentre que la quantitat d'informació es redueix. Per exemple, suposem que en un bar virtual tenim una safata amb 50 gots nets i només 3 cambrers que poden agafar-los. Llavors, té sentit publicar només un nombre menor de gots amb els quals els cambrers puguin interaccionar (p. ex. una quantitat de 6, un per cada mà del cambrer), ja que la interacció amb la resta no suposaria cap diferència. A mesura que els gots siguen emprats, i mentre no s'exhaureixen, la safata publicarà nous gots interactius a l'abast dels cambrers.

Tocant a l'**actuació**, la capa semàntica proveeix tres serveis fonamentals: el desplaçament dels agents per l'escenari, la interacció agent-objecte i la comunicació agent-agent. Quant al **desplaçament**, els agents intel·ligents li demanen el moviment dels avatars que els representen per arribar als punts desitjats d'interacció. Llavors, el món notificarà les col·lisions en cas que es produeixen. Pel que fa a la **interacció agent-objecte**, la capa semàntica és l'encarregada d'executar les accions sol·licitades pels agents intel·ligents. En conseqüència, sobre ella recau la responsabilitat final de modificar la base de dades que emmagatzema l'estat del món. Per tal de fer açò, la capa semàntica conté els procediments que li permeten comprovar l'aplicabilitat, l'èxit o la fallada de tota acció permesa. Quan una acció es realitza amb èxit, la capa semàntica aplica els efectes escaients sobre els objectes involucrats. Per exemple, si continuem amb la safata abans introduïda, l'acció en què un cambrer agafa un got minva en una unitat el nombre de gots continguts en la safata. A més a més, si s'agafa l'últim got disponible, la propietat de l'objecte que marca la seua qualitat d'estar buit (*empty*) serà canviada al valor vertader (*true*). Si, pel contrari, una acció no es pot aplicar o hi ha una fallada durant la seua execució, la capa semàntica ho notificarà a l'agent sol·licitant perquè aquest reaccione a la situació.

El paper que juga la capa semàntica en la **comunicació agent-agent** és el d'un sim-

ple canal de transmissió, de manera similar a l'aire al món real. Malgrat que els entorns virtuals poden estar poblats per agents amb objectius i habilitats diverses, podem considerar-los sistemes tancats a nivell de comunicació; ja que els personatges comparteixen el coneixement sobre com han de comunicar-se. No estem, doncs, davant d'un escenari com Internet; en què interaccionen una barreja d'agents totalment heterogenis i es justifica l'ús d'intermediaris [Benford et al., 1997] o de models de comunicació de tipus pissarra [Omicini and Zambonelli, 1999]. Pel contrari, la comunicació d'igual a igual serà més adequada als món simulats. En segon lloc, cal qüestionar-se si aquesta s'ha de realitzar de forma directa o a través de l'entorn. Considerem que la segona opció és més realista i permet reproduir situacions interessants des del punt de vista de l'animació (p. ex. un actor que tafaneja la conversa d'altres que estan parlant a prop). Per tant, en el nostre marc de simulació la comunicació es realitza a través de l'entorn virtual semàntic. En aquest context, la capa semàntica rep els missatges dels agents i els adreça als destinataris corresponents, de manera que és l'agent l'únic encarregat de produir i gestionar la informació intercanviada (veure la secció 4.4.2).

Després d'haver analitzat l'entorn virtual semàntic, presentem l'arquitectura d'agents socialment intel·ligents que proposem en el nostre marc de simulació. A continuació, analitzem els mòduls principals de l'arquitectura per animar comportaments autònoms intel·ligents. Llavors, en el capítol següent, aprofundirem en les habilitats socials dels actors sintètics.

4.4 ARQUITECTURA D'AGENTS SOCIALMENT INTEL·LIGENTS

Com vérem al llarg de l'estat de l'art, l'animació comportamental dels personatges 3D duu associada la resolució de problemes complexos de natura diversa, com ara: la percepció, el control motor, la presa de decisions, la comunicació, etc. Aquest fet ha conduït al desenvolupament de sistemes en què diferents mòduls, especialitzats en tasques específiques, s'acoblen per aconseguir un comportament global versemblant. Les **arquitectures modulars** són una bona solució per al disseny de 3DIVA, ja que divideixen la complicada animació comportamental en trossos més menuts i abastables. A més a més, aquestes solucions ofereixen la possibilitat d'intercanviar mòduls i experimentar amb diferents tècniques de manera més senzilla. La figura 4.6 mostra un esquema general de l'arquitectura modular utilitzada pels agents socialment intel·ligents del nostre

marc de simulació. Cadascun d'aquests agents rep la informació sensorial de la capa semàntica i calcula la seqüència d'accions apropiada perquè el personatge 3D que governa aconseguisca els seus objectius. Per tal de fer-ho, comptarà amb l'ajuda d'un conjunt de subsistemes que detallarem a continuació.

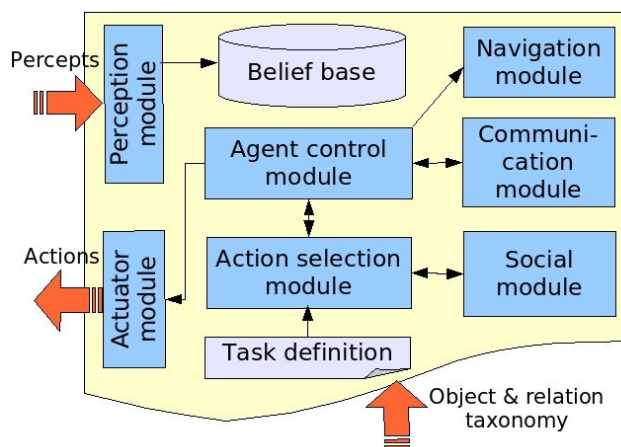


Figura 4.6: Arquitectura dels nostres agents socialment intel·ligents.

El **mòdul de percepció** (*Perception module*) té com a tasca principal el processament dels missatges de percepció rebuts des de l'entorn virtual semàntic. Hom pot distingir tres modes d'enviament per aquests missatges: l'enviament a una freqüència fixa (*on time*), l'enviament sota demanda de l'agent (*on request*) i l'enviament asíncron quan hi ha un canvi sensorial (*on property change*). A causa de la **reactivitat**, demanada pel comportament versemblant dels personatges 3D, hem triat la tercera opció en el nostre marc de simulació. La percepció asíncrona informarà els agents dels canvis produïts en altres agents o objectes en el moment en què aquests es produeixen. Aleshores, el mòdul de percepció compilarà aquestes percepcions i actualitzarà la base de creences de l'agent en conseqüència. Per exemple, en base a les observacions d'eixida i entrada d'un objecte en el camp de visió, el mòdul de percepció pot simular fàcilment mecanismes d'oblit i record, molt interessants des del punt de vista comportamental [Kuffner and Latombe, 1999]. En la secció 5.3.2 analitzarem el paper clau d'aquest mòdul en el manteniment de les creences per a la coordinació de plans en entorns multiagent.

La **base de creences** (*Belief base*), que correspon amb la memòria de l'agent, emmagatzema l'estat del món apercebut, l'estat intern de l'agent i una representació de l'estat

intern d'altres agents. Es tracta d'un contenidor dinàmic d'objectes que rep i actualitza asincrònicament la seua informació a través del mòdul de percepció. La base de creences està estructurada d'acord amb la definició ontològica del model del món, proporcionada per l'entorn virtual semàntic. Fonamentalment, allotja dos nivells de representació: informació numèrica de baix nivell (p. ex. posició, dimensions...) i dades simbòliques que descriuen les propietats o relacions dels objectes i agents (p. ex. la substància continguda per una ampolla en un bar virtual). La memòria de l'agent serà emprada per la resta de mòduls a l'hora de generar l'actuació de l'actor sintètic.

El **mòdul d'actuació** (*Actuator module*) juga un paper d'interfície en els nostres agents socialment intel·ligents, ja que les accions s'executen a l'entorn virtual semàntic (veure secció 4.3.2). En conseqüència, aquest mòdul ofereix als agents els mecanismes per enviar de forma asíncrona les seues peticions d'actuació cap a la capa semàntica. Una vegada realitzada la petició, l'agent restarà a l'espera del resultat de l'acció (*feedback* de l'entorn) mentre la capa semàntica executa el codi comportamental associat a l'acció. Si aquesta pot realitzar-se amb èxit, provocarà els canvis corresponents en els objectes afectats i tornarà èxit. En cas contrari, la capa semàntica retornarà l'agent una fallada de la seua acció. Independentment del resultat de l'acció, l'agent detectarà els canvis produïts al món a través del seu mòdul de percepció, encara que aquests siguin producte de la seua pròpia actuació.

El **mòdul controlador de l'agent** (*Agent control module*) governa l'activitat de l'actor en tot moment i decideix allò que ha de fer depenent del seu estat intern i de l'estat del món que l'envolta. De forma bàsica, està format per una màquina d'estats que atén els aspectes reactius del personatge i que defineix el seu comportament proactiu essencial (p. ex. veure el controlador de la figura 5.2). Llavors, el mòdul controlador se suporta en uns altres mòduls perquè s'encarreguen de tasques específiques com ara la navegació, la presa de decisions i la comunicació.

Junt amb les habilitats d'actuació i interacció comportamentals pròpies de l'actor que representen, els actors sintètics han de combinar habilitats de navegació similars a les estudiades en la robòtica; açò és, la capacitat de trobar rutes lliures d'obstacles. Tanmateix, les facilitats de navegació oferides pels motors gràfics (*pathnodes* o mapes de carrils lliures d'obstacles introduïts off-line) plantegen problemes seriosos d'autonomia i realisme en les rutes recorregudes, que seran sempre les mateixes [Cavazza et al., 2002]. Com a conseqüència, el **mòdul de navegació** aplicat habitualment a personatges autònoms 3D és un

sistema híbrid, que combina un planificador de camins 2D (p. ex. la família d'algorismes A^*) [Kuffner, 1999] amb algun sistema local basat en: regles [Reynolds, 1999], camps potencials [Mamei et al., 2004], xarxes neuronals [Lozano, 2005], etc. La combinació de sistemes globals i locals és una solució comunament acceptada per als personatges 3D [Pelechano et al., 2007, Shao and Terzopoulos, 2005], ja que el sistema local aportarà la reactivitat necessària, mentre que el sistema global solucionarà els problemes (mínims locals) derivats de la complexitat espacial de l'escenari virtual.

El **mòdul de selecció d'accions** (*Action selection module*) és el responsable de la presa de decisions i, per tant, de les accions que finalment seran animades sobre el personatge 3D. Normalment, aquest mòdul està format per un procés de cerca de la millor opció entre un conjunt de tasques possibles. Aquesta cerca té el suport en un **esquema d'acció** per seleccionar quines tasques ha de realitzar un agent a fi d'aconseguir el seus objectius (veure la secció 5.4.2). En aquest context, la intel·ligència dels actors sintètics prové sovint del comportament racional. Segons el criteri de la racionalitat, la idoneïtat d'una acció correspon amb la mesura en què la seua execució apropa l'estat del món a l'estat objectiu desitjat per l'agent.

Nogensmenys, el grau de versemblança dels actors sintètics es veurà limitat sense el comportament social, encarregat de complir amb els sovint difícilment definibles objectius socials (p. ex. coordinació, cooperació, etc.). D'una banda, cal que els agents interaccionen entre ells. D'acord amb açò, els nostres agents socialment intel·ligents incorporen el **mòdul de comunicació** (*Communication module*), encarregat de mantenir els protocols d'interacció per a la comunicació agent-agent. Aquest mòdul genera els missatges que s'envien a l'entorn perquè els repartisca als destinataris corresponents i processa els missatges obtinguts com a resposta (veure la secció 4.4.2). D'una altra banda, cal entendre la selecció d'accions amb un **mòdul social** (*Social module*), que introduïska criteris socials en el mecanisme de presa de decisions; cosa que estudiarem en el capítol 5.

En les següents seccions, analitzem els esquemes d'acció i el mòdul de comunicació incorporats en l'arquitectura d'agents socialment intel·ligents, ja que els considerem de vital importància per a la inclusió d'habilitats socials en el comportament dels actors sintètics. La presa de decisions de tipus social, en què es troben les aportacions principals d'aquesta tesi, serà tractada en el capítol 5.

4.4.1 Esquemes d'acció

En l'arquitectura d'agents proposada, una tasca es defineix com la seqüència d'accions que cal sol·licitar a l'entorn virtual semàntic per animar-la completament. De conformitat amb els mecanismes de selecció dinàmica d'accions introduïts en la secció 2.4, hem provat dues aproximacions per al mòdul de selecció d'accions, cadascuna amb el seu esquema d'acció:

- Selecció mitjançant **planificació heurística**: Dins d'aquesta categoria, hem emprat el planificador *miniMin-HSP* desenvolupat per Miguel Lozano per a l'animació comportamental de personatges intel·ligents 3D [Lozano, 2005]. En aquest context, les tasques disponibles es defineixen a través d'operadors inspirats en el llenguatge d'acció STRIPS (veure la figura 4.7b). Açò és, la descripció d'una tasca es compon d'unes llistes de precondicions (*preconditions*) i efectes (positius *add* i negatius *delete*) que expressen, respectivament, la seua aplicabilitat i els canvis que produeix la seua execució sobre l'estat del món i de l'agent.
- Selecció basada en el **model BDI**: Pel desenvolupament d'aquests agents hem fet servir *Jason* [Bordini and Hübner, 2007], una implementació del llenguatge de programació *AgentSpeak* [Rao, 1996] per a sistemes multiagent. En *AgentSpeak*, els agents gestionen la seua operativitat a través d'un conjunt de plans que descriuen la seqüència d'accions a realitzar per aconseguir les intencions que esdevenen durant el temps de vida de l'agent. La figura 4.7a mostra un exemple d'un pla en *Jason* per servir una substància en un got. El pla té el següent format *esdeveniment* : *context* -> *cos* on: a) l'*esdeveniment* defineix l'objectiu reactiu o proactiu que provoca la seua execució; b) el *context* descriu la situació en què el pla és aplicable; i c) el *cos* estableix les accions a realitzar.

Els esquemes d'acció esmentats abans a sovint no tenen en compte un aspecte molt important dels entorns amb molts objectes d'interacció: la possibilitat de **reutilitzar els operadors** en contextos diferents. Per exemple, els personatges 3D normalment defineixen un operador per cadascuna de les tasques que poden dur a terme. Aleshores, hom pot trobar un barmen virtual capaç de servir un suc de taronja i una copa de vi però sense cap idea de com preparar una tassa de te. Nogensmenys, tal i com demostraren L. Levison [Levison, 1996] i N.I. Badler [Badler et al., 2000] des de la *Universitat de Pennsylvania*, la classificació dels objectes en categories pot ser utilitzada per a definir operadors genèrics que puguin ser aplicats sobre objectes diferents. D'acord amb açò, la definició de

(a) BDI	(b) STRIPS
<pre> +!ServeFromContainer (Glass, Container, Substance) : class (Glass, uncountableContainer) & class (Container, serviceContainer) & class (Substance, substance) & .my_name (MyName) & pickedBy (Glass, MyName) & not full (Glass) & not empty (Container) & containsSubstance (Container, Substance) <- .send (SemanticLayer, achieve, serveFromContainer (Glass, Container, Substance)) . Send action to Semantic Layer +!ServeFromContainer (Glass, Container, Substance) <- +containsSubstance (Glass, Substance); -empty (Glass) . </pre>	<pre> operator ServeFromContainer: parameters: ?glass - UncountableContainer ?container - ServiceContainer ?substance - Substance preconditions: ?glass pickedBy ?me not (?glass full) not (?container empty) ?container containsSubstance ?substance add: ?glass containsSubstance ?substance delete: ?glass empty actions: serve ?glass ?container ?substance </pre>

Figura 4.7: Exemple d'operador genèric expressat mitjançant la parametrització de: (a) un pla BDI o (b) un operador basat en el model STRIPS.

les tasques dels nostres agents intel·ligents ha estat revisada per incorporar la informació semàntica de l'ontologia, la qual representa el model abstracte del món.

En el nostre marc de simulació, la taxonomia d'objectes definida en les ontologies de l'entorn virtual semàntic és usada pels agents socialment intel·ligents per definir **operadors parametritzats**. La inclusió de paràmetres en l'esquema d'acció permet definir operadors que siguin reusables, independentment del formalisme de presa de decisions. Per exemple, la figura 4.7 mostra un operador genèric expressat mitjançant un pla BDI així com un operador basat en el model STRIPS. Aquest operador permet que un actor sintètic, com ara un cambrer virtual, pugui servir qualsevol substància (instància de la classe *Substance*) que estiga continguda en un contenidor de tipus *ServiceContainer* dins d'un contenidor d'objectes incomptables (*UncountableContainer*).

D'una banda, el pla BDI de la figura 4.7a inclou, en les condicions de context, les classes a què han de pertànyer els objectes per poder executar les accions sobre ells. Per exemple, el pla *ServeFromContainer* es pot aplicar tant per a servir un got de vi d'una ampolla com una tassa de cafè d'una cafetera, sempre que aquests objectes siguin instàncies de les classes abans esmentades. La taxonomia d'objectes definida en l'ontologia s'empra, doncs, per classificar els objectes interactius i per obtenir les seues propietats i relacions. En acabant, la capa semàntica serà qui execute les accions i aplique els efectes sobre els objectes involucrats. D'una altra banda, la figura 4.7b mostra la definició de

l'operador *ServeFromContainer* basada en el model STRIPS. De manera similar a PDDL [pdd, 2008], hem inclòs una secció separada de paràmetres (*parameters*), que relaciona els objectes amb les classes adients de l'ontologia. La resta de seccions (*preconditions*, *add*, *delete*) són les clàssiques del model STRIPS. Finalment, la secció anomenada *actions* conté les accions que seran sol·licitades a l'entorn.

La parametrització, la qual fonamentalment introdueix variables i tipus que fan referència a la jerarquia de classes de l'ontologia, estalvia la definició d'un operador per a cada objecte d'interacció, cosa que finalment es veu reflectida en un grau més alt d'interacció. A més a més, la interoperabilitat de l'agent entre escenaris diferents es veu millorada, ja que els agents seran capaços de manipular qualsevol objecte del món, sempre que aquest siga una instància d'una classe definida en l'ontologia.

Tant la planificació heurística com el model BDI són aproximacions que han estat emprades per a l'animació de comportaments racionals en personatges 3D (veure la secció 2.4). En conseqüència, les aportacions principals d'aquesta tesi han estat la inclusió d'habilitats socials en ambdós paradigmes (capítol 5). Com a cloenda d'aquest capítol expliquem breument el mòdul de comunicació, component en el qual s'ajuden aquestes tècniques socials.

4.4.2 Mòdul de comunicació

El mòdul de comunicació és el component que s'encarrega de guiar els personatges durant les interaccions amb els altres actors sintètics. En un entorn virtual poblat per diversos agents, la necessitat d'interaccionar pot provenir d'un comportament reactiu o ser el resultat d'una presa de decisions proactiva. Com a exemple reactiu, quan un agent es dirigeix a agafar un objecte 3D i un altre li l'arrabassa, el primer podria preguntar al segon si li'l deixa o quan l'alliberarà. Com a exemple proactiu, diversos agents emplaçats en un espai comú podrien comunicar els seus objectius de cara a col·laborar en l'assoliment de certes tasques comunes.

En aquest context, la comunicació agent-agent es modelitza mitjançant **protocols d'interacció**. Fonamentalment, aquests protocols són màquines d'estats finits que identifiquen els estadis en què els agents poden estar al llarg de la comunicació, així com els missatges intercanviats entre els interlocutors. Llavors, el mòdul de comunicació em-

magatzema els protocols d'interacció i s'encarrega de la dinàmica. És a dir, genera els missatges (actes de la parla o *speech-acts*) corresponents a l'estat de la comunicació i rep les respostes que fan avançar el protocol d'interacció.

Els protocols inclosos en el mòdul de comunicació han de ser adequats per a la comunicació entre humans virtuals. No estem davant de la comunicació entre agents telemàtics, en què la importància recau sobre l'intercanvi d'informació i la concisió del protocol. Ans al contrari, els agents socialment intel·ligents estan interessats a animar converses i situacions versemblants entre personatges 3D. Això fa que els protocols es definisquen sovint com la combinació d'altres protocols bàsics d'interacció definits per FIPA [FIPA (Foundation for Intelligent Physical Agents), 2008] (p. ex. *Request*, *Propose*, *Query*, etc).

Per exemple, la figura 4.8 conté dos protocols emprats en la coordinació i la col·laboració d'agents basats en planificador heurístic, que descriurem en l'apartat 5.3. La figura 4.8a descriu la conversa que es produeix, de forma reactiva, quan un agent interromp la tasca d'un altre. En aquesta, l'agent afectat pregunta al primer què està fent i usa aquesta informació per evitar col·lisions futures. Com es pot veure, aquest protocol es pot descompondre en dos protocols FIPA bàsics: el protocol de proposta (*Propose*) i el protocol de pregunta (*Query*). La figura 4.8b mostra el protocol que empren els agents per publicar els seus objectius i establir punts de col·laboració. En ambdós exemples, els primers missatges de salutació són, en principi, superflus. Tanmateix, són missatges que aporten realisme a la interacció i, per tant, a l'animació.

Finalment, el mòdul de comunicació també inclou protocols més complexos amb diversos participants. Per exemple, la presa de decisions MADeM, que presentarem en l'apartat 5.4, empra el protocol *FIPA-Contract Net* per a realitzar subhastes entre diferents grups d'agents. Així mateix, el mòdul de comunicació aporta diferents protocols de conversació entre els agents per animar els acords o desacords que s'hagen produït com a resultat de la subhasta (veure la secció 6.4.4).

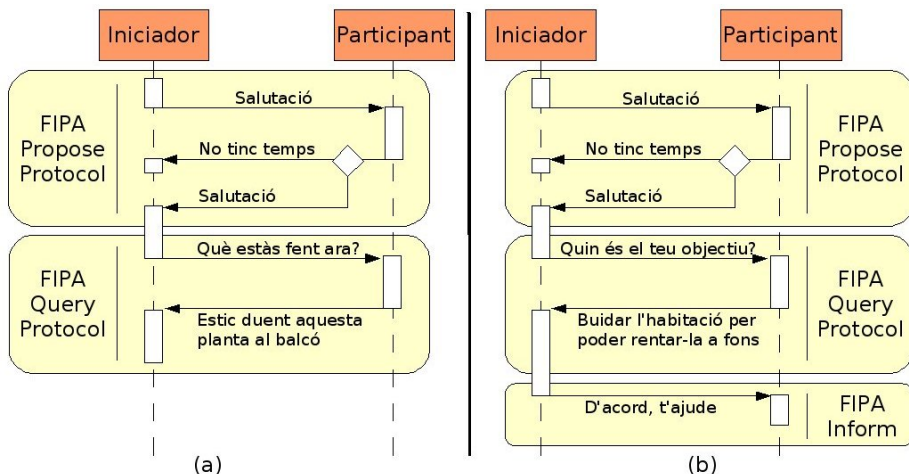


Figura 4.8: Exemples de converses entre dos agents per: (a) conèixer la tasca actual d'un altre, (b) conèixer els objectius d'un altre.

4.5 CONCLUSIONS

En aquest capítol, hem presentat les aportacions menors d'aquesta tesi, arreglades en el **marc de simulació multiagent** dissenyat per al desenvolupament d'entorns virtuals 3D habitats per actors sintètics amb habilitats socials. D'una banda, hem analitzat el model general d'entorn virtual semàntic inclòs en aquest sistema. Aquesta aproximació fa servir les **ontologies** com una solució extensible i reutilitzable per a la representació de l'entorn i de les relacions entre els actors. Llavors, el **coneixement semàntic** beneficia la **producció**, la **percepció** i la **interacció** dels personatges en escenaris amb nombrosos objectes interactius. D'una altra banda, hem presentat una arquitectura d'agent capaç de governar el comportament de personatges 3D socialment intel·ligents (p. ex. percepció, control, navegació, etc.). En aquest context, hem estudiat l'ús d'informació semàntica per a definir **esquemes d'acció generals** per a la interacció agent-objecte, així com els **protocols de comunicació** per a modelitzar la interacció agent-agent.

Tot plegat, hem analitzat el conjunt d'aportacions col·laterals que sostenen les aportacions principals d'aquesta tesi en matèria de presa de decisions socials, les quals tractem en el següent capítol.

PRESA DE DECISIONS SOCIALS

Aquest capítol conté les aportacions principals d'aquesta tesi pel que fa a la integració d'habilitats socials en el comportament dels actors sintètics. Primerament, analitzem els requeriments socials dels actors virtuals intel·ligents 3D. En segon lloc, presentem dues aproximacions per a la presa de decisions de tipus social: una a nivell *micro* (centrada en l'agent) i una altra a nivell *macro* (centrada en la societat). A nivell *micro*, proposem una tècnica de **col·laboració amb planificadors heurístics**. Mitjançant la detecció de conflictes i sinergies amb altres agents, aquesta tècnica proveeix els actors amb l'habilitat d'adaptar-se als entorns amb recursos compartits per diversos caràcters autònoms. Amb una perspectiva *macro*, presentem **MADeM (Multi-modal Agent Decision Making)**, un procés de presa de decisions de tipus social que pot ser emprat per agents de tipus BDI per tal de prendre **decisions socialment acceptables**. Així doncs, MADeM permet els agents reproduir comportaments desitjats a nivell de grup com ara: el benestar social, les actituds envers els altres, etc.

5.1 INTRODUCCIÓ

La presa de decisions duta a terme pels personatges 3D, a l'hora d'animar el seu comportament, sovint ha seguit criteris racionals autointeressats. Aquest fet ha estat conseqüència de la versemblança d'un actor sintètic en un context aïllat (sense la interferència d'altres actors externs) depén, en gran mesura, de l'eficiència en la consecució dels seus objectius. Nogensmenys, als entorns virtuals habitats, els actors necessiten incloure criteris addicionals que representen les seues capacitats socials.

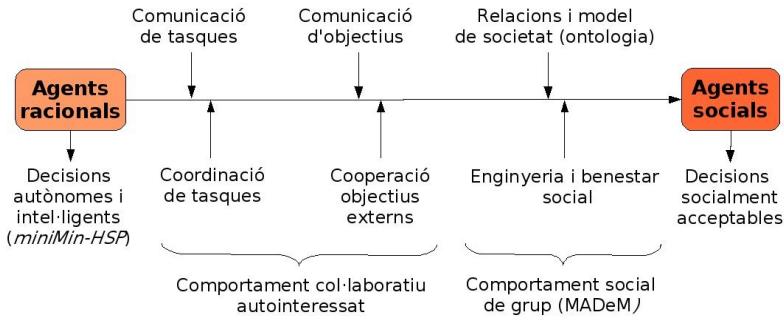


Figura 5.1: Espectre de decisions des dels agents racionals als agents socials.

La figura 5.1 mostra l'espectre de decisions entre els agents racionals i els agents socials. El comportament intel·ligent dels agents racionals és insuficient als entorns amb recursos compartits, en què els conflictes poden aparèixer ràpidament a causa de la voluntat d'interactuar amb els mateixos recursos de manera simultània (veure els exemples del punt 1.1). Llavors, l'intercanvi d'informació al voltant de les tasques i dels objectius dels agents externs pot servir per incorporar **comportaments de grup orientats a l'eficiència** com ara la **coordinació** o la **cooperació**. D'una altra banda, els humanoides estableixen relacions amb altres membres de la societat artificial, no necessàriament orientades a l'eficiència però que poden aportar realisme a l'animació (p. ex. una colla d'amics virtuals que prefereixen xarrar a treballar), que hom espera veure reflectides en el seu comportament. Per tant, els actors intel·ligents necessiten avaluar també l'impacte social de les seues accions i decidir la seua actuació tenint en compte les relacions establides, cosa que els conduirà cap a una presa de **decisions socialment acceptables**.

D'acord amb açò, en l'apartat 5.2 analitzem els **requeriments socials dels actors virtuals intel·ligents 3D**. D'una banda, els agents han de ser capaços de detectar les dependències que es produeixen entre les seues accions i les de la resta de cohabitants (veure la secció 5.2.1). D'una altra banda, cal que els agents realitzen una presa de decisions de tipus social. És a dir, que incloga les accions i els interessos d'altres agents en la selecció de les accions a realitzar (veure la secció 5.2.2). El marc de simulació descrit en el capítol 4 proveeix els mòduls necessaris perquè ambdós aspectes puguen ser tractats.

En aquest capítol, presentem dues aproximacions per a la incorporació de criteris de tipus social en dos mecanismes ben coneguts per a la selecció dinàmica d'accions. En

primer lloc, l'apartat 5.3 defineix una tècnica de **col·laboració basada en planificadors heurístics**, perquè els actors s'adapten de forma eficient en entorns amb recursos compartits. Com mostra la figura 5.1, aquesta tècnica emprà la comunicació directa entre els actors (secció 5.3.1), la coordinació en l'ús dels recursos (secció 5.3.2) i la cooperació en l'assoliment d'objectius externs (secció 5.3.3). Segonament, en l'apartat 5.4 presentem **MADeM (Multi-modal Agent Decision Making)**. MADeM és un procés de presa de decisions que permet avaluar diversos punts de vista (*multimodalitat*), així com consultar les preferències d'altres agents al voltant de les diferents accions a realitzar (seccions 5.4.2 i 5.4.3). Aquest *feedback* social nodreix un mecanisme de selecció de decisions socialment acceptables (secció 5.4.4), que permet reproduir comportaments desitjats a nivell de grup com ara: el benestar social, les actituds envers els altres, etc. (veure la figura 5.1).

Les dues aportacions socials abans esmentades han estat provades de manera independent. Com ja diguérem en el capítol 2, considerem tant la planificació heurística com el model BDI solucions adequades per a la presa de decisions en personatges 3D. De fet, poden ser complementàries. Per exemple, hom pot pensar a combinar la planificació per a la seqüenciació de tasques en el baix nivell i el model BDI per al manteniment dels objectius de més alt nivell. Dissortadament, la integració d'ambdues tècniques queda fora de l'abast d'aquest document i resta com a treball futur d'aquesta tesi (veure el punt 7.2).

5.2 REQUERIMENTS SOCIALS DELS 3DIVA

Malgrat que alguns entorns virtuals tendeixen a simular actors sintètics que realitzen tasques d'una manera independent, aquesta suposició minva la seua versemblança perquè constitueix una simplificació molt forta. Ans al contrari, les tasques dutes a terme pels personatges 3D solen ser dependents, cosa que fa que la seua execució afecte l'operativa d'altres individus i produïska problemes de **coordinació**.

De bon començament, la *Teoria de l'Organització* [Malone and Crowton, 1994] definí la coordinació com la gestió de dependències entre activitats (p. ex. el flux de treball d'una empresa per manufacturar un producte). Més recentment, però, D'Inverno i Luck [D'Inverno and Luck, 2004] han plantejat una formalització de les diferents relacions que es poden donar en un sistema multiagent. Dins d'aquest model, el procés de coordinació consisteix en dues tasques fonamentals que analitzarem a continuació: la **detecció de les**

Categoria	Objectius	Recursos	Habilitats	Tipus de situació
Indiferència	Compatibles	Suficients	Suficients	Independència
Cooperació	Compatibles	Suficients	Insuficients	Col·laboració simple
	Compatibles	Insuficients	Suficients	Obstrucció
	Compatibles	Insuficients	Insuficients	Col·laboració coordinada
Antagonisme	Incompatibles	Suficients	Suficients	Competició individual pura
	Incompatibles	Suficients	Insuficients	Competició col·lectiva pura
	Incompatibles	Insuficients	Suficients	Conflicte individual de recursos
	Incompatibles	Insuficients	Insuficients	Conflicte col·lectiu de recursos

Taula 5.1: Tipus d'interacció en els sistemes multiagent [Ferber, 1999].

dependències i la **presa de decisions** sobre l'acció de coordinació que convé emprar.

5.2.1 Detecció de dependències

En un sistema multiagent, les dependències es manifesten en forma d'interaccions entre dos o més agents, els quals estableixen una mena de relació dinàmica a través d'un conjunt d'accions recíproques. Llavors, les interaccions produiran una sèrie d'accions que influiran el comportament futur de l'agent.

Hom pot identificar diversos **tipus d'interacció**; en funció dels objectius dels agents (compatibles o incompatibles), de les seues habilitats i dels recursos disponibles (suficients o insuficients) [Ferber, 1999]. La taula 5.1 mostra aquesta classificació, en què dos objectius són incompatibles si aconseguir un d'ells comporta que l'altre no podrà ser assolit i són compatibles si no es compleix aquesta condició. Quan els objectius són compatibles estem davant d'una situació de *cooperació*, mentre que si són incompatibles apareix l'*antagonisme*.

La *Independència* no suposa realment un problema de coordinació. Per exemple, 'a priori', no és necessària la interacció entre personatges 3D que es troben en habitacions distintes o que treballen en problemes diferents en un entorn sense manca de recursos. La *Col·laboració simple* sorgeix quan els agents necessiten afegir les habilitats d'altres per complementar la seua insuficiència, com és el cas dels actors que s'ajuden per moure un objecte pesat d'una estança a una altra (veure la figura 3.1b). Succeeix una *Obstrucció* quan un agent s'interposa en el camí d'un altre, malgrat que no es necessiten per complir amb les seues tasques. Un exemple és la disputa de dos fusters virtuals per un mateix martell exposada en el punt 1.1, la qual requerirà l'ús de tècniques de coordinació d'accions. Quan tant els recursos com les habilitats són insuficients, cal una *Col·laboració coordinada*. Açò és, cooperacions més complexes en què es realitze tant la coordinació d'accions com el repartiment de tasques.

Si els agents tenen prou habilitats i recursos però els seus objectius són incompatibles, la situació es reconeix com una *Competició individual pura*. Aleshores els agents han de negociar per superar la incompatibilitat d'objectius. Un exemple podria ser una competició esportiva, en la qual tots els participants competeixen per guanyar. La situació en què els agents no disposen de totes les habilitats necessàries s'anomena *Competició col·lectiva pura*. En aquest context, els agents hauran d'agrupar-se i col·laborar per enfrontar-se amb altres grups; com ara en una carrera de relleus. Quan la mancança es troba als recursos, estem davant d'un *Conflicte individual de recursos*. Per exemple, en el joc del *Risk* el recurs insuficient és el territori. Finalment, si ambdós, habilitats i recursos, són insuficients esdevé un *Conflicte col·lectiu de recursos*. En el cas d'una guerra virtual, cada agrupació necessita una col·laboració coordinada dins del grup perquè les coalicions lluiten per obtenir els recursos.

Les condicions que ha de complir un sistema multiagent per permetre les interaccions suara esmentades es resumeixen en els següents punts:

- Presència d'agents capaços d'actuar i de comunicar-se.
- Existència de situacions en què els agents interactuen (p. ex. moviment d'agents, disputa de recursos limitats, etc.).
- Elements que permeten el desenvolupament de relacions (p. ex. canals de comunicació, accés o denegació d'accés a recursos, etc.).
- Autonomia dels agents per iniciar, mantenir i acabar les susdites relacions.

Com es pot apreciar, el marc de simulació multiagent presentat en el capítol 4 inclou totes aquestes característiques. Per tant, els entorns virtuals habitats creats amb ell poden reproduir del tot les situacions d'interacció que hem definit adés. Això no obstant, l'estat actual dels 3DIVA encara es troba lluny de poder tractar-les. Per tal de fer-ho, els actors necessiten incorporar mecanismes de coordinació de grups a diferents nivells (p. ex. navegació, presa de decisions, etc.).

D'acord amb açò, la recerca duta a terme en aquesta tesi s'ha centrat en aquelles situacions en què els objectius són compatibles i els actors sintètics són benvolents. Aquesta no és una suposició gaire desafortunada als mons simulats, ja que la compatibilitat i la benvolença es poden veure com el resultat d'una negociació prèvia entre personatges amb objectius incompatibles (p. ex. mitjançant un procés de regateig). Tanmateix, malgrat que els agents estiguen d'acord pel que fa a uns objectius compatibles, cal especificar encara els detalls de realització: quines tasques realitza cadascú, quan, a qui informen quan s'aconsegueixen, etc. Aquestes qüestions hauran de ser resoltes pels agents virtuals intel·ligents fent servir una presa de decisions coordinada.

5.2.2 Presa de decisions coordinada

Una volta que els agents són conscients de la necessitat d'adaptar els seus plans a la presència d'altres individus, han de decidir com coordinar la seua actuació; en vista de les dependències detectades. Hom pot concebre la coordinació a diferents nivells de granulació: el nivell *micro* i el nivell *macro* [ANA MAS, 2005].

La coordinació amb una perspectiva *micro* se centra en l'agent i és entesa com una forma d'adaptació a l'entorn. És a dir, cada agent monitoritza les dependències (potencials) amb els objectius, plans o accions de la resta d'agents. Aleshores, determina quines accions ha de realitzar per arribar a l'estat objectiu. Per exemple, considerem un agent que emprava un planificador STRIPS per resoldre un problema en el món de blocs [Nilsson, 2001]. En un context multiagent, aquest agent ha d'acomodar els seus plans per aprofitar les possibles sinergies (p. ex. altres agents poden col·locar blocs per ell) i evitar les dependències perjudicials (p. ex. altres agents poden desaparitar torres de blocs ja construïdes). La presa de decisions coordinada en termes de dependències entre les accions dels agents rep el nom de **raonament social**.

La perspectiva *macro* enfoca el problema de la presa de decisions coordinada des del punt de vista d'un observador extern. La qüestió essencial és com modificar el comportament perquè es produïsquen característiques desitjades en el comportament del sistema i/o dels seus agents (p. ex. benestar social, eficiència, comportament estratègic, etc.). L'**enginyeria social** estudia les diferents maneres d'influir en les decisions dels agents per reproduir els patrons desitjats pels dissenyador (p. ex. normes o lleis socials [Shoham and Tennenholtz, 1995]). És important adonar-se que influir en les decisions no significa determinar totalment el comportament d'un agent, sinó només predisposar-lo en una direcció desitjada. Per tant, aquest tipus d'aproximacions socials no s'oposen de cap manera a la noció fonamental d'autonomia d'un agent [Wooldridge and Jennings, 1995].

Com hem dit al llarg d'aquest document, els criteris racionals autointeressats no són prou a l'hora de simular societats d'actors sintètics amb un comportament versemblant. D'acord amb açò, l'arquitectura dels agents socialment intel·ligents descrita en el capítol 4 inclou el **mòdul social** (veure la figura 4.6). Aquest mòdul és l'encarregat d'incorporar criteris de tipus social o coordinats en la presa de decisions dels personatges 3D. En el que resta de capítol expliquem les dues aportacions fonamentals d'aquesta tesi per a la presa de decisions socials. A continuació, presentem una tècnica a nivell *micro* per aconseguir comportaments col·laboratius en agents basats en planificadors heurístics. En l'apartat 5.4, exposem un procés a nivell *macro* per a la presa de decisions socialment acceptables anomenat MADeM (*Multi-modal Agent Decision Making*).

5.3 COL·LABORACIÓ AMB PLANIFICADORS HEURÍSTICS

En aquest apartat, presentem una tècnica per aconseguir comportaments col·laboratius en aquells actors sintètics que empren els planificadors heurístics [Bonet et al., 1997] com a mecanisme de selecció dinàmica d'accions.

Segons l'anàlisi realitzat en la secció 3.3.3, hom pot afirmar que un conjunt d'agents col·labora quan els individus es comuniquen de manera directa i cooperen amb la resta d'agents per assolir els seus objectius. D'acord amb açò, en la secció 5.3.1, exposem com els personatges 3D intercanvien informació mitjançant **protocols de comunicació directa** per construir una representació dels agents externs; formada per un conjunt de creences sobre les seues accions i els seus objectius. Aquesta informació serveix perquè

els agents coordinen les seues accions, s'agrupen de forma dinàmica en equips de treball i cooperen per complir amb els seus objectius.

En aquest context, expliquem dos mecanismes per incloure criteris socials cooperatius en la presa de decisions dels agents virtuals intel·ligents. Ambdós són independents tant del planificador com de la funció heurística i han estat implementats sobre el planificador *miniMin-HSP* desenvolupat per Miguel Lozano per a l'animació comportamental de personatges intel·ligents 3D [Lozano, 2005]. En primer lloc, la secció 5.3.2 versa sobre la coordinació de tasques basada en una fase de *pre-planificació*, la qual evita els conflictes potencials que es puguen produir en l'ús de recursos compartits. Segonament, en la secció 5.3.3 proposem un mecanisme de cooperació basat en el **sospesament de la funció heurística**. Fonamentalment, aquesta aproximació consisteix a dividir l'objectiu global en subobjectius i assignar un pes a cadascun d'ells, de manera que reflectisca la seua importància social. Així doncs, els agents poden adoptar dinàmicament objectius externs (de menys importància que els propis) i aconseguir comportaments cooperatius sense augmentar la complexitat de les seues decisions.

5.3.1 Comunicació directa i agrupament entre actors

Un conjunt d'actors sintètics, que realitze tasques en un entorn virtual compartit, necessita treballar en equip per augmentar la seua eficiència; associada sovint amb la versemblança dels seus comportaments. El treball en equip (*teamwork*) ha estat estudiat profusament arran de la *teoria de les intencions conjuntes* [Cohen and Levesque, 1991] (veure la secció 3.3.3). Aquesta teoria assevera que la creació d'un equip necessita que cada membre es comprometa amb un objectiu comú i que reba el compromís de la resta d'individus.

Els sistemes que fan treball en equip normalment estableixen els equips d'antuvi i proveeixen els agents amb la consciència de formar part d'un equip [Grosz et al., 1999, Giampapa and Sycara, 2002]. La definició persistent dels equips, emperò, no és escaient als mons virtuals amb un cert grau de dinamisme. En aquests contextos, els caràcters poden canviar els seus objectius així com fer i desfer **associacions temporals** per aconseguir-los d'una manera més eficient (*ephemeral associations* [Bouron and Collinot, 1992]). Considerem, per exemple, un entorn que simula la construcció d'un edifici 3D i que està poblat per un cert nombre de paletes, cadascun ajudat per un manobre. Suposem, així mateix,

que diversos paletes encarreguen el seu manobre que faça una pastera de cement i que els recursos per fer-la són compartits (p. ex. els sacs de cement, la sorra, la llegona, etc.). Al començament, els manobres podrien no ser conscients d'aquesta assignació comú, aleshores, no hi hauria cap grup format i els agents actuarien de forma independent. Tanmateix, a mesura que s'adonaren d'aquest objectiu comú, es comprometrien a aconseguir-lo de forma conjunta. Posteriorment, si algun d'aquests manobres fóra manat a moure un palet de rajoles, hauria de deixar el grup de treball en què estava immers.

El procés d'agrupament entre els actors sintètics necessita la resolució de diversos subproblemes, com ara: la detecció de dependències, l'intercanvi d'informació mitjançant converses i la identificació de sinergies entre els objectius dels personatges. Tocant a la **detecció de dependències**, cal que els personatges s'adonen de quan la seua activitat està essent afectada o interrompuda per les accions d'altres individus. La figura 5.2 mostra la màquina d'estats genèrica que hem implementat per governar l'actuació dels personatges 3D basats en el planificador *miniMin-HSP* i detectar les seues dependències¹. L'estat principal és *SEARCH*, el qual invoca el planificador perquè prenga la decisió de la següent tasca a realitzar. La seua execució (estat *WORKING*) pot necessitar el moviment de l'agent (estat *NAVIGATE*) o la realització d'accions sobre els objectes de l'entorn (p. ex. agafar un objecte), de les quals cal esperar el resultat (estat *WAITING*). Dins d'aquest controlador, la necessitat de comunicació es genera quan una acció falla a causa de la interferència d'un altre agent (p. ex. un actor agafa un objecte que un altre caràcter desitjava). Els actors que defineixen la seua operativa basada en el model STRIPS poden detectar les interferències mitjançant la comprovació de les precondicions de les tasques, no sols durant el procés de cerca sinó també al llarg de la seua execució. Per resoldre aquesta dependència, el caràcter haurà de comunicar-se amb aquell que interferí en l'èxit de l'execució (estat *COMMUNICATE*).

Una vegada identificada la necessitat de comunicació (p. ex. una precondició ha estat violada) és necessari determinar quin tipus de conversa o protocol de comunicació és l'adequat a la situació. Per exemple, el **protocol de pregunta de la tasca actual**, descrit en la figura 4.8a, és interessant quan les interferències entre els personatges són esporàdiques (p. ex. menors a una freqüència fixada). Per mitjà d'aquest diàleg, l'agent interferent comunica la tasca que es disposa a realitzar (p. ex. *moure Cadira_1 des del Dormitori a la Cuina*). Aquesta informació entra en la memòria de l'agent interferit en forma de creences sobre les tasques actuals d'altres agents (veure la figura 5.5). Així, els

¹Per a una definició completa de l'arquitectura dels agents veure la secció 4.4



Figura 5.2: Detecció de dependències en el mòdul controlador dels agents basats en el planificador miniMin-HSP.

actors basteixen la seua memòria interna amb la representació d'altres actors externs.

Quan es treballa amb creences sobre tasques executades per altres personatges 3D, cal resoldre dues qüestions:

A quines circumstàncies una informació ha d'esdevenir una creença?: Qualsevol informació, obtinguda com el resultat d'una comunicació amb un altre agent, no hauria d'esdevenir directament una creença en la memòria d'agents; ja que aquest funcionament podria dur l'actor en un estat mental incoherent. Per exemple, imaginem que en una conversa prèvia, un agent A_0 ha estat informat que un objecte O_1 serà deixat damunt d'una taula T_1 per un agent A_1 . Posteriorment, un altre agent A_2 li tramet la seua intenció de col·locar un objecte diferent O_2 damunt de la mateixa taula T_1 . Aleshores, si suposem que la taula T_1 només pot allotjar un únic objecte, no té sentit que l'agent A_0 crega a ambdós interlocutors. En un context mancat de marques temporals que assegurin quan es realitzen les tasques (com és el cas del planificador *miniMin-HSP*), no és coherent donar validesa a ambdues afirmacions, ja que duen a una situació impossible. Així doncs, la decisió sobre la **credibilitat** implica un procés de raonament sobre la informació rebuda. Una simplificació raonable és el fet de creure que el primer agent que comunicà les seues intencions duu un cert avantatge sobre el segon; de caire físic (p. ex. començà abans la tasca) o de caire personal (p. ex. fou el primer a comunicar que ho faria). Malgrat que aquest criteri no garanteix que la decisió siga la correcta, és prou senzill i funcional. En tot cas, estem parlant de creences, i els personatges 3D podran també estar enganyats,

com sovint ho estem els éssers humans. Per tant, després de rebre un missatge de comunicació, l'agent comprovarà la compatibilitat de la nova informació amb les creences prèviament adoptades. Aquesta comprovació mirarà si els objectes sobre els quals actua la tasca no estaven ja afectats per la tasca d'alguna creença prèvia. En resum, l'actor confia en el primer agent que notifica les seues intencions i descarta les creences posteriors que afecten els mateixos recursos.

Quan una creença deixa de ser-ho?: La durabilitat de les creences és limitada, tanmateix, els personatges que ens les comunicaren no tornaran, habitualment, per informar si han aconseguit o no els seus propòsits. Llavors, és l'agent qui, de forma unilateral i mitjançant l'observació de l'entorn, ha de decidir quan una creença ha de ser esborrada de la memòria d'agents. La figura 5.3 mostra el mecanisme implementat a l'efecte. Cada vegada que una creença és introduïda en la memòria, l'agent marca tots els objectes afectats per ella. Aleshores, quan percep quelcom relacionat amb un objecte marcat, l'agent fa una **validació** de la creença associada. En un context de definició d'operadors de tipus STRIPS, la validació consisteix a comprovar les precondicions de la tasca creguda. Tant se val que la tasca haja estat completada com que haja estat interrompuda; les precondicions no continuaran complint-se. Aquest tipus de validació, per tant, no discrimina l'èxit o el fracàs d'una tasca (cosa innecessària ja que l'agent percep l'estat del món que és allò important), sinó simplement reconeix quan una creença ja no és viable i ha de ser eliminada de la memòria (invalidació en la figura 5.3).

En la secció 5.3.2 exposem com els agents virtuals intel·ligents poden fer servir aquesta representació en memòria dels agents externs per coordinar l'accés als recursos compartits i millorar, doncs, la qualitat de l'animació comportamental.

Formació d'equips

Els personatges que no interfereixen molt a sovint, normalment, no necessiten cooperar per assolir els seus objectius. Tanmateix, convé que aquells que interrompen constantment les seues activitats consideren la possibilitat de formar un equip temporal de treball en què els membres cooperen per aconseguir un objectiu conjunt amb més eficàcia. Aquesta avaluació es farà a través del **protocol de formació d'equips** de la figura 4.8b. En aquest tipus de conversa, els agents intercanvien els seus objectius, compro-

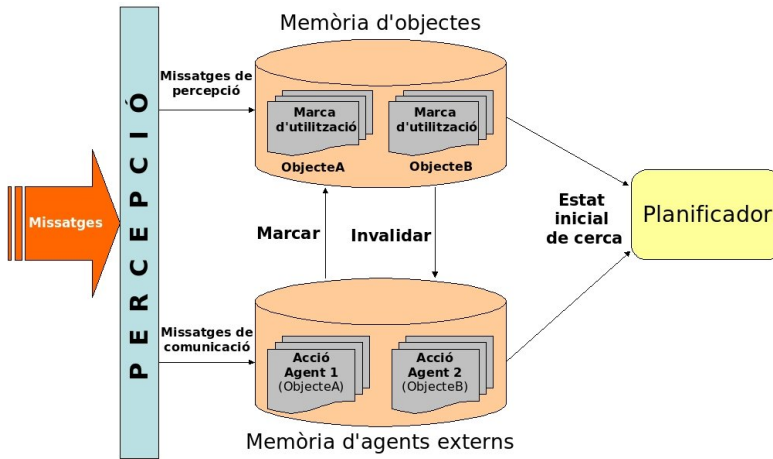


Figura 5.3: Estructura de la memòria dels actors sintètics.

ven el seu nivell d'acoblament i decideixen si ajunten els seus objectius i cooperen per aconseguir-los o no. Siguen A i B dos agents, els objectius dels quals estan formats per una llista de fets ($G_A = \{f_i\}$, $G_B = \{f_j\}$), distingim tres nivells diferents d'acoblament:

- Els objectius estan *totalment relacionats* quan comparteixen el mateix conjunt de fets ($G_A = G_B$) o quan són un subconjunt l'un de l'altre ($G_A \subset G_B$ o $G_A \supset G_B$).
- Els objectius estan *parcialment relacionats* quan la seua intersecció no és completa, açò és, hi ha fets que només pertanyen a un agent ($G_A \cap G_B \neq \phi$).
- Els objectius estan *no relacionats* quan no hi ha intersecció entre ells ($G_A \cap G_B = \phi$).

Els objectius totalment relacionats poden ser problemàtics, ja que, si diversos agents tracten d'aconseguir-los de forma simultània, produiran moltes dependències i conflictes (p. ex. els agents volen els mateixos objectes en el mateix moment). Així doncs, el protocol de formació d'equips proposarà la creació d'un grup quan els dos candidats afectats per la conversa tinguen **objectius totalment relacionats**. Per exemple, considerem una possible extensió del conegut problema *funny dinner-date* [Cavazza et al., 2002], en la qual tres actors 3D han de netejar l'apartament d'un d'ells. En concret, el propietari vol netejar el pis sencer, mentre que els dos amics que l'acompanyen només volen ajudar-lo en certes tasques (veure la figura 5.4). En aquest escenari, el propietari (P) es pot ajuntar

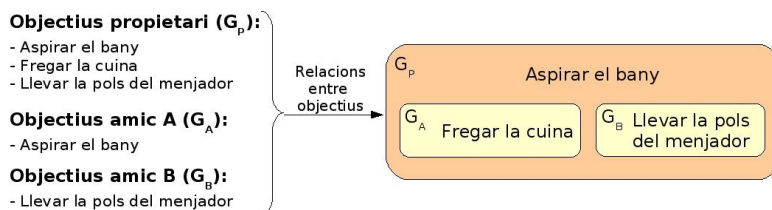


Figura 5.4: Exemple de les relacions entre els objectius de tres agents que estan netejant un apartament.

amb l'amic *A* per fregar la cuina i amb l'amic *B* per llevar la pols del menjador. Tanmateix, la transitivitat no pot ser aplicada quan es formen els grups. Malgrat que els objectius G_p i G_A estan totalment relacionats, G_A i G_B no estan relacionats. Per tant, *A* s'ajunta amb *P* però no amb *B*. És a dir, el protocol de formació d'equips es limita a decidir si dos actors sintètics cooperen, sense afectar els seus compromisos previs. Això no obstant, els equips formats per més de dos agents són possibles de reproduir, ja que els actors poden crear de manera separada tantes parelles com necessiten.

Mentrestant, els personatges formen part d'un equip, comuniquen contínuament les seues intencions als seus companys per facilitar la **coordinació de tasques** (secció 5.3.2). A més a més, els seus objectius passen a formar part de la representació en memòria que tenen els uns sobre els altres. Llavors, el **comportament cooperatiu** dels membres d'un equip s'aconsegueix mitjançant la inclusió d'aquests objectius externs en el procés de planificació, aspecte que es tracta en la secció 5.3.3.

Darrerament, els actors sintètics han de ser capaços de deixar els equips, açò és, de comunicar el cessament de la cooperació per assolir els objectius externs. Anàlogament a la formació dels equips, açò succeirà quan els objectius d'un agent deixen d'estar totalment relacionats amb els d'aquells amb qui estan agrupats.

5.3.2 Coordinació de tasques amb recursos compartits

A l'hora de **reduir els problemes d'obstrucció** característics de les simulacions multi-agent en entorns amb recursos compartits, els actors poden coordinar la seua operativa si coneixen les intencions dels agents que els envolten. La informació sobre les tasques que

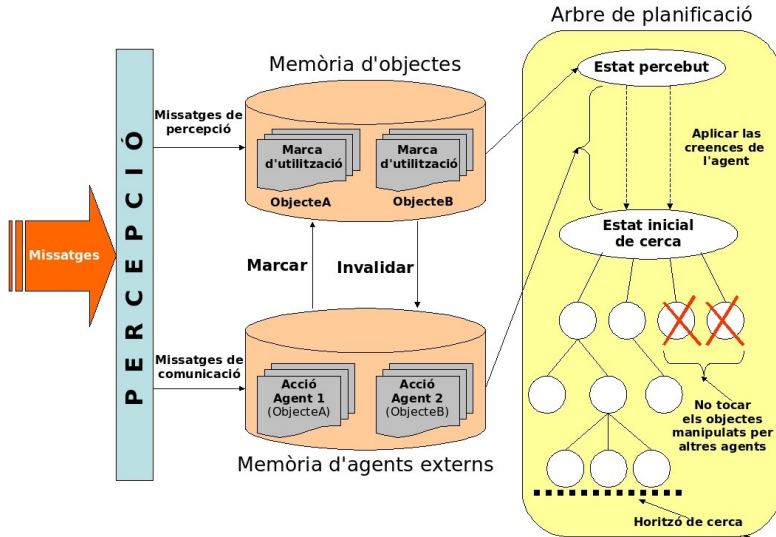


Figura 5.5: Model de coordinació de tasques amb recursos compartits.

altres individus pretenen completar pot ser emprada per gestionar les restriccions imposades per elles (p. ex. un bàrman virtual pot ajornar la resolució d'una comanda i atendre altres clients mentrestant l'ampolla que necessita estiga en possessió d'un altre bàrman). Per representar les intencions dels agents externs, els actors 3D emprun un model estès de memòria que, a banda de l'estat del món apercibut, conté les creences sobre les tasques actuals; rebudes a través de converses amb altres cohabitants del món virtual (veure la figura 5.5). Considerem una creença com una conjectura sobre un canvi del món que un agent extern es disposa a realitzar.

A continuació, presentem com aquesta informació ha estat usada pel formalisme de planificació de l'agent per a realitzar un accés coordinat als recursos compartits. Hom espera que els actors sintètics benvolents coordinen les seues tasques i eviten destorbar-se els uns als altres. La coordinació fa tendir els actors a treballar en tasques separades o sobre diferents objectes 3D; els quals són els recursos del problema. Pareix raonable, doncs, que els personatges planifiquen el seu comportament (les tasques a realitzar) mentre deixen la resta d'individus finalitzar els seus propòsits. Per tal de fer açò, els actors han de completar l'estat del món amb les activitats ja planificades per altres personatges, l'execució de les quals es considera en marxa. Llavors, el formalisme de planificació tindrà

més informació per seleccionar una tasca que reduïska les interferències.

La figura 5.5 mostra esquemàticament com el planificador emprat pels nostres agents socialment intel·ligents (*miniMin-HSP* [Lozano, 2005]) ha estat modificat per començar la cerca des d'un **estat futur virtual**, que és el resultat d'aplicar les creences sobre l'estat actual apercibut. Aquesta execució mental de les tasques d'altres agents externs es realitza aplicant les llistes *add* i *delete* de la definició STRIPS dels operadors. D'aquesta manera, l'agent construeix una situació prospectiva que tracta de passar per damunt de les dependències actuals. L'objectiu és que l'agent planifiqui arran d'un punt en el temps on la resta d'actors ha completat la seua tasca actual i evite, doncs, interrompre'ls; cosa que es reflectirà en una animació més coordinada.

Les creences corresponen amb tasques que encara no han estat completades; de fet, res no garanteix que finalment es completen satisfactòriament. Per tant, començar només la planificació arran d'un estat en què totes les creences han estat acabades no és prou per aconseguir **evitar la competició de recursos**. Per exemple, un agent podria decidir emprar un objecte que està essent utilitzat en aquest moment per un altre per a completar una creença. De cara a evitar aquest tipus de conflictes, cada creença blocarà els objectes que utilitza en el primer nivell de l'arbre de cerca; que és el nivell d'on ix la següent tasca a realitzar. Aquest funcionament prohibeix que un agent comprometa l'èxit d'una tasca prèviament iniciada, ja que no podrà manipular els objectes que estan implicats en ella (veure la figura 5.5). La planificació amb aquests recursos estarà de nou disponible a partir del segon nivell de l'arbre de cerca. En resum, l'agent pot planificar amb els recursos associats a creences, però es prohibeix el seu ús immediat.

El mecanisme de blocatge de recursos només en el primer nivell de l'arbre és suficient en un context de planificació contínua (*continuous planning* [Avradinis et al., 2005]), com el proposat pels nostres 3DIVA. El resultat d'aquest tipus de planificació, que intercala percepció, planificació i acció, no és un pla complet sinó només la següent tasca a executar; una vegada finalitzada la qual es torna a invocar el planificador. D'acord amb açò, si un agent finalitza la primera tasca del seu pla abans d'invalidar una creença prèvia (sense que un altre agent acabe o interrompa la seua tasca objectiu), la restricció de no emprar els recursos en el primer nivell del arbre de cerca tornarà a impedir que seleccionem una tasca que utilitze els objectes afectats per la creença.

La coordinació de tasques dins d'un equip de treball requereix la publicació, en forma

de creences, de les tasques en procés de realització. Segons el procediment descrit abans, la inclusió en el procés de planificació d'aquestes tasques servirà perquè els membres del grup eviten destorbar els seus companys i per relaxar la competició de recursos. Per una altra banda, la participació en un equip implica el compromís d'assolir uns objectius comuns. D'acord amb açò, en la següent secció presentem un mecanisme de sospesament heurístic per incloure els objectius d'altres agents en el procés de planificació i produir comportaments cooperatius entre els membres d'un equip.

5.3.3 Cooperació mitjançant sospesament heurístic

En aquesta secció presentem una segona manera de reduir les dependències entre les activitats dels personatges 3D: la **cooperació** per aconseguir un **objectiu global**. Segons hem explicat en la secció 5.3.1, la formació d'un equip implica que els actors es comprometen a compartir els seus objectius. La planificació sobre un objectiu comú, format pels objectius propis i per altres objectius externs adquirits, pot ser molt profitosa a l'hora de reduir les obstruccions entre els membres d'un grup (p. ex. els agents faran tasques que beneficien els altres). Nogensmenys, cal que els agents utilitzen alguna tècnica per reflectir la importància social dels objectius propis respecte dels objectius de la resta d'actors. A continuació, proposem la divisió de l'objectiu comú en **subobjectius independents**; sobre els quals els agents poden aplicar diferents **pesos heurístics**.

El mecanisme de sospesament heurístic, que estudiem en aquesta secció, pot ser utilitzat sobre heurístiques independents del domini que treballen sobre **problemes de tipus STRIPS**. Un problema STRIPS es defineix com un vector $P = \langle A, O, I, G \rangle$, en què: A és un conjunt de predicats atòmics o fets; O es un conjunt d'accions (operadors instanciats); per últim, $I \subseteq A$ i $G \subseteq A$ corresponen a les situacions inicial i final.

Una acció es representa com el vector $o = \langle pre(o), add(o), del(o) \rangle$, en el qual: $pre(o)$ és la llista amb les precondicions d' o ; $add(o)$ i $del(o)$ són les llistes amb els àtoms que han de ser afegits i esborrats, respectivament, quan s'executa o . D'acord amb aquestes llistes, una acció (o) és aplicable a un estat (S) si i només sí $pre(o) \subseteq S$. Aleshores, el resultat d'executar una acció simple sobre un estat es defineix segons l'equació 5.1. Així mateix, el resultat d'aplicar una seqüència d'accions a un estat es defineix recursivament com mostra l'equació 5.2.

$$Result(S, o) = S \cup add(o) \setminus del(o) \quad (5.1)$$

$$Result(S, \langle o_1, \dots, o_n \rangle) = Result(Result(S, \langle o_1, \dots, o_{n-1} \rangle), o_n) \quad (5.2)$$

Donat un problema STRIPS, un pla (o possible solució del problema) és una seqüència $P = \langle o_1, \dots, o_n \rangle$ d'accions d' O amb l'execució de la qual s'aconsegueix que $G \subseteq Result(I, P)$. Llavors, l'objectiu de la planificació és trobar la mínima seqüència d'accions que transforme l'estat inicial del sistema en un estat objectiu desitjat. Diverses famílies d'algorismes han sigut desenvolupades, entre les més notables: Graphplan [Blum and Furst, 1997], FastForward [Hoffmann and Nebel, 2001], *Heuristic Search Planning* [Bonet and Geffner, 2001], etc. La família dels planificadors heurístics (*Heuristic Search Planner* o HSP [Bonet and Geffner, 2001]) utilitza les **funcions heurístiques** per guiar la cerca d'aquesta solució.

La funció heurística estima el nombre mínim d'accions necessàries per arribar a un estat desitjat i pren la forma de l'equació 5.3. Açò és, una funció definida sobre el conjunt dels reals (\mathbb{R}), que estima la distància que hi ha des de l'estat que s'està avaluant (s) fins l'estat objectiu (G). Hi ha moltes maneres de definir $g(G, s)$. Una forma molt comuna és definir aquesta distància com la suma de les distàncies a cadascuna de les proposicions que formen l'estat objectiu (heurístiques additives). L'equació 5.4 formalitza aquest càlcul de la funció heurística, en què p són les proposicions de G i $g(p, s)$ és el cost d'aconseguir p des de l'estat s .

$$h_G(s) = g(G, s) \quad (5.3)$$

$$g(G, s) = \sum_{p \in G} g(p, s) \quad (5.4)$$

El tipus d'heurística expressat en l'equació 5.4, emperò, no és admissible². La causa radica en què el procés per obtenir una proposició de l'estat objectiu, des de l'estat avalu-

²És a dir, la cerca mitjançant un algorisme de tipus A^* no té garantit trobar la solució òptima

at, pot també aconseguir altres proposicions desitjades com efecte col·lateral. L'equació 5.5 representa una versió admissible de la funció anterior [Haslum and Geffner, 2000], la qual fa servir el quantificador màxim en comptes de la suma de distàncies; cosa que farà que aquesta heurística siga menys informativa. Ambdues heurístiques defineixen el cost d'aconseguir una proposició p des d'un estat s ($g(p, s)$) com 0 si p pertany a l'estat s ($p \in s$). Altrament, li assignen ∞ com el valor inicial i realitzen una revisió incremental, definida per l'equació 5.6, fins que el cost $g(p, s)$ no canvie.

$$g(G, s) = \max_{p \in G} g(p, s) \quad (5.5)$$

$$g(p, s) = \min_{a \in O(p)} [g(p, s), 1 + g(Pre(a), s)] \quad (5.6)$$

Les heurístiques anteriors assumeixen que les proposicions que formen l'objectiu global són **independents**. Açò és, que els plans per aconseguir un subobjectiu no tindran cap efecte en els plans per aconseguir uns altres. D'acord amb aquesta suposició, és possible introduir un pes associat a cada proposició objectiu, de manera que reflectisca la importància de cada proposició, tal i com mostra la següent fórmula:

$$H_w(G, s) = \sum_{p \in G} w(p) * g(p, s) \quad (5.7)$$

Els pesos $w(p)$ varien entre 0 i 1. D'una banda, quan tots els pesos tenen valor 1, l'equació 5.7 és equivalent a l'equació 5.4. D'una altra, quan una proposició rep un pes de valor 0 no tindrà cap efecte en el valor final de l'heurística. És a dir, l'heurística es comportarà com si aquesta proposició no existís en l'estat objectiu. Els pesos $w(p)$ poden ser molt útils a l'hora d'incloure objectius externs de forma dinàmica i combinar-los amb els propis de manera cooperativa. Dins d'aquest context, hom pot concebre el sospesament com un mecanisme per transformar un objectiu global en una agenda, ja que els subobjectius són ordenats d'acord amb la seua significació.

És important considerar com el sospesament de les proposicions afecta el comportament de l'heurística i, en conseqüència, la resposta del planificador; açò és, l'eficiència i l'ordre en què les proposicions són obtingudes. Per exemple, el fet d'assignar un pes

baix a aquelles proposicions que han de ser acomplides necessàriament abans que altres (amb un pes major), pot produir una solució de baixa qualitat i/o un temps de resolució del pla roí. Fins i tot, pot provocar que el planificador no trobe cap solució. La causa d'aquest problema és que, generalment, les proposicions que formen part de l'objectiu no són independents; malgrat que açò fóra suposat en 5.4 i 5.5 com a simplificació del problema.

Per eludir aquest comportament no desitjat, proposem la separació de l'objectiu global en grups de proposicions o **subobjectius independents**. Llavors, els pesos s'aplicaran a cada subobjectiu en comptes de fer-ho a cada proposició. En problemes complexos, la identificació de subobjectius independents pot arribar a ser tan difícil com el mateix procés de planificació. Tanmateix, als entorns virtuals, els problemes de planificació amb objectes 3D solen estar acotats en departaments estancs, ja que cada objecte només es relaciona amb uns pocs objectes de la seua rodalia. Per tant, hem definit un estimador que divideix l'estat objectiu atenent només als objectes que apareixen en les proposicions que el formen, fent servir la següent relació:

$$f_1 \sim f_2 \Leftrightarrow \exists o : (f_1(o, o') \vee f_1(o', o)) \wedge ((f_2(o, o') \vee f_2(o'', o))) \quad (5.8)$$

Açò és, dues proposicions f_1 i f_2 estan relacionades ($f_1 \sim f_2$) si fan referència a un objecte comú. A pesar que l'equació 5.8 defineix la relació mitjançant proposicions d'aritat 2, la definició pot ser estesa a qualsevol proposició de manera trivial. Llavors, la partició de l'estat objectiu (G) vindrà donada com el conjunt quocient d'aquesta relació:

$$\Gamma_G = G / \sim \quad (5.9)$$

Aquest estimador no garanteix que els subobjectius siguin independents, però permet calcular de forma eficient un conjunt finit de **subobjectius quasi independents**. El conjunt d'estats subobjectiu ($SubG_i$), resultat de la partició de l'equació 5.9, serà sospesat per considerar la seua importància segons l'equació següent:

$$H_w(G, s) = \sum_{SubG_i \in \Gamma_G} w(SubG_i) * g(SubG_i, s) \quad (5.10)$$

Quasi qualsevol funció heurística pot ser escrita d'aquesta manera. En el nostre planificador, emprem una heurística similar a l'equació 5.5, però parem esment a la partició en subobjectius de la següent manera:

$$H_w(G, s) = \sum_{SubG_i \in \Gamma_G} (w(SubG_i) * \max_{p \in SubG_i} [g(p, s)]) \quad (5.11)$$

on $g(p, s)$ calcula una estimació del que costa aconseguir la proposició p des de l'estat s , de manera similar a com fou expressat en l'equació 5.6.

A pesar que aquesta heurística és molt similar a l'heurística admissible presentada per Blai Bonet en [Bonet and Geffner, 2001], cal destacar que aquesta nova estimació tracta d'obtenir el valor de les relacions i de les restriccions establides entre els recursos. Convé puntualitzar, així mateix, que aquesta tècnica de sospesament de la funció heurística podria no funcionar com s'espera en alguns algorismes de planificació. En particular, pot tindre problemes si l'algorisme poda algunes accions abans d'avaluar l'heurística o si no avalua tots els estats successors (com succeeix en l'algorisme d'ascens de colines o *enforced hill climbing* [Hoffmann and Nebel, 2001]). Pels experiments duts a terme en aquesta tesi, hem emprat l'algorisme **Minimin** [Lozano, 2005], tot i que també haguera estat possible usar un algorisme de cerca del tipus 'el primer millor' o *best-first search algorithms* (p. ex. A^* , RTA^*).

Aquesta secció deixa obertes certes qüestions, que han requerit d'una fase d'experimentació, com ara: quins valors són els adequats pel sospesament dels subobjectius adquirits dels agents externs i com afecten l'ordre en què les proposicions són aconseguïdes, així com l'eficiència de la planificació. En la secció 6.2 donarem resposta a aquestes qüestions i demostrarem el grau de coordinació i cooperació assolit pels 3DIVA basats en planificadors heurístics.

Fins ara, hem estudiat la introducció de criteris socials en la presa de decisions basada en planificadors heurístics. El que resta de capítol versa sobre un nou tipus de presa de decisions social que hem anomenat MADeM (*Multi-modal Agent Decision Making*) i que pot ser emprada per agents de tipus BDI per tal de prendre decisions socialment acceptables.

5.4 MADEM: MULTI-MODAL AGENT DECISION MAKING

La presa de decisions és el procés cognitiu que permet un agent seleccionar el curs de les seues accions entre un conjunt de possibilitats. Encara que hi ha diversos factors que influencien aquest procés, tal vegada, un dels més importants és el criteri emprat per l'agent a l'hora de realitzar la decisió. Des d'un punt de vista racional, la presa de decisions ha seguit habitualment el criteri de l'eficiència. Açò és, la decisió adoptada ha estat aquella que condueix l'agent a l'estat objectiu d'una manera més ràpida o consumint la quantitat mínima de recursos. Nogensmenys, aquest comportament, tan arrelat en el camp de la robòtica, no és precisament el més versemblant quan simulem humanoides virtuals. Pel contrari, el raonament humà sovint avalua diferents criteris o **punts de vista**, els quals hauran de ser tinguts en compte en el procés de presa de decisions dels actors sintètics. Per exemple, un client que entra en un bar virtual i ha de decidir a quin cambrer fer la seua comanda podria considerar diversos punts de vista, com ara: a) el cansament (p. ex. demanar el cambrer que estiga més a prop); b) l'utilitat (p. ex. demanar el cambrer menys ocupat); o c) l'amistat (p. ex. demanar un cambrer que és amic).

A més a més, la presa de **decisions socialment acceptables** en contextos multiagent requereix que els agents avaluen l'impacte social de les seues accions. Per exemple, els estudis sociològics realitzats per Jean Piaget [Piaget, 1995] determinen que la sociabilitat d'un grup es manifesta com un intercanvi de serveis entre els seus membres. Aquests intercanvis produeixen guanys i pèrdues d'elements energètics (p. ex. temps, diners, energia, etc) i motivadors (p. ex. inversió en serveis fets i serveis esperats en retorn). Llavors, per mesurar aquestes variacions hom ha definit els valors d'intercanvi o *exchange values*. Els valors d'intercanvi poden servir per modelitzar l'intercanvi social entre els agents d'una societat artificial i incloure criteris personals en l'avaluació d'aquests intercanvis [Ribeiro et al., 2003]. Un dels reptes actuals en aquesta línia consisteix a integrar aquest tipus de tècniques socials en els sistemes multiagent, de manera que es puguin obtenir eines per al desenvolupament de simulacions socials multiagent (p. ex. MAS-SOC [Bordini et al., 2005]).

Fonamentat en els requisits anteriors, en aquesta secció presentem **MADeM (Multi-modal Agent Decision Making)**, un procés de presa de decisions perquè els actors sintètics puguin prendre decisions socials multimodals. Definim aquest tipus de decisions com aquelles que consideren diferents punts de vista provinents de diverses fonts d'infor-

mació (agents externs). D'una banda, cada punt de vista o focus d'atenció proporcionarà un criteri diferent a la presa de decisions autointeressada; la qual estarà, doncs, més informada. Per exemple, un agent podria sospesar diferents aspectes com ara: l'eficiència, el cansament, la manya, etc. D'una altra banda, MADeM integra la influència social a través de la consulta a altres agents externs sobre aquells punts de vista en què l'agent està interessat. Aquesta retroalimentació social subministrarà les preferències de la resta d'agents per a cadascuna de les possibles solucions del problema. Amb tota aquesta informació, MADeM realitza una presa de decisions que podem adjectivar com **socials i multimodals**.

En la subsecció següent descrivim la notació i els elements que defineixen una presa de decisions de tipus MADeM. A continuació, les subseccions 5.4.2 i 5.4.3 versen, respectivament, sobre la representació de les solucions d'un problema i sobre l'expressió de preferències mitjançant funcions d'utilitat. Finalment, en la subsecció 5.4.4 expliquem el procediment de presa de decisions realitzat per MADeM.

5.4.1 Elements i notació

La presa de decisions social duta a terme per MADeM ha estat inspirada per la teoria de la **repartició de recursos en entorns multiagent (Multi-Agent Resource Allocation** o MARA [Chevalyere et al., 2006]); que introduïrem en la secció 3.4.3. MARA és el procés de distribució d'un nombre d'ítems (recursos) entre un conjunt d'agents. La definició d'aquest procés necessita resoldre qüestions com ara quin és el tipus de recursos que es distribuïran o quin és el procediment d'assignació emprat. MADeM comparteix el mateix domini de definició que MARA, però afegint-hi algunes característiques noves. Els elements que defineixen el procés de presa de decisions de tipus MADeM es poden resumir en els punts següents:

- Un conjunt d'agents $A = \{a_1, \dots, a_n\}$, on cada a_i representa un agent particular involucrat en la decisió. Associat a cadascun d'aquests agents hi ha un vector de pesos $\vec{w} = \langle w_1, \dots, w_n \rangle$, el qual representa la seua actitud personal envers la resta d'agents (p. ex. l'egoisme, l'altruisme, la indiferència, etc). D'acord amb aquests pesos, MADeM és capaç de sospesar la informació rebuda des d'altres agents (veure la secció 5.4.4).
- Un conjunt de recursos que els agents han d'assignar $R = \{r_1, \dots, r_m\}$. En MA-

DeM, les solucions a un problema donat són representades com assignacions de recursos. La definició dels recursos que fa servir MADeM és lleugerament diferent a la que trobem en la literatura clàssica de MARA i serà tractada amb detall en la secció 5.4.2.

- Un conjunt de funcions d'utilitat $\{U^1, U^2, \dots, U^q\}$, perquè cada agent avaluï les assignacions de recursos suara esmentades des de diferents punts de vista ³. Les propietats que han de complir aquestes funcions d'utilitat són estudiades en la secció 5.4.3. Relacionat amb aquestes funcions, cada agent fa servir un vector de pesos d'utilitat $\vec{w}_u = \langle w_{u1}, \dots, w_{uq} \rangle$, el qual representa la importància que dona l'agent a cada punt de vista en la presa de decisions multimodal. En l'exemple de la secció anterior, on un client d'un bar virtual havia de decidir a quin cambrer fer-li la seua comanda, els pesos d'utilitat expressarien tendències personals, com ara: la mandra, la impaciència o el foment de l'amistat.

5.4.2 Tipus de recursos

MADeM proporciona als agents l'habilitat de preguntar l'opinió dels altres sobre les diferents solucions d'un problema (p. ex. quan un agent rep un encàrrec, pot avaluar si li interessa més realitzar-lo o passar la tasca a un altre agent). D'acord amb la filosofia MARA, MADeM representa cadascuna d'aquestes solucions com un problema d'assignació de recursos, en què els recursos considerats són les tasques. L'assignació de tasques ha estat estudiada àmpliament en la comunitat de sistemes multiagent però la seua aplicació als agents virtuals socials no és gaire comuna i necessita tindre en compte nous aspectes.

La principal novetat de la nostra aproximació radica en què MADeM no considera únicament l'execució d'una tasca, com es pot trobar normalment a la literatura. Pel contrari, el tipus de recurs que serà assignat és el que nosaltres anomenem **ranures de tasques** o *task slots*. Considerem les ranures de tasques com paràmetres que necessiten ser assignats perquè una tasca pugui ser executada (de conformitat amb els esquemes d'acció presentats en la secció). Associats a qualsevol tasca, distingim dos tipus principals de ranures: a) les ranures d'agents o *agent slots*, les quals corresponen amb els agents que interpreten algun rol en la tasca (p. ex. l'executor o el beneficiari de la tasca); i b) les ranures d'objectes

³Al contrari dels problemes clàssics de MARA, en què els agents només avaluen una única funció d'utilitat

o *object slots*, que fan referència als objectes involucrats en la tasca (p. ex. per clavar un clau cal un martell o algun objecte amb què poder colpejar).

Atenent al rol que juguen en l'execució de la tasca, podem identificar tres ranures de tasques generals. Hi ha una ranura que estarà present a qualsevol tipus de tasca, que és l'agent que la duu a terme (Ag_e). A més a més, podem considerar també com casos generals dues ranures que estan presents en moltes tasques: l'objecte principal de l'acció (Obj_m) i el receptor, en qualsevol sentit, dels efectes de l'acció (Ag_d). Per exemple, en la tasca *Donar*, Ag_e és l'agent que té l'objecte a l'inici, Obj_m és l'objecte que està essent transferit i Ag_d és l'agent que rep l'objecte. Amb tot i això, sempre és possible considerar tantes ranures de tasques com siguin necessàries per a un problema particular.

Com es pot apreciar, l'assignació clàssica de tasques es troba subsumida per la nostra aproximació, ja que correspon a l'assignació de la ranura Ag_e . Així mateix, la repartició d'objectes entre un conjunt d'agents també és possible si la modelem com una ranura d'una tasca adequada (p. ex. la ranura Obj_m de l'acció *Posseir*).

En resum, MADeM representa els recursos en forma de ranures de tasques ($r = task(slot)$). L'assignació d'un element (agent o objecte) a una ranura és representada, aleshores, com $task(slot) \leftarrow element$. D'acord amb açò, una distribució (P) d'elements en ranures de tasques té el següent aspecte:

$$P = \{t_1(s_1) \leftarrow e_1, t_1(s_2) \leftarrow e_2, \dots, t_1(s_n) \leftarrow e_n, t_2(s_1) \leftarrow e_{n+1}, \dots, t_m(s_n) \leftarrow e_{n*m}\} \quad (5.12)$$

MADeM representa cadascuna de les solucions considerades per un problema de decisió com una distribució de l'estil de l'equació 5.12. Per exemple, suposem que en un bar virtual modelem la tasca *Servir*(*Cambrer*, *Producte*, *Client*), la qual és utilitzada pels cambrers a l'hora de servir un producte a un client. Aquesta tasca conté dues ranures que poden ser subhastades: la ranura Ag_e , que correspon amb el cambrer que serveix; i la ranura Ag_d , que és el client que rep el producte. Aquestes ranures serien representades com *Servir*(Ag_e) i *Servir*(Ag_d). Llavors, una possible assignació per aquestes ranures podria ser $\{Servir(Ag_e) \leftarrow a_1, Servir(Ag_d) \leftarrow a_5\}$, considerant que a_1 fóra un cambrer i a_5 un client.

5.4.3 Expressió de preferències

Els agents involucrats en un procés de presa de decisions MADeM han d'expressar les seues preferències per a cadascuna de les solucions d'un problema donat. A l'hora de fer-ho, els agents fan servir les **funcions d'utilitat** [Fishburn, 1970]. Aquestes funcions avaluen, de forma numèrica, l'interès que té un agent per una distribució d'elements en ranures de tasques (veure l'equació 5.12).

El tipus de representació de les funcions d'utilitat és un aspecte a tindre en compte. Per exemple, l'enumeració dels valors d'utilitat per cadascuna de les assignacions possibles és normalment inviable, ja que el nombre de combinacions creix de manera exponencial. D'acord amb açò, MADeM emprava un tipus de representació més concisa i eficient, les funcions d'utilitat *k-additives*. Una funció d'utilitat, per a una assignació de recursos P , és *k-additiva* si es pot expressar com un sumatori de valors d'utilitat assignats a subconjunts T de grandària màxima k (veure l'equació 5.13).

$$U(P) = \sum_{T \in P} u(T) \quad |T| \leq k \quad (5.13)$$

La representació *k-additiva* de les funcions d'utilitat és totalment expressiva, ja que pot descriure qualsevol funció d'utilitat sempre que k siga suficientment gran. Això no obstant, segons afirma [Chevalerey et al., 2006], en molts dominis d'aplicació és raonable assumir un valor reduït de k . En conseqüència, els agents MADeM fan servir **funcions d'utilitat 2-additives no negatives**, tal i com especifica l'equació 5.14.

$$U(P) = \sum_{p \in P} u(p) + \sum_{p_1, p_2 \in P} u(p_1, p_2) \quad (5.14)$$

Açò és, la utilitat d'una distribució d'elements P es calcula com la suma de les utilitats de cada assignació ($u(p)$) més els valors d'utilitat extra que aporten les assignacions en parella $u(p_1, p_2)$. De cara a obtenir les funcions d'utilitat normalitzades ($U(P) \in [0, 1]$), les funcions d'utilitat de tots els agents han de ser dividides per la mateixa constant, la qual dependrà del problema particular.

Per últim, les funcions d'utilitat en MADeM expressen benefici, malgrat que els tipus

de recursos són ranures de tasques. Per tant, a l'hora de prendre la decisió, l'objectiu dels agents serà maximitzar la seua utilitat (veure la secció següent). L'ús de funcions d'utilitat que expressaren costs també hagués estat possible. Llavors, els agents haurien de minimitzar aquests costs en la seua presa de decisions.

5.4.4 Procediment de presa de decisions

Per arreplegar les preferències externes sobre les diferents solucions a un problema, MADeM fa servir les **subhastes combinatòries d'una ronda amb apostes no visibles** (*one-round sealed-bid combinatorial auctions*). En aquest context, els rols de subhastador i apostador no són jugats per uns agents fixos del sistema. Al contrari, els agents adoptaran dinàmicament aquests rols, depenent de les seues necessitats o interessos. Per exemple, un agent que vol decidir si realitza una tasca o passa l'execució a un altre, actuarà com el subhastador. Aleshores, els agents que reberen la subhasta apostarien amb els seus valors d'utilitat, sempre que estigueren interessats en les ranures de tasques que s'estan subhastant. Nogensmenys, com que les decisions complexes poden necessitar la consideració de més d'un punt de vista, el subhastador pot requerir que la resta d'agents expressen les seues preferències mitjançant diverses funcions d'utilitat. Per tant, MADeM permet executar més d'una subhasta alhora. Així mateix, els agents poden participar en diverses subhastes de manera simultània. En conseqüència, la presa de decisions de tipus MADeM es troba entremig de l'assignació de recursos basada en el mercat centralitzada (perquè hi ha un subhastador central que decideix) i distribuïda (perquè tots els agents poden ser subhastadors).

La figura 5.6 mostra l'esquema del **procediment de presa de decisions** dut a terme per MADeM per generar decisions socials multimodals. Aquest procediment se subdivideix, fonamentalment, en els següents passos:

- 1 *Fase de subhasta*: Aquesta fase és realitzada per un agent (a_1) que desitja resoldre un problema de decisió amb informació social (p. ex. a quina taula seure en un bar virtual). Llavors, l'agent construeix el conjunt de distribucions que representen les possibles solucions al problema ($\{P_1, P_2, \dots, P_m\}$). Cadascuna d'aquestes distribucions estarà formada per un conjunt d'assignacions d'elements a ranures de tasques, com ara $Seure(Obj_m) \leftarrow taula_1$. A continuació, l'agent subhasta aquestes distribucions a un grup particular d'agents, que anomenarem agents objectiu.

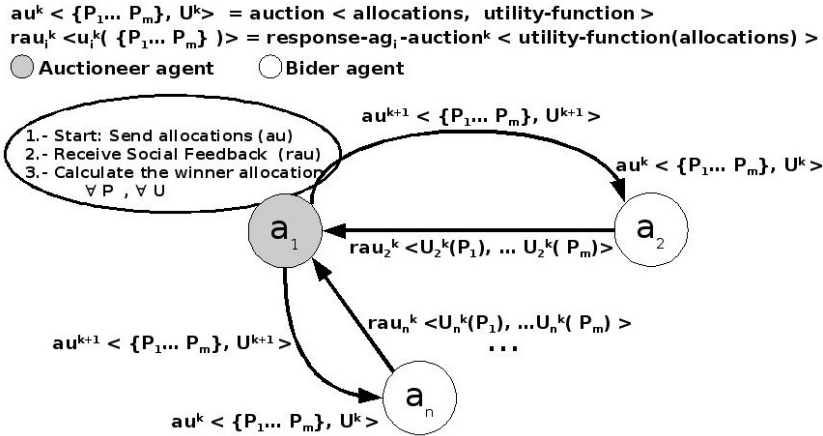


Figura 5.6: Procediment de presa de decisions MADeM.

Cada subhasta, a més, inclou el tipus de funció d'utilitat o punt de vista considerat ($au^k(\{P_1, P_2, \dots, P_m\}, U^k)$). Com que els processos de decisió complexos sovint requereixen la consideració de més d'un punt de vista, l'agent subhastador pot encetar diferents subhastes per a les mateixes distribucions però amb diferents funcions d'utilitat a avaluar (des d' au^1 fins au^q).

El mecanisme de selecció dels agents objectiu d'una subhasta varia depenent de les eines que el subhastador tinga a l'abast. Si l'agent disposa d'un sistema de classificació dels objectes de l'entorn (p. ex. la base ontològica definida en la secció 4.3), cada ranura pot ser considerada com un paràmetre amb un cert tipus. Llavors, quan es treballa amb ranures d'agent, els agents objectiu poden ser extrets fent servir el tipus de ranura que s'estiga subhastant. Per exemple, la ranura *Fer_Cafe*(*Age*) hauria de ser subhastada només als agents de la classe *Cambrer*, ja que són els únics que poden realitzar aquesta tasca en un bar virtual. D'una altra banda, quan s'assignen ranures d'objectes, els agents objectiu podrien ser aquells agents que estan relacionats d'alguna manera amb algun objecte del tipus de la ranura que s'està subhastant. Per exemple, la ranura de tasca *Seure*(*Obj_m*), si *Obj_m* correspon amb la taula on l'agent ha de seure, podria ser subhastada a tots els agents que estan relacionats amb alguna taula de l'entorn. Si no hi ha un sistema de tipus disponible, els agents objectius poden extraure's atenent a criteris de visibilitat o, en una situació extrema, ser el conjunt de tots els agents.

2 Fase d'apostes: Quan s'inicia una subhasta, el subhastador informa tant sobre les

distribucions (solucions possibles) que vol avaluar com sobre la funció d'utilitat que ha de ser considerada. Per tant, els agents objectiu (apostadors) simplement han de computar la funció d'utilitat sol·licitada i enviar els valors obtinguts per cada distribució en forma d'aposta ($rau_i^k = \langle U_i^k(P_1), \dots, U_i^k(P_m) \rangle$).

3 Fase de resolució del guanyador: Aquesta fase consta de dues parts, referides com la determinació del guanyador de cada subhasta i la presa de decisions multimodal. La primera part serveix perquè el subhastador seleccione quina és la distribució guanyadora de cadascuna de les subhastes llançades. Aquesta decisió seguirà el criteri de maximitzar el benestar de la societat artificial (veure [Chevaleyre et al., 2004] per un bon conjunt d'exemples de benestar social). En acabant, la presa de decisions multimodal triarà la distribució guanyadora final (solució adoptada) del conjunt de distribucions guanyadores de les subhastes.

A continuació, analitzem amb detall com MADeM realitza la determinació del guanyador de cada subhasta i la presa de decisions multimodal.

Determinació del guanyador de cada subhasta

El procés de determinació del guanyador d'una subhasta comença una vegada tots els apostadors han enviat la seua aposta ⁴. Com a informació d'entrada, l'agent subhastador disposa dels valors d'utilitat ($U_i(P_j)$) enviats pels agents de la societat ($a_i \in A$) per cada distribució que s'estiga avaluant (P_j). L'equació 5.15 agrupa aquests valors d'utilitat en un conjunt de **vectors d'utilitat**, un per cada possible solució.

$$\overrightarrow{U(P_j)} = \langle U_1(P_j), \dots, U_n(P_j) \rangle \quad \forall j \in [1..m] \quad (5.15)$$

Així mateix, recordem que cada agent disposa d'un vector de pesos que representen la seua actitud personal envers els altres (veure l'equació 5.16). Llavors, l'agent subhastador pot aplicar un diferencial als vectors d'utilitat mitjançant la multiplicació component a component amb el vector d'actituds, com demostra l'equació 5.17.

⁴Hom pot considerar que aquell qui no responga no té cap preferència al respecte, és a dir, la seua utilitat és zero

$$\vec{w} = \langle w_1, \dots, w_n \rangle \quad (5.16)$$

$$\overline{U_w(P_j)} = \overline{U(P_j)} * \vec{w} \quad \forall j \in [1..m] \quad (5.17)$$

Els **pesos d'actitud** s'empren per modelar el comportament social a nivell personal, açò és, entre l'agent subhastador i la resta d'agents de la societat. Per exemple, l'egoisme extrem es pot representar a través d'un pes de valor 1 per les utilitats del mateix agent i 0 per la resta d'agents. De fet, hom pot modelar tot un rang de comportaments entre l'egoisme i l'altruisme mitjançant un vector de pesos com el que apareix en l'equació 5.18. En aquest model, $p = 0$ correspon amb el comportament definit abans, $p = 1$ representa l'altruisme màxim i $p = 0.5$ representa un comportament igualitari o indiferent cap a la resta d'agents. Darrerament, amb el vector de pesos \vec{w} també és possible modelar actituds recíproques. L'equació 5.19 mostra un exemple senzill, en què els pesos es calculen en funció del nombre de favors intercanviats entre els agents.

$$\begin{aligned} \text{Egoisme} - \text{Altruisme} : \quad \vec{w} &= \langle p, \dots, p, 1 - p, p, \dots, p \rangle \\ w_i &= 1 - p, w_{j \neq i} = p, i = \text{Myself} \end{aligned} \quad (5.18)$$

$$\text{Reciprocitat} : \quad w_i = \frac{\text{Favors_from}(i)}{\text{Favors_to}(i)} \quad (5.19)$$

En aquest context de preferències individuals dels agents, la determinació del guanyador d'una subhasta consisteix a trobar aquella distribució P_j que es puga considerar òptima respecte d'alguna mètrica global a la societat. La definició de mètriques globals com agregació de preferències individuals ha estat tractada, a sovint, pel **benestar social** (*social welfare*); concepte estudiat per l'economia del benestar (*Welfare Economics*) i per la teoria de la decisió social (*Social Choice Theory*). En el cas habitual en què les preferències s'expressen mitjançant funcions d'utilitat, el benestar social es defineix a través de **funcions d'utilitat col·lectives** (*Collective Utility Functions* o CUFs). Així doncs, l'equació 5.20 defineix el benestar social en MADeM en funció de les utilitats sospesades de l'equació 5.17. MADeM permet triar entre diferents CUFs a l'hora d'avaluar el benestar social d'una assignació (veure l'equació 5.21). Cadascuna d'aquestes funcions està rela-

cionada amb un tipus de societat [Sandholm and Suri, 2001], respectivament: utilitària, igualitària, elitista i tipus Nash.

$$sw(P) = \text{cuf}(\overrightarrow{U_w(P)}) \quad (5.20)$$

$$\begin{aligned} \text{cuf}_{\text{utilitarian}} &= \Sigma u_w(i) \\ \text{cuf}_{\text{egalitarian}} &= \min\{u_w(i)\} \\ \text{cuf}_{\text{elitist}} &= \max\{u_w(i)\} \\ \text{cuf}_{\text{nash}} &= \Pi u_w(i) \end{aligned} \quad (5.21)$$

La funció de benestar social (sw) defineix un ordre social tal que: una distribució Q és socialment preferible a una distribució P si i solament sí $sw(P) \leq sw(Q)$. D'acord amb açò, la distribució guanyadora d'una subhasta serà aquella que maximitze el benestar de la societat (veure l'equació 5.22).

$$\text{Winner} = P_w \longleftrightarrow sw(P_w) = \max_{j \in [1..m]} sw(P_j) \quad (5.22)$$

Preses de decisions multimodal

La presa de decisions multimodal es l'últim pas que es realitza i el seu resultat és la solució MADeM a un problema donat. Com hem analitzat abans, un agent que executa una presa de decisions de tipus MADeM considera les preferències d'altres agents segons diferents punts de vista (p. ex. l'eficiència, el cansament, etc.). Per tal de fer-ho, l'agent llança diverses subhastes, cadascuna amb una funció d'utilitat diferent (veure el paràmetre U^k de les subhastes en la figura 5.6). Una vegada que totes aquestes subhastes han estat resoltes, l'agent subhastador té la distribució guanyadora segons cada punt de vista, així com el benestar social que s'obté amb ella (veure l'equació 5.23).

$$\begin{aligned}
\text{auction}_1(\{P_1, P_2, \dots, P_m\}, U^1) &\longrightarrow (P_{w1}, sw(P_{w1})) \\
&\dots \\
\text{auction}_k(\{P_1, P_2, \dots, P_m\}, U^k) &\longrightarrow (P_{wk}, sw(P_{wk}))
\end{aligned} \tag{5.23}$$

Aleshores, la selecció de la distribució guanyadora final es realitza fent servir el vector de pesos d'utilitat \vec{w}_u definit en la secció 5.4.1. La presa de decisions multimodal utilitza els valors de \vec{w}_u com els pesos assignats a cada punt de vista o funció d'utilitat. En conseqüència, la distribució guanyadora final (P_f) serà aquella que maximitze el benestar de la societat, després d'haver multiplicat el benestar obtingut amb cada punt de vista pel seu corresponent pes d'utilitat (veure l'equació 5.24).

$$P_f = P_{wi} \iff sw(P_{wi}) = \max_{i \in [1..k]} w_u(i) * sw(P_{wi}) \tag{5.24}$$

En resum, l'ús de subhastes és una aproximació molt flexible per a l'avaluació de les diferents solucions a un problema en un entorn multiagent. Nogensmenys, és clar que l'animació de totes aquestes subhastes produiria un comportament del tot inversemblant (p. ex. humanoides que subhasten tasques contínuament i reben respostes numèriques inintel·ligibles). Per tant, cal separar el procés de presa de decisions de l'animació de la decisió adoptada. D'acord amb açò, MADeM pot ser utilitzat com el mòdul social per a la presa de decisions en l'arquitectura d'agents presentada en la secció 4.4. Tanmateix, MADeM s'executa de manera transparent a l'observador extern. Açò és, les comunicacions establides entre els agents per dur a terme les subhastes no s'animen sinó que es realitzen de forma interna. Llavors, una vegada ha estat seleccionada la decisió guanyadora, l'agent ha de fer-se càrrec d'animar la situació corresponent (p. ex. mitjançant protocols d'animació de converses entre personatges 3D). En aquest context, MADeM és una font d'informació molt útil per a la simulació d'acords, desacords, i comportaments socials purs (p. ex. xats o diàlegs banals). En la secció 6.4.4, mostrem com ha estat resolta l'animació d'acords socials en un entorn 3D on els personatges fan servir MADeM per a la presa de decisions socials.

5.5 CONCLUSIONS

En aquest capítol hem analitzat les dues aportacions principals d'aquesta tesi pel que fa a la integració d'habilitats socials en la presa de decisions dels actors sintètics.

En primer lloc, hem presentat una tècnica per aconseguir comportaments col·laboratius en aquells personatges 3D que empren els planificadors heurístics (model STRIPS) com a mecanisme de selecció dinàmica d'accions. Aquesta tècnica proveeix els actors amb l'habilitat d'adaptar-se i actuar de forma eficient als entorns poblats per diversos caràcters autònoms (p. ex. jocs, *storytelling*, simulació civil, etc.). En aquests contextos, la detecció de dependències i la comunicació directa entre els personatges serveix per construir una representació dels agents externs. Llavors, hem proposat dos mecanismes que utilitzen aquesta informació per incloure criteris socials en el procés de planificació. D'una banda, hem incorporat una fase de pre-planificació que evita obstruir les accions d'altres agents. D'una altra banda, hem desenvolupat un mecanisme de sospesament de la funció heurística que permet els agents cooperar en la consecució d'objectius externs.

Segonament, hem presentat MADeM (*Multi-modal Agent Decision Making*). MADeM és un procés de presa de decisions multimodal i social. Açò és, que permet avaluar diversos punts de vista (p. ex. eficiència, cansament, etc.) així com consultar les preferències d'altres agents al voltant de les diferents solucions a un problema donat. Les preferències individuals són convertides en preferències col·lectives a través del benestar social, el qual defineix el tipus de societat simulada. MADeM és capaç de simular diferents tipus de societats (p. ex. elitistes, igualitàries, utilitàries, etc.), així com actituds socials entre els seus membres (p. ex. egoisme, altruisme, indiferència o reciprocitat).

En el capítol següent analitzarem els efectes socials que provoquen ambdues aportacions en el comportament dels personatges 3D. Per tal de fer-ho, repassarem tres exemples de complexitat creixent: a) el sospesament heurístic en el món de blocs clàssic; b) la col·laboració en el món de blocs 3D multipersonatge; c) la presa de decisions socialment acceptables dins d'un bar virtual.

RESULTATS

En aquest capítol presentem tres exemples que mostren els efectes produïts pels mecanismes de presa de decisions socials proposats en aquesta tesi. En primer lloc, definim un conjunt d'experiments basats en el món de blocs clàssic, amb vista a estudiar l'impacte del sospesament d'objectius en els planificadors de cerca heurística. Segonament, hem creat diversos escenaris 3D inspirats en el món de blocs multipersonatge. En aquests entorns dinàmics, els actors sintètics (basats en planificadors heurístics) mostren la seua sociabilitat a través de la **coordinació** de tasques i de la **cooperació** amb els objectius d'altres personatges. Finalment, hem desenvolupat un entorn dinàmic i amb un grau de complexitat major, consistent en un bar virtual on actors que executen el rol de cambres serveixen les comandes d'altres actors amb el rol de clients. En aquest context, ambdós tipus d'actors fan servir MADeM per expressar la seua **sociabilitat** segons diversos criteris (p. ex. eficiència, cansament, etc.), així com per modelar diferents actituds personals (p. ex. reciprocitat, altruisme, etc.).

6.1 DESCRIPCIÓ DELS EXPERIMENTS PLANTEJATS

Les dificultats a les quals s'ha hagut d'enfrontat l'animació comportamental d'un actor sintètic individual (p. ex. fidelitat visual, percepció, navegació, etc.) han fet que tot just s'estiguen començant a integrar comportaments socials complexos en la presa de decisions dels personatges 3D [Prada and Paiva, 2005, Pelechano et al., 2007]. En aquest context, un problema important és com validar els models socials incorporats en els agents. Tanmateix, la recerca en actors sintètics manca hui dia de mètodes de validació acceptats comunament (*benchmarks*), per poder comparar el comportament social d'actors sintètics.

tics governats per mecanismes diferents. Al contrari, cada sistema ha utilitzat el conjunt d'experiments adients per demostrar els efectes socials que es poden aconseguir amb ell. D'acord amb açò, en aquest capítol presentem tres exemples, de complexitat creixent, que ens permeten demostrar l'efecte de les tècniques socials desenvolupades en aquesta tesi de forma quantitativa, així com comprovar que els mecanismes proposats permeten controlar la integració de diferents habilitats socials en el comportament dels 3DIVA.

Els dos primers exemples estan relacionats amb la tècnica de col·laboració mitjançant l'ús d'un planificador heurístic. Primerament, hem realitzat un conjunt d'**experiments basats en el món de blocs clàssic** (veure el punt 6.2). En aquests problemes monoagent, hem definit l'estat objectiu de l'agent com un conjunt de subobjectius propis i externs. Així doncs, podem analitzar la influència que té el sospesament heurístic dels objectius externs en els entorns multiagent, però simplificant el problema amb un entorn estàtic on no es produeixen canvis exògens. El segon exemple que considerem ha estat l'**animació 3D del món de blocs multipersonatge** (veure l'apartat 6.3). El dinamisme d'aquests escenaris durà els actors a enfrontar-se a situacions conflictives, que hauran de ser resoltes de manera reactiva. En aquests entorns orientats a la consecució de tasques, mesurem la sociabilitat dels personatges en funció de la coordinació de tasques (secció 6.3.1) i de la cooperació amb els objectius de la resta d'agents (secció 6.3.2).

En tercer lloc, hem construït un entorn 3D més complex que simula un **bar universitari virtual** poblat per cambriers i clients (veure l'apartat 6.4). Els nombrosos objectes interactius considerats (p. ex. gots, plats, etc.), han estat representats seguint les definicions d'una ontologia específica del domini la qual també permet la definició de relacions socials entre els actors (seccions 6.4.1 i 6.4.2). Llavors, els personatges utilitzen MADeM per prendre decisions socials multimodals. En la secció 6.4.3, mostrem les funcions d'utilitat que hem dissenyat perquè els actors expressen les seues preferències segons punts de vista diferents (p. ex. l'eficiència, el cansament, les ganes de xarrar, la mandra, etc.). Finalment, en les seccions 6.4.5 i 6.4.6 repassem els resultats socials obtinguts per diversos tipus de personatges (p. ex. coordinats, xerraires, mandrosos, etc.) i per diferents models d'actitud (p. ex. egoisme, altruisme, reciprocitat, etc.).

Com a detall d'implementació, els resultats mostrats al llarg d'aquest capítol han estat obtinguts fent servir un Pentium IV a 2.66 GHz i amb 1 Gb de RAM. Per tant, podem afirmar que els models socials proposats no necessiten una gran potència de còmput i que poden ser aplicats als entorns virtuals 3D que animen els ordinadors personals hui dia.

6.2 SOSPESAMENT HEURÍSTIC EN EL MÓN DE BLOCS

En aquest apartat mostrem els resultats d'un conjunt d'experiments que hem realitzat per estudiar l'impacte del sospesament d'objectius en els planificadors de cerca heurística. Els experiments estan basats en el món de blocs clàssic, el qual consisteix a què un agent moga (apile/desapile) un conjunt de blocs per aconseguir una distribució objectiu de la manera més eficient possible. Tanmateix, per poder avaluar el grau de cooperació en entorns multiagent, hem completat l'estat objectiu de l'agent amb un nombre finit de subobjectius externs els quals representen els objectius d'altres agents. Llavors, hem introduït el mecanisme de sospesament heurístic explicat en la secció 5.3.3 per modular la influència d'aquests subobjectius externs en la presa de decisions de l'agent.

El conjunt d'experiments emprat consta d'una bateria de 10 problemes per a cadascuna de les següents grandàries: 8, 12, 16, 20 i 24 blocs. La disposició inicial i final (objectiu) dels blocs ha estat generada de forma aleatòria. Hem dividit l'objectiu en un conjunt de subobjectius independents. D'aquests subobjectius, considerem una part com objectius propis de l'agent i una altra part com objectius externs. La proporció utilitzada ha estat 1:1, açò és, una meitat d'objectius externs i l'altra d'objectius propis. Aleshores, hem executat tots aquests problemes fent servir 11 pesos diferents pels objectius externs ($w \in [0, 0.1, 0.2, \dots, 1.0]$). Amb vista a calcular una mitjana estable dels resultats, hem resolt cada problema 10 vegades per pes. Els problemes són resolts per un únic agent que utilitza *miniMin-HSP* [Lozano, 2005] per calcular la seqüència d'accions que li permeta aconseguir els objectius propis. Encara que en aquests experiments no es produeixen interferències entre els agents, la quantitat d'objectius externs aconseguits pot ser un bon estimador del **grau de cooperació**.

La figura 6.1a mostra la mitjana d'objectius externs (mesurats en nombre de fets o proposicions) aconseguits per un agent mentre compleix amb els seus objectius propis. La influència dels pesos aplicats sobre l'heurística dels objectius externs modificarà la seua importància en el procés de presa de decisions i, per tant, la quantitat d'objectius externs assolits. Per exemple, quan $w_{ext} = 0$, els objectius externs no són considerats per l'heurística i el nombre de fets externs assolits és molt baix. Hom podria pensar que aquest valor hauria de ser 0. Emperò, és fàcilment comprensible que alguns fets externs es puguin aconseguir de manera casual (p. ex. com efectes col·laterals de les accions realitzades per complir amb els objectius propis). A mesura que augmenten els pesos,

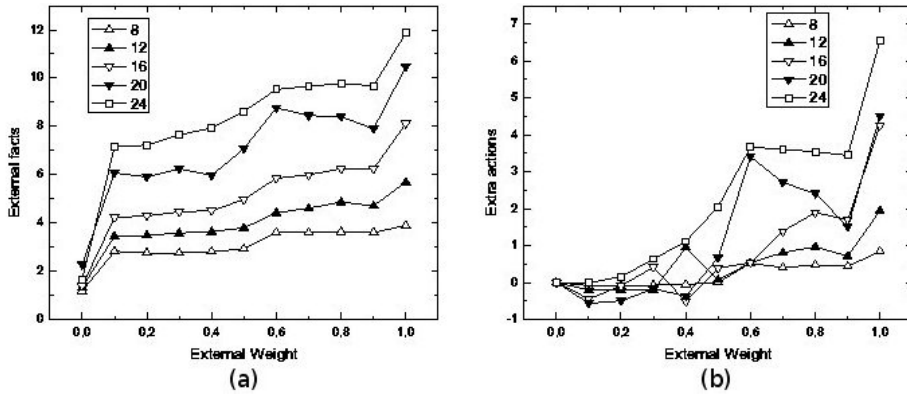


Figura 6.1: Cooperació en funció del pes heurístic: a) mitjana d'objectius externs aconseguits; i b) mitjana d'accions extres realitzades.

també ho fa la proporció d'objectius externs aconseguits. Per exemple, quan $w_{ext} \in [0.1..0.5]$, el nombre de fets externs assolits es mou entre un 30% i un 35% dels objectius externs considerats. Aquesta creixença es produeix en tots els problemes, que apleguen al 50% dels objectius totals quan la importància entre els objectius externs i els propis s'igualen ($w_{ext} = 1$); resultat que correspon amb la ràtio entre els objectius propis i externs.

La figura 6.1b mostra la mitjana d'accions extres executades per causa de la cooperació amb els objectius d'agents externs. El propòsit d'aquesta gràfica és avaluar l'esforç addicional que realitza un agent depenent dels pesos emprats pel sospesament d'objectius externs. Com es pot apreciar, quan els pesos són baixos ($w_{ext} \in [0.1..0.3]$), l'agent no realitza més accions que aquelles que necessita quan no hi ha cooperació ($w_{ext} = 0$). Recordem, però, que amb aquest rang de pesos ja s'aconseguien algunes proposicions associades a objectius externs (com vérem en la figura 6.1a). Aquest fet es produeix gràcies a l'oportunitat de certes accions. Per exemple, quan un agent ha de llevar un bloc de damunt d'un altre perquè vol agafar el bloc de baix, pot decidir moure'l a qualsevol posició disponible o bé, deixar-lo en una posició que servisca per aconseguir algun objectiu extern. Si açò no és possible, almenys, l'agent pot deixar el bloc en algun lloc que no destorbe altres agents, és a dir, que no allunye la consecució d'altres objectius externs. La influència del sospesament heurístic en l'activitat de l'agent es manifesta d'una manera més clara quan $w_{ext} > 0.5$. Per exemple, els problemes de 8 blocs normalment es resolen amb plans formats per una mitjana de 6 accions. Tanmateix, la consideració d'objectius

externs amb la mateixa prioritat que els propis ($w_{ext} = 1$) fa que els agents executen una acció extra per terme mitjà. Aquest comportament es reproduïx també en la resta d'experiments. Per exemple, en els problemes de 24 blocs, la mitjana d'accions extres arriba quasi a 7 accions. Aquests resultats mostren com els agents, en el seu intent d'actuar de forma cooperativa, poden incórrer en un esforç innecessari, ja que aquestes accions extres es realitzen únicament per aconseguir els objectius associats a altres agents (veure també la figura 6.1a).

El sospesament heurístic no sols facilita la cooperació entre els agents sinó que, en certes circumstàncies, pot tindre un impacte positiu en l'eficiència del procés de presa de decisions. La figura 6.2 mostra la mitjana de temps de CPU que necessita el planificador per obtenir un pla complet que duga l'agent a aconseguir el seus objectius propis. De manera general, es pot constatar que el temps consumit pel planificador és superior per a $w_{ext} = 0$ que quan $w_{ext} > 0$. La raó d'aquest comportament radica en què, quan es consideren només els objectius propis per guiar la cerca, poden haver més camins de la mateixa llargària que arriben a l'estat objectiu. Aleshores, el planificador expandirà més nodes a l'hora de buscar el millor camí cap a l'objectiu. Aquest fet encara és més notori quan augmenta el nombre d'objectes de l'entorn, ja que hi ha molts objectes que no tenen cap influència en els objectius propis (veure l'alt temps d'execució dels problemes amb 24 blocs i $w_{ext} = 0$). Nogensmenys, si es consideren els objectius externs en el càlcul de l'heurística, aquells camins que aconseguisquen més objectius externs seran afavorits i, en conseqüència, el nombre de nodes explorats serà menor. Això no obstant, el fet d'augmentar molt el pes dels objectius externs implica reduir la prioritat de resolució dels objectius propis respecte dels externs. En un cas extrem, l'agent podria començar a perdre temps tractant d'aconseguir altres objectius externs en comptes dels propis, que són els que fan que finalitze el procés de planificació. La figura 6.2 reflecteix aquest fet en forma d'una lleugera tendència a l'alça en els temps de CPU quan $w_{ext} = 1$.

En resum, l'estudi anterior ha demostrat que la inclusió d'objectius externs en el procés de planificació permet aconseguir **comportaments cooperatius** sense augmentar la complexitat de les decisions. En aquest context, el pes heurístic utilitzat pels objectius externs determinarà el grau de cooperació assolit. D'una banda, a l'hora de desenvolupar agents que cooperen amb d'altres sense incrementar el seu esforç (el nombre d'accions realitzades), un bon valor per els pesos w_{ext} es trobarà en el rang [0.1..0.3]. D'una altra banda, si hom vol implementar un agent que coopere en un grau major, encara que això supose un esforç extra, pot usar valors de w_{ext} en el rang [0.6..1.0].

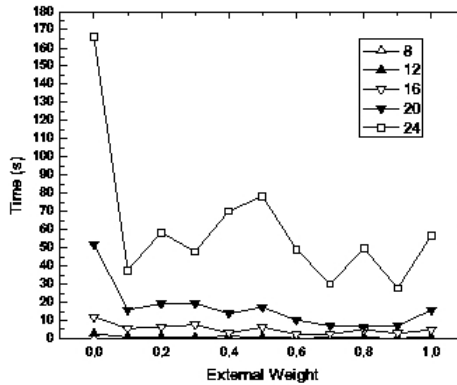


Figura 6.2: Comparativa dels temps de resolució en funció del pes heurístic.

6.3 MÓN DE BLOCS 3D MULTIPERSONATGE

Per avaluar la tècnica de col·laboració proposada pels actors sintètics basats en planificadors heurístics en un entorn dinàmic, hem creat diversos escenaris 3D inspirats en el món de blocs. Aquests entorns estan poblats per un conjunt d'actors que manipulen els objectes de l'entorn, fonamentalment, per situar-los en una posició final desitjada. L'operativa de cadascun dels actors segueix el bucle típic que intercala percepció, planificació i actuació. En aquest context, els personatges 3D utilitzen el planificador miniMin-HSP amb les extensions socials presentades en la secció 5.3 per decidir la següent tasca a realitzar (p. ex. agafar o deixar un objecte damunt un altre). Pel que fa a la visualització gràfica, en aquest exemple hem fet servir el motor del joc *Unreal Tournament* [Epic games, 2008].

La simulació 3D del món de blocs multipersonatge s'enfronta sovint amb situacions conflictives, a causa de la forta competència pels recursos compartits. Per exemple, encara que dos actors decidisquen agafar el mateix objecte, només un d'ells ho pot aconseguir finalment. Per detectar les interferències que es produeixen amb altres agents, els actors comproven contínuament les precondicions de les tasques (operadors STRIPS) que executen. Llavors, la detecció de dependències provoca el llançament dels protocols de pregunta de la tasca actual i de formació d'equips, descrits en la secció 5.3.1. El propòsit d'aquestes converses és l'intercanvi d'informació sobre la tasca i els objectius actuals.

Agent	Plans interromputs		Tasques executades	
	Simple	Coordinat	Simple	Coordinat
A	5	0	14	8
B	5	4	12	4

Taula 6.1: Resultats de coordinació per a dos agents en el món de blocs 3D.

Aquesta informació representa el **context social**, el qual és utilitzat posteriorment pel planificador.

En aquest tipus d'entorns orientats a la consecució de tasques, la sociabilitat dels personatges s'associa amb l'eficiència de la seua actuació. Per tant, el grau de comportament social assolit es pot mesurar a través de la **coordinació de tasques** i la **cooperació amb els objectius de la resta d'agents**; aspectes que estudiem en les seccions 6.3.1 i 6.3.2, respectivament.

6.3.1 Coordinació de tasques

La manca de recursos i els conflictes que sorgeixen per la seua manipulació compartida poden reduir la versemblança dels personatges 3D. Per il·lustrar aquest fet, comencem amb un senzill entorn on l'objectiu siga apilar quatre caixes segons un cert ordre establert (veure la figura 6.3). Mentre que un únic agent necessita 12 tasques (agafar/soltar) per completar la pila, seria desitjable que dos agents es repartiren les tasques per muntar la pila en un temps inferior. Tanmateix, quan els agents no estan coordinats, el nombre de tasques executades per cada agent supera el cas monoagent i arriba a un total de 26 tasques (veure la taula 6.1). La causa la trobem en què, quan els agents planifiquen de manera independent, l'actuació d'un agent interrompra els plans d'altres agents (fins a 10 vegades en el senzill exemple plantejat). Aquesta sensació de **descoordinació** minvarà la versemblança del comportament dels actors.

En la tècnica col·laborativa presentada en aquesta tesi, les interferències es resolen mitjançant una conversa on els agents es pregunten sobre les seues tasques actuals. Aquesta informació, que entra en forma de creença en la memòria de l'agent, s'emprarà en el procés de planificació per començar la cerca des d'un possible estat futur en el qual no es



Figura 6.3: Problema d'apilar quatre caixes entre dos personatges.

destorbe l'altre agent. La taula 6.1 mostra també els resultats obtinguts quan els agents comuniquen les seues accions i coordinen les seues tasques futures d'acord amb la informació intercanviada. En aquest cas, el nombre total d'accions completades entre els dos agents és 12, com era d'esperar. L'agent *A* no ha vist interrompuda cap de les seues tasques, mentre que l'agent *B* ha utilitzat les seues interrupcions per coordinar-se amb l'agent *A*. La figura 6.4 mostra la seqüència de passos realitzada pels agents per apilar les caixes. Les instantànies *b*, *e* i *g* mostren les converses establides entre els agents per comunicar-se la seua tasca actual.

Per estudiar la **coordinació de tasques** en un exemple més complex, hem animat un món de blocs 3D amb 21 caixes i poblat per 4 actors que comparteixen un objectiu comú. Segons estableix Jacques Ferber [Ferber, 1999], la coordinació entre un conjunt d'agents pot ser mesurada a través de tres indicadors: la **supervivència**, la **resolució de conflictes** i l'**augment de l'eficiència**. L'indicador de supervivència no té sentit en aquest problema, ja que tots els agents aconseguiran el seu objectiu. El nivell de resolució de conflictes pot ser estimat mitjançant el nombre de plans interromputs. L'increment d'eficiència pot ser estimat atenent a la quantitat de tasques executades. Finalment, el nombre de crides al planificador és un estimador de coordinació que barreja ambdues mètriques, ja que correspon amb la suma dels plans interromputs més les tasques completades amb èxit.

La taula 6.2 mostra els valors que prenen els tres estimadors anteriors per cadascun

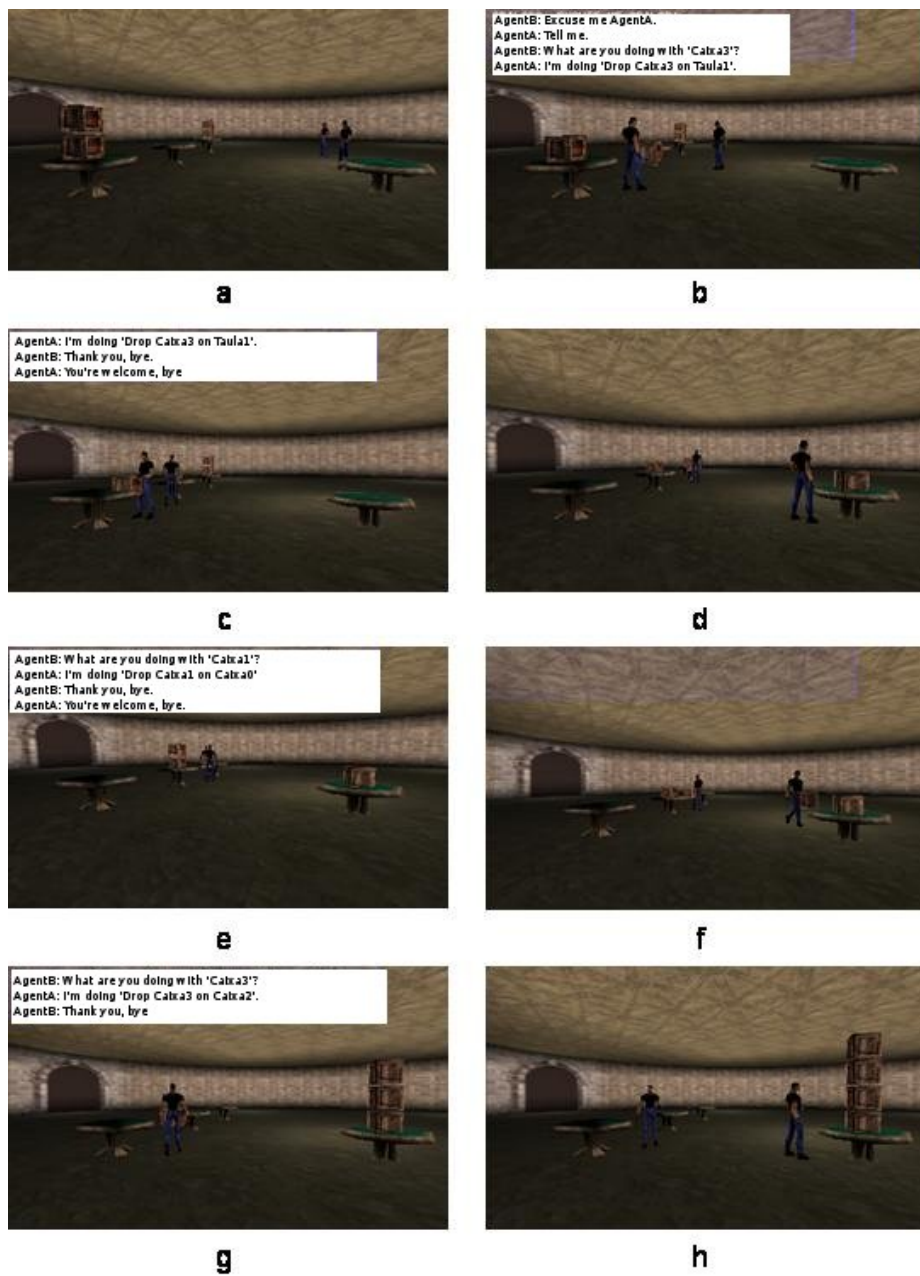


Figura 6.4: Animació 3D de la construcció de la pila de caixes entre dos actors.

Agent	Plans interromputs		Tasques executades		Crides al planificador	
	Simple	Coordinat	Simple	Coordinat	Simple	Coordinat
A	4	5	10	8	14	13
B	3	2	6	3	9	5
C	4	2	11	7	15	9
D	7	2	8	4	15	6

Taula 6.2: Resultats de coordinació per a quatre agents en el món de blocs 3D.

dels actors sintètics. Com es pot apreciar a la columna *Tasques executades*, els actors simples o no coordinats necessiten executar 35 tasques per arribar a l'objectiu. Tanmateix, els actors coordinats redueixen aquesta quantitat a 22 tasques. A més a més, aquestes tasques són no conflictives, ja que no fan servir cap recurs que estiga essent utilitzat per un altre agent o que vaja a ser emprat en un futur proper. Per tant, l'objectiu comú pot ser assolit d'una manera més ràpida per un conjunt d'actors coordinats, ja que poden executar les seues tasques de forma paral·lela. Pel que fa a la resolució de conflictes, el nombre de plans interromputs també es redueix de manera general en els agents coordinats, ja que les seues tasques interfereixen amb menys freqüència. Tot plegat, els personatges coordinats necessitaran invocar menys vegades el planificador per a la selecció de la següent acció, cosa que augmentarà la fluïdesa del seu comportament (veure la columna *Crides al planificador*).

En resum, la inclusió de les tasques dels agents externs en el procés de planificació redueix els conflictes i produeix un conjunt coordinat de tasques, l'execució de les quals pot ser simultaniejada per diversos agents. A continuació, analitzem l'impacte addicional aconseguit amb la inclusió dels objectius externs diferents dels propis en el procés de cerca heurística.

6.3.2 Cooperació amb objectius externs

La cooperació amb els objectius d'altres personatges pot servir per **reduir les obstruccions i millorar l'eficàcia** dels actors sintètics. La traça de la figura 6.5 mostra un exemple senzill d'obstrucció entre dos actors que volen moure les caixes que es troben dessota les piles, respectivament, sobre les taules anomenades `table1` i `table2`. Per tal

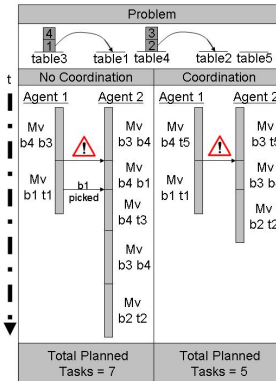


Figura 6.5: Traça simple de cooperació entre dos agents en el món de blocs 3D.

de fer-ho disposen, a més a més, d'una taula auxiliar (table5). Emperò, aquesta taula és massa lluny perquè l'actor la considere com la primera opció a l'hora de desapilar la caixa que es troba al cim de la pila, ja que hi ha moviments possibles que requereixen menys esforç. Llavors, quan els personatges no són conscients dels objectius externs, la seua actuació pot perjudicar els interessos dels actors que els envolten (veure la part esquerra de la figura 6.5). Pel contrari, la part dreta de la figura 6.5 mostra l'actuació cooperativa dels actors quan coneixen els objectius de l'altre. En aquesta situació, els personatges decideixen usar la taula auxiliar per evitar obstruir l'altre company, aleshores, el nombre total de tasques planificades es veu reduït.

Els actors socials basats en planificadors heurístics, presentats en aquesta tesi, publiquen els seus objectius mitjançant el protocol de formació d'equips explicat en la secció 5.3.1. Llavors, segons la relació que hi haja entre els objectius, els agents decideixen si volen o no adquirir els objectius externs i cooperar en la seua consecució. Aquesta aproximació, més flexible que la definició d'una base de coneixement global per part del dissenyador, s'adequa molt al dinamisme dels entorns virtuals habitats (p. ex. permet la creació dinàmica d'equips de treball).

Per estudiar l'**adquisició d'objectius externs** hem desenvolupat un escenari 3D on tres personatges han d'organitzar els objectes d'un pis compost per 4 estances (veure la figura 6.6a). Els objectius dels personatges són els següents: *Agent_1* vol dur tots els llibres a l'habitació número 4; *Agent_2* desitja ajuntar totes les plantes a l'habitació 3;

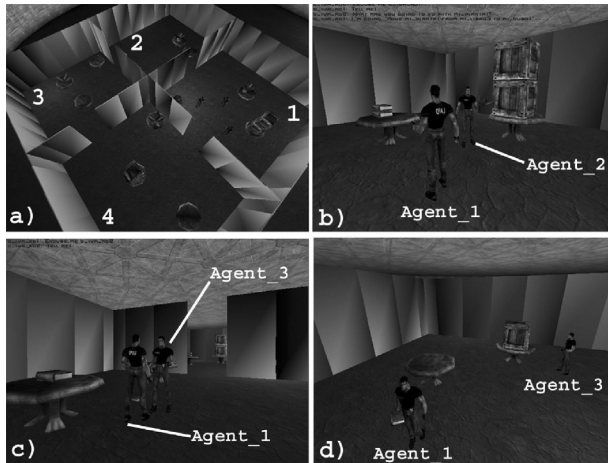


Figura 6.6: Cooperació entre tres actors sintètics en un entorn 3D.

per últim, *Agent_3* té un objectiu més gran format pels dos objectius previs i l'objectiu d'apilar totes les caixes a l'habitació 4.

En començar, els actors no són conscients dels objectius de la resta. Tanmateix, mentre avança la simulació i la seua actuació es veu interrompuda, els personatges aniran comunicant-se'ls. Per exemple, en la instantània 6.6b, *Agent_1* tenia la intenció d'aparcar una planta que estava damunt dels llibres, tanmateix, *Agent_2* l'ha agafada primer. En conseqüència, els actors inicien els protocols que intercanvien tant la seua tasca actual com els seus objectius. En aquesta situació, però, *Agent_1* i *Agent_2* rebutgen la cooperació ja que els seus objectius no estan relacionats i consideren que només la coordinació amb la tasca actual en serà prou. Nogensmenys, quan els objectius estan totalment relacionats, els personatges formen un equip de treball cooperatiu, ja que els personatges no volen estar contínuament destorbant-se. Per exemple, en la fotografia 6.6c, *Agent_1* i *Agent_3* competeixen per un llibre. La figura 6.7 mostra parcialment els objectius intercanviats i com *Agent_1* adquireix els dos subobjectius associats a les plantes com objectius externs. Mentrestant els actors formen part d'un equip, la coordinació de tasques s'aplica contínuament. Per exemple, quan *Agent_1* mou el llibre a l'habitació 4 en la fotografia 6.6d, *Agent_3* canviarà a un altre subobjectiu per evitar interferències (p. ex. apilar caixes en comptes d'arreglar els llibres).

Per mesurar el benefici de la **cooperació** en funció del pes heurístic associat als ob-

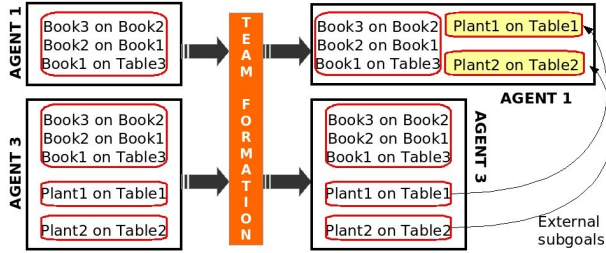


Figura 6.7: Exemple d'adquisició d'objectius externs.

jectius externs, hem considerat tres estimadors en l'exemple anterior: a) el nombre de tasques executades; b) el nombre d'interrupcions o fallades; i c) el nombre d'interaccions (I) que cada agent realitza sobre el mateix objecte. Els dos primers estan relacionats, com ja raonarem en la secció anterior, amb l'eficiència i amb la resolució de conflictes. L'últim estimador s'empra com una aproximació de la quantitat de tasques repetides per un agent, les quals seran innecessàries la majoria de vegades.

La taula 6.3 mostra els valors obtinguts per a tres pesos diferents, que hom pot associar a diversos graus de cooperació. La primera fila de la taula ($w_{ext} = 0$), correspon amb la situació en què els actors no paren cap atenció als objectius externs, és a dir, no cooperen. Sense cooperació, els personatges necessiten executar nombroses tasques per organitzar els objectes del pis. La causa d'açò la trobem en l'alta quantitat d'interrupcions i d'accions repetides. Per terme mitjà, els actors interaccionen amb els objectes al voltant d'1.5 vegades, cosa que ens duu a pensar que la meitat de les tasques les han de repetir dues vegades (veure les columnes $I(Avg)$ i $I(Max)$). La segona fila de la taula ($w_{ext} = 1$), correspon amb la situació en què els objectius externs reben la mateixa importància que els propis. Aquesta cooperació màxima redueix les fallades, la quantitat d'interaccions repetides i, per tant, el nombre de tasques executades pels personatges. Com discutírem a la secció 6.2, l'ús de pesos elevats pot provocar que els personatges comencen a treballar per aconseguir els objectius dels altres i deixen de banda els seus objectius. En conseqüència, hom pot ajustar el pes heurístic dels objectius externs per prioritzar la consecució dels objectius propis, cosa que donarà joc en la generació d'altres rols amb diferents nivells d'eficiència. Per exemple, la taula 6.3 mostra que, si rebaixem w_{ext} a 0.2, els actors redueixen lleugerament el nombre de tasques executades i aconsegueixen el seu objectiu més ràpidament.

w_{ext}	Agent	Tasques	Fallades	I(Avg)	I(Max)
0	A	10	3	1.4	2
	B	15	7	1.6	2
	C	12	3	1.5	2
1	A	7	1	1	1
	B	5	1	1.3	2
	C	8	4	1	1
0.2	A	4	0	1	1
	B	5	1	1	1
	C	6	0	1.2	2

Taula 6.3: Diferents graus de cooperació amb objectius externs.

En resum, la inclusió dels objectius externs en el procés de planificació permet que un actor sintètic coopere amb la resta de personatges que l'envolten. En aquest context, el sospesat de l'heurística per els objectius externs permet ajustar el **grau de cooperació**, cosa que afectarà l'eficiència, la capacitat de resolució de conflictes i, doncs, la qualitat dels rols interpretats pels actors sintètics.

6.3.3 Conclusions a la col·laboració amb planificadors heurístics

La tècnica de col·laboració amb planificadors heurístics, presentada en aquesta tesi, ajunta la presa de decisions racional amb un model social de **treball col·laboratiu**, de manera que permet desenvolupar actors sintètics que mostren ambdues capacitats: racionalitat i sociabilitat.

El comportament social en entorns multipersonatge apunta, normalment, a la capacitat d'un actor d'adaptar-se a l'actuació dels actors que l'envolten. D'acord amb açò, el mecanisme de **detecció de dependències** inclòs en els 3DIVA permet els actors reaccionar a les interferències externes. Açò és, no és el dissenyador qui decideix quan interaccionen els actors ni tampoc qui estableix els detalls concrets de la col·laboració. Pel contrari, el comportament col·laboratiu sorgeix de manera dinàmica com el resultat de les interferències entre els actors; cosa que proporcionarà una flexibilitat superior als

mecanismes d'animació comportamental comuns per a personatges 3D (p. ex. guions, màquines d'estats, HTN, etc.).

A l'hora de col·laborar, els actors utilitzen dos mecanismes: la **coordinació de tasques** i la **cooperació amb els objectius externs**. D'una banda, la inclusió de les tasques dels agents externs, en una fase de pre-planificació redueix els conflictes i produeix un conjunt coordinat de tasques en sistemes que intercalen planificació i acció; en els quals la construcció d'un pla conjunt coordinat, en una fase de post-planificació, no és viable per raó de l'alta variabilitat de l'entorn. D'una altra banda, la incorporació dels objectius externs en el procés de cerca genera decisions cooperatives. En aquest context, el pes heurístic que els personatges donen als objectius externs serveix per ajustar el seu grau de cooperació. La capacitat de graduar la quantitat de cooperació és una característica fonamental per animar comportaments versemblants en actors sintètics situats en entorns virtuals habitats. Llavors, el mecanisme de sospesament heurístic permetrà la simulació de diferents models de personalitat d'una manera senzilla. Per exemple, un personatge generós utilitzarà un pes heurístic alt per donar importància a la consecució dels objectius externs.

La col·laboració mitjançant els planificadors heurístics és una bona tècnica per seleccionar accions en contextos socials orientats a l'execució de tasques, tanmateix, la cerca heurística té uns handicaps que convé discutir. En primer lloc tenim el seu alt cost computacional, que creix de manera exponencial en funció del nombre d'objectes d'interacció de l'escena. Llavors, si es vol mantenir la presa de decisions dins d'un temps raonable per a la versemblança dels personatges 3D, cal realitzar una percepció intel·ligent que excloua els objectes innecessaris de l'estat de cerca. En segon termini es troba el criteri de decisió, que sempre està relacionat amb l'eficiència; és a dir, els millors plans són aquells que són més curts. A nivell social, aquest criteri es manifesta amb la coordinació i la cooperació amb la resta d'agents. Nogensmenys, com que els actors sintètics no són robots ni han de comportar-se com a tals, hom pot trobar criteris que siguin socials però no eficients. Per exemple, en un grup d'amics, els actors podrien preferir competir per un recurs per poder estar junts i xarrar, en comptes de coordinar-se i realitzar tasques separades. D'acord amb açò, en l'exemple de la secció següent tractem ambdós aspectes: la representació semàntica de mons amb molts objectes d'interacció i la presa de decisions social multimodal mitjançant MADeM.

6.4 BAR UNIVERSITARI VIRTUAL

En aquesta secció presentem un entorn 3D complex que simula un bar universitari virtual, on un grup de cambrers serveixen les comandes d'un conjunt de clients. El marc de simulació per animar aquest escenari ha estat implementant fent servir *Jason* [Bordini and Hübner, 2007], un sistema multiagent de propòsit general que permet la definició d'agents de tipus BDI. Per a la visualització 3D, hem emprat la llibreria gràfica *OpenSceneGraph* [OSG, 2008].

El bar virtual conté un gran nombre d'objectes interactius (p. ex. gots, plats, etc.), els quals han estat representats com instàncies de classes definides en una **ontologia** específica del problema (veure la secció 6.4.1). Per exemple, alguns objectes típics en un bar han estat modelats com dispensadors que tenen un temps associat per proveir els seus productes (p. ex. una cafetera triga 2 minuts a fer un cafè). En aquest context, els personatges han de coordinar-se per evitar els conflictes per l'ús de recursos compartits; que no puguin ser utilitzats per dos agents alhora. Així mateix, els actors poden estar relacionats socialment mitjançant els conceptes definits en l'ontologia (veure la secció 6.4.2). Per exemple, els cambrers estableixen relacions d'amistat amb els seus companys de feina. Aquesta xarxa social haurà de ser tinguda en compte a l'hora de prendre decisions socials (p. ex. realitzar favors, promoure trobades, etc.).

Tot plegat, considerem l'exemple del bar virtual com un entorn escaient per comprovar els efectes de **MADeM**, com el procés per a la presa de decisions social dels actors sintètics (veure el punt 5.4). En la secció 6.4.3, mostrem les funcions d'utilitat que hem dissenyat perquè els actors expressen les seues preferències segons punts de vista diferents (p. ex. l'eficiència, el cansament, les ganes de xarrar, la mandra, etc.). Aleshores, en funció de la decisió guanyadora, els agents animaran l'acord social corresponent (veure la secció 6.4.4). En la secció 6.4.5, repassem els resultats socials obtinguts per diversos tipus de personatges (p. ex. coordinats, xerraires, mandrosos, etc.) i per diferents models d'actitud (p. ex. egoisme, altruisme, reciprocitat, etc.). En la secció 6.4.6, analitzem amb més detall l'efecte de la coordinació i la sociabilitat en les tasques realitzades pels actors sintètics. Darrerament, enunciem les conclusions de la presa de decisions social multimodal duta a terme per MADeM.

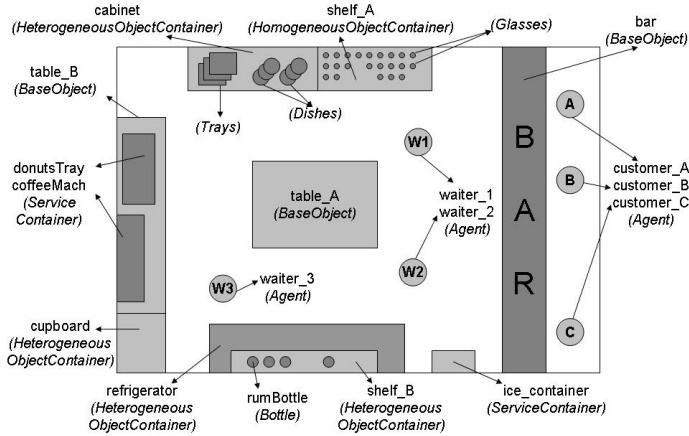


Figura 6.8: Definició semàntica de l'entorn del bar virtual.

6.4.1 Definició de l'entorn

A l'hora de construir el bar virtual hem fet servir l'**ontologia base** que fou presentada en la secció 4.3.1 (per a una definició completa en OWL veure l'annex A.1). Aquesta base semàntica ens permet classificar els objectes interactius del món, així com les seues interrelacions. Per exemple, la figura 6.8 mostra una possible configuració d'un bar virtual, la instanciació del qual es troba a l'annex A.3. En aquest escenari, la classe *ServiceContainer* ha estat usada per modelar tant una cafetera, com una safata amb pastissos o una glaçonera. Així mateix, diversos objectes han estat representats com instàncies de la classe *HeterogeneousObjectContainer*, com ara *shelf-B*, un estant que conté ampolles de begudes alcohòliques. A més a més, hem implementat una **ontologia específica del domini** per definir els objectes típics que hom pot trobar en un bar, com ara ampolles (classe *Bottle*), plats (classe *Dish*), gots (classe *Glass*), etc. (veure l'annex A.2).

La definició semàntica de l'entorn pot servir per millorar la **percepció** d'escenes complexes, açò és, amb molts objectes d'interacció. Per exemple, alguns objectes contenidors (p. ex. *refrigerator*, *cabinet* o *cupboard*) han estat configurats amb el filtre de percepció *CLASSES-ONLY*, de manera que, mentre estiguen tancats, només publiquen les classes dels objectes que contenen. Un altre exemple de reducció sensorial el trobem a l'objecte *shelf-A*, un estant amb 20 gots nets, que hem modelat com una instància de la classe *HomogeneousObjectContainer*. La representació completa d'aquest objecte inundaria fàcilment qualsevol estat mental (a causa de la gran quantitat d'informació associada als

objectes). A més a més seria innecessària, ja que tots els gots són similars. Per tant, la capa semàntica resumeix la informació percebuda en temps de simulació. De manera que, malgrat que informa sobre la quantitat total de gots continguts, només proporciona una descripció completa d'un conjunt reduït de gots. L'escenari mostrat en la figura 6.8 conté 68 objectes interactius. Sense cap tipus de filtratge, la quantitat d'informació necessària per descriure en lògica proposicional les propietats i les relacions de tots aquests objectes seria 612 literals. Tanmateix, el filtratge basat en l'ontologia realitzat per la capa semàntica redueix aquesta quantitat a 252 literals, referits a 28 objectes interactius. Per tant, produeix una reducció del 60% en el flux d'informació que descriu l'escena. Així doncs, podem garantir estats mentals d'una grandària raonable per els 3DIVA que la perceben.

6.4.2 Definició dels actors sintètics

A l'igual dels objectes que formen l'escenari, els actors sintètics s'interrelacionen mitjançant els conceptes (classes, propietats i relacions) definits en l'ontologia. Per exemple, la figura 6.9 mostra alguns exemples de **relacions socials** definides entre els personatges que habiten el bar universitari virtual. Tots els cambrers es relacionen a través d'una *groupSocialRelation* amb el grup *Waiters*, una classe que representa el seu rol. Així mateix, els cambrers poden relacionar-se individualment amb altres cambrers que siguen amics de treball (*workFriend*). Aquesta és una relació bidireccional però no transitiva. Per exemple, en la figura 6.9, Albert és amic de Dough i John, però aquests dos no en són. Tocant als clients, hem especificat tres grups o classes socials: professors, estudiants no graduats i graduats. La xarxa social definida per aquestes relacions és emprada per promoure trobades socials entre els personatges de l'entorn.

Els controladors dels agents han estat definits mitjançant plans implementats en la versió estesa del llenguatge *AgentSpeak(L)* [Rao, 1996], que proporciona *Jason* per a la definició dels agents. D'acord amb aquests plans, l'**actuació dels cambrers** està governada per la màquina d'estats finits de la figura 6.10a. Els cambrers serveixen les comandes, fonamentalment, en dos passos: a) usen el dispensador corresponent per obtenir el producte demanat (p. ex. la planxa per fer un sandvitx); i b) entreguen el producte al client. Emperò, l'execució d'aquestes tasques és subhastada fent servir MADeM, amb vista a determinar bones assignacions socials. La taula 6.4 resumeix les tasques executades pels actors i les ranures considerades per a cadascuna d'elles. Segons el que hem dit abans, els encarregats de les tasques *Use* i *Give* són els cambrers, que tracten d'assignar-les al

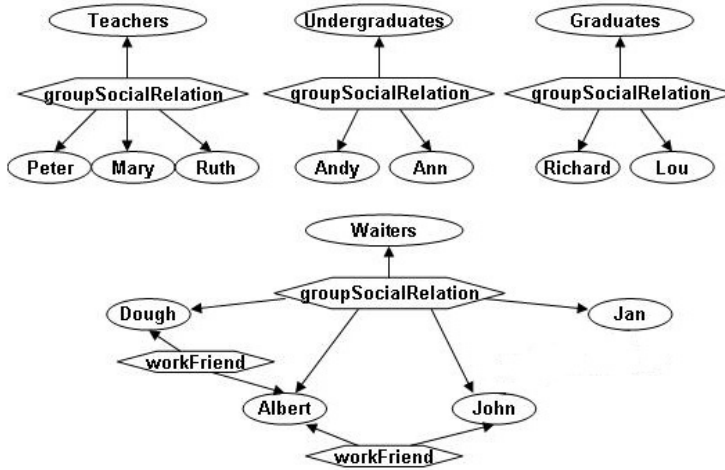


Figura 6.9: Relacions socials entre els personatges que habiten el bar virtual.

Tasks/Slots	Ag_e	Obj_m	Ag_d
Use	Waiter	ProductDispenser	-
Give	Waiter	Product	Customer
SitAt	Customer	SeatingPlace	-

Taula 6.4: Tasques subhastades pels agents en el bar virtual.

millor candidat. Per tant, en ambdós casos, la ranura que se subhastarà serà l'executor de la tasca (Ag_e). El conjunt d'assignacions entre les quals decidir està representat per l'equació 6.1. És a dir, els cambrers avaluen la possibilitat de realitzar cada tasca contra la possibilitat de passar-la a un altre actor, de manera que puguin encarregar-se de la següent tasca pendent. En la secció 6.4.3, mostrem les funcions d'utilitat utilitzades pels cambrers per expressar les seues preferències sobre cadascuna d'aquestes possibilitats.

$$\begin{cases} P_0 = \{t(Ag_e) \leftarrow Myself\} \\ P_i = \{t(Ag_e) \leftarrow a_i, t_{next}(Ag_e) \leftarrow Myself\} \quad \forall a_i \in A \end{cases} \quad (6.1)$$

En segon lloc, la figura 6.10b mostra el **controlador dels clients** del bar virtual. Els clients fan les seues comandes a la barra i consumeixen els productes una vegada els cambrers els han servits. En aquesta ocasió, estem interessats a animar comportaments socials

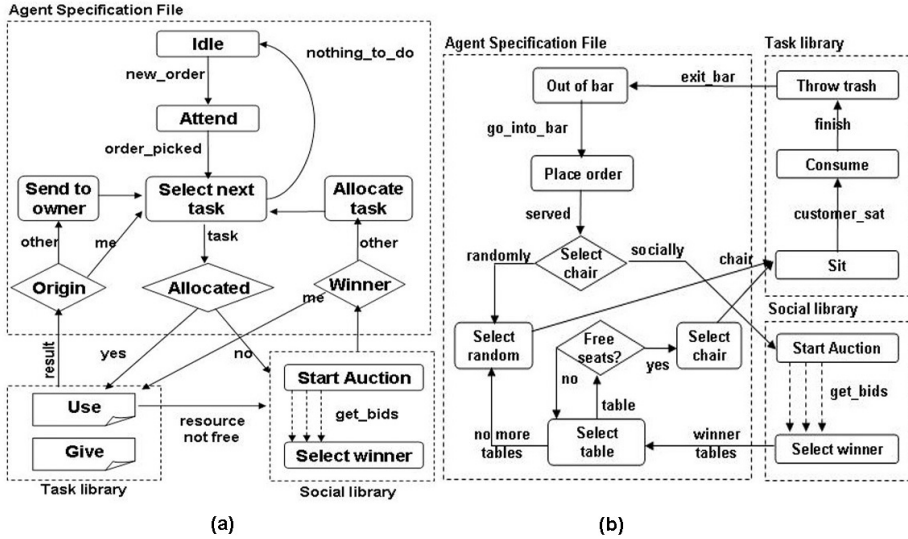


Figura 6.10: Controladors dels agents: (a) cambrers i (b) clients.

de classe, d'acord amb els quals els clients desitgen consumir junt amb aquells actors que són de la mateixa classe social. Per tant, els clients utilitzen MADeM per decidir on seure a consumir. Com que el problema de decisió dels clients serà triar entre diferents seients (*SeatingPlaces*) per consumir, els clients subhasten la ranura Obj_m de l'acció *SitAt*, que correspon amb el lloc on seure (veure la taula 6.4). Les classes dels objectes, usades per descriure les ranures de les tasques, s'extrauen de la taxonomia d'objectes definida en l'ontologia del bar virtual. Açò és, objectes com les taules o la barra del bar seran implementades com instàncies de la classe *SeatingPlace* i, doncs, seran considerades com a llocs possibles on seure a consumir. L'equació 6.2 defineix el conjunt de possibilitats entre les quals decidir, çò és, els llocs candidats on seure són totes les taules de l'entorn així com la pròpia barra del bar.

$$\begin{cases} P_0 = \{SitAt(Obj_m) \leftarrow bar\} \\ P_j = \{SitAt(Obj_m) \leftarrow table\} \quad \forall table \in Tables \end{cases} \quad (6.2)$$

A continuació, mostrem el conjunt de funcions d'utilitat utilitzades pels cambrers i pels clients del bar virtual. Aquestes funcions expressen, segons diversos punts de vista,

les **preferències** sobre cadascuna de les possibles solucions a un problema donat, representades en les equacions 6.1 i 6.2.

6.4.3 Expressió de preferències

En l'exemple del bar virtual, els cambrers tenen en compte tres punts de vista a l'hora d'expressar les seues preferències: l'**eficiència**, la **sociabilitat** i el **cansament**. Les equacions 6.3 i 6.4 defineixen els valors d'utilitat retornats per la funció d'utilitat d'eficiència per a les tasques *Use* i *Give*. Aquesta funció té el propòsit de maximitzar el nombre de tasques realitzades simultàniament i representa el desig dels cambrers de servir les comandes tan aviat com puguem. El comportament social, definit pels cambrers, està orientat a animar xats amb el seus amics del treball (aquells amb qui tenen una relació de tipus *workFriend*). Per tal de fer-ho, els cambrers implementen la funció d'utilitat social que mostren les equacions 6.5 i 6.6; en les quals *Near* computa la distància que hi ha entre els agents mentre executen un parell de tasques. La funció d'utilitat social avalua l'interès social com la possibilitat de trobar un amic en un futur proper i, per tant, poder animar una conversa amb ell. Per últim, l'equació 6.7 defineix la funció d'utilitat de cansament d'un cambrer. Aquesta funció implementa el principi bàsic de mínima energia, que els humans apliquem sovint en la feina. El tipus de societat simulada pels cambrers és **elitista**, en conseqüència, MADeM triarà aquelles assignacions que maximitzen les funcions d'utilitat suara esmentades.

$$U^{perf}(\text{Use}(Ag_e) \leftarrow a) = \begin{cases} 0 & \text{if } a \neq \text{Myself} \\ 1 & \text{if } (a = \text{Myself}) \wedge \\ & \text{IsUsing}(\text{Myself}, \text{Obj}_m) \wedge \\ & \text{not}(\text{IsComplete}(\text{Obj}_m)) \\ \frac{1}{t_{\text{to be free}} + t_{\text{queue}}} & \text{Otherwise} \end{cases} \quad (6.3)$$

$$U^{perf}(\text{Give}(Ag_e) \leftarrow a) = \begin{cases} 0 & \text{if } a \neq \text{Myself} \\ 1 & \text{if } (a = \text{Myself}) \wedge \\ & \text{CurrentTask} = \text{'Give'} \wedge \\ & \text{not}(\text{HandsBusy}(\text{Myself}) < 2) \\ \frac{1}{t_{\text{to be free}}} & \text{Otherwise} \end{cases} \quad (6.4)$$

$$U^{soc}(t_1(Ag_e) \leftarrow a_1, t_2(Ag_e) \leftarrow a_2) = \begin{cases} 0 & \text{if } a_1 \neq Myself \vee a_2 \neq Auctioneer \\ 1 & \text{if } (a_1 = Myself) \wedge a_2 = Auctioneer \wedge \\ & IsFriend(a_1, a_2) \wedge Near(Pos(t_1), Pos(t_2)) \wedge \\ & ExecTime(t_2) > RemainTime(CurrentTask) \\ 0 & \text{Otherwise} \end{cases} \quad (6.5)$$

$$U^{soc}(t(Ag_e) \leftarrow a) = \begin{cases} 0 & \text{if } a \neq Auctioneer \\ 1 & \text{if } a = Auctioneer \wedge IsFriend(a, Myself) \\ & \wedge Near(Pos(CurrentTask), Pos(t)) \wedge \\ & \wedge TimeToStart(t, a) = Now \\ 0 & \text{Otherwise} \end{cases} \quad (6.6)$$

$$U^{tir}(t(Ag_e) \leftarrow a) = \begin{cases} 1 - \frac{tasks_done}{total_tasks} & \text{if } a = Myself \\ 0 & \text{Otherwise} \end{cases} \quad (6.7)$$

Pel que fa als clients, considerem dos punts de vista: la **sociabilitat** i la **mandra**. Les equacions 6.8 i 6.9 corresponen a la funció d'utilitat social definida pels clients. Aquesta funció assigna el valor màxim a una taula en cas que, assegut en ella, hi haja un client de la mateixa classe social (professor, estudiant graduat o no graduat). El fet de consumir a soles en la barra no té gaire interès social, llavors, el seu valor d'utilitat és zero. La funció d'utilitat de mandra està representada per les equacions 6.10 and 6.11. Aquesta funció avalua les taules en funció de la seua distància al client i, al contrari de la sociabilitat, que el client consumisca en la barra es considera la millor opció. El tipus de societat simulat pels clients és **utilitària**, així doncs, MADeM triarà aquella assignació que maximitze la suma dels valors d'utilitat prèviament definits.

$$U^{soc}(SitAt(Obj_m) \leftarrow table) = \begin{cases} 1 & \text{if } AtTable(Myself, table) \wedge \\ & IsSameClass(Myself, Auctioneer) \wedge \\ & AvailablePlace(table) \\ 0 & \text{Otherwise} \end{cases} \quad (6.8)$$

$$U^{soc}(SitAt(Obj_m) \leftarrow bar) = 0 \quad (6.9)$$

$$U^{laz}(SitAt(Obj_m) \leftarrow table) = \begin{cases} \frac{\min_{t \in Tables} \{DistanceTo(t)\}}{DistanceTo(table)} & \text{if } Auctioneer = Myself \wedge \\ & AvailablePlace(table) \\ 0 & \text{Otherwise} \end{cases} \quad (6.10)$$

$$U^{laz}(SitAt(Obj_m) \leftarrow bar) = \begin{cases} 1 & \text{if } Auctioneer = Myself \wedge \\ & AvailablePlace(bar) \\ 0 & \text{Otherwise} \end{cases} \quad (6.11)$$

Les funcions d'utilitat, definides adés pels cambrers i els clients, expressen una preferència diferent de zero només sota certes condicions. Com que un valor d'utilitat zero és similar a no expressar cap preferència, les utilitats zero no s'envien al subhastador. Mitjançant aquest detall d'implementació, el problema de determinació del guanyador es veu simplificat, ja que no considera apostes que no aporten res a la decisió.

6.4.4 Animació d'acords socials

Els controladors dels cambrers i dels clients invoquen MADeM quan l'actor ha de prendre una decisió social (veure la figura 6.10). Llavors, cal arreplegar la decisió guanyadora i animar l'**acord social** que corresponga. Per animar els acords socials, el mòdul de comunicació proporciona diversos protocols d'animació de converses entre personatges 3D. Per exemple, els diàlegs *tell* i *shout* serveixen perquè els agents es sol·liciten l'execució de tasques, respectivament, amb un to de veu normal o a crits.

Com a cas d'ús, considerem que un cambrer (*A*) avalua dos punts de vista a l'hora de subhastar l'execució d'una tasca (*t*): l'eficiència i la sociabilitat. La taula 6.5 resumeix els plans animats pels cambrers depenent del criteri guanyador. La meitat superior de la taula correspon amb l'eficiència, açò és, quan el guanyador és aquell qui realitza la tasca en el menor temps. Si és un agent extern *B* qui guanya la subhasta, l'agent subhastador *A* pot animar l'acord de diverses maneres. Per exemple, pot acostar-se i encetar un diàleg on li sol·licite que execute la tasca *t*, o bé, notificar-li-ho a crits (p. ex. l'agent *A* llança el crit: "*Per favor B, fes-me un cafè!*"). En ambdós diàlegs, cal una resposta positiva per part de *B* perquè l'animació siga consistent amb el resultat obtingut de MADeM. A sovint, quan l'agent subhastador és també el guanyador, els agents no animen cap acord social sinó que simplement l'agent *A* comença a executar la tasca. Això no obstant, l'animació

Performance winner			
Winner	Preconditions	Action	Response
A	None	None	None
	near(B,Resource) not(near(B,A))	shout(A,B,make(t))	shout(B,A,no)
	near(B,Resource) near(B,A)	tell(A,B,make(t))	tell(B,A,no)
B	noise(high)	approach(B), tell(A,B,make(t))	tell(B,A,yes)
	noise(low)	shout(A,B,make(t))	shout(B,A,yes)
Sociability winner			
A	None	plan_meeting(A,B),chat(A,B)	chat(B,A)
B	noise(high)	approach(B), tell(A,B,make(t)), plan_meeting(A,B),chat(A,B)	tell(B,A,yes) chat(B,A)
	noise(low)	shout(A,B,make(t)), plan_meeting(A,B),chat(A,B)	shout(B,A,yes) chat(B,A)

Taula 6.5: Exemples de converses per animar acords socials.

d'aquestes situacions pot ser també molt útil per reflectir desacords entre els agents, és a dir, converses amb una resposta negativa (p. ex. quan l'agent *B* respon: "*Ho sent, no puc fer-lo perquè estic molt enfeinat*").

La meitat inferior de la taula 6.5 mostra un exemple dels diàlegs que es poden animar quan el criteri guanyador ha estat la sociabilitat. En aquest cas, els personatges planifiquen una cita mitjançant l'acció *plan_meeting*. La planificació de cites és un diàleg en què els actors acorden un encontre per poder parlar (p. ex. "*Per favor, vine a la cafetera i parlem una estona*"). Posteriorment, l'animació de la conversa (*chat*) entre els personatges s'iniciarà quan els actors tinguin una cita prèvia i estiguin a la distància adequada. Com hem vist en ambdós casos, la decisió sobre l'animació a utilitzar depèn d'unes precondicions, com ara el nivell de soroll ambient (*noise*) o la distància entre els agents i els recursos (*near*). La satisfacció d'aquestes precondicions es realitzarà en funció de l'estat de l'escena. Per exemple, el nivell de soroll pot ser fàcilment calculat atenent al nombre d'agents de l'escena o només aquells que estan prop del guanyador.

6.4.5 Resultats socials

En aquesta secció mostrem un conjunt de resultats socials, obtinguts amb diferents cambrers i clients que utilitzen MADeM com el seu mecanisme de presa de decisions social. Els valors que analitzem a continuació corresponen amb simulacions en què 10 cambrers serveixen les comandes de 100 clients al bar universitari virtual.

Com presentàrem en la secció 6.4.3, hem modelat una societat elitista de cambrers en la qual els actors tenen en compte tres punts de vista (eficiència, sociabilitat i cansament); cadascun representat per la seua funció d'utilitat. En aquest context, els pesos d'utilitat poden ser ajustats per crear cambrers amb diversos tipus de **personalitat**. Per exemple, un cambrer *coordinat* podria donar el triple d'importància al criteri d'eficiència que als criteris de sociabilitat i cansament. D'acord amb açò, un vector de pesos d'utilitat per a un cambrer *coordinat* seria $\vec{w}_u = \langle 0.75, 0.125, 0.125 \rangle$, en què cada component representa la importància donada a cadascuna de les funcions d'utilitat avaluades. De manera similar, hem creat els cambrers *socials* (o xarrais) com actors que utilitzen el següent vector de pesos d'utilitat $\vec{w}_u = \langle 0.125, 0.75, 0.125 \rangle$ i els cambrers *igualitaris* (o calculadors) com els que utilitzen el vector $\vec{w}_u = \langle 0.125, 0.125, 0.75 \rangle$.

La taula 6.6 resumeix les tasques realitzades pels cambrers anteriors (*coordinats*, *socials* i *igualitaris*) en comparació amb un conjunt de cambrers *autointeressats*; açò és, que no inclouen cap mecanisme de presa de decisions social. Els cambrers *coordinats* mostren una millor eficiència (veure la columna *Tasques/s*), ja que la majoria de conflictes causats per la voluntat d'usar el mateix recurs (p. ex. la cafetera) es resolen mitjançant l'**especialització**. Açò és, delegant l'execució de la tasca a un altre cambrer que estiga ja emprant l'esmentat recurs. D'una altra banda, els cambrers *socials* triguen més a servir les comandes dels clients, ja que animen més xats entre els cambrers (veure la mitjana de xats animats en la columna \overline{NXats}). Els cambrers *igualitaris* donen més importància a la funció d'utilitat de cansament i, per tant, tracten d'assignar l'execució de la tasca al cambrer que estiga més descansat. En conseqüència, la desviació estàndard en el nombre de tasques executades per cada agent tendeix a zero per aquests actors (veure la columna $\sigma_{NTasques}$). Darrerament, els cambrers *autointeressats* demostren treballar de forma menys eficient que qualsevol tipus de cambrer amb habilitats socials. Com que aquests actors no realitzen reassignació de tasques ni xats socials, les columnes $\sigma_{NTasques}$ i \overline{NXats} no han estat considerades.

Tipus d'actor	$\sigma_{NTasques}$	\overline{NXats}	Tasques/s
Coordinat	6.73	5	0.91
Social	4.37	29.4	0.65
Igualitari	2.74	6.6	0.62
Autointeressat	-	-	0.17

Taula 6.6: Tasques realitzades per diversos tipus de cambriers.

A banda de definir la importància de cada punt de vista mitjançant el vector de pesos d'utilitat (\vec{w}_u), MADeM permet la definició d'un vector de pesos personals (\vec{w}), els quals modelen l'actitud d'un actor envers la resta d'individus. La taula 6.7 mostra els resultats obtinguts pels cambres definits prèviament usant els quatre **models d'actitud** presentats en la secció 5.4.4: **indiferència, reciprocitat, altruisme i egoisme**. Els actors que fan servir el model d'indiferència no realitzen cap modificació sobre les utilitats rebudes de la resta de personatges, per tant, considerem els resultats obtinguts per la indiferència com els valors base amb què comparar la resta d'actituds. La reciprocitat sospesa les utilitats atenent al conjunt de favors intercanviats entre els agents. Aquesta actitud produeix l'equilibri entre el nombre de favors intercanviats, cosa que es pot apreciar en el baix valor que pren σ_{Favors} pels tres tipus de cambriers recíprocs ($< 1.15, 1.76, 2.4 >$); en comparació amb els valors obtinguts amb el model d'indiferència ($< 7.57, 3.52, 7.58 >$). L'altruisme ha estat implementat de manera que el pes que els agents donen a la seua utilitat és 0.25, mentre que el pes donat a les utilitats de la resta de personatges és 0.75. Com a conseqüència, els agents altruistes realitzen un major nombre de favors, ja que la importància que donen a les seues opinions és tres vegades inferior a la donada a les preferències externes (veure els alts valors de la mitjana de favors intercanviats \overline{Favors} $< 17, 12.7, 17.9 >$). Pel contrari, l'egoisme sospesa amb 0.75 les preferències pròpies i amb 0.25 les externes, així doncs, els actors rarament fan favors (veure els valors tan baixos de la columna \overline{Favors} per a l'egoisme $< 0.7, 0.4, 0.1 >$).

Les actituds personals poden, de vegades, anar en contra de les preferències dels actors, i viceversa. Per exemple, mentre que la reciprocitat tracta de balancejar el nombre de favors, el cansament tendeix a assignar les tasques al cambrer menys cansat (veure l'alta σ_{Favors} dels personatges igualitaris recíprocs a la taula 6.7). Un altre exemple el trobem a l'egoisme aplicat als cambriers igualitaris, que no intercanvien pràcticament l'execució de cap tasca ($\overline{Favors} = 0.1$). Nogensmenys, les actituds personals també poden

Actitud	Coordinat		Social		Igualitari	
	σ_{Favors}	\overline{Favors}	σ_{Favors}	\overline{Favors}	σ_{Favors}	\overline{Favors}
Indiferència	7.57	6.9	3.52	8.7	7.58	13.6
Reciprocitat	1.15	8.8	1.76	7.8	2.4	15.5
Altruisme	5.94	17	6.66	12.7	4.44	17.9
Egoisme	1.41	0.7	0.81	0.4	0.47	0.1

Taula 6.7: Favors intercanviats amb diferents models d'actitud personal.

enfortir les preferències dels actors. Per exemple, l'altruisme aplicat als cambrers coordinats produeix un alt nivell d'especialització. Aquest tipus d'actors genera grans valors de σ_{Favors} , ja que els agents que comencen a usar un recurs (p. ex. una màquina de fer suc) romandran extraient productes d'acord amb la seua política de comportament altruista i coordinat. Malgrat aquestes variacions en els efectes dels models d'actitud, els pesos d'actitud han demostrat produir efectes similars en els personatges, independentment del tipus de cambrer que es considere: coordinat, social o igualitari.

Al contrari dels cambrers, els clients prenen les seues decisions dins d'una societat utilitària, en què consideren dos punts de vista (sociabilitat i mandra). La figura 6.11 mostra una comparativa del comportament obtingut en funció del pes d'utilitat assignat a cadascun d'aquests criteris. En aquesta gràfica comparem dues mètriques: a) la mitjana de trobades socials produïts entre els clients; i b) la distància mitjana recorreguda per un client a l'hora de consumir. Els clients mandrosos, que apliquen un pes d'utilitat baix a la sociabilitat, decideixen consumir preferiblement en la barra o seure en una taula que estiga a prop (veure el punt $\vec{w}_u = \langle 0.3, 0.7 \rangle$). En aquest cas, la distància recorreguda fins el punt de consum és curta però només uns poques trobades socials són animades. D'una altra banda, els clients socials, que apliquen pesos més alts a la sociabilitat, duen a terme més trobades socials però també necessiten recórrer distàncies més llargues per trobar els seus companys. Per exemple, els punts $\vec{w}_u = \langle 0.6, 0.4 \rangle$ i $\vec{w}_u = \langle 0.7, 0.3 \rangle$ de la figura 6.11 són un exemple dels resultats obtinguts amb clients socials.

Els efectes de la **sociabilitat** sobre els clients es poden veure en la figura 6.12. Aquesta instantània mostra un bar virtual poblat per 7 cambrers i 17 clients. Els clients pertanyen als tres grups socials definits prèviament: professors, estudiants no graduats i graduats. En aquesta simulació, els clients usen el vector de pesos d'utilitat $\vec{w}_u = \langle 0.7, 0.3 \rangle$.

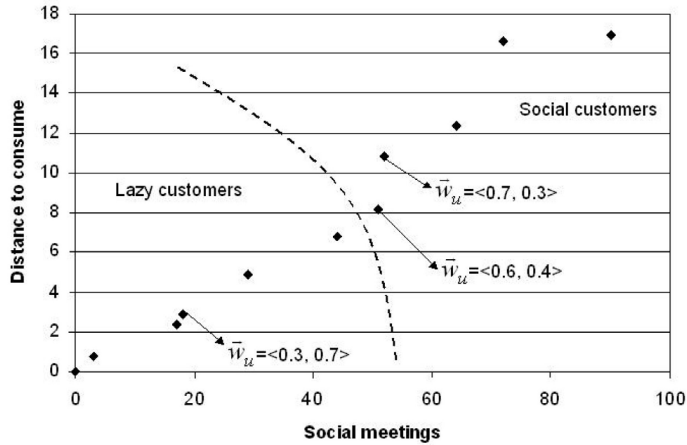


Figura 6.11: Comparativa entre els clients mandrosos i els clients socials.

Aleshores, una vegada han estat servits, els actors normalment decideixen consumir junt amb altres clients del seu grup social. Per distingir els membres de cada grup social, emprarem avatars diferents. En la figura 6.12 es veu com els avatars del mateix tipus s'ajunten per a consumir en la mateixa taula sempre que poden.

En resum, en aquesta secció hem presentat els efectes de la **multimodalitat** de criteris i de les **actituds personals** en la presa de decisions dels actors sintètics. A continuació, analitzem més pregonament els resultats obtinguts per a dos criteris molt importants dins del comportament en grup: la coordinació i la sociabilitat.

6.4.6 Coordinació vs. Sociabilitat

En el camp dels sistemes multiagent, el comportament social ha estat associat sovint amb la capacitat dels agents d'actuar de forma coordinada. Tanmateix, els personatges 3D poden mostrar actituds socials amb **comportaments no necessàriament orientats a l'eficiència**. Per exemple, els cambrers i els clients descrits en les seccions anteriors implementen la sociabilitat com l'habilitat de conversar amb altres personatges. D'acord amb açò, en aquesta secció comparem els efectes de la coordinació i de la sociabilitat en el comportament dels actors sintètics. Per tal de fer-ho, utilitzem els cambrers definits amb anterioritat, però no tenim en compte el criteri del cansament ($w_{tir} = 0$). Llavors, el

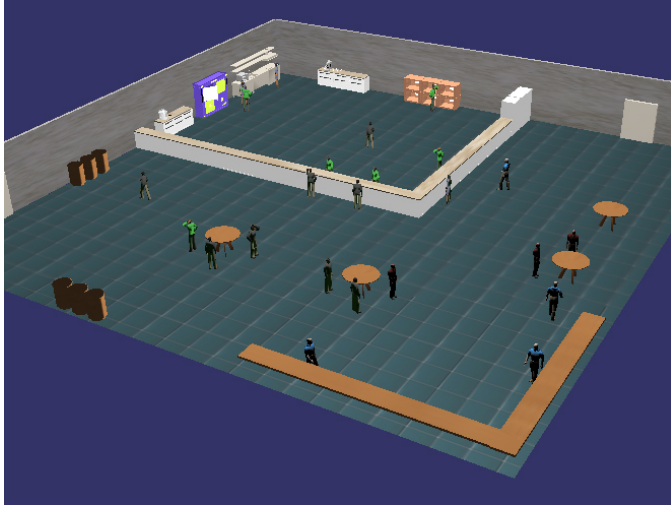


Figura 6.12: Entorn 3D habitat que simula un bar virtual.

pes d'utilitat assignat a la sociabilitat w_{soc} , que anomenem ací *factor de sociabilitat*, és el paràmetre amb què balancejarem eficiència i sociabilitat, ja que $w_{perf} = 1 - w_{soc}$.

Primerament, considerem unes traces senzilles de comportament on intervenen un conjunt reduït de cambrers i clients. La figura 6.13 esquematitza dues simulacions en les quals 2 cambrers reben les comandes de 4 clients en el bar virtual. En aquestes simulacions, tots els clients demanen un sandvitx però la planxa, recurs necessari per preparar-los, és compartida pels cambrers. A pesar que la planxa no pot ser emprada per més d'un cambrer alhora, hem proporcionat aquest objecte amb la capacitat de fer 4 sandvitxos de manera simultània. Llavors, els cambrers poden coordinar-se per servir els clients més ràpidament. La figura 6.13a mostra la traça resultant quan els cambrers tenen un grau de coordinació màxima ($w_{perf} = 1$). En aquest cas, *Waiter1* s'especialitza a fer sandvitxos, mentre que *Waiter2* s'engarrega d'agafar les comandes i de servir els clients. Açò és, els cambrers treballen de la manera més eficient possible.

Tanmateix, un excés d'eficiència no és més comú en el comportament dels robots que en les societats humanes. Aleshores, el factor de sociabilitat pot ser ajustat per incorporar interaccions socials que responguen a les relacions definides entre els actors, com ara l'amistat en el treball (*workFriends*). Per exemple, la figura 6.13b correspon a la situació en què els cambrers utilitzen $w_{soc} = 0.6$. En aquesta configuració, *Waiter2* primerament

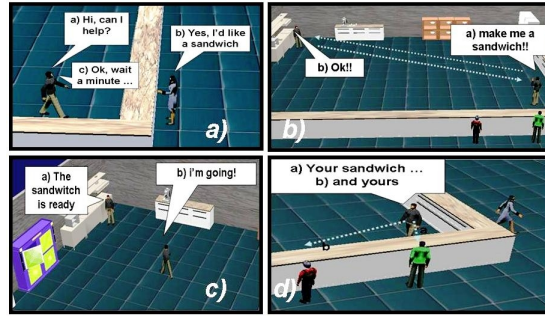


Figura 6.14: Animació de diverses situacions interactives: (a) Atendre un client, (b) Demanar un favor a un company, (c) Informar del resultat d'una acció (d) Servir un client.

altre. Una vegada que la tasca reassignada ha estat completada, el cambrer que la realitza informa el sol·licitador del resultat. Per exemple, en la vinyeta 6.14c, *Waiter1* comunica *Waiter2* que el sandvitx està preparat. Anàlogament a usar un recurs per resoldre diverses tasques de manera simultània, els cambres poden usar les seues mans per dur més d'un producte alhora. Per exemple, en la vinyeta 6.14d, *Waiter2* carrega els dos sandvitxos de *Customer2* i *Customer3*.

Per estimar numèricament els efectes del **balanceig entre la coordinació i de la sociabilitat** hem creat un conjunt de simulacions que varien el factor de sociabilitat dels cambres, des del valor mínim $w_{soc} = 0$ al valor màxim $w_{soc} = 1$. Mesurem l'eficiència d'un grup d'actors mitjançant l'estimador *Throughput*. Aquest estimador és un indicador que pren valors en el rang $[0, 1]$ i que estima com de prop es troba una simulació de la situació ideal, en què la càrrega de treball pot ser distribuïda entre tots els membres del grup i no es produeix cap col·lisió. D'acord amb açò, l'equació 6.12 defineix *Throughput* com la relació entre el temps de simulació ideal (T_{sim}^*) i el temps de simulació real (T_{sim}), on N_{tasks} i N_{agents} és el nombre de tasques i d'actors respectivament mentre que $\overline{T_{task}}$ és el temps mitjà que costa executar una tasca.

$$Throughput = \frac{T_{sim}^*}{T_{sim}} = \frac{N_{tasks} * \overline{T_{task}} / N_{agents}}{T_{sim}} \quad (6.12)$$

La figura 6.15a mostra els valors de *Throughput* obtinguts amb els diferents cambres socials, en comparació amb els cambres autointeressats (sense cap mecanisme so-

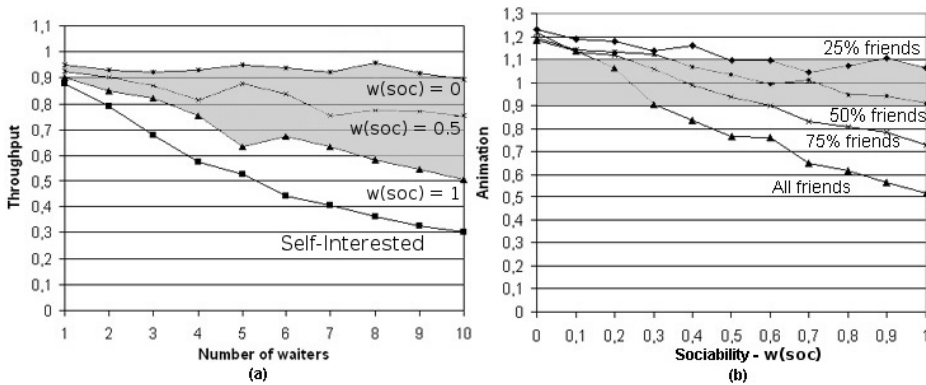


Figura 6.15: Estimadors de coordinació i sociabilitat per a un grup de cambrers: (a) *Throughput* i (b) *Animation*.

cial inclòs). Com ja sabem, els actors autointeressats col·lisionen constantment a causa de la competició per l'ús dels recursos compartits. Aquestes col·lisions produeixen temps d'espera elevats que redueixen l'eficiència en tant que el nombre d'actors creix, cosa que produirà una animació comportamental de baixa qualitat. L'eficiència en el comportament pot ser millorada amb personatges totalment coordinats ($w_{soc} = 0$), els quals intercanvien les tasques amb uns altres que les puguin executar en paral·lel i, per tant, redueixen els temps d'espera dels recursos. Tanmateix, aquest comportament extremadament coordinat és més propi dels robots (que estan contínuament treballant si tenen l'oportunitat) que dels humans (que manifesten les seues relacions socials, per exemple, mitjançant les converses amb els amics). Llavors, el factor de sociabilitat, w_{soc} , pot ser usat per balancejar coordinació amb sociabilitat i garantir, doncs, comportaments més humans que robòtics. En aquest context, el valor de *Throughput* del tipus d'animacions que considerem interessants es troba entre el comportament totalment coordinat ($w_{soc} = 0$) i el comportament totalment social ($w_{soc} = 1$), com destaca la zona ombrejada de la figura 6.15a.

Throughput és un estimador del grau de coordinació assolit pels personatges 3D. Emperò, a pesar que l'eficiència és un requeriment bàsic a l'hora de simular grups d'actors sintètics, aquest no és l'únic criteri que convé avaluar quan hom pretén crear simulacions versemblants. D'acord amb açò, hem definit un estimador que té en compte el temps que el dissenyador de la simulació desitja que els personatges dediquen a l'animació d'interaccions socials. L'equació 6.13 conté la definició d'aquest estimador, el qual anomenem

Animation. En aquesta equació, T_{social} representa el temps dedicat a l'animació de converses i acords socials entre els personatges. En el nostre bar virtual, hem definit aquest temps com el 35% del temps ideal de simulació (T_{sim}^*).

$$Animation = \frac{T_{sim}^* + T_{social}}{T_{sim}} \quad (6.13)$$

La figura 6.15b mostra els valors d'animació obtinguts pel grup de cambrers quan es consideren 4 configuracions d'amistat: tots els cambrers són amics, el 75% dels cambrers són amics, la meitat dels cambrers són amics i només el 25% dels cambrers són amics. Com ja havíem esmentat abans, els valors baixos de sociabilitat produeixen simulacions amb un baix nivell de versemblança, cosa que es reflecteix en la funció d'animació, que es troba per damunt del valor de referència ($Animation = 1$). Així mateix, valors molt alts de sociabilitat també condueixen a simulacions de baixa versemblança, especialment quan el grau d'amistat és alt. En aquestes configuracions, el nombre de converses socials animades és massa gran per ser realista i el valor d'animació es troba lluny del valor de referència. L'estimador d'animació pot ser emprat per extraure el rang de valors adequat pel factor de sociabilitat (w_{soc}), depenent de la situació que s'estiga simulant. Per exemple, en el nostre bar virtual hem considerat com simulacions de bona qualitat aquelles que cauen dins de la franja del $\pm 10\%$ del valor de referència (veure la zona ombrejada de la figura 6.15b). D'acord amb açò, quan tots els cambrers són amics, les bones animacions emergeixen quan $w_{soc} \in [0.1, 0.3]$ (veure la línia *All friends* de la figura 6.15b).

Darrerament, la taula 6.8 compara la quantitat de temps dedicat a executar cada tipus de tasca per cadascun dels 10 cambrers amb els dos valors extrems del factor de sociabilitat: coordinació màxima ($w_{soc} = 0$) i sociabilitat màxima ($w_{soc} = 1$). En el cas de la coordinació màxima, els valors irregulars de les columnes T_{use} i T_{give} de la part esquerra de la taula demostren com alguns cambrers s'han especialitzat en l'execució de certes tasques. Per exemple, els cambrers 2, 5, 9 i 10 destinen la majoria del seu temps a entregar productes als clients, mentre que els cambrers 3 i 7 es dediquen principalment a usar els recursos del bar (p. ex. la màquina de fer sucs). A pesar que l'especialització és un efecte desitjable en molts sistemes multiagent, les societats humanes, a sovint de caire igualitari, solen balancejar la càrrega de treball assignada a cada agent. Segons mostra la part dreta de la taula, els cambrers completament socials equilibren els temps en què usen els recursos de l'entorn i duen productes als clients (veure les columnes T_{use} i T_{give}). A més a més, si el model d'actitud personal és la reciprocitat, la diferència de favors intercanviats ten-

Agent	$w_{soc} = 0$					$w_{soc} = 1$				
	T_{wait}	T_{chat}	T_{use}	T_{give}	$Balan$	T_{wait}	T_{chat}	T_{use}	T_{give}	$Balan$
1	0	0	32	19	-6	16	36	69	34	-2
2	3	0	4	26	-3	18	62	58	24	-2
3	14	0	52	1	28	41	66	45	16	0
4	3	0	16	28	-3	48	61	60	27	3
5	0	0	7	30	-16	34	68	58	12	-1
6	3	0	37	17	-1	48	74	64	14	-2
7	0	0	67	4	21	18	66	48	24	1
8	0	0	45	17	1	33	76	45	24	4
9	7	0	5	23	-11	46	58	36	21	0
10	1	0	6	41	-10	27	69	56	20	-1

Taula 6.8: Distribució temporal (en segons) de les tasques executades per 10 cambres.

deix a zero (comparar les columnes anomenades *Balan*). Un efecte col·lateral d'aquest equilibri és l'increment en els temps d'espera, ja que els cambres socials de vegades prefereixen esperar que l'altre cambre allibere el recurs que passar la tasca (comparar les columnes T_{wait}). Com a conseqüència, sorgeix un nou percentatge de temps (T_{chat}) en el qual els cambres poden animar interaccions socials com ara xats entre cambres que són amics.

En conclusió, malgrat que la coordinació és un comportament molt desitjat en contextos multiagent, generalment, no és l'únic camí per expressar la sociabilitat. Al contrari, el **balanceig** de la coordinació amb comportaments socials no orientats a l'eficiència (p. ex. que responguen a relacions socials d'índole diversa com ara l'amistat, el rol o la classe social), aporta eixa **variabilitat** que distingeix el raonament humà del robòtic i que, per tant, millorarà la versemblança dels actors sintètics.

6.4.7 Conclusions MADeM

La presa de decisions de tipus MADeM proporciona diverses característiques interessants a l'hora d'incorporar raonament social en l'animació de personatges 3D. Prime-

rament, MADeM permet que els agents consideren diferents punts de vista per avaluar les seues accions (**multimodalitat**). L'ús de les funcions d'utilitat per expressar aquests punts de vista és una aproximació modular que permet definir diferents criteris de manera senzilla. Per exemple, hem desenvolupat grups d'actors sintètics que prenen les seues decisions d'acord amb criteris com ara l'eficiència, la sociabilitat, el cansament o la mandra. En aquest context, el sospesament de les diferents funcions d'utilitat permet definir **personalitats** diverses que atenen més a un punt de vista que a un altre (p. ex. actors coordinats, xarrais, calculadors, mandrosos, etc.). En segon lloc, MADeM arreplega les preferències de la resta de membres de la societat mitjançant la subhasta de les possibles solucions a un problema de decisió. En aquest context, l'ús de distintes funcions de benestar social servirà per simular diferents **models de societat**. En l'exemple del bar virtual, la societat simulada pels cambrers corresponia amb el benestar elitista mentre que els clients seguien un benestar igualitari. Tercerament, per mitjà de l'aplicació d'un diferencial a les preferències rebudes, MADeM proporciona als personatges la possibilitat de modelar **actituds personals** envers la resta d'individus de la societat artificial. Aquestes actituds personals poden variar des de la indiferència, passant per la reciprocitat, fins a l'altruisme i l'egoisme.

Tot plegat, MADeM ha demostrat la seua capacitat per reproduir comportaments socials com ara: la coordinació, l'especialització, l'intercanvi de favors, el foment d'encontres socials, el balanceig de tasques executades, etc. Comptat i debatut, MADeM proporciona un conjunt d'eines adequades per proveir els actors sintètics d'un mecanisme de **presa de decisions socialment acceptables**.

CONCLUSIONS, APORTACIONS I TREBALLS FUTURS

7.1 CONCLUSIONS GENERALS I APORTACIONS

La recerca duta a terme en aquesta tesi, per aconseguir la integració d'habilitats socials en l'animació comportamental dels actors sintètics, ha generat els resultats i aportacions que resumim en els punts següents:

- La simulació d'entorns virtuals habitats s'enfronta a la resolució de problemes complexos, provinents del camp de l'animació gràfica i de la intel·ligència artificial. Per poder abastar aquest conjunt de problemes, hem proposat un **marc de simulació multiagent** que separa el model semàntic, el model comportamental i el model geomètric. D'acord amb açò, el sistema de simulació està format per: a) un **entorn virtual semàntic**, que crea i manté l'estat del món; b) un conjunt d'**agents socialment intel·ligents**, que genera el comportament dels personatges; i c) un **motor gràfic 3D**, encarregat de la visualització.
- L'evolució de l'animació comportamental dels personatges 3D ha anat lligada al nivell de representació de l'estat del món. En aquest context, preconitzem un **model general d'entorn virtual semàntic basat en ontologies** amb el qual es poden aconseguir tres objectius bàsics per a l'animació d'actors sintètics: a) la millora de la sensorització d'escenes complexes; b) la definició d'operatives generals que els personatges puguin reutilitzar en situacions diverses; i c) la definició de les relacions socials i d'organització establides entre els membres d'una societat artificial.
- El desenvolupament de personatges 3D comporta la resolució d'una multitud de problemes complexos, com ara: la percepció, el control motor, la navegació, la

comunicació, la selecció d'accions, etc. En conseqüència, proposem una **arquitectura d'agent socialment intel·ligent** formada per un conjunt de mòduls a càrrec d'aquests aspectes. El grau de versemblança comportamental d'aquests agents dependrà, fonamentalment, de les seues capacitats racionals i socials. Per tant, les aportacions principals d'aquesta tesi s'han centrat en la inclusió de criteris socials en dos formalismes ben coneguts per a la selecció dinàmica d'accions en 3DIVA: la planificació de tasques de tipus STRIPS i el model BDI.

- En primer lloc, hem presentat una **tècnica per aconseguir comportaments de tipus col·laboratiu en aquells personatges 3D que empren els planificadors heurístics com a mecanisme de selecció dinàmica d'accions**. Aquesta tècnica proveeix els actors amb l'habilitat d'adaptar-se i actuar de forma eficient als entorns poblats per diversos caràcters autònoms (p. ex. jocs, *storytelling*, simulació civil, etc.). En aquests contextos, hem usat la comunicació directa entre els personatges per construir la representació en memòria dels agents externs. Llavors, hem incorporat una fase de pre-planificació que afavoreix la **coordinació** amb les accions d'altres agents i hem desenvolupat un mecanisme de sospesament de la funció heurística que permet els agents **cooperar** en la consecució d'objectius externs.
- En segon lloc, hem presentat **MADeM (Multi-modal Agent Decision Making), un procés de presa de decisions socialment acceptables per a actors sintètics**. D'una banda, MADeM permet avaluar diversos punts de vista o preferències personals (p. ex. l'eficiència, la sociabilitat, el cansament, etc.). D'una altra, MADeM és capaç d'incorporar les preferències d'altres agents externs mitjançant una tècnica basada en l'ús de subhastes de tasques. En aquest context, MADeM permet definir diversos ordres socials en una societat (p. ex. l'elitista, l'utilitari, etc.) així com actituds personals dels seus membres (p. ex. l'egoisme, l'altruisme, etc.). Finalment, MADeM ha demostrat la seua capacitat per reproduir habilitats socials en actors de tipus BDI com ara: la coordinació, l'especialització, l'intercanvi de favors, el foment d'encontres socials, el balanceig de tasques executades, etc.

En resum, hem dissenyat, implementat i provat un marc de simulació d'entorns virtuals habitats per actors socialment intel·ligents. Aquest sistema emprà les ontologies com una solució genèrica per a la representació semàntica de l'estat del món. Pel que fa a la integració d'habilitats socials, hem provat dues aproximacions, una basada en la col·laboració amb planificadors heurístics i una altra en un procés de presa de decisions socialment acceptables per a agents de tipus BDI. Per tant, podem considerar assolits els

objectius proposats a l'inici d'aquest document; malgrat que han quedat obertes una sèrie de línies de recerca, que resumim en el punt següent.

7.2 TREBALLS FUTURS

Moltes són les línies de treball que resten obertes com el resultat d'aquesta tesi. A continuació resumim aquelles que considerem més interessants:

- **Incorporació de sistemes de normes a la dinàmica de les relacions que lliguen els membres d'una societat artificial.** Els sistemes de normes poden ser aplicats, per exemple, a la definició del contracte social, que estableix el conjunt de regles que governen les agrupacions de personatges 3D. D'acord amb aquest contracte, l'actuació d'un actor sintètic pot implicar l'activació d'una sèrie de regles que modifiquen les actituds de la resta de membres envers ell (p. ex. si un personatge perjudica el grup amb la seua actuació, la resta d'actors podrien recriminar-lo verbalment o fins i tot modificar les relacions social que tenen amb ell). Aquesta evolució social afavorirà la variabilitat en el comportament dels actors sintètics i augmentarà, doncs, el seu grau de versemblança.
- Els mecanismes de presa de decisions socials presentats en aquesta tesi responen a situacions en què els objectius són compatibles i els actors sintètics són benvolents. La compatibilitat i la benvolença es poden veure com el resultat d'una negociació prèvia entre personatges amb objectius incompatibles. Per tant, convé estudiar l'**aplicació de mecanismes de negociació com ara el regateig o les votacions**. La negociació és un aspecte interessant als entorns 3D multiusuari d'última generació (p. ex. SecondLife), en els quals la simulació no està sota el control d'un únic dissenyador i els objectius incompatibles apareixen de manera natural.
- Al llarg d'aquesta tesi, hem presentat la planificació heurística i el model BDI com a formalismes adequats per a la presa de decisions en personatges 3D. Considerem, a més a més, que ambdues aproximacions poden emprar-se de forma complementària. Per exemple, hom pot pensar a combinar la planificació per a la seqüenciació de tasques en el baix nivell i el model BDI per al manteniment dels objectius de més alt nivell. Llavors, la **integració de la col·laboració amb planificadors heurístics i MADeM** permetrà ajuntar els comportaments socials orientats a l'eficiència (p. ex. coordinació, cooperació, etc.) amb comportaments orientats a garantir el benestar

social d'un grup d'actors sintètics (p. ex. elitisme, reciprocitat, altruisme, egoisme, etc.).

- En el capítol 6, hem demostrat l'efecte de les tècniques socials desenvolupades de forma quantitativa. Tanmateix, una **validació qualitativa de la sociabilitat apreciada pels usuaris humans** pot servir per extraure dades importants. En aquest sentit, hom pot realitzar proves clàssiques en què un conjunt d'individus visualitza una selecció de vídeos i respon a les preguntes d'una enquesta; la qual avalua les qualitats socials dels actors sintètics des del punt de vista d'un observador extern. Una altra possibilitat més sofisticada consisteix a realitzar proves de presència d'última generació, en les quals l'usuari se submergeix en l'escena mitjançant algun dispositiu immersiu (p. ex. ulleres estereoscòpiques). Llavors, l'usuari podrà avaluar l'experiència social resultant de la interacció amb els humanoides virtuals.

7.3 PUBLICACIONS DERIVADES D'AQUESTA TESI

La integració d'habilitats socials en la presa de decisions dels actors sintètics ha donat lloc a les següents publicacions, ordenades en relació amb els capítols anteriors:

- **Semantic Virtual Environments for interactive planning agents.** Fran Grimaldo, Fernando Barber, Miguel Lozano, Juan M. Orduña. *International Digital Games Conference (iDiG'06)*. Portalegre (Portugal), setembre de 2006.
- **An ontology-based approach for IVE+VA.** Fran Grimaldo, Fernando Barber, Miguel Lozano. *International Conference on Intelligent Virtual Environments and Virtual Agents (IVEVA'06)*. Aguascalientes (Mexico), octubre de 2006.
- **A multiagent framework to animate socially intelligent agents.** Francisco Grimaldo, Miguel Lozano, Fernando Barber. *The 2nd Hybrid Artificial Intelligence Systems Workshop (CAEPIA-TTIA'2007)*. Springer Advances in Soft Computing, Salamanca (Spain), novembre de 2007.
- **Integrating social skills in task-oriented 3D IVA.** Fran Grimaldo, Miguel Lozano, Fernando Barber, Juan M. Orduña. *The 5th International Working Conference on Intelligent Virtual Agents (IVA'05)*. Springer-Verlag LNAI, Kos (Greece), setembre de 2005.

- **Coordination and sociability for intelligent virtual agents.** Francisco Grimaldo, Miguel Lozano, Fernando Barber. *The MALLOW Workshop on Coordination, Organization, Institutions and Norms in agent systems (COIN'07)*. Springer Verlag LNAI, Durham (UK), setembre de 2007.
- **Balancing social and task oriented behaviors for animation agents.** Francisco Grimaldo, Miguel Lozano, Fernando Barber. *III Taller en Desarrollo de Sistemas Multiagente (CEDI'2007)*. Zaragoza (Spain), setembre de 2007.
- **Social animation in complex environments.** Francisco Grimaldo, Miguel Lozano, Fernando Barber. *The 7th International Conference on Intelligent Virtual Agents (IVA'07)*. Springer-Verlag LNAI, Paris (France), setembre de 2007.
- **Animating groups of socially intelligent agents.** Francisco Grimaldo, Miguel Lozano, Fernando Barber, Guillermo Viguera. *Proceedings of the Cyberworlds 2007*. IEEE Computer Society Press. Hannover (Germany), octubre de 2007.
- **MADeM: a multi-modal decision making for social MAS.** Francisco Grimaldo, Miguel Lozano, Fernando Barber. *The 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'08)*. ACM Press. Estoril (Portugal), maig de 2008.
- **Simulating socially intelligent agents in Semantic Virtual Environments.** Francisco Grimaldo, Miguel Lozano, Fernando Barber, Guillermo Viguera. *The Knowledge Engineering Review*. ISSN: 0269-8889. Cambridge University Press (acceptat i pendent de publicació al setembre de 2008).

BIBLIOGRAFIA

- [pdd, 2008] (2008). Writing planning domains and problems in PDDL. Available at <http://www.ida.liu.se/TDDA13/labbar/planning/2003/writing.html>. 4.4.1
- [Activeworlds Corporation, 2008] Activeworlds Corporation (2008). Active worlds. Available at <http://www.activeworlds.com/>. 2.3.1
- [Agotnes et al., 2007] Agotnes, T., der Hoek, W. V., Rodriguez-Aguilar, J. A., Sierra, C., and Wooldridge, M. (2007). On the logic of normative systems. In *Proc. of IJCAI07: International Joint Conference on AI*, pages 1175–1180. AAAI Press. 3.4
- [ANA MAS, 2005] ANA MAS (2005). *Agentes software y sistemas multiagente: Conceptos, arquitecturas y aplicaciones*. Pearson Prentice Hall. 5.2.2
- [Anastassakis et al., 2001] Anastassakis, G., Panayiotopoulos, T., and Ritchings, T. (2001). Virtual agent societies with the mVITAL intelligent agent system. In *Proc. of the 3rd International Workshop on Intelligent Virtual Agents*. 3.2
- [Avradinis et al., 2005] Avradinis, N., Panayiotopoulos, T., and Aylett, R. (2005). *Intelligent techniques for planning*, chapter Continuous Planning for Virtual Environments, pages 162–193. Idea Grup Publishing. 2.4.1, 5.3.2
- [Aylett and Cavazza, 2001] Aylett, R. and Cavazza, M. (2001). Intelligent virtual environments: A state of the art report. In *Proc. of Eurographics*. 2.1
- [Badler, 1997] Badler, N. (1997). Virtual humans for animating ergonomics and simulation. In *Proc. of the Workshop on non-rigid and articulation motion*, Puerto Rico. IEEE. 2.3.1

- [Badler and Allbeck, 2000] Badler, N. and Allbeck, J. (2000). Towards behavioral consistency in animated agents. In *Proc. of Deformable avatars*. 2.1
- [Badler et al., 2000] Badler, N., Bindiganavale, R., Bourne, J., Palmer, M., Shi, J., and Schuler, W. (2000). *A parametrized action representation for virtual human agents*, pages 256–284. MIT press. 2.2.2, 4.4.1
- [Badler et al., 1993] Badler, N., Philips, C., and Webber, B. (1993). *Simulating Humans: Computer Graphics, Animation and Control*. Oxford University Press. 2.1
- [Bates, 1992] Bates, J. (1992). The nature of character in interactive worlds and the oz project. Technical Report CMU-CS-92-200, School of Computer Science, Carnegie Mellon University, Pittsburg, PA. 2.4.1
- [Benford et al., 1997] Benford, S., Greenhalgh, C., and Lloyd, D. (1997). Crowded collaborative virtual environments. In *Proc. of SIGCHI: Conference on human factors in computing systems*. 3.1, 4.3.2
- [Bergenty and Ricci, 2002] Bergenty, F. and Ricci, A. (2002). Three approaches to the coordination of multi-agent systems. In *Proc. of Symposium on Applied Computing*, pages 367–372. ACM Press. 3.3.2
- [Bickmore and Cassell, 2001] Bickmore, T. and Cassell, J. (2001). Relational agents: A model and implementation of building user trust. In *Proc. of CHI'2001: Conference on Human Factors in Computing Systems*, London. ACM. 3.1
- [Bioware, 2003] Bioware (2003). Starwars: Knights of the old republic. Available at <http://lucasarts.com/products/swkotor>. 3.1
- [Blum and Furst, 1997] Blum, A. and Furst, M. (1997). Fast planning through planning graph analysis. *Artificial Intelligence*, 90:281–300. 2.4.1, 5.3.3
- [Bonet and Geffner, 2001] Bonet, B. and Geffner, H. (2001). Planning as heuristic search. *Artificial Intelligence*, 129(1–2):5–33. 2.4.1, 2.4.1, 5.3.3, 5.3.3
- [Bonet et al., 1997] Bonet, B., Loerincs, G., and Geffner, H. (1997). A fast and robust action selection mechanism for planning. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*. AAAI Press. 5.3
- [Bordini et al., 2005] Bordini, R. H., da Rocha, A. C., Hübner, J. F., Moreira, A. F., Okuyama, F. Y., and Vieira, R. (2005). MAS-SOC: a social simulation platform based

- on agent-oriented programming. *Journal of Artificial Societies and Social Simulation*, 8(3). 5.4
- [Bordini and Hübner, 2007] Bordini, R. H. and Hübner, J. F. (2007). Jason. Available at <http://jason.sourceforge.net/>. 2.4.2, 3.2, 4.4.1, 6.4
- [Bouron and Collinot, 1992] Bouron, T. and Collinot, A. (1992). SAM: a model to design computational social agents. In *Proc. of ECAI'92: European Conference on AI*, pages 239–243. 5.3.1
- [Boutilier and Brafman, 2001] Boutilier, C. and Brafman, R. I. (2001). Partial order planning with concurrent interacting actions. *Journal of Artificial Intelligence Research*, 14:105–136. 3.3.2
- [Brainov and Sandholm, 1999] Brainov, S. and Sandholm, T. (1999). Power, dependence and stability in multiagent plans. In *Proc. of AAAI '99/IAAI '99: 6th national conference on Artificial intelligence and 11th Innovative Applications of Artificial Intelligence conference*, pages 11–16, Menlo Park, CA, USA. American Association for Artificial Intelligence. 3.4.3
- [Bratman, 1987] Bratman, M. (1987). *Intention, Plans, and Practical Reason*. Harvard University Press. 2.4.2
- [Brenner, 2003] Brenner, M. (2003). A multi-agent planning language. In *Proc. of ICAPS'03 Workshop on PDDL*. 3.3.2
- [Caicedo and Thalmann, 2000] Caicedo, A. and Thalmann, D. (2000). Virtual humanoids: Let them be autonomous without losing control. In *Proc. of 3IA2000: 4th Conf. Computer Graphics and Artificial Intelligence*. (document), 2.4.2, 2.14
- [Calderón et al., 2003] Calderón, C., Cavazza, M., and Díaz, D. (2003). Interactive problem solving in an intelligent virtual environment. In *Proc. of IUI'03: International conference on Intelligent User Interfaces*, pages 319–319, New York, USA. ACM Press. 2.1
- [Cassell et al., 1994] Cassell, J., Pelachaud, C., Badler, N., Steedman, M., Achorn, B., Becket, T., Douville, B., Prevost, S., and Stone, M. (1994). Animated conversation: Rule-based generation of facial expression, gesture and spoken intonational for multiple conversational agents. In *Proc. of SIGGRAPH'94*. 3.1

- [Castelfranchi, 1998] Castelfranchi, C. (1998). Modeling social action for ai agents. *Artificial Intelligence*, 103:157–182. 3.4
- [Castelfranchi and Falcone, 1998] Castelfranchi, C. and Falcone, R. (1998). Principles of trust in MAS: cognitive anatomy, social importance and quantification. In *Proc. of ICMAS'98*, pages 72–79, Paris. 3.4.2
- [Castelfranchi et al., 1992] Castelfranchi, C., Miceli, M., and Cesta, A. (1992). Dependence relations among autonomous agents. In *Proc. of MAAMAW'92*, pages 215–227, Amsterdam. Elsevier Science Publishers. 3.4.1
- [Cavazza et al., 2002] Cavazza, M., Charles, F., and Mead, S. J. (2002). Character based interactive storytelling. *IEEE Intelligent systems*, 17(4):17–24. (document), 2.4.1, 2.10, 3.1, 4.4, 5.3.1
- [Chang et al., 2005] Chang, P. H.-M., Chien, Y.-H., Kao, E. C.-C., and Soo, V.-W. (2005). A knowledge-based scenario framework to support intelligent planning characters. In LNCS, S., editor, *Proc. of the 5th International Workshop of Intelligent Virtual Agents (IVA05)*, pages 134–145. (document), 2.2.2, 2.2
- [Chevalere et al., 2006] Chevalere, Y., Dunne, P. E., Endriss, U., Lang, J., Lemaitre, M., Maudet, N., Padget, J., Phelps, S., Rodriguez-Aguilar, J. A., and Sousa, P. (2006). Issues in multiagent resource allocation. *Informatica*, 30:3–31. 3.4.3, 3.4.3, 5.4.1, 5.4.3
- [Chevalere et al., 2004] Chevalere, Y., Endriss, U., Estivie, S., and Maudet, N. (2004). Welfare engineering in practice: On the variety of multiagent resource allocation problems. In *ESAW*, pages 335–347. 3
- [Ciger, 2005] Ciger, J. (2005). *Collaboration with agents in VR environments*. PhD thesis, École Polytechnique Fédérale de Lausanne. (document), 2.4.1, 3.1, 3.1
- [Cohen and Levesque, 1991] Cohen, P. and Levesque, H. (1991). Teamwork. *Nous, Special Issue on Cognitive Science and AI*. 3.3.3, 5.3.1
- [Conte, 1999] Conte, R. (1999). Artificial social intelligence: a necessity for agent systems' developments. *Knowledge Engineering Review*, 14:109–118. 3.3.3, 3.4.1
- [Conte and Castelfranchi, 1995] Conte, R. and Castelfranchi, C. (1995). *Cognitive and Social Action*. UCL Press, London. 3.3.3

- [Costa de Paiva et al., 2005] Costa de Paiva, D., Vieira, R., and Musse, S. R. (2005). Ontology-based crowd simulation for normal life situations. In *Proceedings of Computer Graphics*. IEEE. (document), 2.2.1, 2.1
- [Cox and Durfee, 2005] Cox, J. S. and Durfee, E. H. (2005). An efficient algorithm for multiagent plan coordination. In *Procs. of AAMAS'05: Autonomous Agents and Multi-agent Systems*, pages 828–827. 3.3.2
- [de Weerd, 2003] de Weerd, M. M. (2003). Plan coordination. In *Proceedings of the Doctoral Consortium of the International Conference on AI Planning and Scheduling*, pages 142–145. 3.3.2
- [Decker, 1998] Decker, K. S. (1998). *Simulating Organizations: Computational Models of Institutions and Groups*, chapter Task environment centered simulation, pages 105–128. AAAI Press / MIT Press, Menlo Park. 2.2.3
- [Decker and Lesser, 1997] Decker, K. S. and Lesser, V. R. (1997). *Readings in Agents*, chapter Designing a family of coordination algorithms. Elsevier. 3.3.2
- [Delgado-Mata and Aylett, 2004] Delgado-Mata, C. and Aylett, R. (2004). Emotion and action selection: Regulating the collective behaviour of agents in virtual environments. In *AAMAS*, pages 1304–1305. 3.1
- [D’Inverno and Luck, 2004] D’Inverno, M. and Luck, M. (2004). *Understanding Agent Systems*. Springer-Verlag, 2nd edition. 5.2
- [Distributed Systems and Information Systems Group - University of Hamburg, 2008] Distributed Systems and Information Systems Group - University of Hamburg (2008). JADEx: BDI agent system. Available at <http://vsis-www.informatik.uni-hamburg.de/projects/jadex/>. 2.4.2
- [Doyle, 2002] Doyle, P. (2002). Believability through context using "knowledge in the world" to create intelligent characters. In Press, A., editor, *First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS02)*, pages 342–349. 2.2.2
- [Durfee, 1999] Durfee, E. H. (1999). *A modern approach to distributed artificial intelligence: Distributed Problem Solving and Planning*, chapter 3. The MIT Press. 3.3.2
- [Elliott et al., 1999] Elliott, C., Rickel, J., and Lester, J. (1999). Lifelike pedagogical agents and affective computing: An exploratory synthesis. *Lecture Notes in Computer Science*, 1600. (document), 2.1, 2.4.2, 2.13

- [Ephrati and Rosenschein, 1994] Ephrati, E. and Rosenschein, J. S. (1994). Divide and conquer in multi-agent planning. In *Procs. of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pages 375–380, Menlo Park, CA. AAAI Press. 3.3.2
- [Epic games, 2008] Epic games (2008). Unreal tournament. Available at <http://www.unreal.com>. 2.2, 4.2, 6.3
- [Falcone et al., 2004] Falcone, R., Pezzulo, G., Castelfranchi, C., and Calvi, G. (2004). Why a cognitive trustier performs better: Simulating trust-based contract nets. In *Proc. of AAMAS'04: Autonomous Agents and Multi-Agent Systems*, pages 1392–1393. ACM. (document), 3.4.2, 3.7
- [Farenc et al., 1999] Farenc, N., Boulic, R., and Thalmann, D. (1999). An informed environment dedicated to the simulation of virtual humans in urban context. In Brunet, P. and Scopigno, R., editors, *Proc. of EUROGRAPHICS'99*, volume 18(3), pages 309–318. The Eurographics Association and Blackwell Publishers. 2.2.1
- [Ferber, 1999] Ferber, J. (1999). *Multi-Agent Systems: An introduction to Distributed Artificial Intelligence*. Addison-Wesley Publishing Company, Longman. (document), 3.2, 3.2, 3.4.3, 5.1, 5.2.1, 6.3.1
- [Ferber and Gutknecht, 1998] Ferber, J. and Gutknecht, O. (1998). A meta-model for the analysis and design of organizations in multi-agents systems. In *Proc. of the 3rd International Conference on Multi-Agent Systems (ICMAS'98)*, pages 128–135. IEEE Press. 2.2.3
- [Fikes and Nilsson, 1971] Fikes, R. and Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208. 2.4.1
- [Finin et al., 1994] Finin, T., Fritzson, R., McKay, D., and McEntire, R. (1994). KQML as an agent communication language. In *Proc. of CIKM'94*. ACM Press. 3.3.2, 4.2
- [FIPA (Foundation for Intelligent Physical Agents), 2008] FIPA (Foundation for Intelligent Physical Agents) (2008). FIPA ACL specifications. Available at <http://www.fipa.org>. 3.3.2, 4.2, 4.4.2
- [Fishburn, 1970] Fishburn, P. (1970). *Utility Theory for Decision Making*. Robert E. Krieger Publishing Co., Huntington, NY. 5.4.3

- [Fox et al., 1998] Fox, M. S., Barbuceanu, M., Gruninger, M., and Lon, J. (1998). *Simulating Organizations: Computational Models of Institutions and Groups*, chapter An organizational ontology for enterprise modeling., pages 131–152. AAAI Press / MIT Press, Menlo Park. 2.2.3
- [Funge et al., 1999] Funge, J., Tu, X., and Terzopoulos, D. (1999). Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In *Proc. of SIGGRAPH'99*, pages 29–38. ACM. (document), 2.3, 2.4, 2.4.1, 2.11
- [Gardner, 1970] Gardner, M. (1970). MATHEMATICAL GAMES: The fantastic combinations of john conway's new solitaire game "life". *Scientific American*, 223:120–123. 3.2
- [Geib et al., 1994] Geib, C., Levison, L., and Moore, M. (1994). Sodajack: An architecture for agents that search and manipulate objects. Technical Report MS-CIS-94-16 / LINC LAB265, Dpt. Computer and Information Science, University of Pennsylvania. (document), 2.4.1, 2.9
- [Georgeff, 1983] Georgeff, M. P. (1983). Communication and interaction in multi-agent planning. In *Proceedings of the Third National Conference on Artificial Intelligence (AAAI-83)*, pages 125–129, Menlo Park, CA. AAAI Press. 3.3.2
- [Giampapa and Sycara, 2002] Giampapa, J. A. and Sycara, K. (2002). Team-oriented agent coordination in the RETSINA multi-agent system. Tech. report CMU-RI-TR-02-34, Robotics Institute-Carnegie Mellon University. 3.3.3, 5.3.1
- [Gibson, 1984] Gibson, W. (1984). *Neuromancer*. Ace Books. 1.1
- [Gil, 2005] Gil, Y. (2005). Description logics and planning. *AI Magazine*, 26(2):73–84. 2.2.2
- [Giménez-Lugo et al., 2005] Giménez-Lugo, G., Sichman, J., and Hübner, J. (2005). Addressing the social components of knowledge to foster communitary exchanges. *International Journal on Web Based Communities*, 1(2):176–194. 2.2.3
- [Grosz et al., 1999] Grosz, B., Hunsberger, L., and Kraus, S. (1999). Planning and acting together. *AI Magazine*, 20(4):23–34. 3.3.3, 5.3.1
- [Gutierrez, 2006] Gutierrez, M. (2006). *Semantic Virtual Environments*. PhD thesis, École Polytechnique Fédérale de Lausanne. 2.2.1

- [Haslum and Geffner, 2000] Haslum, P. and Geffner, H. (2000). Admissible heuristics for optimal planning. *Artificial Intelligence Planning Systems*, pages 140–149. 5.3.3
- [Hayes-Roth, 2008] Hayes-Roth, B. (2008). Barbara hayes-roth website. Available at <http://www.ksl.stanford.edu/people/bhr/>. 3.1
- [Helbing et al., 2005] Helbing, D., Buzna, L., Johansson, A., and Werner, T. (2005). Self-organized pedestrian crowd dynamics. *Transportation Science*, 39(1):1–24. 3.1
- [Hexmoor, 2001] Hexmoor, H. (2001). From inter-agents to groups. In *Proc. of ISAI'01: International Symposium on Artificial Intelligence*. 3.3.3
- [Hoffmann and Nebel, 2001] Hoffmann, J. and Nebel, B. (2001). The FF planning system: Fast plan generation through heuristic search. *Artificial Intelligence Research*, 14:253–302. 2.4.1, 5.3.3, 5.3.3
- [Hogg and Jennings, 2001] Hogg, L. M. and Jennings, N. (2001). Socially intelligent reasoning for autonomous agents. *IEEE Transactions on System Man and Cybernetics*, 31(5):381–393. 3.1
- [Iglesias and Luengo, 2004] Iglesias, A. and Luengo, F. (2004). Intelligent agents in virtual worlds. In *Proc. of CW04: International Conference on Cyberworlds*, pages 62–69. IEEE Computer Society. 2.3
- [Ioeger and Johnson, 2001] Ioeger, T. and Johnson, J. (2001). A formal model of responsibilities in agent-based teamwork. *Applied Artificial Intelligence*, 15(10):875–916. 3.3.3
- [JAVA3D, 2008] JAVA3D (2008). JAVA 3D API. Available at <http://java.sun.com/products/java-media/3D/>. 2.2
- [Jena, 2008] Jena (2008). A semantic web framework for java. Available at <http://jena.sourceforge.net/>. 4.3
- [Jennings and Campos, 1997] Jennings, N. and Campos, J. (1997). Towards a social level characterisation of socially responsible agents. In *Proc. of Software Engineering*, volume 144, pages 11–25. IEEE. 3.4
- [Joint US/EU ad hoc Agent Markup Language Committee, 2008] Joint US/EU ad hoc Agent Markup Language Committee (2008). DAM+OIL 2001 release. Available at <http://www.daml.org/2001/03/daml+oil-index.html>. 2.2.1, 4.3

- [Kallmann and Thalmann, 1999] Kallmann, M. and Thalmann, D. (1999). Direct 3D interaction with smart objects. In Press, A., editor, *VRST '99: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 124–130. 2.2.1
- [Kambhampati et al., 1991] Kambhampati, S., Tanenbaum, M., and Lee, S. (1991). Combining specialized reasoners and general purpose planners. In *Proceedings of the 9th National Conference on Artificial Intelligence*. 3.3.2
- [Kao et al., 2005] Kao, E. C.-C., H-M., C. P., Chien, Y.-H., and Soo, V.-W. (2005). Using ontology to establish social context and support social reasoning. In *Proc. of IVA'05: International Conference on Intelligent Virtual Agents*. Springer-Verlag LNAI. (document), 2.3, 2.2.3
- [Kim, 2003] Kim, I.-C. (2003). KGBot: A BDI agent deploying within a complex 3D virtual environment. In T. Rist, R. Aylett, D. B. and Rickel, J., editors, *Intelligent Virtual Agents*, pages 192–196. Springer-Verlag LNAI. 2.4.2
- [Kuffner, 1999] Kuffner, J. (1999). *Autonomous Agents for Real-time Animation*. PhD thesis, Stanford University, Stanford, CA. 4.4
- [Kuffner and Latombe, 1999] Kuffner, J. and Latombe, J. (1999). Perception-based navigation for animated characters in real-time virtual environments. *The Visual Computer: Real-Time Virtual Worlds*, 3 (4). 4.4
- [Laird et al., 1987] Laird, J., Newel, A., and Rosenbloom, P. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence*, pages 1–64. 2.3.2, 2.4.2
- [Laird and Duchi, 2000] Laird, J. E. and Duchi, J. C. (2000). Creating human-like synthetic characters with multiple skill levels: A case study using the soar quakebot. *AAAI Fall Symposium Series: Simulating Human Agents*. 2.1, 2.4.2
- [Lang, 2005] Lang, J. (2005). Some representation and computational issues in social choice. In *Proc. of . ECSQARU'05*, pages 15–26. 3.4.3
- [Levison, 1996] Levison, L. (1996). *Connecting planning and acting via object-specific reasoning*. PhD thesis, University of Pennsylvania, PA, USA. 2.2.2, 4.4.1
- [Lewis and Jacobson, 2002] Lewis, M. and Jacobson, J. (2002). Introduction. *SPECIAL ISSUE: Game engines in scientific research. Commun. ACM*, 45(1):27–31. 2.2
- [Linden Lab, 2008] Linden Lab (2008). Second life. Available at <http://secondlife.com/>. 2.3.1, 3.1

- [Lozano, 2005] Lozano, M. (2005). *Animación comportamental de personajes inteligentes 3D basada en miniMin-HSP (Heuristic Search Planning)*. PhD thesis, Escola Tècnica Superior d'Enginyeria. Universitat de València. (document), 1.1, 2.4.1, 2.12, 4.4, 4.4.1, 5.3, 5.3.2, 5.3.3, 6.2
- [Lozano and Calderón, 2004] Lozano, M. and Calderón, C. (2004). Entornos virtuales 3D clásicos e inteligentes: hacia un nuevo marco de simulación para aplicaciones gráficas 3D interactivas. *Revista Ibero-Americana de Inteligencia Artificial*, 23. 4.2
- [Luck and Aylett, 2000] Luck, M. and Aylett, R. (2000). Applying artificial intelligence to virtual reality: Intelligent virtual environments. *Applied Artificial Intelligence*, 14(1):3–32. 2.1
- [Malone and Crowton, 1994] Malone, T. and Crowton, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119. 5.2
- [Mamei et al., 2004] Mamei, M., Zambonelli, F., and Leonardi, L. (2004). Co-fields: A physically inspired approach to motion coordination. *IEEE Pervasive Computing*, 3(2):52–61. 4.4
- [Marsh, 1994] Marsh, S. (1994). *Formalizing Trust As a Computational Concept*. PhD thesis, University of Stirling, U.K. 3.4.2
- [Martin et al., 1999] Martin, D., Cheyer, A., and Moran, D. (1999). The open agent architecture: A framework for building distributed software systems. *Applied Artificial Intelligence*, 13(1/2):91–128. 3.3.3
- [Meneguzzi et al., 2004] Meneguzzi, F., Zorzo, A., and Da Costa Mi $\frac{1}{2}$ a, M. (2004). Propositional planning in BDI agents. In *SAC'04: Proceedings of the 2004 ACM Symposium on Applied Computing*, pages 58–63. ACM Press. 2.4.2
- [Molet et al., 1997] Molet, M., Aubel, A., Capin, T., Carion, S., Lee, E., Thalmann, N., Noser, H., and Thalmann, D. (1997). Anyone for tennis? *Presence*, 8(2):140–156. 2.3.2
- [Molyneux, 2001] Molyneux, P. (2001). *Black&white*. Postmortem: Lionhead studio. 2.4.2
- [Nebel et al., 1997] Nebel, B., Dimopoulos, Y., and Koehler, J. (1997). Irrelevant facts and operators in plan generation. Technical report, Institut für Informatik Freiburg. 2.4.1

- [Niewiadomski and Pelachaud, 2007] Niewiadomski, R. and Pelachaud, C. (2007). Fuzzy similarity of facial expressions of embodied agents. In Pelachaud, C., Martin, J.-C., André, E., Chollet, G., Karpouzis, K., and Pelé, D., editors, *Proc. of IVA'07: Intelligent Virtual Agents*, number 4722, pages 86–98, Paris. Springer-Verlag LNAI. 2.3.1
- [Nilsson, 2001] Nilsson, N. (2001). *Inteligencia Artificial, una nueva síntesis*. McGraw-Hill. 2, 5.2.2
- [Noma and Badler, 1997] Noma, T. and Badler, N. (1997). A virtual human presenter. In *Proc. of the IJCAI'97. Workshop on Animated Interface Agents*, Nagoya, Japan. 2.3.1
- [Okuyama et al., 2005] Okuyama, F. Y., Bordini, R. H., and da Rocha, A. C. (2005). ELMS: An environment description language for multi-agent simulations. In Weyns, D., van Dyke Parunak, H., and Michel, F., editors, *Proc. of the 1st International Workshop on Environments for Multiagent Systems (E4MAS), held with AAMAS'04*, number 3374, pages 67–83. Springer-Verlag LNAI. 2.2.2
- [Omicini and Zambonelli, 1999] Omicini, A. and Zambonelli, F. (1999). Tuple centres for the coordination of internet agents. In *Proc. of the ACM Symposium on Applied Computing*, pages 183–190. ACM Press. 4.3.2
- [Osborne, 2004] Osborne, M. (2004). *An Introduction to Game Theory*. Oxford University Press. 3.4.3
- [OSG, 2008] OSG (2008). Openscenegraph website. Available at <http://www.openscenegraph.org/projects/osg>. 2.2, 4.2, 6.4
- [Otto, 2005] Otto, K. A. (2005). Towards semantic virtual environments. In *Workshop Towards Semantic Virtual Environments (SVE'05)*. 2.2.2
- [Pandzic et al., 1998] Pandzic, I., Capin, T., Lee, E., Magnenat-Thalmann, N., and Thalmann, D. (1998). Autonomous actors in networked collaborative virtual environments. In *Proc. of MultiMedia Modeling'98*. IEEE Computer Society Press. 3.1
- [Parunak et al., 2002] Parunak, H. V. D., Fleishcher, M., Brueckner, S., and Odell, J. (2002). Co-x: Defining what agents do together. In *Workshop on Teamwork and Coalition Formation, AAMAS*. 3.3
- [Pelachaud, 2005] Pelachaud, C. (2005). Multimodal expressive embodied conversational agents. In *Proc. of MULTIMEDIA '05: The 13th annual ACM international*

- conference on Multimedia*, pages 683–689, New York, NY, USA. ACM. (document), 2.1, 2.3.1, 2.6
- [Pelechano et al., 2007] Pelechano, N., Allbeck, J., and Badler, N. (2007). Controlling individual agents in high-density crowd simulation. In Metaxas, D. and Popovic, J., editors, *Proc. of SCA'07: Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*. ACM. 3.1, 4.4, 6.1
- [Pellens et al., 2005] Pellens, B., Bille, W., De Troyer, O., and Kleinermann, F. (2005). VR-wise: A conceptual modelling approach for virtual environments. In *Methods and Tools for Virtual Reality (MeTo-VR 2005) workshop*. 2.2.1
- [Perlin and Goldberg, 1996] Perlin, K. and Goldberg, A. (1996). IMPROV: a system for scripting interactive agents in virtual worlds. In *Proc. of SIGGRAPH'96*, pages 205–216. ACM. 2.3.2
- [Piaget, 1995] Piaget, J. (1995). *Sociological studies*. Routledge. 5.4
- [Prada and Paiva, 2005] Prada, R. and Paiva, A. (2005). Believable groups of synthetic characters. In *Proc. of AAMAS '05: Autonomous Agents and Multi-Agent Systems*, pages 37–43, New York, NY, USA. ACM. (document), 3.1, 3.1, 6.1
- [Protégé Wiki, 2008] Protégé Wiki (2008). Protégé ontology library. Available at http://protegewiki.stanford.edu/index.php/Protege_Ontology_Library. 4.3
- [Racer Systems, 2008] Racer Systems (2008). RACER: Renamed abox and concept expression reasoner. Available at <http://www.sts.tu-harburg.de/~r.f.moeller/racer/>. 4.3
- [Rao, 1996] Rao, A. S. (1996). AgentSpeak(L): BDI agents speak out in a logical computable language. In Verlag, S., editor, *Proc. of MAAMAW'96*, number 1038 in LNAI, pages 42–55. 4.4.1, 6.4.2
- [Raupp and Thalmann, 2001] Raupp, S. and Thalmann, D. (2001). Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):152–164. 2.4.2
- [Reilly, 1996] Reilly, W. S. N. (1996). *Believable Social and Emotional Agents*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. 3.1

- [Reynolds, 1987] Reynolds, C. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proc. of SIGGRAPH'87*, pages 25–34. ACM. (document), 2.3.2, 2.8, 3.1
- [Reynolds, 1999] Reynolds, C. (1999). Steering behaviors for autonomous characters. In *Proc. of Games Developer Conference*, pages 763–782. 2.3.2, 4.4
- [Ribeiro et al., 2003] Ribeiro, M., da Rocha, A. C., and Bordini, R. H. (2003). A system of exchange values to support social interactions in artificial societies. In *Proc. of AAMAS'03: Autonomous Agents and Multi-agent Systems*. ACM. 5.4
- [Rodríguez-García, 2004] Rodríguez-García, R. (2004). *Actores sintéticos en tiempo real: Nuevas estructuras de datos y métodos para su integración en aplicaciones de simulación*. PhD thesis, Escola Tècnica Superior d'Enginyeria. Universitat de València. 2.2
- [Sabater and Sierra, 2002] Sabater, J. and Sierra, C. (2002). Reputation and social network analysis. In *Proc. of AAMAS'02: International Conference on Autonomous Agents and Multi-Agent Systems*, pages 475–482. 3.4.2
- [Sandholm, 1998] Sandholm, T. (1998). Contract types for satisficing task allocation: I theoretical results. In *Proc. of AAAI Spring Symposium: Satisficing Models*. 3.4.3
- [Sandholm and Suri, 2001] Sandholm, T. and Suri, S. (2001). Side constraints and non-price attributes in markets. In *Proc. of IJCAI Workshop on Distributed Constraint Reasoning*. 3.4.3, 5.4.4
- [Schmitt and Rist, 2003] Schmitt, M. and Rist, T. (2003). Avatar arena: Virtual group-dynamics in multicharacter negotiation scenarios. In *Procs. of IVA'03: International Conference on Intelligent Virtual Agents*. Springer LNAI. 3.1
- [Scott, 1991] Scott, J. (1991). *Social network analysis: A handbook*. Sage publications, London. 3.4
- [Shao and Terzopoulos, 2005] Shao, W. and Terzopoulos, D. (2005). Autonomous pedestrians. In K. Anjyo, P. F., editor, *Proc. of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*. 4.4
- [Shoham and Tennenholtz, 1995] Shoham, Y. and Tennenholtz, M. (1995). On social laws for artificial agent societies: Off-line design. *Artificial Intelligence*, 73(1–2):231–252. 3.3.2, 3.4, 5.2.2

- [Sichman et al., 1994] Sichman, J., Conte, R., Castelfranchi, C., and Demazeau, Y. (1994). A social reasoning mechanism based on dependence networks. In Cohn, A., editor, *Proc. of ECAI'94: European Conference on Artificial Intelligence*, pages 188–192. Wiley. 3.4.1
- [Sichman and Demazeau, 2001] Sichman, J. and Demazeau, Y. (2001). On social reasoning in multi-agent systems. *Revista Ibero-Americana de Inteligencia Artificial*, 13:68–84. 3.4.1
- [Soto and Allongue, 2002] Soto, M. and Allongue, S. (2002). Modeling methods for reusable and interoperable virtual entities in multimedia virtual worlds. *Multimedia Tools and Applications*, 16:161–177. 2.2.2
- [Stanford Medical Informatics, 2006] Stanford Medical Informatics (2006). Protégé. Available at <http://protege.stanford.edu/>. 4.3, 1
- [Stanney, 2002] Stanney, K., editor (2002). *Handbook of Virtual Environments*. Lawrence Erlbaum Associates. 2.1
- [Stephenson, 1992] Stephenson, N. (1992). *Snow crash*. Bantam Books. 1.1
- [Sycara, 1998] Sycara, K. (1998). Multiagent systems. *AI Magazine*, 10(2):79–93. 3.2
- [Tambe, 1997] Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124. (document), 3.3.3, 3.5
- [Thalmann et al., 1998] Thalmann, N., Kalra, P., and Escher, M. (1998). Face to virtual face. In *Proceedings of the IEEE*, volume 86, pages 870–883. 2.3.1
- [Thomas and Donikian, 2000] Thomas, G. and Donikian, S. (2000). Virtual humans animation in informed urban environments. In *Computer Animation 2000*. 2.2.1
- [Tomlinson and Blumberg, 2002] Tomlinson, B. and Blumberg, B. (2002). Social synthetic characters. In *Proc. of Computer Graphics*, volume 26. 3.1
- [Torres et al., 2003] Torres, J. A., Nedel, L. P., and Bordini, R. H. (2003). Using the BDI architecture to produce autonomous characters in virtual worlds. In *Procs. of IVA'03: International Conference on Intelligent Virtual Agents*. 3.2
- [Tu and Terzopoulos, 1994] Tu, X. and Terzopoulos, D. (1994). Artificial fishes: physics, locomotion, perception, behavior. In *Proc. of SIGGRAPH'94*, pages 43–50. ACM. 2.3, 2.4.1

- [van Kokswijk, 2007] van Kokswijk, J. (2007). *Digital Ego: Social and Legal Aspects of Virtual Identity*. 1.1, 2.3.1
- [Vercouter et al., 2007] Vercouter, L., Casare, S. J., Sichman, J. S., and Brandao, A. A. F. (2007). An experience on reputation models interoperability based on a functional ontology. In *Proc. of IJCAI07: International Joint Conference on Artificial Intelligence*, Hyderabad, India. 3.4.2
- [Viroli et al., 2006] Viroli, M., Ricci, A., and Omicini, A. (2006). Operating instructions for intelligent agent coordination. *Knowledge Engineering Review*, 21:49–69. 2.2.3
- [Vosinakis and Panayiotopoulos, 2001] Vosinakis, S. and Panayiotopoulos, T. (2001). Simhuman: A platform for real time virtual agents planning capabilities. In *Proc. of IVA'01: International Conference on Intelligent Virtual Agents*. Springer-Verlag LNAI. 3.2
- [Vosinakis and Panayotopoulos, 2003] Vosinakis, S. and Panayotopoulos, T. (2003). A task definition language for virtual agents. *Journal of WSCG*, 11:512–519. 2.2.2
- [W3C, 2004] W3C (2004). OWL web ontology language guide. Available at <http://www.w3.org/TR/owl-guide/>. 2.2.3, 4.3
- [Wooldridge and Jennings, 1994] Wooldridge, M. and Jennings, N. (1994). Towards a theory of cooperative problem solving. In *Proc. of MAAMAW094*, pages 15–26. 3.3.3
- [Wooldridge and Jennings, 1995] Wooldridge, M. J. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152. 1, 5.2.2
- [Yang, 1997] Yang, Q. (1997). *Intelligent Planning*. Springer. 3.3.2
- [Young, 2001] Young, R. (2001). An overview of the mimesis architecture integrating intelligent narrative control into an existing gaming environment. Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, AAAI Press (2001). 2.4.1

PART IV

ANNEXOS

ONTOLOGIES DE L'ENTORN VIRTUAL SEMÀNTIC

Aquest annex arreplega l'ontologia base de l'entorn virtual semàntic, un exemple d'ontologia específica del domini que defineix alguns objectes necessaris en un bar virtual i una possible instanciació d'aquest bar.

A.1 SVE CORE ONTOLOGY

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns="http://www.uv.es/~agentes/SVECore#"
  xml:base="http://www.uv.es/~agentes/SVECore">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="InterGroupSocialRelation">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="SocialRelation"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="MovableObject">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="GraphicalEntity"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Group"/>
  <owl:Class rdf:about="#GraphicalEntity">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:cardinality
          rdf:datatype="http://www.w3.org/2001/XMLSchema#int"

```

```

    >1</owl:cardinality>
    <owl:onProperty>
      <owl:FunctionalProperty rdf:ID="posx"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:cardinality
      rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</owl:cardinality>
    <owl:onProperty>
      <owl:FunctionalProperty rdf:ID="ry"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty>
      <owl:FunctionalProperty rdf:ID="height"/>
    </owl:onProperty>
    <owl:cardinality
      rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</owl:cardinality>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:cardinality
      rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</owl:cardinality>
    <owl:onProperty>
      <owl:FunctionalProperty rdf:ID="objectFree"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:cardinality
      rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</owl:cardinality>
    <owl:onProperty>
      <owl:FunctionalProperty rdf:ID="posy"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty>
      <owl:FunctionalProperty rdf:ID="rx"/>
    </owl:onProperty>
    <owl:cardinality

```

```

        rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</owl:cardinality>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
    <owl:Restriction>
        <owl:onProperty>
            <owl:FunctionalProperty rdf:ID="posz"/>
        </owl:onProperty>
        <owl:cardinality
            rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</owl:cardinality>
        </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
    <owl:Restriction>
        <owl:cardinality
            rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</owl:cardinality>
        <owl:onProperty>
            <owl:FunctionalProperty rdf:ID="yaw"/>
        </owl:onProperty>
    </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Agent">
    <rdfs:subClassOf>
        <owl:Class rdf:ID="FilterClass"/>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#GraphicalEntity"/>
</owl:Class>
<owl:Class rdf:ID="CountableContainer">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:cardinality
                rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
                >1</owl:cardinality>
            <owl:onProperty>
                <owl:FunctionalProperty rdf:ID="num"/>
            </owl:onProperty>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Class rdf:ID="Container"/>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:cardinality
                rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
                >1</owl:cardinality>
            <owl:onProperty>
                <owl:FunctionalProperty rdf:ID="max"/>
            </owl:onProperty>
        </owl:Restriction>

```

```

    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ObjectContainer">
    <rdfs:subClassOf rdf:resource="#CountableContainer"/>
  </owl:Class>
  <owl:Class rdf:ID="ServiceContainer">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:FunctionalProperty rdf:about="#num"/>
        </owl:onProperty>
        <owl:cardinality
          rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
          >1</owl:cardinality>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf rdf:resource="#CountableContainer"/>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:cardinality
            rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</owl:cardinality>
          <owl:onProperty>
            <owl:ObjectProperty rdf:ID="containsSubstance"/>
          </owl:onProperty>
        </owl:Restriction>
      </rdfs:subClassOf>
    </owl:Class>
  <owl:Class rdf:ID="FriendshipRelation">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:DatatypeProperty rdf:ID="favoursReceived"/>
        </owl:onProperty>
        <owl:maxCardinality
          rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
          >1</owl:maxCardinality>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty>
            <owl:DatatypeProperty rdf:ID="favoursDone"/>
          </owl:onProperty>
          <owl:maxCardinality
            rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</owl:maxCardinality>
          </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
          <owl:Class rdf:ID="AgentSocialRelation"/>
        </rdfs:subClassOf>
      </owl:Class>
    <owl:Class rdf:ID="HeterogeneousObjectContainer">

```

```

    <rdfs:subClassOf rdf:resource="#ObjectContainer"/>
  </owl:Class>
  <owl:Class rdf:about="#Container">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:FunctionalProperty rdf:ID="full"/>
        </owl:onProperty>
        <owl:maxCardinality
          rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
          >1</owl:maxCardinality>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf rdf:resource="#GraphicalEntity"/>
      <rdfs:subClassOf rdf:resource="#FilterClass"/>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:maxCardinality
            rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</owl:maxCardinality>
          <owl:onProperty>
            <owl:FunctionalProperty rdf:ID="empty"/>
          </owl:onProperty>
        </owl:Restriction>
      </rdfs:subClassOf>
    </owl:Class>
    <owl:Class rdf:ID="HomogeneousObjectContainer">
      <rdfs:subClassOf rdf:resource="#ObjectContainer"/>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty>
            <owl:FunctionalProperty rdf:ID="currBaseObjects"/>
          </owl:onProperty>
          <owl:cardinality
            rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</owl:cardinality>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:cardinality
            rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</owl:cardinality>
          <owl:onProperty>
            <owl:FunctionalProperty rdf:ID="objectBaseName"/>
          </owl:onProperty>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:cardinality
            rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</owl:cardinality>
          <owl:onProperty>

```

```

        <owl:FunctionalProperty rdf:ID="numBaseObjects"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="BaseObject">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality
        rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:ID="baseFree"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#FilterClass"/>
  <rdfs:subClassOf rdf:resource="#GraphicalEntity"/>
</owl:Class>
<owl:Class rdf:ID="Substance">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="NonGraphicalEntity"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="GroupSocialRelation">
  <rdfs:subClassOf rdf:resource="#SocialRelation"/>
</owl:Class>
<owl:Class rdf:ID="UncountableContainer">
  <rdfs:subClassOf rdf:resource="#Container"/>
</owl:Class>
<owl:Class rdf:about="#AgentSocialRelation">
  <rdfs:subClassOf rdf:resource="#SocialRelation"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="domainAgentSocialRelation">
  <rdfs:range rdf:resource="#Agent"/>
  <rdfs:domain rdf:resource="#AgentSocialRelation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="targetAgentSocialRelation">
  <rdfs:domain rdf:resource="#AgentSocialRelation"/>
  <rdfs:range rdf:resource="#Agent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="pickedBy">
  <rdfs:domain rdf:resource="#MovableObject"/>
  <rdfs:range rdf:resource="#Agent"/>
  <rdfs:subPropertyOf>
    <owl:FunctionalProperty rdf:ID="senseDepends"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="domainGroupSocialRelation">
  <rdfs:domain rdf:resource="#Agent"/>
  <rdfs:range rdf:resource="#GroupSocialRelation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="domainInterGroupSocialRelation">
  <rdfs:range rdf:resource="#InterGroupSocialRelation"/>

```



```

    <rdfs:domain rdf:resource="#Group"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="in">
    <rdfs:subPropertyOf>
      <owl:FunctionalProperty rdf:about="#senseDepends"/>
    </rdfs:subPropertyOf>
    <rdfs:domain rdf:resource="#MovableObject"/>
    <rdfs:range rdf:resource="#Container"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#containsSubstance">
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#ServiceContainer"/>
          <owl:Class rdf:about="#UncountableContainer"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <rdfs:range rdf:resource="#Substance"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="on">
    <rdfs:subPropertyOf>
      <owl:FunctionalProperty rdf:about="#senseDepends"/>
    </rdfs:subPropertyOf>
    <rdfs:range rdf:resource="#BaseObject"/>
    <rdfs:domain rdf:resource="#MovableObject"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="targetInterGroupSocialRelation">
    <rdfs:range rdf:resource="#InterGroupSocialRelation"/>
    <rdfs:domain rdf:resource="#Group"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="targetGroupSocialRelation">
    <rdfs:domain rdf:resource="#Group"/>
    <rdfs:range rdf:resource="#GroupSocialRelation"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:about="#favoursReceived">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:domain rdf:resource="#FriendshipRelation"/>
    <rdfs:subPropertyOf>
      <owl:DatatypeProperty rdf:ID="socialProperty"/>
    </rdfs:subPropertyOf>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="#socialProperty">
    <rdfs:domain rdf:resource="#SocialRelation"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="graphicalProperty"/>
  <owl:DatatypeProperty rdf:about="#favoursDone">
    <rdfs:subPropertyOf rdf:resource="#socialProperty"/>
    <rdfs:domain rdf:resource="#FriendshipRelation"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  </owl:DatatypeProperty>
  <owl:FunctionalProperty rdf:about="#objectBaseName">
    <rdfs:range
      rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>

```

```

    <rdfs:domain rdf:resource="#HomogeneousObjectContainer"/>
    <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:ID="filterType">
    <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:range>
      <owl:DataRange>
        <owl:oneOf rdf:parseType="Resource">
          <rdf:first
            rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >ALL</rdf:first>
          <rdf:rest rdf:parseType="Resource">
            <rdf:first
              rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
              >NOTHING</rdf:first>
            <rdf:rest rdf:parseType="Resource">
              <rdf:first
                rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                >CLASSES-ONLY</rdf:first>
              <rdf:rest
                rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
            </rdf:rest>
          </rdf:rest>
        </owl:oneOf>
      </owl:DataRange>
    </rdfs:range>
    <rdfs:domain rdf:resource="#FilterClass"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#full">
    <rdfs:domain rdf:resource="#Container"/>
    <rdfs:range
      rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
    <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#posz">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdfs:domain rdf:resource="#GraphicalEntity"/>
    <rdfs:subPropertyOf rdf:resource="#graphicalProperty"/>
    <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#posy">
    <rdfs:subPropertyOf rdf:resource="#graphicalProperty"/>
    <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain rdf:resource="#GraphicalEntity"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#numBaseObjects">
    <rdfs:domain rdf:resource="#HomogeneousObjectContainer"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>

```

```

    <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#objectFree">
    <rdfs:domain rdf:resource="#GraphicalEntity"/>
    <rdfs:range
      rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
    <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#yaw">
    <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:subPropertyOf rdf:resource="#graphicalProperty"/>
    <rdfs:domain rdf:resource="#GraphicalEntity"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#senseDepends">
    <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:range rdf:resource="#FilterClass"/>
    <rdfs:comment
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Sensorial dependency</rdfs:comment>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#baseFree">
    <rdfs:domain rdf:resource="#BaseObject"/>
    <rdfs:range
      rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
    <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#currBaseObjects">
    <rdfs:domain rdf:resource="#HomogeneousObjectContainer"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#ry">
    <rdfs:subPropertyOf rdf:resource="#graphicalProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdfs:domain rdf:resource="#GraphicalEntity"/>
    <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#rx">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdfs:domain rdf:resource="#GraphicalEntity"/>
    <rdfs:subPropertyOf rdf:resource="#graphicalProperty"/>
    <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:ID="closed">
    <rdfs:range

```

```

    rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:about="#max">
  <rdfs:domain rdf:resource="#CountableContainer"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:about="#num">
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#CountableContainer"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:comment
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Number of objects or services in the container</rdfs:comment>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:about="#empty">
  <rdfs:comment xml:lang="en">Number of objects or services in
    the container</rdfs:comment>
  <rdfs:domain rdf:resource="#Container"/>
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="typeObject">
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#HomogeneousObjectContainer"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:about="#posx">
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:subPropertyOf rdf:resource="#graphicalProperty"/>
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#GraphicalEntity"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:about="#height">
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#GraphicalEntity"/>
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:subPropertyOf rdf:resource="#graphicalProperty"/>
</owl:FunctionalProperty>
</rdf:RDF>

```

A.2 DOMAIN SPECIFIC ONTOLOGY: BAR VIRTUAL

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.uv.es/agentes/sve/DSOBar.owl#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:core="http://www.uv.es/~agentes/SVECore#"
  xml:base="http://www.uv.es/agentes/sve/DSOBar.owl">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://www.uv.es/~agentes/SVECore"/>
  </owl:Ontology>
  <rdfs:Class rdf:about="http://www.uv.es/~agentes/
  SVECore#HeterogeneousObjectContainer"/>
  <rdfs:Class
  rdf:about="http://www.uv.es/~agentes/
  SVECore#MovableObject"/>
  <rdfs:Class rdf:about="http://www.uv.es/~agentes/
  SVECore#HomogeneousObjectContainer"/>
  <rdfs:Class rdf:about="http://www.uv.es/~agentes/
  SVECore#BaseObject"/>
  <rdfs:Class rdf:about="http://www.uv.es/~agentes/
  SVECore#UncountableContainer"/>
  <rdfs:Class rdf:about="http://www.uv.es/~agentes/
  SVECore#ServiceContainer"/>
  <owl:Class rdf:ID="Cup">
    <rdfs:subClassOf rdf:resource="http://www.uv.es/~agentes/
    SVECore#UncountableContainer"/>
    <rdfs:subClassOf rdf:resource="http://www.uv.es/~agentes/
    SVECore#MovableObject"/>
  </owl:Class>
  <owl:Class rdf:ID="Dish">
    <rdfs:subClassOf rdf:resource="http://www.uv.es/~agentes/
    SVECore#MovableObject"/>
    <rdfs:subClassOf rdf:resource="http://www.uv.es/~agentes/
    SVECore#UncountableContainer"/>
  </owl:Class>
  <owl:Class rdf:ID="TrayOfGlasses">
    <rdfs:subClassOf rdf:resource="http://www.uv.es/~agentes/
    SVECore#HomogeneousObjectContainer"/>
  </owl:Class>
  <owl:Class rdf:ID="Tray">
    <rdfs:subClassOf rdf:resource="http://www.uv.es/~agentes/
    SVECore#BaseObject"/>
    <rdfs:subClassOf rdf:resource="http://www.uv.es/~agentes/
    SVECore#MovableObject"/>
    <rdfs:subClassOf rdf:resource="http://www.uv.es/~agentes/
    SVECore#HeterogeneousObjectContainer"/>
  </owl:Class>
  <owl:Class rdf:ID="Glass">
    <rdfs:subClassOf rdf:resource="http://www.uv.es/~agentes/

```

```

    SVECore#UncountableContainer"/>
    <rdfs:subClassOf rdf:resource="http://www.uv.es/~agentes/
    SVECore#MovableObject"/>
  </owl:Class>
  <owl:Class rdf:ID="Cabinet">
    <rdfs:subClassOf rdf:resource="http://www.uv.es/~agentes/
    SVECore#HeterogeneousObjectContainer"/>
  </owl:Class>
  <owl:Class rdf:ID="Bottle">
    <rdfs:subClassOf rdf:resource="http://www.uv.es/~agentes/
    SVECore#ServiceContainer"/>
    <rdfs:subClassOf rdf:resource="http://www.uv.es/~agentes/
    SVECore#MovableObject"/>
  </owl:Class>
</rdf:RDF>

```

A.3 INSTANCIACIÓ DEL MÓN: BAR VIRTUAL

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns="http://www.uv.es/~agentes/WorldBar#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dso="http://www.uv.es/~agentes/DSOBar#"
  xmlns:core="http://www.uv.es/~agentes/SVECore#"
  xml:base="http://www.uv.es/~agentes/WorldBar">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://www.uv.es/~agentes/DSOBar"/>
  </owl:Ontology>
  <dso:Cabinet rdf:ID="refrigerator">
    <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >50</core:max>
    <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >15</core:num>
    <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >-1.0</core:posx>
    <core:objectFree
    rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:objectFree>
    <core:filterType
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >CLASSES-ONLY</core:filterType>
    <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >-4.7</core:posy>
    <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.3</core:ry>

```

```
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>2.0</core:rx>
<core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:full>
<core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>1.0</core:height>
<core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.5</core:posz>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:empty>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
</dso:Cabinet>
<dso:Dish rdf:ID="dish_1">
<core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:ry>
<core:in>
  <dso:Cabinet rdf:ID="cabinet_1">
    <core:full
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
      >false</core:full>
    <core:max
      rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >20</core:max>
    <core:ry
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >0.5</core:ry>
    <core:filterType
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >CLASSES-ONLY</core:filterType>
    <core:posy
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >4.5</core:posy>
    <core:rx
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >1.5</core:rx>
    <core:posx
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >-2.0</core:posx>
    <core:height
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >1.0</core:height>
    <core:yaw
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >0.0</core:yaw>
    <core:posz
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >0.5</core:posz>
```

```

    <core:empty
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:empty>
    <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >9</core:num>
    <core:objectFree
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:objectFree>
  </dso:Cabinet>
</core:in>
<core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.01</core:height>
<core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
<core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:full>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:rx>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:objectFree>
<core:empty
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>true</core:empty>
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
</dso:Dish>
<dso:Dish rdf:ID="dish_4">
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posy>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posx>
  <core:filterType
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ALL</core:filterType>
  <core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:objectFree>
  <core:empty
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >true</core:empty>
  <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.2</core:rx>
  <core:in rdf:resource="#cabinet_1"/>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posz>
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:yaw>
  <core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.01</core:height>

```



```

    <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.2</core:ry>
    <core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:full>
</dso:Dish>
<dso:Tray rdf:ID="tray_1">
  <core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:objectFree>
  <core:filterType
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ALL</core:filterType>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posx>
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posy>
  <core:empty
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >true</core:empty>
  <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.25</core:rx>
  <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >5</core:num>
  <core:baseFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:baseFree>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posz>
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:yaw>
  <core:full
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:full>
  <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >5</core:max>
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.25</core:ry>
  <core:height
  rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.01</core:height>
  <core:in rdf:resource="#cabinet_1"/>
</dso:Tray>
<dso:Bottle rdf:ID="beerBottle_4">
  <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >1</core:num>
  <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >1</core:max>
  <core:height
  rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.2</core:height>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posz>
  <core:filterType
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```

```

>ALL</core:filterType>
<core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:full>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:rx>
<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:objectFree>
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:in rdf:resource="#refrigerator"/>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:empty rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:empty>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:ry>
<core:containsSubstance>
  <core:Substance rdf:ID="Beer"/>
</core:containsSubstance>
</dso:Bottle>
<core:ServiceContainer rdf:ID="coffeeMachine">
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.5</core:ry>
  <core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >true</core:full>
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posy>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posx>
  <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >100</core:max>
  <core:containsSubstance>
    <core:Substance rdf:ID="Coffee"/>
  </core:containsSubstance>
  <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.2</core:rx>
  <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >100</core:num>
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:yaw>
  <core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:objectFree>
  <core:filterType
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ALL</core:filterType>
  <core:empty rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:empty>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posz>
  <core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"

```

```

    >0.5</core:height>
  </core:ServiceContainer>
  <core:HomogeneousObjectContainer rdf:ID="shelf_A">
    <core:filterType
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ALL</core:filterType>
    <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.7</core:posz>
    <core:empty rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:empty>
    <core:typeObject
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Glass</core:typeObject>
    <core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</core:full>
    <core:objectBaseName
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Glass</core:objectBaseName>
    <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >1.0</core:posx>
    <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.5</core:ry>
    <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:yaw>
    <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >50</core:max>
    <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >50</core:num>
    <core:objectFree
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:objectFree>
    <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >4.5</core:posy>
    <core:numBaseObjects
      rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >3</core:numBaseObjects>
    <core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >1.4</core:height>
    <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >1.5</core:rx>
  </core:HomogeneousObjectContainer>
  <dso:Dish rdf:ID="dish_6">
    <core:empty rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</core:empty>
    <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.2</core:rx>
    <core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.01</core:height>
    <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posz>
    <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.2</core:ry>
    <core:objectFree
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"

```

```

>false</core:objectFree>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:full>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
<core:in rdf:resource="#cabinet_1"/>
</dso:Dish>
<dso:Dish rdf:ID="dish_3">
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
<core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:full>
<core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:objectFree>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:in rdf:resource="#cabinet_1"/>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:ry>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:empty>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:rx>
<core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.01</core:height>
</dso:Dish>
<dso:Bottle rdf:ID="orangeBottle_3">
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
<core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
<core:containsSubstance>
<core:Substance rdf:ID="Orange"/>
</core:containsSubstance>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:rx>
<core:in rdf:resource="#refrigerator"/>

```

```

<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:objectFree>
<core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:full>
<core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
<core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:num>
<core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:height>
<core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:max>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:ry>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:empty>
</dso:Bottle>
<dso:Bottle rdf:ID="rumBottle">
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
<core:in>
<core:HeterogeneousObjectContainer rdf:ID="shelf_B">
<core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:empty>
<core:posz
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>1.7</core:posz>
<core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:full>
<core:max
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>10</core:max>
<core:rx
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>1.5</core:rx>
<core:posy
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>-4.9</core:posy>
<core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>4</core:num>
<core:objectFree

```

```

    rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:objectFree>
  <core:yaw
    rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:yaw>
  <core:ry
    rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.1</core:ry>
  <core:posx
    rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >-1.0</core:posx>
  <core:height
    rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.05</core:height>
  </core:HeterogeneousObjectContainer>
</core:in>
<core:empty
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:empty>
<core:posz
  rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posz>
<core:height
  rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.3</core:height>
<core:containsSubstance>
  <core:Substance rdf:ID="Rum"/>
</core:containsSubstance>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.03</core:ry>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.03</core:rx>
<core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >8</core:max>
<core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >8</core:num>
<core:filterType
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ALL</core:filterType>
<core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:objectFree>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posy>
<core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >true</core:full>
</dso:Bottle>
<dso:Bottle rdf:ID="beerBottle_2">
  <core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >true</core:full>
  <core:filterType
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ALL</core:filterType>
  <core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"

```

```

>0.2</core:height>
<core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
<core:empty rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:empty>
<core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:num>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:rx>
<core:containsSubstance rdf:resource="#Beer"/>
<core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:max>
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:objectFree>
<core:in rdf:resource="#refrigerator"/>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:ry>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
</dso: Bottle>
<dso:Tray rdf:ID="tray_3">
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posz>
  <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >5</core:max>
  <core:in rdf:resource="#cabinet_1"/>
  <core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >>false</core:full>
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.25</core:ry>
  <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >5</core:num>
  <core:filterType
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ALL</core:filterType>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posx>
  <core:baseFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >true</core:baseFree>
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posy>
  <core:on>
    <dso:Tray rdf:ID="tray_2">
      <core:on rdf:resource="#tray_1"/>
      <core:posz
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >0.0</core:posz>
      <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"

```

```

>5</core:max>
<core:in rdf:resource="#cabinet_1"/>
<core:baseFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:baseFree>
<core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:full>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.25</core:ry>
<core:posx
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
<core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>5</core:num>
<core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
<core:posy
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.01</core:height>
<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:objectFree>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.25</core:rx>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>true</core:empty>
<core:yaw
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
</dso:Tray>
</core:on>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.25</core:rx>
<core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.01</core:height>
<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:objectFree>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>true</core:empty>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
</dso:Tray>
<dso:Cup rdf:ID="cup_3">
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"

```



```

>0.03</core:rx>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
<core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.07</core:height>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.03</core:ry>
<core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:full>
<core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>true</core:empty>
<core:in>
  <dso:Cabinet rdf:ID="cupboard">
    <core:height
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >1.0</core:height>
    <core:yaw
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >0.0</core:yaw>
    <core:full
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
      >>false</core:full>
    <core:posz
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >0.8</core:posz>
    <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >5</core:num>
    <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >40</core:max>
    <core:objectFree
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
      >>false</core:objectFree>
    <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >0.75</core:ry>
    <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >0.5</core:rx>
    <core:empty
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
      >>false</core:empty>
    <core:filterType
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >CLASSES-ONLY</core:filterType>
    <core:posx
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >-4.5</core:posx>
    <core:posy
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >-1.75</core:posy>

```

```

    </dso:Cabinet>
  </core:in>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posx>
  <core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:objectFree>
</dso:Cup>
<core:BaseObject rdf:ID="table_A">
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posy>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >-0.5</core:posx>
  <core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:objectFree>
  <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >1.5</core:rx>
  <core:filterType
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ALL</core:filterType>
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:yaw>
  <core:height
  rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >1.5</core:height>
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >1.0</core:ry>
  <core:baseFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >true</core:baseFree>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.75</core:posz>
</core:BaseObject>
<core:BaseObject rdf:ID="table_B">
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >2.0</core:ry>
  <core:height
  rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >1.5</core:height>
  <core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:objectFree>
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >2.0</core:posy>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >-4.5</core:posx>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.75</core:posz>
  <core:filterType
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ALL</core:filterType>
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:yaw>

```

```

    <core:baseFree
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:baseFree>
    <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.5</core:rx>
  </core:BaseObject>
  <core:Substance rdf:ID="Coke"/>
  <dso:Cup rdf:ID="cup_2">
    <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:yaw>
    <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.03</core:rx>
    <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posy>
    <core:filterType
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ALL</core:filterType>
    <core:height
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.07</core:height>
    <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.03</core:ry>
    <core:full
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:full>
    <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posz>
    <core:empty
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</core:empty>
    <core:in rdf:resource="#cupboard"/>
    <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posx>
    <core:objectFree
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:objectFree>
  </dso:Cup>
  <core:Agent rdf:ID="customerB">
    <core:filterType
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ALL</core:filterType>
    <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >4.5</core:posx>
    <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.25</core:ry>
    <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >2.5</core:posy>
    <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:yaw>
    <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >1.0</core:posz>
    <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.25</core:rx>
    <core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"

```

```

    >2.0</core:height>
    <core:objectFree
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</core:objectFree>
  </core:Agent>
  <dso:Bottle rdf:ID="beerBottle_3">
    <core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</core:full>
    <core:filterType
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ALL</core:filterType>
    <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posz>
    <core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.2</core:height>
    <core:empty rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:empty>
    <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</core:num>
    <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.015</core:rx>
    <core:containsSubstance rdf:resource="#Beer"/>
    <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</core:max>
    <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posx>
    <core:objectFree
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:objectFree>
    <core:in rdf:resource="#refrigerator"/>
    <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.015</core:ry>
    <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posy>
    <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:yaw>
  </dso:Bottle>
  <dso:Bottle rdf:ID="beerBottle_5">
    <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</core:max>
    <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</core:num>
    <core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.2</core:height>
    <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posz>
    <core:filterType
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ALL</core:filterType>
    <core:full rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</core:full>
    <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.015</core:rx>
    <core:objectFree

```

```

    rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:objectFree>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posx>
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:yaw>
  <core:in rdf:resource="#refrigerator"/>
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posy>
  <core:empty
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:empty>
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.015</core:ry>
  <core:containsSubstance rdf:resource="#Beer"/>
</dso:Bottle>
<dso:Bottle rdf:ID="orangeBottle_5">
  <core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:objectFree>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posx>
  <core:containsSubstance rdf:resource="#Orange"/>
  <core:full
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >true</core:full>
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.015</core:ry>
  <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.015</core:rx>
  <core:in rdf:resource="#refrigerator"/>
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posy>
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:yaw>
  <core:empty
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:empty>
  <core:filterType
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ALL</core:filterType>
  <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >1</core:num>
  <core:height
  rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.2</core:height>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posz>
  <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >1</core:max>
</dso:Bottle>
<dso:Cup rdf:ID="cup_5">
  <core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"

```

```

>false</core:objectFree>
<core:in rdf:resource="#cupboard"/>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:empty>
<core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
>0.0</core:posz>
<core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
>0.07</core:height>
<core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:full>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
>0.0</core:yaw>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
>0.0</core:posy>
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
>0.0</core:posx>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
>0.03</core:ry>
<core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
>0.03</core:rx>
</dso:Cup>
<core:ServiceContainer rdf:ID="donutsTray">
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
  >0.0</core:yaw>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
  >0.0</core:posx>
  <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
  >0.2</core:rx>
  <core:on rdf:resource="#table_B"/>
  <core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:objectFree>
  <core:filterType
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ALL</core:filterType>
  <core:containsSubstance>
    <core:Substance rdf:ID="Donut"/>
  </core:containsSubstance>
  <core:empty
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:empty>
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
  >0.4</core:ry>
  <core:height
  rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
  >0.01</core:height>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
  >0.0</core:posz>

```

```

<core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>15</core:num>
<rdf:type
rdf:resource="http://www.uv.es/~agentes/SVECore#MovableObject"/>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>15</core:max>
<core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:full>
</core:ServiceContainer>
<core:ServiceContainer rdf:ID="iceContainer">
  <core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
  <core:containsSubstance>
    <core:Substance rdf:ID="Ice"/>
  </core:containsSubstance>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.5</core:posz>
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
  <core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:objectFree>
  <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>50</core:num>
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>-4.75</core:posy>
  <core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>1.0</core:height>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>2.5</core:posx>
  <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>50</core:max>
  <core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:empty>
  <core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:full>
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.25</core:ry>
  <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.25</core:rx>
</core:ServiceContainer>
<dso:Bottle rdf:ID="cokeBottle_3">
  <core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
  <core:containsSubstance rdf:resource="#Coke"/>
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"

```

```

>0.0</core:posy>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:rx>
<core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:height>
<core:in rdf:resource="#refrigerator"/>
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
<core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:full>
<core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:max>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:ry>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:empty>
<core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
<core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:num>
<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:objectFree>
</dso:Bottle>
<dso:Dish rdf:ID="dish_2">
  <core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:full>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
  <core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
  <core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:objectFree>
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
  <core:in rdf:resource="#cabinet_1"/>
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:ry>
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
  <core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:empty>
  <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"

```



```

    >0.2</core:rx>
    <core:height
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.01</core:height>
  </dso:Dish>
  <dso:Bottle rdf:ID="ginBottle">
    <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.03</core:rx>
    <core:empty
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:empty>
    <core:filterType
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ALL</core:filterType>
    <core:in rdf:resource="#shelf_B"/>
    <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posz>
    <core:height
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.3</core:height>
    <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >8</core:num>
    <core:full
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</core:full>
    <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posy>
    <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >8</core:max>
    <core:objectFree
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:objectFree>
    <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.03</core:ry>
    <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:yaw>
    <core:containsSubstance>
      <core:Substance rdf:ID="Gin"/>
    </core:containsSubstance>
    <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posx>
  </dso:Bottle>
  <dso:Cup rdf:ID="cup_1">
    <core:filterType
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ALL</core:filterType>
    <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.03</core:ry>
    <core:full
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:full>
    <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posz>
    <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"

```

```

>0.03</core:rx>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
<core:in rdf:resource="#cupboard"/>
<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:objectFree>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:empty>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.07</core:height>
</dso:Cup>
<dso:Bottle rdf:ID="cokeBottle_4">
  <core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:height>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
  <core:in rdf:resource="#refrigerator"/>
  <core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
  <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:rx>
  <core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:empty>
  <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:num>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
  <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:max>
  <core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:objectFree>
  <core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:full>
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:ry>
  <core:containsSubstance rdf:resource="#Coke"/>
</dso:Bottle>
<dso:Cup rdf:ID="cup_4">

```

```

<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:objectFree>
<core:in rdf:resource="#cupboard"/>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:empty>
<core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
<core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.07</core:height>
<core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:full>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.03</core:ry>
<core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.03</core:rx>
</dso:Cup>
<core:Substance rdf:ID="Milk"/>
<dso:Bottle rdf:ID="whiskyBottle">
  <core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:objectFree>
  <core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:empty>
  <core:containsSubstance>
    <core:Substance rdf:ID="Whisky"/>
  </core:containsSubstance>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
  <core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
  <core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.3</core:height>
  <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>8</core:max>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
  <core:in rdf:resource="#shelf_B"/>
  <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"

```

```

>0.03</core:rx>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>8</core:num>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:full>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.03</core:ry>
</dso:Bottle>
<dso:Bottle rdf:ID="cokeBottle_5">
<core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:height>
<core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
<core:in rdf:resource="#refrigerator"/>
<core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:rx>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:empty>
<core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:num>
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:max>
<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:objectFree>
<core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:full>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:ry>
<core:containsSubstance rdf:resource="#Coke"/>
</dso:Bottle>
<core:BaseObject rdf:ID="bar">
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>5.0</core:ry>
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>3.5</core:posx>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>

```

```

<core:height rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>1.5</core:height>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.5</core:rx>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:objectFree>
<core:baseFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>true</core:baseFree>
<core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.75</core:posz>
</core:BaseObject>
<dso:Bottle rdf:ID="orangeBottle_1">
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:ry>
<core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:empty>
<core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
<core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:height>
<core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:max>
<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:objectFree>
<core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:num>
<core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>true</core:full>
<core:containsSubstance rdf:resource="#Orange"/>
<core:posx
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
<core:in rdf:resource="#refrigerator"/>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:rx>
</dso:Bottle>

```

```

<core:Agent rdf:ID="customerC">
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >-2.5</core:posy>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >1.0</core:posz>
  <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.25</core:rx>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >4.5</core:posx>
  <core:filterType
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ALL</core:filterType>
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.25</core:ry>
  <core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >true</core:objectFree>
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:yaw>
  <core:height
  rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >2.0</core:height>
</core:Agent>
<dso:Bottle rdf:ID="orangeBottle_2">
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posx>
  <core:containsSubstance rdf:resource="#Orange"/>
  <core:filterType
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ALL</core:filterType>
  <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.015</core:rx>
  <core:in rdf:resource="#refrigerator"/>
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:yaw>
  <core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:objectFree>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posz>
  <core:full
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >true</core:full>
  <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >1</core:num>
  <core:height
  rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.2</core:height>
  <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >1</core:max>
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.015</core:ry>
  <core:posy
  rdf:datatype="http://www.w3.org/2001/XMLSchema#float"

```

```

    >0.0</core:posy>
    <core:empty
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:empty>
  </dso:Bottle>
  <dso:Bottle rdf:ID="vodkaBottle">
    <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posx>
    <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.03</core:rx>
    <core:filterType
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ALL</core:filterType>
    <core:full
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</core:full>
    <core:posz
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posz>
    <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >8</core:num>
    <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.03</core:ry>
    <core:containsSubstance>
      <core:Substance rdf:ID="Vodka"/>
    </core:containsSubstance>
    <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:yaw>
    <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >8</core:max>
    <core:empty
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:empty>
    <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:posy>
    <core:height
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.3</core:height>
    <core:objectFree
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:objectFree>
    <core:in rdf:resource="#shelf_B"/>
  </dso:Bottle>
  <core:Agent rdf:ID="customerA">
    <core:filterType
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ALL</core:filterType>
    <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.25</core:ry>
    <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >4.5</core:posx>
    <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.0</core:yaw>
    <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"

```

```

>1.0</core:posz>
<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:objectFree>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.25</core:rx>
<core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>2.0</core:height>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>4.5</core:posy>
</core:Agent>
<dso:Bottle rdf:ID="cokeBottle_1">
  <core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:full>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
  <core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:objectFree>
  <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:ry>
  <core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
  <core:containsSubstance rdf:resource="#Coke"/>
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
  <core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</core:empty>
  <core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:height>
  <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:rx>
  <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:max>
  <core:in rdf:resource="#refrigerator"/>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
  <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:num>
</dso:Bottle>
<dso:Dish rdf:ID="dish_5">
  <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
  <core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```



```

>ALL</core:filterType>
<core:objectFree
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:objectFree>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:empty>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:rx>
<core:in rdf:resource="#cabinet_1"/>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posz>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:ry>
<core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.01</core:height>
<core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:full>
</dso:Dish>
<dso:Bottle rdf:ID="beerBottle_1">
<core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:max>
<core:in rdf:resource="#refrigerator"/>
<core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:yaw>
<core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posy>
<core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.0</core:posx>
<core:filterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>ALL</core:filterType>
<core:empty
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</core:empty>
<core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</core:num>
<core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:rx>
<core:height
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.2</core:height>
<core:full
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</core:full>
<core:containsSubstance rdf:resource="#Beer"/>
<core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.015</core:ry>
<core:posz
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"

```

```

    >0.0</core:posz>
    <core:objectFree
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:objectFree>
  </dso:Bottle>
  <dso:Bottle rdf:ID="orangeBottle_4">
    <core:objectFree
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:objectFree>
    <core:posx
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
    >0.0</core:posx>
    <core:containsSubstance rdf:resource="#Orange"/>
    <core:full
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</core:full>
    <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
    >0.015</core:ry>
    <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
    >0.015</core:rx>
    <core:in rdf:resource="#refrigerator"/>
    <core:posy
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
    >0.0</core:posy>
    <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
    >0.0</core:yaw>
    <core:empty
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >false</core:empty>
    <core:filterType
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ALL</core:filterType>
    <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int "
    >1</core:num>
    <core:height
      rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
    >0.2</core:height>
    <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int "
    >1</core:max>
    <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
    >0.0</core:posz>
  </dso:Bottle>
  <dso:Bottle rdf:ID="cokeBottle_2">
    <core:filterType
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ALL</core:filterType>
    <core:containsSubstance rdf:resource="#Coke"/>
    <core:posy rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
    >0.0</core:posy>
    <core:rx rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
    >0.015</core:rx>
    <core:yaw rdf:datatype="http://www.w3.org/2001/XMLSchema#float "
    >0.0</core:yaw>
    <core:height

```

```
    rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.2</core:height>
  <core:posx rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posx>
  <core:in rdf:resource="#refrigerator"/>
  <core:full
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >true</core:full>
  <core:max rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >1</core:max>
  <core:empty
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:empty>
  <core:ry rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.015</core:ry>
  <core:posz rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</core:posz>
  <core:num rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >1</core:num>
  <core:objectFree
  rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</core:objectFree>
</dso:Bottle>
</rdf:RDF>
```