



Modèle global et paramétrable, pour la gestion des foules d'agents en animation comportementale

Hakim Soussi

► To cite this version:

Hakim Soussi. Modèle global et paramétrable, pour la gestion des foules d'agents en animation comportementale. Sociologie. Université de Bourgogne, 2011. Français. NNT : 2011DIJOS069 . tel-00762319

HAL Id: tel-00762319

<https://theses.hal.science/tel-00762319>

Submitted on 7 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE BOURGOGNE
U.F.R SCIENCES ET TECHNIQUES
ÉCOLE DOCTORALE E2S
LE2I - UMR CNRS 5158

Thèse

présentée pour obtenir le grade de Docteur de l'Université de Bourgogne
Spécialité : INFORMATIQUE
par

Hakim Soussi

MODÈLE GLOBAL ET
PARAMÉTRABLE, POUR LA GESTION DES
FOULES D'AGENTS EN ANIMATION
COMPORTEMENTALE.

Soutenue le 06/12/2011 devant le jury composé de :

Rapporteurs :	Yves Duthen	Professeur à l'Université de Toulouse 1 Capitole
	Celine Loscos	Professeur à l'Université de Reims Champagne-Arden
Examineurs :	Behzad Shariat	Professeur à l'Université de Lyon 1
	Fabrice Lamarche	Maitre de conférences à l'Université de Rennes 1
Directeur de thèse :	Marc Neveu	Professeur à l'Université de Bourgogne, Dijon
Co-Directeur :	Joël Savelli	Maitre de conférences à l'Université de Bourgogne, Dijon

Table des matières

Résumé	v
Abstract	vii
Remerciements	ix
1 Introduction	1
2 Niveaux macroscopique et microscopique	5
2.1 Introduction	6
2.2 Réalisme	6
2.2.1 Réalisme microscopique	6
2.2.2 Réalisme macroscopique	7
2.2.3 Combinaison des deux niveaux de réalisme	8
2.3 Types de foules	8
2.3.1 Foules homogènes	9
2.3.2 Collection d'agents	9
2.3.3 Groupe	10
2.3.4 Combinaison des types de foules	11
2.4 Types de modèles	11
2.4.1 Modèles macroscopiques	12
2.4.2 Modèles microscopiques	12
2.4.3 Synthèse	17
2.5 Contrôle versus autonomie	17
2.5.1 Contrôle	18
2.5.2 Communication	20
2.5.3 Variabilité	21
2.5.4 L'adaptabilité	22
2.6 Synthèse	23
2.7 Conclusion	24

3	L'environnement virtuel	25
3.1	Introduction	25
3.2	Acquisition de l'information	26
3.2.1	La perception	28
3.2.2	La communication	30
3.3	Représentation de l'environnement	32
3.3.1	Représentation des environnements simples	32
3.3.2	Représentation des environnements complexes	33
3.3.3	Les algorithmes de navigation	36
3.4	Conclusion	38
4	Modèles comportementaux de foules	39
4.1	Introduction	40
4.2	Les smart objects	40
4.2.1	Propriétés physiques de l'objet	41
4.2.2	Modélisation des formes d'interaction	41
4.2.3	Exemples d'utilisation des smart objects	42
4.2.4	Discussion	46
4.3	Système MACES	48
4.4	Système HiDAC	52
4.5	Le système de classeurs (LCS)	54
4.6	Le modèle ViCrowd	56
4.6.1	Types de comportements dans ViCrowd	57
4.6.2	La structure des groupes dans ViCrowd	58
4.7	Modèle de comportements scalables	58
4.8	Le système MAGS	60
4.9	Synthèse	61
4.9.1	Smart objects	62
4.9.2	MACES	63
4.9.3	HiDAC	64
4.9.4	Le système de classeurs (LCS)	64
4.9.5	Le modèle ViCrowd	65
4.9.6	Modèle de comportements scalable	66
4.9.7	Système MAGS	67
4.10	Conclusion	68
5	Présentation du modèle	69
5.1	Introduction	69
5.2	Composants principaux du modèle	70
5.2.1	Agent et attributs de caractère	71
5.2.2	Contexte	72

5.2.3	Environnement	84
5.3	Fonctionnement du modèle	86
5.3.1	Calcul de l'ensemble des contextes	86
5.3.2	Calcul de la combinaison des contextes	86
5.3.3	Calcul des valeurs des attributs de caractère	88
5.3.4	Activation des comportements	88
5.3.5	Exécution des comportements	88
5.4	Expérimentations	90
5.4.1	Deuxième expérimentation : soumission à plusieurs contextes	94
5.4.2	Troisième expérimentation : contexte propageur . . .	99
5.5	Conclusion	102
6	Conclusion et perspectives	105

Résumé

Le réalisme d'une application traitant de l'animation comportementale de foules est fondé d'une part sur le rendu graphique des scènes produites par l'application, mais aussi sur le réalisme du comportement lui-même. C'est ce dernier point qui est notre objet d'étude. Le réalisme du comportement d'une foule est avant tout global (réalisme macroscopique) : elle doit avoir des propriétés statistiques (densité, dispersion, vitesse moyenne, ...) proches de celles d'une foule réelle. Il est aussi local (réalisme microscopique), c'est-à-dire que les agents ou groupes d'agents doivent idéalement avoir des comportements proches de ceux des humains ou groupes humains pris pour référence, tout au moins dans le domaine d'application restreint considéré.

L'objet de cette thèse est de proposer un modèle générique pour effectuer des simulations comportementales de foules, pour pouvoir satisfaire simultanément les deux types de réalisme macroscopique et microscopique et rendre compte de la plupart des types de foules (foules homogènes, groupes, collection d'agents). Pour ce faire, nous avons dégagé quelques principes simples et peu coûteux en ressources. Nous introduisons la notion de contexte (global, localisé, et propagateur). Les contextes globaux et localisés appliqués à un groupe d'agents tendent à leur donner un même comportement et constituent ainsi un moyen de contrôle global de ces acteurs (réalisme macroscopique). Le contexte propagateur donne un moyen de propagation d'informations entre les agents de la foule (communication). Les agents eux-mêmes sont dotés de tendances qui, en les différenciant dans un même contexte, leur donne une variété de comportements qui concourt au réalisme microscopique.

Mots-clés : simulation comportementale de foules, contextes, attributs de caractère, réalisme macroscopique et microscopique.

Abstract

The realism of crowd behavioral animation is based on one hand on a rendering of graphic scenes generated by the application, and on the other hand on the realism of the behavior. This is last point which is our object of study. The realism of crowds' behavior is essentially global (macroscopic realism). It must respect the required statistical characteristics of the crowd (density, dispersal, speed...) similar to those of a real crowd. The realism is also local (microscopic realism), i.e. agents should ideally behave like humans or human groups taken as reference, at least in the focus of the considered application.

The aim of our research is to propose a generic model to perform crowd behavior simulation in order to simultaneously satisfy the criterion of macroscopic and microscopic realism and that may be used to build various crowd behavioral simulations (homogeneous crowds, groups, collections of agents). To do this, we identified some simple principles and inexpensive resources ; we introduce the notion of context (global, localized, and propagator). Global contexts applied to a group of agents, it tends to give them the same behavior and thus constitutes a means for global control of those agents (macroscopic realism). Propagator contexts allow the propagation of information among agents (communication). Agents are themselves equipped with tendencies that differentiate them within the context, and give them a variety of behaviors that contributes to microscopic realism.

Keywords : crowd behavioral simulation, contexts, character attributes, macroscopic and microscopic realism.

Remerciements

Je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de cette thèse.

Je tiens à remercier en premier lieu les personnes qui m'ont suivi durant cette thèse. Tout d'abord je souhaite citer mes deux responsables de thèse, M. Marc NEVEU, Professeur à l'Université de Bourgogne et M. Joël SAVELLI, Maître de conférences à l'Université de Bourgogne, qui se sont toujours montrés à l'écoute et disponibles tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'ils ont bien voulu me consacrer.

Je tiens à remercier aussi Dominique MICHELUCCI et Dominique FAUDOT qui ont très gentiment fait partie du comité de suivi de ma thèse.

Je remercie l'ensemble de l'équipe Synthèse d'Images au sein du Laboratoire Le2i pour son accueil chaleureux durant la durée de cette thèse.

Je tiens à remercier également Monsieur Yves DUTHEN, Professeur à l'Université de Toulouse 1 Capitole et Madame Céline LOSCOS, Professeur à l'Université de Reims Champagne-Ardenne pour avoir accepté d'être les rapporteurs de ce travail. Leurs commentaires, leurs critiques et leurs questions sont autant d'encouragements à poursuivre ce travail.

Je remercie également Behzad SHARIAT, Professeur à l'Université de Lyon 1 et Fabrice LAMARCHE, Maître de conférences à l'Université de Rennes 1 d'avoir fait l'honneur d'examiner ma thèse.

Je n'oublie pas non plus les nombreux doctorants que j'ai pu croiser aussi bien au Laboratoire Le2i que durant les conférences auxquels j'ai participé.

Je remercie aussi toute ma famille, ma mère, mon père, mes frères, et surtout ma femme, qui m'ont soutenu et ont toujours cru à ma réussite.

Merci à tous

Chapitre 1

Introduction

L'animation comportementale prend de plus en plus d'ampleur au sein de la simulation et de la modélisation en réalité virtuelle, que ce soit dans l'industrie du jeu vidéo ou encore dans l'industrie cinématographique. Le point commun entre ces industries est le fait que les deux cherchent un moyen d'obtenir dans leurs simulations des comportements de plus en plus réalistes. Ces moyens ne cessent d'évoluer au cours du temps pour satisfaire à toutes les exigences. L'un des domaines qui permet cette évolution et sans doute l'intelligence artificielle (IA) par l'intermédiaire des recherches dans le domaine des multi-agents. Un système multi-agents est composé d'un ensemble d'entités situées dans un environnement. Parmi ces entités, certaines représentent des agents. Ces agents sont autonomes, ils sont capables d'agir, de communiquer, et de percevoir leur environnement, dans le but d'assurer une tâche spécifique précédemment affectée à l'agent. Cependant, les techniques mises en œuvre en IA sont peu opératoires, trop complexes pour pouvoir être intégrées facilement à des applications réelles. De plus, elles sont coûteuses en terme de calculs à cause du nombre d'informations qui caractérisent chaque entité de l'environnement.

En animation comportementale de foules, on cherche à utiliser des techniques plus simples et moins coûteuses. Vu la place que commence à prendre l'animation comportementale de foules au sein de l'industrie ou de la recherche, un modèle global et paramétrable s'avère nécessaire. Ce modèle doit posséder les propriétés suivantes :

- le modèle doit être assez simple à mettre en œuvre et peu coûteux (quantité d'information à traiter pour chaque agent).
- il doit satisfaire au réalisme d'un point de vue macroscopique ou microscopique.

- il doit être capable de s'intégrer facilement à d'autres architectures.
- il doit pouvoir simuler les différents types de foules que se soient des agents isolés ou des groupes d'agents (une famille par exemple).
- il doit aussi permettre aux utilisateurs de contrôler les agents de la foule et d'interagir avec eux.

Le travail effectué dans cette thèse a pour objet d'étudier et d'intégrer toutes sortes de catégories de simulations de foules dans un modèle unique, flexible qui soit facile à mettre en œuvre, et qui permette de satisfaire le réalisme de la simulation au niveau macroscopique ou microscopique. En d'autres termes, ce modèle doit pouvoir offrir un cadre simple pour définir les agents. Il sera possible de leur affecter des tendances qui permettent de donner des indications sur leurs personnalités (caractères), afin que chaque agent puisse déterminer de manière personnelle les comportements à adopter face à chaque situation. De cette manière, on souhaite obtenir une variété dans les comportements adoptés par les agents dans une même situation. Mais la foule doit aussi avoir un comportement attendu dans certaines situations. Il doit donc être possible de la contrôler en la soumettant à des contextes consécutifs ou simultanés. Certains contextes (contextes globaux) concerneront toute la foule (par exemple, une sirène d'alarme) et les contextes localisés ne concerneront que la partie de la foule concernée directement (par exemple un obstacle statique ou dynamique). Un agent de la foule doit s'adapter aux situations créées via les contextes (c'est-à-dire, qu'il doit prendre en compte chaque nouvelle situation).

Organisation du mémoire

Ce document est constitué de six parties :

Dans le chapitre 2, nous présentons les deux types de réalisme (macroscopique et microscopique), auxquels on associe plusieurs critères nécessaires pour leur réalisation. On verra et on classera aussi les différents types de foules (foules homogènes, groupes, et collections d'agents) par rapport à ces critères.

Dans le chapitre 3, on parlera de la nature de l'information que peut contenir un environnement virtuel, et comment un agent virtuel peut percevoir et récupérer cette information (perception et communication).

Différents modèles classiques présentés et classés par rapport aux différents critères de réalisme seront vus au chapitre 4.

Dans le chapitre 5, on présentera les différents éléments du modèle issus de

la recherche effectuée au cours de cette thèse.

Enfin, dans le chapitre 6 nous présentons une synthèse de notre étude et nous énonçons des perspectives pour développer ce travail.

Chapitre 2

Niveaux macroscopique et microscopique

Sommaire

2.1	Introduction	6
2.2	Réalisme	6
2.2.1	Réalisme microscopique	6
2.2.2	Réalisme macroscopique	7
2.2.3	Combinaison des deux niveaux de réalisme	8
2.3	Types de foules	8
2.3.1	Foules homogènes	9
2.3.2	Collection d'agents	9
2.3.3	Groupe	10
2.3.4	Combinaison des types de foules	11
2.4	Types de modèles	11
2.4.1	Modèles macroscopiques	12
2.4.2	Modèles microscopiques	12
2.4.3	Synthèse	17
2.5	Contrôle versus autonomie	17
2.5.1	Contrôle	18
2.5.2	Communication	20
2.5.3	Variabilité	21
2.5.4	L'adaptabilité	22
2.6	Synthèse	23
2.7	Conclusion	24

2.1 Introduction

Dans [Par07] et [Don04] la distinction est clairement faite entre le niveau microscopique et macroscopique d’une animation comportementale de foules. Le comportement au niveau microscopique caractérise le comportement individuel de chaque agent. Le comportement au niveau macroscopique est celui **émergent** de la foule dans sa globalité : c’est *le comportement statistique provenant de la composition des comportements individuels des agents qui sont membres de la foule*.

2.2 Réalisme

Comme cela a été déjà été précisé, nous ne considérons pas l’animation comportementale sous l’angle du rendu graphique des scènes produites. De même, notre travail ne porte pas sur l’aspect physiologique des comportements illustré par les travaux de Demetri Terzopoulos [FTT99]. Notre étude porte sur le choix des actions effectuées par les agents en fonction de l’environnement dans lequel ils évoluent. C’est aussi sous cet angle que nous envisageons la notion de réalisme dans l’animation comportementale de foule. Celui-ci caractérise alors le réalisme du comportement par apport à l’environnement courant.

Dans le cas particulier de l’animation comportementale de foules, le réalisme doit être considéré au niveau *macroscopique* autant qu’au niveau *microscopique*.

2.2.1 Réalisme microscopique

Le **réalisme microscopique** concerne le réalisme du comportement de chaque individu virtuel (agent de la foule) dans un jeu de contextes, où le comportement doit ressembler à celui d’un humain appartenant à une foule dans les mêmes contextes.

Dans la plupart des cas, il est alors nécessaire que ce comportement ne soit pas trop stéréotypé, c’est-à-dire qu’il y ait une **variabilité** de comportement d’un agent à l’autre. De plus, les agents doivent être dotés d’une certaine **autonomie**, c’est-à-dire pouvoir résoudre certains problèmes à leur échelle de manière autonome.

Le réalisme microscopique est exigible dans les comportements génériques, communs à la plupart des agents (éviter des collisions par exemple) et dans des comportements spécifiques (pertinence des comportements dans un contexte donné, réalisation de tâches individuellement ou en collaboration

avec d'autres agents,...). La n  cessit   d'un r  alisme microscopique est modul  e par le degr   de granularit   (niveau de d  tails) de l'animation ; plus la vision est globale (moins elle se focalise sur les agents), moins le r  alisme des comportements individuels est important.

Focaliser sur le r  alisme microscopique conduit    r  aliser des mod  les o   les agents sont dot  s de capacit  s cognitives   lev  es, s'inspirant souvent des mod  les de l'intelligence artificielle et utilisant parfois des r  sultats issus de la psychologie ou de la sociologie. L'arch  type de ce genre d'approche est rencontr   chez Musse et Thalmann et al. [MT97], [MGT99], [MT01], [SMGT99] ou chez Pelechano et al. [PAB07], [PB06], [PG06]. Les agents y sont dot  s d'  tats   motionnels ou d'  tats mentaux et de capacit  s    g  rer des relations « sociales » locales avec d'autres agents. Il est alors possible de mod  liser des comportements sophistiqu  s associ  s    des individus isol  s ou    des groupes d'individus (ensembles structur  s d'individus qui partagent des informations et dont chacun a un r  le dans la structure).

2.2.2 R  alisme macroscopique

Le comportement macroscopique repr  sente l'« histoire » racont  e par l'animation comportementale de foule consid  r  e.

Parfois, il n'y a pas d'attente particuli  re sur le comportement macroscopique de la foule. L'histoire racont  e par la foule est limit  e    la juxtaposition des *histoires individuelles*. C'est le cas, par exemple, dans [MT01][Par07], o   des agents autonomes   voluent individuellement dans une gare. Dans beaucoup de cas, cependant, l'animation est *sc  naris  e* et la foule doit ob  ir    des contraintes macroscopiques, en termes de propri  t  s (densit  , dispersion, volume, vitesse moyenne de d  placement,...) ou de comportement (par exemple suivre une trajectoire) ; elle doit aussi r  agir d'une mani  re pr  visible    des   v  nements potentiels pr  d  finis (par exemple, comportement statistique de fuite en cas d'alarme). Dans le cadre d'une animation comportementale donn  e, le mot sc  nario sera d  fini ici comme *l'ensemble des contraintes attendus sur le comportement d'une foule* dans cette animation.

Le *r  alisme macroscopique* d'une animation comportementale de foule est satisfait lorsque la foule *raconte l'histoire pr  vue*, c'est-  -dire lorsqu'elle a le comportement macroscopique pr  vu par le sc  nario li      une animation donn  e, ce qui lui permet de ressembler    une foule r  elle.

Pour satisfaire au r  alisme macroscopique, les agents doivent jouer un r  le pr  cis dans le comportement   mergent de la foule.

2.2.3 Combinaison des deux niveaux de réalisme

Il est délicat de combiner les deux niveaux de réalisme et peu de chercheurs tentent de le faire. Néanmoins, Musse et Thalmann s'y sont attaché [MT01] en constituant des groupes d'agents aux comportements déterminés par leurs relations sociales et en mettant les groupes en relation sous le contrôle d'une entité logique qui représente la foule. Dans les démonstrations qui sont faites, la foule n'a cependant pas d'unité, de cohérence globale (elle est représentée par une collection de groupes qui, eux, peuvent être cohérents). De plus, le système est rigide et complexe. Il est difficile de faire varier dynamiquement l'autonomie des agents alors que certaines réalités le nécessitent (situation d'urgence qui s'impose à tous, même aux agents auparavant autonomes).

D'une manière plus générale, il y a le plus souvent une antinomie entre le réalisme macroscopique et le réalisme microscopique. En effet, pouvoir imposer à une foule des propriétés statistiques nécessite de *contrôler* des agents qui la composent. A contrario, le réalisme microscopique repose souvent sur l'**autonomie** des agents, qui leur permet d'adopter un comportement adapté dans des circonstances particulières. La nécessité de la modulation conjointe du contrôle et de l'autonomie des agents est au cœur des systèmes d'animation comportementale de foules.

Cela étant dit, le réalisme microscopique et macroscopique peuvent se rejoindre dans certains cas. Par exemple, il peut être exigé qu'une foule soit hétérogène ; les agents qui la composent doivent alors être variés en termes de natures et de comportements. Cela n'est possible que si, à l'échelle individuelle, les agents possèdent des paramètres qui les distinguent les uns des autres. Les paramètres de ce genre donnent une « personnalité » aux agents, ce qui concourt aussi à un réalisme microscopique. C'est dans ce cadre que se situent la plupart des travaux de Musse et al. [MT97], [MGT99], [MT01].

2.3 Types de foules

La plateforme qui est l'objet de cette étude devra rendre compte des modèles déjà réalisés et donc de tous les types de foules qu'ils permettent de représenter. Le principal problème à résoudre est encore (voir réalisme microscopique et macroscopique) de faire le lien entre le comportement des agents eux-même (échelle microscopique) et le comportement émergent de la foule (échelle macroscopique). Les relations entre ces deux niveaux de comportements varient en fonction du type de foules étudié.

2.3.1 Foules homogènes

Dans une foule homogène, les agents partagent la plupart de leurs caractéristiques, en particulier leurs buts et leurs comportements potentiels ; le regroupement est en général implicite, imposé par une communauté d'objectifs et les placements initiaux des agents.

En général, dans les foules de ce type, le *réalisme macroscopique* est primordial et les comportements des agents — en général très simples — sont souvent limités à ce qui est nécessaire pour qu'ils soient à même de jouer leur rôle dans le scénario. Par exemple, dans les Boids de Reynolds [Rey99], la foule doit respecter un facteur donné de cohésion et de dispersion. Ces contraintes sont traduites au niveau microscopique des agents et déterminent leur comportement commun (par exemple, se situer à une distance minimale et maximale des autres agents ou suivre un leader matérialisé ou non).

2.3.2 Collection d'agents

Dans ce que nous appelons une collection, chaque agent a son propre comportement et ses propres buts. Le scénario est faible ; il n'y a pas d'attente particulière sur le comportement macroscopique de la foule. En d'autres termes, « l'histoire racontée » par la foule est limitée à la juxtaposition des « histoires individuelles » (figure 2.1) (Voir par exemple, dans [MGT99], les agents autonomes qui évoluent dans une gare).



FIGURE 2.1 – Collection d'agent [MGT99].

Dans ces conditions, le réalisme microscopique ne contredit pas le réalisme macroscopique : comme les agents n'ont rien à faire en commun, presque aucun problème ne peut survenir quand on les met ensemble — à part les problèmes de partage de ressources (notamment les problèmes de collision) et les situations de blocage qui sont communs à toutes les foules. Le concepteur

a alors la liberté de construire des modèles d'agents complexes ; des modèles cognitifs avec des capacités d'apprentissage par exemple (voir [MGT99]).

En fait, l'animation comportementale de collections d'agents revient quasiment à animer les agents individuellement. Dès lors qu'une collection d'agents ne possède pas de propriétés macroscopiques, nous considérons que ces situations ne relèvent pas vraiment de la problématique de l'animation comportementale de foules — bien que des collections puissent ressembler à des foules et participer à les constituer.

2.3.3 Groupe

Les *groupes* sont des ensembles d'agents explicitement circonscrits (au contraire des foules homogènes) qui partagent de l'information et/ou communiquent. Ses membres doivent coopérer pour donner au groupe le comportement requis (pour respecter le scénario)(figure 2.2).

Dans [Pel03] on trouve quelques aspects fondamentaux caractérisant un



FIGURE 2.2 – Groupe (édité par Electronic Arts).

groupe ;

- Les différents rôles qu'y jouent les membres caractérisent totalement le groupe.
- La notion de cohésion d'un groupe, soit la « force » qui pousse les membres à demeurer ensemble.

En général, dans le cadre de l'animation comportementale, soit le scénario est *faible* (il ne nécessite pas de comportement spécifique. Par exemple, des personnes qui se déplacent ensemble : familles, groupes d'amis avec un « meneur ».), soit le groupe est fonctionnellement structuré préalablement. Cela signifie que les rôles qu'il est nécessaire d'attribuer aux membres du groupe pour satisfaire le scénario sont donnés dès le départ par le concepteur de

l'application. Dans les deux cas, il n'y a pas de conflit entre le réalisme microscopique et macroscopique.

Cependant, il est envisageable que le scénario soit contraignant (par exemple une tâche à résoudre par le groupe) et qu'aucune structure fonctionnelle ne soit précisée par le concepteur. Les membres du groupe ont alors à négocier leurs rôles respectifs pour remplir les conditions posées par le scénario. Ce genre de tâche requiert des capacités cognitives de haut niveau, ce qui conduit les auteurs de [MT01] à considérer que les groupes sont les foules les plus complexes. Les groupes de ce type sont surtout traités en intelligence artificielle et les modèles produits sur cette base sont peu opératoires à cause de leur complexité.

2.3.4 Combinaison des types de foules

De nombreuses foules réelles ne peuvent pas être placées directement dans l'une des catégories précédentes. Par exemple, certains modèles de nuées (d'oiseaux) font appel à un meneur (leader) à qui sont donnés une direction et une vitesse ou un but ([DDL02], [Rey99]). Les autres agents ont un même objectif : suivre le meneur en évitant les collisions (ainsi que d'autres contraintes qui ne concernent pas cette réflexion). Ce genre de foule est un groupe, dès lors que deux rôles différents y apparaissent. Par ailleurs, tous les agents sauf le meneur partagent souvent les mêmes caractéristiques et comportements ; ils forment donc ce que nous avons appelé une foule homogène. Dans une animation particulière, il sera en général nécessaire de faire appel à plusieurs des catégories de foules distinguées ci-dessus et de pouvoir considérer une foule comme une unité (un agent) d'une autre foule. Un modèle qui se veut général (c'est le cas pour celui sous-jacent à une plateforme) doit donc comporter de quoi représenter et traiter chaque type de foule, mais aussi proposer les moyens de combiner des foules de plusieurs types.

2.4 Types de modèles

Dans [Par07] et [Don04] notamment, sont distingués les modèles de foules macroscopique et microscopique qui mettent respectivement l'accent sur les propriétés émergentes de la foule et celles, individuelles, des agents. Les modèles macroscopiques ont été établis pour la plupart hors de l'animation comportementale, par l'étude de foules réelles. Ils ont permis de mettre en exergue des propriétés macroscopiques de foules qui devront être respectées par les modèles informatiques. Les modèles de l'animation comportementale sont pour la plupart des modèles microscopiques (autrement, les agents n'auraient pas

d'existence propre). Suivant le type de foules modélisé, ils sont plus ou moins sophistiqués.

2.4.1 Modèles macroscopiques

Dans un **modèle macroscopique**, les agents n'ont aucune autonomie ; leurs caractéristiques comportementales (en général leur position) est déterminée suivant une loi (en général issue de la physique) qui régit la foule qu'ils composent. En d'autres termes, ils sont vus comme des points matériels qui constituent un flux.

C'est dans ce cadre que se situent les systèmes à base de particules très utilisés dans d'autres champs de l'informatique graphique (modélisation de fumées, de nuages, de flammes...). Les modèles à base de particules forment une sous catégorie des modèles fondés sur la mécanique des fluides, lesquels font suite aux recherches de Henderson [Hen71] qui ont démontré qu'il était possible de modéliser un flux de piétons réels en les remplaçant par des molécules de gaz. Partant de cette idée, Dirk Helbing [HMFB01] a effectué des observations sur les similitudes entre moyenne et haute densité de la foule des piétons et le mouvement des particules de gaz et des fluides.

Les modèles fondés sur la mécanique des fluides (ceux sont à la base de particules) sont essentiellement adaptés à l'observation des flux d'écoulement de foules dans des bâtiments non encore réalisés, ce qui permet de calculer le temps d'évacuation d'une foule et de détecter par exemple les ralentissements et les endroits où elle s'agglomère (là où elle subit la plus grande pression). Pour ce faire, ces modèles se basent sur des observations de foules réelles ainsi que sur des calculs statistiques.

Dans [Par07], une étude plus détaillée de l'approche macroscopique a été effectuée (Voir le tableau 2.1). Étant caractérisés par une approche macroscopique, ces modèles ne tiennent pas compte du comportement individuel des agents, permettant ainsi une simulation rapide.

Ne nécessitant qu'une quantité assez faible de calculs, fonction linéaire du nombre d'agents, et faciles à répartir entre eux, les modèles macroscopiques sont les premiers à avoir été réalisés en animation comportementale et ils permettent de gérer des foules très volumineuses.

2.4.2 Modèles microscopiques

Contrairement aux modèles macroscopiques, l'approche microscopique permet des simulations dans lesquelles chaque entité est gérée individuellement. Un individu aura donc ici son fonctionnement propre qui prend en compte son environnement dont les entités (agents ou objets) autour de lui ainsi que

Modèles	Caractéristiques
Togawa [Tog55]	En se fondant sur les observations relatives à une foule réelle, l'auteur propose une formule permettant de calculer la vitesse de déplacement d'une foule par rapport à sa densité (une densité normale) dans les bâtiments à plusieurs étages (la vitesse sera la même pour toute la foule du bâtiment).
Melinek & Booth [MB75]	Proposent un modèle permettant de calculer la vitesse d'évacuation d'une foule pour chaque étage d'un bâtiment, où sont considérés deux types de densités soit une densité maximale à chaque étage. Soit une densité normale.
Pauls [PoBR75]	Se fondant sur l'observation d'exercices d'évacuation d'immeubles de bureaux de grande hauteur, l'auteur a établi une loi permettant de simuler l'évacuation au niveau des escaliers en fonction du débit unitaire en personne /seconde/mètre de largeur d'escalier et du nombre de personnes par mètre de largeur d'escalier.
Predtechenskii & Milinskii [PM78]	Propose un modèle fluide basé sur des calculs statistiques dont le but est de gérer les mouvements de foules de grande densité. Il permet de calculer la vitesse de déplacement de la foule en fonction de sa densité et des conditions d'évacuations (normal, urgent, confortable).
Takahashi, Tanaka & Kose [TTK88]	- Propose aussi un modèle fluide permettant de calculer la vitesse d'évacuation. Ce modèle repose sur une description simplifiée de l'environnement avec une caractérisation des espaces (escaliers, couloirs,...).

TABLE 2.1 – Modèles analysés dans [Par07].

des informations éventuelles sur la situation courante.

Les capacités internes dont disposent les individus peuvent varier en terme de complexité allant des comportements les plus simples, totalement réactifs (évitement simple des collisions) à des comportements très sophistiqués.

C'est dans le cadre des comportements réactifs que se situent les modèles basés règles tels que les boids [Rey87], où, à chaque comportement est associé une règle de type (*Si Condition Alors Comportement*).

Il y a aussi les comportements moyennement complexes, par exemple les groupes dans [MT01]. Les agents se déplacent en groupes et chaque agent dispose de critères lui permettant, suivant ses besoins, de rester dans un groupe ou d'en sortir au profit d'un autre groupe. Afin de garantir un certain réalisme visuel, les agents doivent aussi respecter certaines contraintes de cohésion imposées, comme par exemple une vitesse associée au groupe. Un agent peut aussi devenir chef de son groupe.

Les comportements les plus complexes sont généralement traités par les modèles développés en Intelligence Artificielle. Il s'agit par exemple de réaliser une tâche, d'atteindre un objectif en effectuant le « calcul » qui permet de définir la succession de comportements élémentaires à adopter en partant des ressources disponibles. Ce sont des processus de planification ou de raisonnement qui ne se limitent pas à un calcul de trajectoire — qui n'est déjà pas un problème simple.

Mais la plus grande difficulté consiste à réaliser collectivement une tâche donnée, chaque agent adoptant un rôle précis dans son accomplissement. Il est alors nécessaire de mettre en place des mécanismes d'interaction sociale, de négociation entre agents, qui nécessitent de doter les agents de capacités de communication évoluées et de gérer l'ordonnancement des sous-tâches.

2.4.2.1 Modèles basé règles

Les modèles à base de règles ont été largement utilisés dans la communauté infographique pour simuler les troupes d'animaux. Parmi les modèles les plus connus il y a les modèles basés sur les règles locales de Reynolds « boids » ; Il y a tout d'abord le modèle basique [Rey87] constitué de trois règles simples (voir figure 2.3) pour assurer « la séparation », « l'alignement » et « la cohésion ». Chaque boid effectue l'action par rapport à sa perception de son voisinage, laquelle permet d'obtenir une distance et un angle. (voir figure 2.3).

Chaque boid doit éviter la collision avec les objets et les autres boids de son voisinage. Afin d'appliquer ce modèle à des environnements plus complexes, Reynolds a ajouté un concept plus général au modèle basique précédent (comportement directionnel, permettant à une entité de se déplacer d'un endroit à un autre). Pour cela, en plus des trois règles précédentes, Reynolds a défini

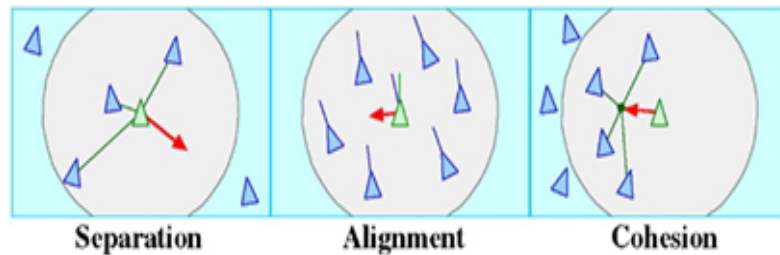


FIGURE 2.3 – Règles locales de Reynolds [Rey87].

de nouvelles règles, dont chacune correspond à une réaction spécifique dans l'environnement (par exemple, fuir, poursuivre, chercher, éviter des obstacles, suivre un chemin, ...). Il est possible aussi de combiner plusieurs de ces règles afin de réaliser une tâche plus complexe [Rey99].

Dans [OD02], des règles de comportements ont été définies, lesquelles vont servir à gérer d'une manière appropriée les différents types d'interactions entre les individus de la foule (par exemple, bavarder, se tenir la main, ou encore s'éviter). Dans [NT96], l'animation des agents autonomes exécutants des comportements complexes est complètement définie par les règles (où des règles simples peuvent conduire à des comportements émergents).

2.4.2.2 Les modèles cognitifs

Les modèles cognitifs s'intéressent à la représentation des connaissances d'un agent (croyances, intentions). Les intentions permettent à un agent de raisonner sur son état interne et sur celui des autres. Les modèles cognitifs ne dépendent pas seulement de stimuli provenant de l'environnement, mais aussi d'un but courant poursuivi par chaque agent. Ce dernier est doté de capacité d'apprentissage, qui permettent aux agents virtuels de prendre des décisions en fonction de leurs expériences en créant par exemple une carte mentale¹ de l'environnement. On trouve aussi l'approche connexionniste qui représente un réseau de neurones. L'agent dans ce cas est capable d'apprendre automatiquement l'exécution d'une tâche en utilisant une répartition de l'information sur l'ensemble du réseau. La prise de décision nécessite de raisonner ou de planifier pour accomplir des tâches à long terme afin de parvenir à une pleine autonomie de l'agent. J. Funge et al. [FTT99] a proposé un langage de modélisation de la cognition (CML), qui précise la connaissance du domaine en termes d'actions. Les auteurs fournissent une interface de haut niveau pour la

1. La carte mentale représente la connaissance de l'architecture de l'environnement (voir [PB05])

description des objectifs souhaités. D'autre part, le langage peut aussi servir comme un langage de programmation traditionnel, permettant la description précise de la façon dont le personnage doit agir. Certaines études intègrent les émotions, les modèles psychologiques et les rôles dans les simulations. Par exemple, Silverman et al. proposent le système PMFServ [SJCO06], [SBOC06] qui fait appel à des facteurs psychologiques qui affectent le comportement humain.

Pelechano et al. ont intégré des modèles psychologiques dans la simulation de foules [PB05](figure 2.4). Ils ont proposé un processus d'orientation basé sur les forces sociales qui permet aux individus d'explorer et d'apprendre la structure interne d'un bâtiment. Ainsi, les agents peuvent générer une carte mentale pour la navigation et trouver leur chemin dans un environnement qu'ils ne connaissent pas au préalable. La communication et les rôles sont ajoutés pour obtenir des comportements individualistes et diffuser des informations sur l'environnement.

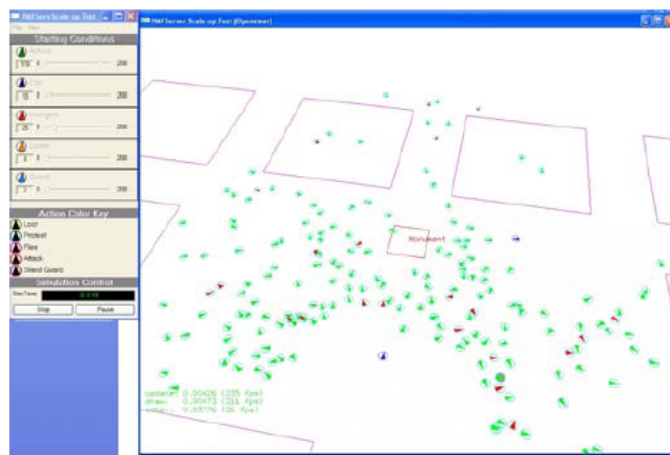


FIGURE 2.4 – Capture d'écran d'une scène de protestation développée en utilisant PMFServ et OpenSteer [PB05].

Des modèles cognitifs ont été employés en combinaison avec les modèles basés sur les règles telles que dans le modèle de Shao, W. et Terzopoulos [ST05] pour obtenir des comportements plus réalistes pour la simulation piétonnière. Cependant le fait que les règles soient indépendantes les unes des autres a pour effet que les comportements issus concernent plus les comportements réactifs (des situations particulières), que les comportements destinés à réaliser une tâche particulière.

On trouve aussi les comportements orientés buts, connus sous le nom de BDI² [RG91], où les agents exécutent des tâches via un système expert. Il y a aussi le système HTN³ qui est une approche de planification automatisée dans laquelle la dépendance entre les actions peut être donnée sous la forme de réseaux [EHN94].

2.4.3 Synthèse

Nous avons vu qu'il était difficile de concilier les réalismes microscopique et macroscopique. Dans les modèles de foules abordés ci-dessus, le problème n'est pas traité. Soit un scénario impose des contraintes fortes (réalisme macroscopique exigeant), mais alors les agents de la simulation ont un comportement réduit à leur rôle dans ce scénario ; ils n'ont pas d'autonomie (réalisme microscopique faible). Soit les contraintes associées au scénario sont très faibles et la « personnalité » des agents peut alors être développée (réalisme microscopique fort).

2.5 Contrôle versus autonomie

Nous avons vu que le réalisme d'une animation comportementale de foules était à considérer à l'échelle microscopique (celle des agents) et à l'échelle macroscopique (celle de la foule dans sa globalité). Or, pour que le réalisme soit assuré à l'échelle macroscopique, il est nécessaire que les agents soient « au service » du scénario de l'animation, et donc qu'ils puissent être **contrôlés** « d'en haut » pour remplir leur rôle dans ce scénario. Mais, le réalisme microscopique — s'il y a lieu — nécessite que les agents aient une individualité marquée, c'est-à-dire, en terme de comportement, qu'ils aient des comportements variés, non stéréotypés. Il est aussi nécessaire qu'ils soient capables de résoudre des problèmes simples à leur échelle qui ne soient pas partie prenante du scénario (le problème des collisions par exemple). Il est donc nécessaire que les agents disposent d'une certaine **autonomie**.

Il y a a priori antinomie entre la nécessité d'un contrôle et celle de l'autonomie des agents. Afin de mieux classer chaque type de foules dans ce sens, nous allons définir des critères qui sont essentiels pour résoudre ce problème. Ces mêmes critères vont nous servir par la suite (au chapitre 4) à analyser quelques modèles classiques de l'animation comportementale de foules.

2. Beliefs Desired Intentions

3. Hierarchical Task Network.

2.5.1 Contrôle

Le contrôle de la foule est un critère important dans la simulation et notamment pour satisfaire le réalisme macroscopique. Il représente la manière dont les utilisateurs peuvent interagir ou influencer les membres de la foule, tout en prenant en compte leur autonomie et les nombreuses interactions possibles au court du temps.

On peut distinguer deux types de contrôles : interactif qui donne un contrôle partiel à l'utilisateur, et prédéfini, qui à l'inverse du premier offre un contrôle total.

2.5.1.1 Contrôle interactif

Il permet de fournir un contrôle interactif aux utilisateurs de telle manière qu'ils puissent facilement influencer le comportement des agents ainsi que le flux de la foule durant la simulation (figure 2.5). Le contrôle interactif des personnages a été surtout appliqué aux jeux. Toutefois, seules quelques approches dans la littérature ont été consacrées au contrôle interactif de la foule. Dans [UCT04], les auteurs contrôlent la foule en utilisant une sorte de brosse, laquelle est commandée par l'utilisateur via la souris (sélection d'une zone susceptible d'être affectée — soumettre la foule dans cette zone à un événement — ou encore spécifier le chemin de la foule en dessinant des points à suivre) ainsi que le clavier (changer les paramètres des actions de la brosse, tel que la vitesse du déplacement). Dans [LGN⁺09], les auteurs présentent une approche dans laquelle les utilisateurs ont la possibilité de diriger et de contrôler les foules en spécifiant des zones de déplacement (navigation field) par des chemins dessinés directement dans la scène via une interface de création. Dans [MT01], les auteurs donnent la possibilité à l'utilisateur de contrôler le mouvement de la foule de différentes manières : soit en interagissant avec la foule en soumettant cette dernière à des événements afin d'obtenir des comportements du type événement / réaction, soit en guidant la foule en imposant des comportements à cette dernière. Dans [AMC03], les auteurs ont présenté une méthode permettant d'imposer des contraintes sur les trajectoires des agents à des moments spécifiques, tout en conservant les caractéristiques globales des autres agents qui ne sont pas soumis aux contraintes (c'est-à-dire, que les autres agents qui ne sont pas concernés par ces contraintes vont continuer leur route sans tenir compte de ces derniers).

2.5.1.2 Contrôle prédéfini

A l'inverse du contrôle interactif, l'animateur dispose ici d'un contrôle total puisqu'il est obligé de spécifier les trajectoires et comportements des

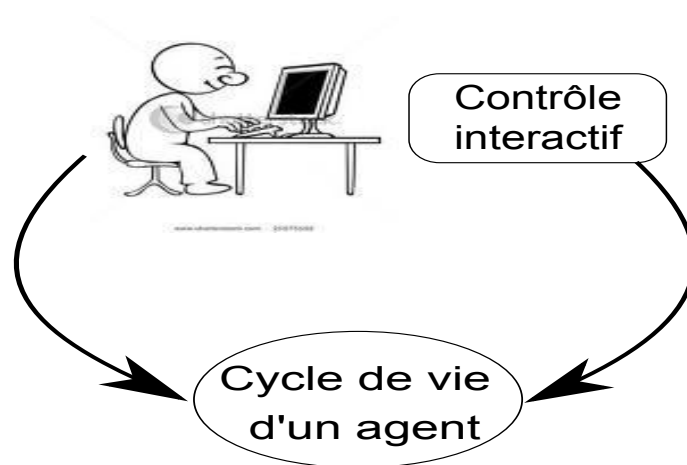


FIGURE 2.5 – Le contrôle interactif.

différents objets de l'animation.

On entend par contrôle prédéfini, des comportements scriptés, où l'agent est défini dans l'environnement en ayant un rôle ou un but qui ne peut être simulé via un mécanisme de décision, par exemple, des agents ayant un rôle de figurant dans la simulation. Dans ce cas, l'utilisateur va définir au préalable leurs histoires dans l'environnement tout au long de leur cycle de vie, et ne peut interagir avec eux durant la simulation.

On trouve dans cette thématique les modèles d'automates à états finis ; par exemple, HPTS⁴ [DR95] qui est un langage de scénarisation fondé sur des automates à état finis, il qui permet un contrôle prédéfini du fait que tous les états que peut prendre un agent seront tous définis à l'avance. Ce qui pose un problème dans le cas où l'agent aura beaucoup d'états, dans ce cas, on peut avoir des automates très difficiles à concevoir et à interpréter.

Les outils de scénarisation sont importants en ce qui concerne l'industrie des effets spéciaux, surtout depuis que, ces dernières années, les studios de production se sont mis à utiliser abondamment l'animation de foule pour remplacer les milliers de figurants dans les films (par exemple, dans le film *Titanic*, où des milliers de d'agents virtuels faisant un signe de la main remplacent de vrais voyageurs).

4. Hierarchical Parallel Transition System

2.5.2 Communication

La communication est l'un des critères essentiels pour réaliser une simulation réaliste que ce soit sur un plan microscopique ou macroscopique. La communication est utilisée pour faire circuler une information (figure 2.6). Le « témoin » d'un événement localisé (départ d'incendie par exemple) peut ainsi informer les individus qui n'y sont pas directement confrontés, le chemin pour atteindre une issue peut aussi être indiqué par certains agents à d'autres agents [PB05]. La communication peut aussi revêtir un caractère plus élaboré, lorsqu'il s'agit d'effectuer une tâche collective en négociant entre agents les rôles respectifs de chacun dans l'accomplissement de cette tâche. Les agents forment alors un groupe dédié à la tâche et des changements de groupes peuvent être envisagés au gré des tâches à effectuer [MT01].

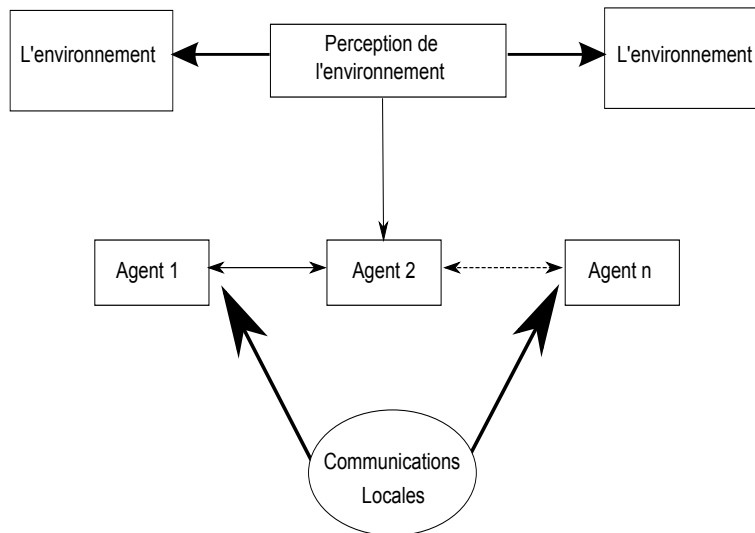


FIGURE 2.6 – La communication au niveau local entre les agents.

La communication au niveau microscopique permet ainsi aux comportements émergents de la foule de satisfaire aux buts correspondants au niveau macroscopique attendus par les utilisateurs. En effet, le réalisme macroscopique d'une simulation de foule est aussi satisfait par le biais des interactions locales entre les agents. Par exemple, dans [PB05], où le but est d'évacuer la foule en un minimum de temps, ce sont les communications locales entre les agents expérimentés et non expérimentés qui permettent de diriger le flux de la foule vers la sortie. C'est-à-dire, que l'information chemine par un relais, d'agent à agent, via un mécanisme de communication permettant à l'information de

circuler entre les membres de la foule (voir chapitre 3).⁵

2.5.3 Variabilité

Assurer la variabilité au sein d'une foule nécessite que les comportements des agents puissent être différenciés, ce qui implique que les agents soient tous définis sur des bases différentes (c'est très laborieux dans un modèle de foule). La difficulté est alors de ne pas tenter de modéliser d'une manière trop approfondie la nature humaine, ce qui est illusoire et inutile dans le cadre de la modélisation des foules — où le réalisme le plus essentiel à respecter reste le réalisme macroscopique. Les modèles risquent d'être surchargés de données à prendre en compte, ce qui peut induire des problèmes de performance sur un plan quantitatif (quantité de calculs à effectuer) ou qualitatif (difficulté de garder cohérente la « personnalité » des acteurs). Pour résoudre ce problème, les agents peuvent être simplement dotés d'une forme de « personnalité », de « caractère » qui module leur comportement par rapport à un comportement standard. Deux agents ainsi différenciés peuvent alors avoir des comportements différents dans le même contexte.

Cette personnalité est matérialisée dans les modèles par des attributs dédiés. Dans [Wig96], les auteurs ont présenté un système de personnalité OCEAN qui se base sur cinq traits de personnalités : franchise, conscience, extraverti, agréable, névrotique. Chaque trait est composé de plusieurs facteurs qui représentent les adjectifs qui sont employés pour décrire les personnes (par exemple, le facteur C^+ qui est représenté par des facteurs de personnalités correspondants à la conscience, tels que, l'*organisation*. Par contre, le facteur C^- est exactement le contraire de C^+).

D'autres travaux caractérisent la personnalité des agents de manière plus générale, pour cela ils utilisent des notions comme celle des états émotionnels ([MT97], [MGT99], [PB06], [SMGT99]) ou encore d'états mentaux [Nyg07]. Cela étant dit, le facteur *stress* semble primordial pour influencer le comportement des agents. De plus, l'animation comportementale de foules est souvent appliquée aux problèmes d'évacuation en cas d'urgence où ce facteur est particulièrement pertinent. L'augmentation du stress réduit la conscience, et spécialement la capacité d'orientation dans le bâtiment lors d'une urgence. Beaucoup d'agents réagissent à une forte pression par une augmentation de la vitesse de leurs actions. Selon le degré du stress ainsi que le changement dû à l'environnement, les agents non expérimentés peuvent être incapables de prendre la bonne décision, contrairement aux agents expérimentés qui se

5. L'utilisation de différents mécanismes de perception ne remettent pas en cause les principes établis.

basent sur leurs perceptions ainsi que leurs connaissances pour prendre la bonne décision [PB05]. Peu ou trop de stress peut affecter les performances négativement [SFG⁺05]. En effet, peu de stress peut provoquer une certaine négligence de la part de l'agent. Trop de stress peut mener à la panique avec les conséquences que celle-ci peut provoquer.

Les individus ayant un degré de stress très élevé ont tendance à la grégarité lors d'une évacuation, ce qui peut provoquer des accumulations aux niveaux des sorties. Dans [PG06], les auteurs ont simulé des agents respectueux qui ont tendance à laisser d'autres agents passer en premier. Pour ce faire, ils intègrent un facteur qui est *l'impatience* afin d'obtenir deux types d'agents différents (celui qui est patient, et celui qui ne l'est pas). Ils ont intégré aussi les *règles d'arrêts* afin de traiter les problèmes dûs aux blocages. Cependant ces règles ne permettent pas un réalisme parfait, elles augmentent le temps d'évacuation du fait que ces règles seront appliquées à chaque interaction entre les agents (par exemple, lors d'une évacuation d'urgence).

En conclusion, la modélisation de certains éléments de personnalité peut avoir une utilité, même dans l'animation comportementale de foules.

2.5.4 L'adaptabilité

Dans une simulation réaliste d'agents virtuels, chaque agent est en général défini au début de la simulation par un rôle dans son environnement et un but qu'il va essayer de réaliser. Cependant, lors du cycle de vie de l'agent, il se peut qu'il doive faire face à des situations imprévues qui l'empêchent de se rapprocher de son objectif, telles qu'une situation d'urgence (par exemple un feu). Dans ce cas, l'agent doit d'abord traiter cette situation avant de poursuivre son objectif principal. De ce fait, le besoin de s'adapter aux différentes situations qui peuvent se présenter durant la simulation devient rapidement primordiale. Afin de gérer ce problème, un mécanisme de prise de décision doit permettre à l'agent de réagir aux nouvelles situations suivant son état interne courant.

Certaines approches modélisent des agents cognitifs (avec des quantités de calculs et un nombre de paramètres assez conséquent) possédant des capacités très développées pour réaliser une tâche bien précise, et capable de s'adapter à des situations imprévues, telles que l'évitement de collisions ou encore un changement d'itinéraire. Dans [MT01], les auteurs ont simulés des agents déambulant dans un musée avec chacun une personnalité et un rôle (que se soit, un père de famille ou encore un guide). Ces agents vont faire face à une situation inattendue qui est le réveil d'une statue, laquelle se met à bouger et crée un mouvement de panique chez les agents qui vont réagir à cette situation selon la personnalité de chacun.

Une autre contrainte qui peut être traitée dans une simulation, est que l'agent peut effectuer une tâche complexe qui peut être nouvelle pour lui (n'ayant pas d'informations sur la manière de l'exécuter). Dans ce cas, au fur et à mesure que le temps passe, l'agent va apprendre à réaliser cette tâche d'une manière autonome via un système d'apprentissage qui va lui permettre de faire évoluer ces connaissances. Dans ce cas on trouve les systèmes de classeurs (LCS⁶) [SGC04], lesquels, utilisent un algorithme génétique évolutionniste à base de règles permettant de faire évoluer les classeurs de telle façon que, l'agent puisse effectuer des tâches dont lesquelles il est novice.

D'autres approches modélisent des agents réactifs tel que, les systèmes d'automates. Dans ce cas, on trouve HPTS++ [LD02] qui est une approche d'automates parallèles hiérarchiques. Il représente une évolution du système HPTS⁷ [Don01][DR95], où chaque état peut être composé d'un nombre d'automates fils, lesquels s'exécutent en parallèle (l'automate père choisit l'action à entreprendre parmi les propositions des sous-automates via une fonction, que se soit de manière cohérente ou encore aléatoire.). HPTS++ permet de gérer automatiquement la concurrence et l'adaptation des comportements selon leurs importances. Pour se faire, quelques notions ont été ajoutées, telles que les ressources, la priorité, et le degré de préférence. En d'autres termes, il offre un mécanisme permettant de dérouler des comportements en parallèle, sans que l'utilisateur ait à synchroniser à la main leur exécution. Par exemple, un agent qui écrit une lettre de la main droite, et qui boit de la main gauche. Si par exemple, le téléphone sonne, alors l'agent va suspendre temporairement son activité de la main gauche pour répondre au téléphone et la reprendre juste après. Une des applications connues représentant les fonctionnalités de ce système, qui permet de montrer la coordination de plusieurs comportements, est l'application du lecteur-buveur-fumeur, où l'acteur utilise plusieurs ressources afin de réaliser quatre comportements : boire, fumer, lire et manipuler les feuilles de papiers.

2.6 Synthèse

L'analyse faite dans ce chapitre montre une grande dispersion de modèles dont aucun ne traite simultanément les problèmes de réalisme macroscopique et microscopique. Soit les modèles d'acteurs sont riches et le réalisme microscopique est fort, mais c'est probablement au prix d'une certaine complexité de mise en œuvre qui ne rend pas très pertinente leur utilisation dans l'animation comportementale de foules — où le comportement des individus n'est

6. Learning Classifier System

7. Hierarchical Parallel Transition System

pas primordial. De plus, sauf d'une manière ad-hoc, les agents ne peuvent alors pas être contrôlés pour se mettre « au service » d'un scénario global contraignant (d'un réalisme macroscopique exigeant) — ce qui ne pose pas de problèmes dans certaines applications mais réduit le champ d'application des modèles considérés. Soit les modèles d'acteurs sont simples, réduits à leur rôle dans l'animation (dans le scénario de cette animation). Ils ne peuvent alors pas représenter des situations très riches (elles se limitent souvent à des foules homogènes). Il manque la possibilité de passer facilement dans un même modèle de l'autonomie au contrôle pour tout ou partie des agents.

2.7 Conclusion

Les deux types de réalisme (microscopique et macroscopique) ne sont pas facilement compatibles. En effet, le réalisme macroscopique nécessite le contrôle du comportement émergent de la foule, qui est d'autant plus facile que les acteurs qui la composent sont faiblement individualisés ; à la limite, les acteurs n'ont alors pour caractéristiques que leur rôle par rapport à la foule. Par contre, pour satisfaire le réalisme microscopique, les modèles donnent en général aux acteurs une individualité très forte (modèles cognitifs par exemple). Dans ce cas, le comportement émergent de la foule n'est pas nécessairement (globalement) réaliste.

Chapitre 3

L'environnement virtuel

Sommaire

3.1	Introduction	25
3.2	Acquisition de l'information	26
3.2.1	La perception	28
3.2.2	La communication	30
3.3	Représentation de l'environnement	32
3.3.1	Représentation des environnements simples	32
3.3.2	Représentation des environnements complexes . . .	33
3.3.3	Les algorithmes de navigation	36
3.4	Conclusion	38

3.1 Introduction

Sur un plan purement comportemental (dénué de considérations graphiques), l'environnement d'un agent se limite pour lui à un pourvoyeur d'informations à prendre en compte pour « choisir » les comportements futurs à adopter. Par exemple, un agent doit pouvoir réagir à un événement global (sirène d'alarme par exemple), gérer les collisions avec des obstacles fixes ou mobiles, poursuivre un objectif en fonction de la situation présente, participer à l'évitement des situations de blocage, jouer son rôle dans un groupe auquel il appartient.

L'environnement d'un agent peut être statique (bâtiments, panneaux de signalisation, . . .) ou dynamique. La propriété principale des éléments qui constituent l'environnement dynamique est leur aptitude à changer d'état au cours

de l'animation. C'est bien entendu le cas des autres agents — qui se déplacent —, mais aussi celui de certains objets (feux de signalisation par exemple) qui sont souvent modélisés par des smart objects dans les applications d'animation comportementale [KT98].

Par essence, un environnement est relatif à un agent donné. Cependant, un même environnement (la sirène d'alarme par exemple) peut concerner (influencer sur) plusieurs agents. Dès lors que l'environnement d'un agent n'est vu que comme un pourvoyeur d'information, il concerne d'une manière générale tous les agents qui peuvent accéder à cette source, c'est-à-dire la percevoir. Certaines sources d'information sont globales, c'est-à-dire qu'elles sont perceptibles par une grande partie des agents — souvent sur des critères de portée d'un signal sonore ou lumineux. D'autres sont locales (présence d'un obstacle, d'une situation de blocage) et certaines sont même internes (état d'avancement dans le parcours vers un objectif).

Nous allons voir dans ce chapitre comment un agent virtuel peut acquérir l'information de son environnement via la perception (comment un agent virtuel peut percevoir son environnement) ainsi que la communication (comment les agents communiquent entre eux).

3.2 Acquisition de l'information

L'acquisition de l'information représente un aspect fondamental dans la simulation des foules virtuelles. L'information collectée peut être interne ; elle représente alors l'information personnelle concernant la définition de l'agent en lui-même (sa personnalité, son état ou encore, son rôle dans l'environnement ou dans le groupe auquel il appartient). Dans [MT01][Par07], les auteurs ont simulé des agents dans une gare, qui disposent d'informations sur leur environnement, leur permettant d'effectuer des tâches bien précises (par exemple, acheter un ticket, ou encore téléphoner). L'information peut être aussi locale, obtenue grâce à la perception ou à la communication avec d'autres agents. Elle concerne les informations qui sont obtenues dynamiquement dans l'entourage de l'agent, comme par exemple celles qui lui permettent d'éviter un obstacle. Dans [PB06], les auteurs ont simulé l'interaction entre des agents à l'intérieur d'un bâtiment. Les agents n'ont pas une « grande » connaissance des lieux et c'est la communication entre eux qui leur permet de procéder collectivement à l'évacuation du bâtiment. L'information peut aussi provenir de l'utilisateur, via des opérateurs de contrôle dont il dispose (voir figure 3.1), par exemple, l'application de contextes globaux ou locaux sur la foule [SSN08].

L'acquisition de l'information nécessite un système permettant aux agents

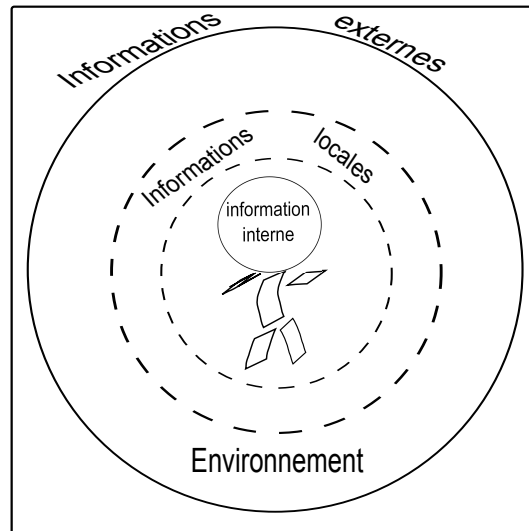


FIGURE 3.1 – Les informations collectées

de pouvoir percevoir l'environnement dans lequel ils se trouvent, ainsi qu'un système de communication entre les agents. En d'autres termes, dans chaque catégorie de foule, un agent doit disposer de moyens de perception pour pouvoir détecter (percevoir) d'autres agents ou obstacles dans son entourage¹. D'autre part, dans ce que nous appelons un groupe, chaque membre doit être au courant de l'état et des actions des autres membres pour pouvoir ajuster son propre comportement. Il doit les percevoir et/ou pouvoir communiquer avec eux. Une manière habituelle de modéliser la perception (en particulier la vision) est de définir une zone géométrique de perception associée à l'agent. Régulièrement, l'intersection entre cette zone et l'environnement est remise à jour et son contenu est scruté [PG06]. Une autre manière plus générique de modéliser à la fois la communication et la perception peut être de définir un médiateur entre deux entités : l'expéditeur et le récepteur dans une communication, ou l'objet perçu et l'agent qui perçoit dans le processus de perception. Le rôle du médiateur est de convoier l'information du fournisseur d'information (un obstacle, un agent...) à l'agent qui l'utilisera. Ce médiateur peut prendre la forme d'un message du fournisseur d'information aux agents indiqués [PAB07][TLCC01][LMM03]. Il est aussi possible que l'environnement « porte » l'information, c'est-à-dire que celle-ci soit écrite par l'émetteur « autour de lui », dans des zones où peut se trouver un récepteur dont la charge est de scruter l'environnement à l'emplacement qu'il occupe [PG06] [J95].

1. Ce qui peut être le rôle de l'espace d'influence.

3.2.1 La perception

La perception dans l'environnement constitue un critère essentiel du mécanisme de prise de décision.

Chaque agent effectue des actions via un mécanisme de prise de décision, ce dernier se base principalement sur les informations collectées durant un cycle de vie de chaque agent. Ces informations concernent l'évolution de l'état local et global de l'environnement dans lequel l'agent évolue.

Dans ce cadre, on peut distinguer deux types de perceptions : la perception visuelle et la perception basée sur les environnements informés.

3.2.1.1 Perception visuelle

Ce type de perception se base principalement sur l'état interne de l'agent, c'est-à-dire, que l'information n'est pas communiquée via l'environnement mais directement via les capacités internes et sensorielles de l'agent lui-même. Les techniques de perception pour déterminer comment un agent perçoit son environnement peuvent être classées en deux catégories : Approche zonale et sensorielle.

1. Approche zonale

Cette approche, connue aussi sous le nom d'espaces d'influence [PG06], consiste à entourer le personnage par une région qui représente la zone de perception de l'agent. Sa perception est alors limitée aux seuls objets se trouvant dans cette zone (région).

Chaque agent en mouvement a son propre espace d'influence. Dans [PAB07], les auteurs ont considéré un espace d'influence sous forme d'un rectangle (voir figure 3.2), où chaque agent reçoit des informations sur les obstacles dynamiques (d'autres agents), ou statiques qui se trouvent dans son espace. De cette manière, le coût des calculs est limité puisque les agents réagissent seulement aux obstacles qui se trouvent dans leur espace d'influence ; par exemple, lorsqu'une entité statique ou dynamique apparaît dans l'espace d'influence d'un agent (un autre agent par exemple), des forces tangentielles seront appliquées afin de modifier légèrement la direction du mouvement pour éviter la collision. Le mouvement local d'un agent est basé sur le modèle de Helbing [DHV00] qui décrit le comportement d'une foule humaine par une combinaison de forces physiques et socio-psychologique. Ce modèle permet de produire des mouvements réalistes tels que les courbures de trajectoires à l'approche d'une porte.

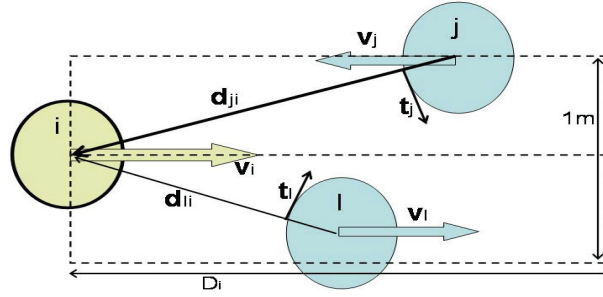


FIGURE 3.2 – Le modèle de l'espace d'influence rectangulaire [PAB07][PG06].

Le fait de limiter le champ de vision à une zone prédéfinie, diminue considérablement le coût de calcul. Par contre, augmenter le champ de vision signifie l'élargissement de la zone d'influence, ce qui implique plus de temps de calcul.

2. Approche sensorielle

Ce type d'approche consiste à placer différents types de capteurs sur des agents synthétiques. Par exemple, des capteurs pour l'odorat, ou encore la vue. Dans [OPOD10], les auteurs ont proposé une approche de vision synthétique pour la simulation de foule. Il existe un type de capteurs spécifique à chaque type de stimuli ou d'événements qui peuvent surgir dans l'environnement. Dans [BMMV04], les auteurs ont représenté ce type de perception par une fonction de perception qui, selon chaque type d'agent, définit les capacités perceptives de celui-ci. Funge et al. ont créé des agents virtuels qui sont dotés d'une vision synthétique capable de percevoir l'environnement dans lequel ils évoluent [FTT99].

Un tel processus appliqué à chaque entité de la simulation, alourdit considérablement le mécanisme de prise de décision. Ce qui fait que ce type de techniques s'applique seulement à des simulations contenant un nombre limité d'entités dynamiques.

3.2.1.2 Perception basée sur les environnements informés

Généralement, la terminologie des environnements informés désigne des environnements étiquetés (associant une information sémantique à des zones ou des objets.). La perception basée sur les environnements informés stocke l'information dans l'environnement. En d'autres termes, les agents récupèrent l'information relative à l'état actuel (local et global) de l'environnement directement dans la zone dans laquelle ils sont situés, ou encore dans un endroit

spécifique de l'environnement servant à indiquer aux agents les éventuels objets susceptibles de les heurter se trouvant dans leur voisinage [Sep07]. Wong et al. [WTML07] ont stocké les informations concernant les entités de l'environnement ainsi que la position des agents dans une carte 2D.

L'utilisation des environnements informés représente un bon concept pour remplacer les perceptions visuelles en raisons de leurs coûts, et cela en utilisant l'accès aux bases de données. Dans [DF92], les auteurs ont proposé un mécanisme permettant aux agents de déposer des marques dans l'environnement afin d'informer les autres agents d'une évolution ou de l'état de la résolution d'un problème. Cependant, leur technique souffre d'un problème d'efficacité machine et nécessite de très nombreux agents afin que les informations puissent se propager rapidement et efficacement.

3.2.2 La communication

La communication entre agents est l'autre moyen d'obtenir des informations — par rapport à la perception. Dans [PB06], les auteurs ont développé un modèle basé sur la communication entre les agents (formés et non formés), où chaque agent peut développer sa carte mentale au fur et à mesure qu'il explore le bâtiment à l'aide d'un processus de communication entre les individus se trouvant dans les pièces. Se basant sur des études de psychologie pour obtenir des comportements proches de la réalité, ils ont simulé deux foules dotées de deux degrés de communications différents (un degré élevé² et l'autre faible). Ils ont remarqué une différence dans le temps d'évacuation (le temps le plus court correspondant au degré de communication le plus élevé). En effet, lorsque le degré de communication est grand, cela signifie qu'un agent a plus de chance d'avoir des informations venues des autres agents ; par exemple, ils lui communiquent directement le chemin le plus court pour sortir. Mais il est nécessaire que la foule soit volumineuse pour que l'aspect « contributif » de l'approche soit opérant.

Plus spécifiquement, il est aussi possible que des agents soient en relation permanente (c'est le cas dans un groupe). Une possibilité de représentation est alors de matérialiser le groupe par une entité destinée à jouer le rôle d'environnement pour tous les membres du groupe (voir technique précédente) le fournisseur écrit des informations dans cette zone, et les autres agents y lisent ce qui les concerne ([JNR98]).

Une autre technique permettant la communication entre agents est celle des tableaux noirs (blackboard).

2. Un nombre important d'agents expérimentés possédant une parfaite connaissance des lieux et ayant pour rôle de communiquer aux autres agents la sortie la plus proche

Les tableaux noirs représentent une structure de données partagées. Les agents peuvent utiliser un tableau noir pour communiquer entre eux, en écrivant simplement sur la structure de données. Le premier système de tableau noir est le système HEARSAY-II [EL80]. Il représente à la fois une solution spécifique au problème de compréhension de la parole, et un cadre général pour la coordination de processus indépendants. Par la suite plusieurs systèmes ont été développés. On trouve les travaux de Lesser et Corkill [CL83] qui ont utilisé deux tableaux, l'un pour les données, et l'autre pour les objectifs des agents. Dans [Nii86], les auteurs ont présenté une description des modèles basiques de tableaux noirs en distinguant les trois composantes principales suivantes : Les sources de connaissances, le contrôle et la structure de données des tableaux noirs.

1. Les sources de connaissances

Les connaissances requises pour résoudre le problème sont divisées en sources de connaissances, qui sont séparées et indépendantes.

2. Contrôle

Les données d'état pour la résolution de problèmes sont conservées dans une base de données globale. Les sources de connaissances produisent des changements au sein des tableaux noirs (figure 3.3). Ces changements constituent des états successifs (étapes) dans la résolution des problèmes. La communication et l'interaction entre les sources de connaissances ont lieu uniquement à travers les tableaux noirs.

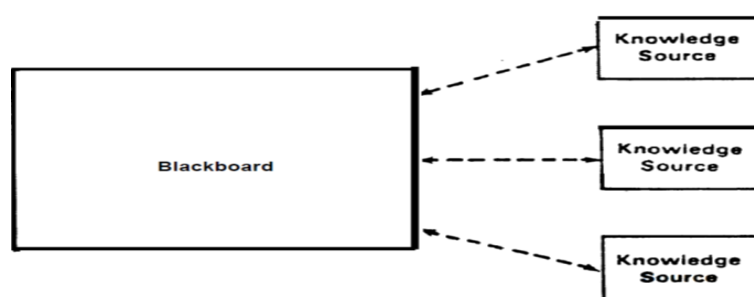


FIGURE 3.3 – Modèle de tableau noir [EL80].

3. La structure de données des tableaux noirs

Il y a une base de données globale appelée tableau noir, et il y a des sources de connaissances indépendantes qui répondent de façon opportuniste aux changements au sein des systèmes des tableaux noirs. Les sources de connaissances s'auto-activent et réagissent aux changements sur le tableau noir.

3.3 Représentation de l'environnement

Nous avons vu que l'environnement joue un rôle crucial pour apporter des informations aux agents. La représentation de l'environnement est donc un point important dans la constitution des applications d'animation comportementale et notamment dans le domaine de la robotique [GGL⁺98]. Parmi les principes de représentation étudiés, certains sont dédiés aux environnements simples (champs de potentiels, décomposition en une grille de cellules). D'autres traitent des environnements complexes (décomposition en utilisant la triangulation de Delaunay ou encore modèles multi-couches).

3.3.1 Représentation des environnements simples

3.3.1.1 Représentation basée sur les grilles

Les grilles sont des modèles simples à mettre en application et qui induisent des temps de traitement courts. Ils permettent de traiter aisément les problèmes de collision (les agents ne peuvent se déplacer que lorsque la cellule adjacente est libre [Che04][KS03][TLCC01].). Ils peuvent aussi contenir des informations calculées par le système ou laissées par d'autres agents (informations de direction pour atteindre un objectif par exemple [LMM03].). La décomposition à base de grille est utilisée par les modèles d'automates cellulaires dans lesquels l'environnement est représenté par une grille de cellules uniformes. Chaque agent occupe une cellule, et se déplace entre les cellules selon le système de modélisation. Chaque cellule contient des variables qui sont mises à jour à régulièrement. Leur valeur est fonction des valeurs des variables des cellules voisines par l'application de règles locales situées aussi dans les cellules [Wol83]. La taille des cellules détermine la densité de la population par mètre carré. La densité maximal où les agents peuvent encore se déplacer est fixée à 7,4 personnes / mètre carré [Vas04]. L'utilisation des grilles de cellules est efficace pour les petites et moyennes foules (un nombre petit ou moyen d'agents), mais pour les foules à grande densité cela pose quelques problèmes. Le caractère intercellulaire des contraintes provoque des

interblocages — un déplacement ne pouvant être autorisé à partir d'une cellule A que si au moins une cellule voisine est libre dans un sens proche de celui du déplacement, laquelle ne peut être libérée que si, entre autres, la cellule de A est libre. Pour des raisons voisines, il y a aussi une mauvaise exploitation des espaces (figure 3.4), cela donne un effet peu réaliste (des espaces inaccessibles visibles à l'intérieur des foules quand ces dernières sont bloquées). Par ailleurs, les trajectoires ne sont pas toujours réalistes, étant données l'aspect discret des angles de déplacement possibles : 0, 45, 90, 135, 180, -45, -90, -135. La longueur d'une trajectoire peut ainsi être artificiellement accrue.

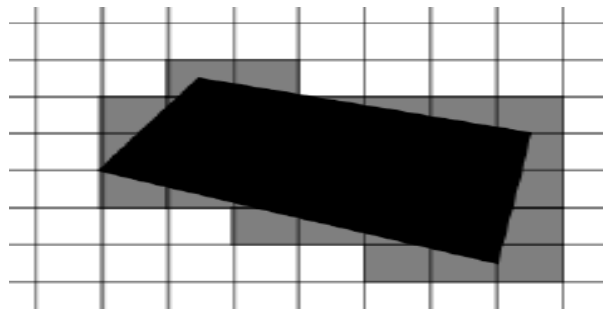


FIGURE 3.4 – Représentation en utilisant une grille de cellules

3.3.1.2 Représentation basée sur les champs de potentiels

Introduite par Khatib [Kha86], les méthodes basées sur les champs de potentiels sont des méthodes simples, où chaque point de l'environnement est affecté d'un gradient de force [RKBB94][TCP06][TBS04]. Les obstacles de l'environnement sont caractérisés par une force de répulsion et les buts par une force d'attraction (figure 3.5). Ainsi, l'objet essaie de se déplacer vers son objectif tout en étant repoussé par les obstacles. En raison de l'utilisation des propriétés locales seulement, l'objet peut se déplacer dans la mauvaise direction, ce qui peut entraîner une situation de blocage.

3.3.2 Représentation des environnements complexes

3.3.2.1 Représentation basée sur les Triangulations de Delaunay

Les triangulations de Delaunay offrent une représentation de l'environnement plus précise que les représentations à base de grille. En effet, avec la représentation à base de grilles pour les environnements complexes, il peut

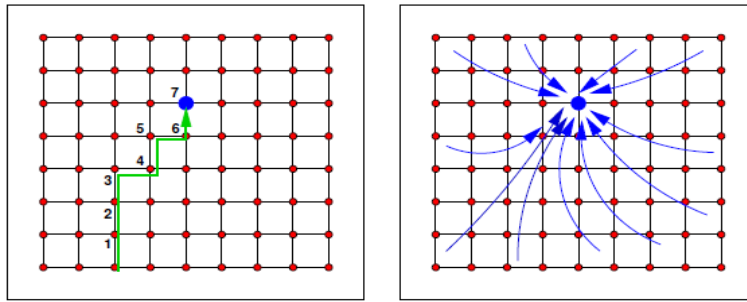


FIGURE 3.5 – La figure de gauche montre la trajectoire des déplacements observés et les potentiels attribués. La figure de droite montre des forces résultantes des champs de potentiels [TBS04].

y avoir des pertes d'espace dues au fait que les cellules sont de tailles uniformes — ce qui donne une représentation approximative de l'environnement (figure 3.4). La triangulation de Delaunay crée un ensemble de triangles en fonction d'un degré de précision qui dépend de la courbure de l'espace [BY98]. Le fait que chaque point est relié à son plus proche voisin, cela permet de l'utiliser pour représenter l'espace de navigation (figure 3.6).



FIGURE 3.6 – Capture d'écran d'une application interactive où un humain virtuel est capable de marcher à n'importe quel endroit sélectionné sans entrer en collision avec des boîtes à l'intérieur de la salle. Les boîtes sont également capables de changer de position. En bas à droite sa représentation correspondante par triangulation de delaunay. [KBT03].

Cependant, dans les environnements très complexes, la représentation devient compliquée (nombre de cellules linéaire en fonction du nombre de sommets en

entrée). C'est le cas dans les environnements très bosselés, avec des degrés de courbures important.

3.3.2.2 Représentation multi-couches

Les modèles multi-couches ont pour objet de représenter les environnements ayant des structures topologiques complexes, tels que les environnements extérieurs. Dans [JXM⁺09], les auteurs présentent un modèle multi-couche subdivisé en trois couches (figure 3.7) :

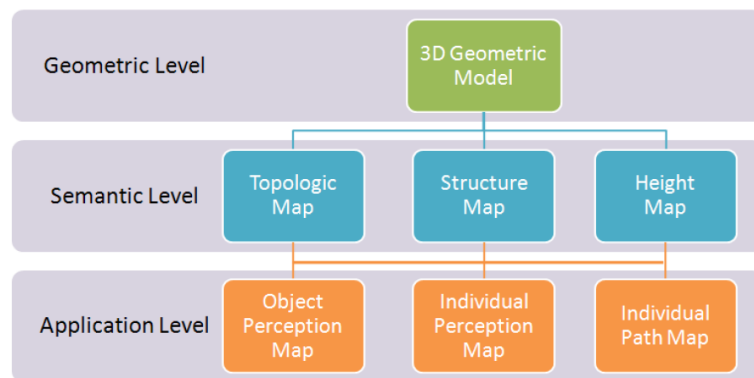


FIGURE 3.7 – Les trois niveaux hiérarchiques « géométrique, sémantique, application » [JXM⁺09]

1. Niveau géométrique

Le niveau géométrique représente le noyau de ce modèle. Il se base sur un modèle géométrique en 3D de l'environnement, lequel, est utilisé pour l'affichage et l'extraction des informations sémantiques.

2. Niveau sémantique

La couche sémantique est représentée par :

- des cartes de structures, où tous les objets placés dans la même région sont inclus dans un bloc. Tous les blocs ainsi que leurs objets sont organisés en cartes de structures.
- des cartes topologiques qui sont utilisés pour stoker et récupérer les relations entre les régions.
- des cartes de hauteurs qui sont utilisées pour stocker l'information concernant les élévations de la surface.

3. Niveau application

Afin de permettre une interaction entre les piétons et l'environnement, l'information récupérée au niveau sémantique permet de générer des cartes de haut niveaux telles que :

- la carte de perception des objets dans laquelle les blocs sont divisés en sous régions en utilisant une grille de cellules uniformes ; le but est de limiter la recherche des objets à une liste de cellules au lieu de tout le bloc.
- la carte de perception individuelle, où, comme pour la carte de perception des objets, une grille de cellules uniformes est associée à chaque bloc afin de stocker les informations concernant tous les individus se trouvant dans le bloc.
- le plan du parcours individuel, où chaque agent a un objectif initial. La carte topologique à base de blocs est utilisée pour générer une carte de voies individuelles.

3.3.3 Les algorithmes de navigation

Les algorithmes de navigation sont très importants dans l'animation comportementale de foules dès lors qu'il est rare de rencontrer des simulations (dans ce domaine) où les agents ne se déplacent pas. Le choix de ces algorithmes dépend fortement de la représentation de l'environnement. Dans beaucoup de cas, il s'agit de trouver le chemin le plus court pour atteindre un but précis (Par exemple, chercher la sortie la plus proche dans le cas d'une évacuation). Dans [BLA03] la modélisation de comportements complexes de groupes passe par l'utilisation de l'équivalent d'une carte routière, qui contient des informations globales sur l'environnement, telles que des informations topologiques et des poids évolutifs sur les noeuds et arêtes de la carte. Ces derniers sont par exemple utilisés pour indiquer à des agents confrontés à un carrefour (noeud) quelle direction est la meilleure pour atteindre leur objectif (mécanismes d'attraction-répulsion). L'efficacité relative des parcours des agents est analysée pour faire évoluer dynamiquement les poids qui serviront aux futurs parcours (figure 3.8).

Dans [SKG05], les auteurs ont présenté un algorithme capable de générer les trajectoires d'un ensemble d'agents (ayant un point de départ et point d'arrivée). L'algorithme permet de créer une trajectoire tout en évitant les collisions avec les autres objets et agents. Il se base sur un graphe représentant l'équivalent d'une carte de cheminement, où le choix de l'orientation des agents dans l'environnement est basé sur un système probabiliste (figure 3.9).

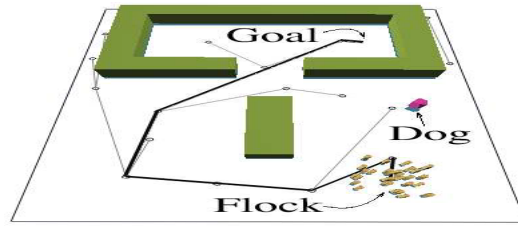


FIGURE 3.8 – La force répulsive du chien est plus élevée, ce qui oblige le groupe à choisir un autre chemin [BLA03]

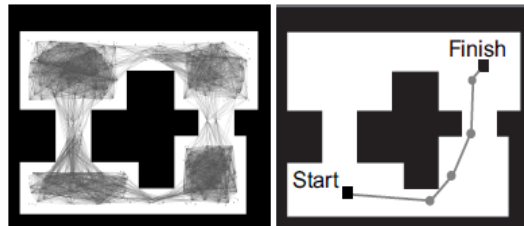


FIGURE 3.9 – A gauche : Une carte de cheminement probabiliste. A droite : Une série de points de passages qui relie deux positions. [SKG05]

Dans [LD04], les auteurs ont présenté un algorithme pour la planification de chemin permettant d'optimiser la trajectoire des agents (figure 3.10), et de gérer la visibilité entre les différentes entités afin d'éviter les collisions. Sa structure est basée sur une topologie construite à partir de la base de données géométriques de l'environnement virtuel.

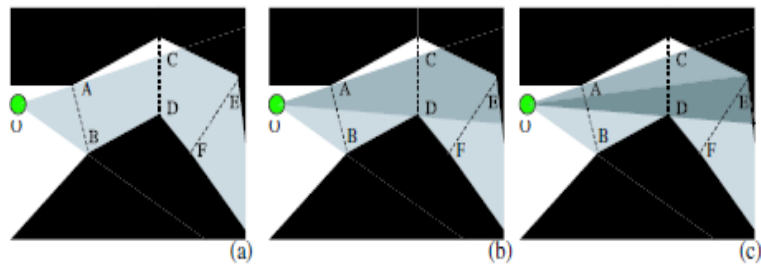


FIGURE 3.10 – Optimisation visuelle de la trajectoire [LD04]

D'autres systèmes de planifications existent, par exemple, dans [LRDG90], les auteurs ont proposé un algorithme permettant de trouver le chemin le plus court parmi un ensemble de chemins entre deux points correspondants

à deux nœuds dans un graphe. Dans [Nil93], les auteurs ont proposé un algorithme pour trouver le plus court chemin où à chaque nœud visité, un coût qui représente le nombre de nœuds qui ont mené à lui lui sera affecté. Dans [MYMT07], les auteurs ont proposée une architecture hybride pour la navigation des agents dans les environnements qui exploite plusieurs algorithmes selon les points d'intérêts des régions.

3.4 Conclusion

On a vu au cours de ce chapitre l'importance de la perception ainsi que la communication dans l'acquisition de l'information. La perception visuelle se base sur les capacités internes de l'agent lui-même, ce qui offre un effet visuel très réaliste, notamment, dans la vie artificielle [FTT99]. Les techniques de ce type utilisent des mécanismes complexes et leur mise en oeuvre est algorithmiquement très coûteuse. Une autre façon de représenter la perception consiste à définir une zone de perception visuelle [PAB07][PG06], ce qui permet de limiter les calculs à ce qui interagit avec cette zone. Cependant, bien que cette méthode soit plus facile à mettre en oeuvre que la précédente, elle nécessite aussi beaucoup de calculs pour un nombre assez conséquent d'entités et la qualité de la perception est réduite (champ visuel peu réaliste). L'utilisation des environnements informés permet de simplifier le mécanisme de perception, et de diminuer la quantité, ce qui permet de l'appliquer à des environnements contenant des foules volumineuses. L'utilisation des environnements informés nécessite une représentation de l'environnement. Pour ce faire, nous avons vu plusieurs représentations, qui vont de la représentation par une grille uniforme aux champs de potentiels pour les environnements simples, jusqu'à la représentation en utilisant la triangulation de Delaunay ou la représentation multi-couches pour les environnements complexes. Chacune des méthodes a ses avantages, les premières, sont simples à mettre en oeuvre. Par contre, pour les endroits complexes avec éventuellement des courbures, on constate une perte d'espace lors de la représentation. Les méthodes de la seconde catégorie représentent mieux les environnements complexes mais sont plus coûteuses en termes d'informations. L'architecture hybride combine les différentes techniques selon les zones de l'environnement. Mais le problème des surfaces complexes existe toujours et la combinaison de plusieurs architectures alourdit le système. Il est plus judicieux de choisir une technique spécifique selon l'environnement à simuler (selon sa complexité typologique).

Chapitre 4

Modèles comportementaux de foules

Sommaire

4.1	Introduction	40
4.2	Les smart objects	40
4.2.1	Propriétés physiques de l'objet	41
4.2.2	Modélisation des formes d'interaction	41
4.2.3	Exemples d'utilisation des smart objects	42
4.2.4	Discussion	46
4.3	Système MACES	48
4.4	Système HiDAC	52
4.5	Le système de classeurs (LCS)	54
4.6	Le modèle ViCrowd	56
4.6.1	Types de comportements dans ViCrowd	57
4.6.2	La structure des groupes dans ViCrowd	58
4.7	Modèle de comportements scalables	58
4.8	Le système MAGS	60
4.9	Synthèse	61
4.9.1	Smart objects	62
4.9.2	MACES	63
4.9.3	HiDAC	64
4.9.4	Le système de classeurs (LCS)	64
4.9.5	Le modèle ViCrowd	65
4.9.6	Modèle de comportements scalable	66
4.9.7	Système MAGS	67
4.10	Conclusion	68

4.1 Introduction

Les modèles comportementaux existants ont pour objet de simuler le comportement des entités vivantes qu'elles soient humaines ou animales. La plupart de ces modèles représentent des travaux effectués dans des domaines et situations spécifiques, tel que l'évacuation ou encore l'interaction avec des objets de l'environnement. Le comportement des entités peut être scripté ou soumis à un processus décisionnel afin de réaliser des tâches individuelles ou collectives.

Ce chapitre est consacré au passage en revue de plusieurs modèles de simulation de comportements d'agents. Les modèles seront évalués par apport aux critères de simulation vus au chapitre 2.

4.2 Les smart objects

Un smart object est un objet qui contient toutes les informations — et souvent les comportements — qui permettent aux agents qui le perçoivent d'interagir avec lui [KT98]. Un agent qui perçoit un smart object peut alors adapter son comportement pour tenir compte de sa présence et, éventuellement, de son état. Selon un autre point de vue, un smart object peut être considéré comme un objet qui induit un comportement chez les agents qui le perçoivent.

Par exemple, un agent prêt à sortir d'une pièce, peut commencer à marcher au hasard dans une pièce jusqu'à la découverte d'un smart object correspondant à une porte.

Entre autres, un smart object contient le « mode d'emploi » de l'objet, c'est-à-dire l'ensemble des actions qui peuvent être effectuées sur lui (par exemple, les actions d'ouverture et de fermeture d'une porte). L'utilisation du modèle des smart objects permet de distinguer clairement les comportements généraux d'un agent et ceux qui sont spécifiques aux situations particulières représentées par les smart objects. Cette séparation entre les comportements généraux et spécifiques permet d'alléger la description des agents et apporte une souplesse dans la modélisation.

4.2.1 Propriétés physiques de l'objet

Les smart object contiennent des informations sur celles de leurs propriétés qui sont utiles à l'interaction potentielle avec des agents. Par exemple leur forme, leur poids ou leur centre de masse.

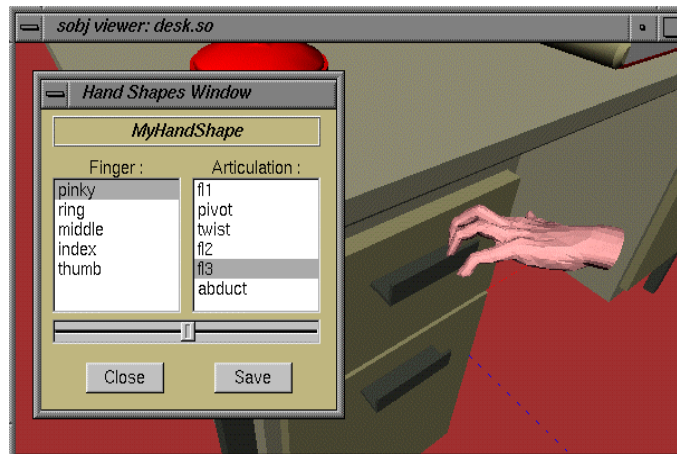
L'approche smart objects introduit les caractéristiques principales suivantes dans la simulation :

1. **La délocalisation de l'information ;** l'information concernant l'interaction avec un objet est stockée dans ce dernier. Il est ainsi possible d'enrichir une modélisation sans modifier la description des agents.
2. **Une souplesse de modélisation.** Pour traiter des situations similaires situées à divers emplacements dans l'espace de la simulation, il est possible de définir un premier smart-object et d'en créer d'autres sur son modèle, en adaptant éventuellement leur description [KT98]. Il est aussi potentiellement possible de créer des classes de smart objects et de les raffiner selon les principes classiques de la modélisation objet.
3. **La décentralisation du contrôle de l'animation.** La décentralisation du contrôle de l'animation. Même si, techniquement, l'agent accède au contenu d'un smart object, tout se passe comme si ce dernier induisait des comportements chez les agents. En effet, ce sont les smart objects qui contiennent les informations (données physiques ou état par exemple) qui permettent l'interaction avec eux. Les smart objects procurent aussi leur propre « mode d'emploi », c'est-à-dire les procédures qui permettent d'interagir correctement avec eux. Par exemple, lors de l'interaction avec un casier (figure 4.1), le smart object qui le représente doit indiquer à l'agent où se positionner, ou encore les différents éléments qui le constituent (par exemple, sa poignée). Il indique aussi son mode d'emploi (ouverture de la porte du casier vers l'intérieur ou l'extérieur), et le comportement que l'agent utilisateur doit adopter (par exemple, tendre la main vers la poignée et la faire tourner).

4.2.2 Modélisation des formes d'interaction

En s'inspirant de [KT02], il est possible de distinguer les informations suivantes à prendre en compte pour interagir avec un objet :

1. **Propriétés intrinsèques utiles pour l'interaction :** ce sont certaines des propriétés qui sont relatives à la constitution de l'objet, qui le définissent, par exemple la description des mouvements de ses pièces mobiles ou encore la description physique de l'objet quand elle joue

FIGURE 4.1 – Interaction avec un smart object *casier* [KT02].

un rôle dans l'interaction (poids, encombrement dans le cadre d'une manutention).

2. **Propriétés extrinsèques compatibles avec l'interaction** : ce sont les propriétés qui caractérisent l'état de l'objet. Certains objets sont immuables (pans de murs) mais beaucoup peuvent passer d'un état à un autre (feu tricolore, ascenseur, imprimante, porte,...). La possibilité d'interagir avec l'objet dépend de son état courant. Les parties de l'objet dédiées à l'interaction (l'interface de l'objet). Celles qui sont destinées à le manipuler (poignées, boutons,...).
3. **Le « mode d'emploi » de l'objet** : L'attitude à adopter (positionnement, posture,...) et les actions à effectuer pour pouvoir interagir avec l'objet (ouvrir, avancer, puis refermer une porte).

Via sa description, un smart object contient tout ou partie de ces informations. Celles-ci doivent être accessibles aux agents qui interagissent avec lui.

4.2.3 Exemples d'utilisation des smart objects

Voilà quelques exemples d'application du modèle des smart-objects :

1. Dans [KT98], est décrite une application permettant à l'utilisateur de contrôler un agent pour le faire interagir avec un smart object en lui soumettant une instruction via un interpréteur de commandes. Le programme parcourt tous les smart objects de la scène pour identifier le plus proche d'entre-eux qui référence l'instruction, et exécute le comportement correspondant. Dans l'exemple de la (figure 4.2), une simple instruction « fermer » peut

être donnée. Après une recherche, le programme identifie le smart object ayant le comportement « fermer » le plus proche disponible, et l'exécute.

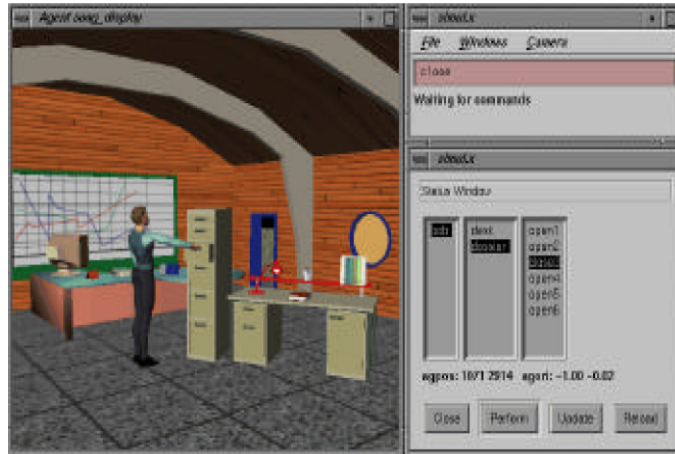


FIGURE 4.2 – Une application basée sur du texte [KT98].

2. ACE¹ [KMCT00] est une plateforme qui donne la possibilité de :
- Créer et de placer des agents virtuels ainsi que des smart objects.
 - Appliquer des moteurs de mouvements aux agents (par exemple, pour les déplacements, ou encore les expressions faciales), lesquels peuvent se déclencher en parallèle selon les situations (par exemple, marcher et parler en même temps).
 - Déclencher l'interaction entre les agents et les smart objects.
 - Retourner pour un agent une liste avec tous les objets perçus à l'intérieur de son champ de vision.

ACE a été appliquée à un environnement virtuel (un laboratoire d'informatique [KMCT00]) contenant un agent ainsi que 90 smart objects. Dans cet exemple de simulation, les auteurs ont défini un scénario au cours duquel un agent (une femme) pénètre dans le laboratoire un jour de congé afin de voler des informations. Elle entre dans la pièce, allume les lumières, ouvre un livre (figure 4.3) où elle trouve la disquette qu'elle voudrait voler, puis elle prend la disquette, éteint les lumières et sort du laboratoire. L'agent ici va essayer d'atteindre son but en étant influencé par l'évolution de son état émotionnel interne.

La plateforme ACE a aussi été appliquée à une animation où un agent déambule dans une pièce tout en collectant des informations concernant la position

1. agent common environment



FIGURE 4.3 – Un agent (voleur) interagissant avec des smart objects dans un laboratoire. [KMCT00].

des smart objects qui se trouvent dans son environnement [SKT01] (tel qu'un hamburger ou encore du café). Il est doté d'états internes (états émotionnels) qui évoluent au cours du temps et de la situation concernant par exemple l'envie de manger ou de se reposer. Par exemple, si l'agent a faim, il se dirige vers le smart object « hamburger » et exécute les actions imposées par ce dernier.

3. SOMOD² est une application qui a été développée principalement pour modéliser les smart objects. Elle permet d'importer les composant géométriques des objets, et offre la possibilité de réutiliser les smart objects existants ainsi que leurs fonctionnalités précédemment définies [KT02],[Kal04]. SOMOD permet aussi d'afficher via une interface graphique les détails des actions (script) lors de l'interaction entre l'objet et l'agent. Par exemple, pour l'action « ouvrir la porte » SOMOD affiche les différentes étapes pour effectuer cette tâche (se mettre en position, tendre la main en direction de la poignée,...)(voir par exemple dans la figure 4.4 qui représente une simulation d'un agent en interaction avec des toilettes, où à chaque action l'ensemble des étapes correspondant sera affichées [SKT01]).

SOMOD permet de programmer les comportements simples (primitifs), lesquels peuvent être composés pour créer des comportements plus complexes. Dans l'exemple de comportement d'ascenseur [KT02] de la figure 4.5 (à gauche), il y a une interaction simple afin d'accéder à l'ascenseur, où l'ensemble du processus de prise de l'ascenseur est programmé à l'intérieur des états enter_12 et

2. Smart object modeler

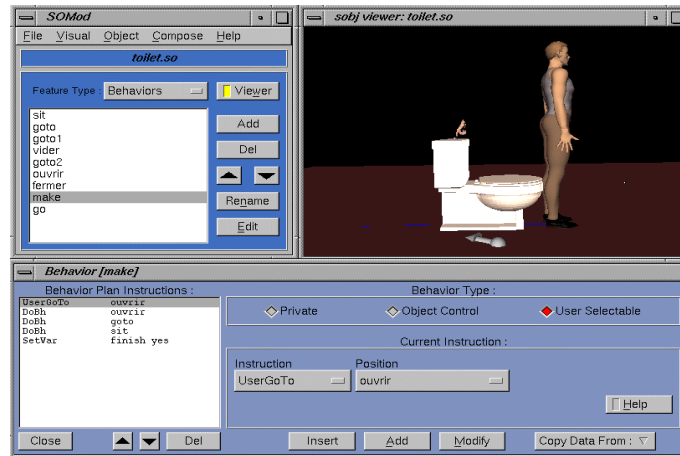


FIGURE 4.4 – Interaction avec un smart object « toilette »[SKT01].

enter_21. Dans la figure 4.5, l'état représenté par un double cercle est *l'état courant*, et les deux rectangles montrent l'interaction requise pour changer d'état. Par exemple, pour changer de floor_1 à l'état floor_2, l'interaction enter_21 est exigée. La figure 4.5 (à droite) montre un état machine plus complexe, qui modélise les fonctionnalités de l'ascenseur en tenant compte des états intermédiaires possibles. La figure 4.5 montre également que des comportements sont associés à des états ; une fois que l'objet se trouve dans un état alors un comportement se déclenche.

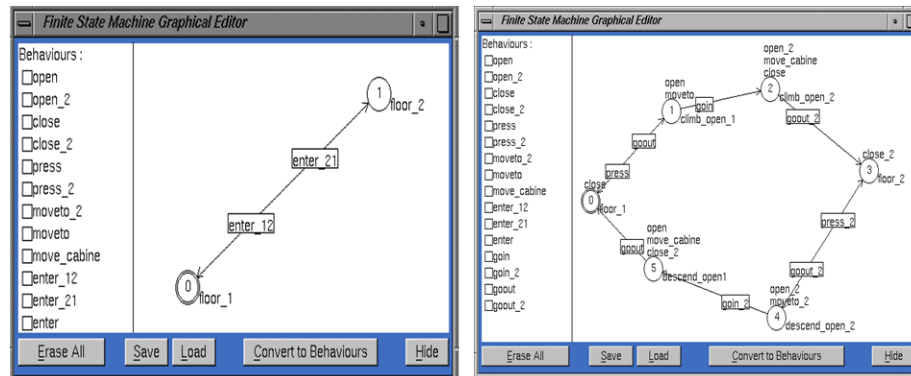


FIGURE 4.5 – (à gauche) Un état machine simple pour une fonctionnalité de l'ascenseur. (à droite) Un état machine plus complexe pour l'ascenseur, où des états intermédiaires sont considérés [KT02].

SOMOD offre la possibilité de programmer les comportements d'objets simples.

Cependant, lorsque les comportements deviennent complexes, la quantité des comportements secondaires (les états intermédiaires) augmente considérablement, ce qui implique qu'il sera difficile de garantir la concordance au but final. En plus SOMOD n'offre pas de moyens pour préciser une priorité lors de l'interaction d'un objet avec plusieurs personnes en même temps.

D'autres modèles utilisent l'approche des smart objects dans leurs simulations. Dans [PDMO03], les auteurs définissent des objets ayant les caractéristiques des smart objects auxquels ils ajoutent selon la nature du smart object une autre caractéristique « point d'attention » qui permet d'attirer l'attention et particulièrement le regard des agents, tel qu'un agent de sécurité qui surveille le smart object « moniteur de surveillance ». Les auteurs ont notamment présenté un exemple avec des agents (barmans et consommateurs) en interaction avec des smart objects « bar (le comptoir) » qui contient un point d'attention afin que le consommateur se serve immédiatement une fois que sa commande lui a été servie.

Dans [ACT05], les auteurs se basent sur le principe des Smart Objects en prenant en compte deux nouvelles caractéristiques : « les pré-conditions » et « les effets des actions » (utilisation du formalisme STRIPS pour planifier l'action). En d'autres termes, avant de pouvoir interagir avec un objet, il faut que l'agent en question satisfasse à certaines conditions. Par exemple, pour déplacer un smart object ayant un poids important, ce dernier impose au moins deux agents pour pouvoir exécuter l'action de déplacement. Une fois qu'une interaction a été effectuée avec un smart object, l'effet de l'action peut conduire à la suppression de ce smart object de l'environnement. Par exemple « un hamburger » qui sera mangé va disparaître de l'environnement. Il est aussi possible qu'un smart object soit créé. Par exemple une canette après l'interaction avec un distributeur automatique de boissons.

4.2.4 Discussion

Les modèles fondés sur les smart-objects permettent de gérer les comportements d'un agent confronté à un objet, en général un objet sur lequel l'agent peut opérer (qu'il peut actionner). Ces comportements sont décrits dans le smart object lui-même, qui les transmet à l'agent. Les applications existantes n'utilisent la technique des smart-objects que pour traiter le cas spécifique de la relation d'agent à objet. Elle comporte des techniques complémentaires pour gérer les relations d'agent à agent ou d'agent à objet « statique » (un obstacle par exemple). Cela induit une dispersion conceptuelle des modèles (beaucoup de notions différentes et sophistiquées à considérer conjointement). Or il nous semble possible de définir un modèle sur le même principe que les smartobjects (déportation des comportements dans les « objets » qui les

causent) qui pourrait traiter l'essentiel des comportements d'un agent. Un tel modèle serait conceptuellement beaucoup plus simple que les modèles existants.

Par ailleurs, les comportements inclus dans un smart object donné sont destinés à être adoptés par tous les agents confrontés à ce smart object. Cela peut donner un caractère stéréotypé et uniforme aux comportements des agents (mauvais réalisme microscopique). Mais il est tout à fait possible de différencier les agents en les dotant d'une « personnalité » qui leur donne une plus ou moins grande « résistance » à l'influence d'un smart object.

Ce sont ces deux idées qui sont l'objet de notre travail et que nous développerons au chapitre suivant. L'approche des smart objects est beaucoup plus orientée vers le réalisme graphique et visuel que vers le réalisme comportementale à long terme [Par07] avec un processus décisionnel et des buts. Afin de pouvoir simuler les comportements de foules, d'autres approches sont nécessaires qui sont caractérisées par :

1. le nombre possible d'agents durant la simulation, on trouve dans ce cas le modèle HiDAC³ proposé par Pelechano et al. [PG06] qui a été conçu pour simuler le mouvement local des agents dans les larges foules, où le but est de trouver son chemin.
2. le niveau d'apprentissage des agents (intelligence ou encore leurs capacités à s'adapter), tel que le système de classeurs (LSC)⁴. Introduits par John Holland (Holland, 1976), qui utilise un système à base d'algorithmes et de règles qui sont capables d'évoluer au cours de la simulation.
3. le niveau de contrôle, par exemple les automates à états finis. Dans ce cas, on trouve le système HPTS⁵ [Don01], où un but est représenté par un ensemble d'états.
4. leur capacité à communiquer entre eux. Le système MACES [PB05] illustre bien cette approche. Comportant deux niveaux, MACES est un système de simulation de foules destiné à l'évacuation et basé sur la communication entre les agents expérimentés et agents non expérimentés, dont le but est de permettre une évacuation plus efficace.

3. High-Density Autonomous Crowds

4. Learning Classifier Systems

5. Hierarchical Parallel Transition System

4.3 Système MACES

MACES⁶ est un système de simulation pour l'évacuation de foules se situant dans des bâtiments. Le modèle met en jeu les notions de leader et d'expérience. Les agents leaders (par exemple des pompiers) — qui maîtrisent leur stress — et sont expérimentés (ils ont une connaissance complète de la structure du bâtiment) peuvent guider les autres agents non leaders et non expérimentés vers la sortie. Le modèle met aussi l'accent sur la communication entre agents (communication du chemin le plus court menant à une sortie ou encore d'un chemin qui ne mène pas vers une sortie afin de ne pas l'emprunter). Pour représenter leur connaissance, les agents sont dotés d'une « carte mentale ». Chaque agent a sa propre carte mentale de l'environnement (construction ou bâtiment), qui est représentée par un graphe (où les nœuds représentent des pièces, et les arcs représentent les chemins entre les pièces) [PB05]. Chaque agent développe sa carte mentale au fur et à mesure qu'il explore la construction et à l'aide d'un processus de communication entre les agents. Comme beaucoup de modèles comportementaux, MACES offre la possibilité de choisir le caractère des agents, ce qui permet d'avoir une certaine variété dans les comportements. Dans MACES, on trouve trois grands genres d'individus qui peuvent apparaître dans une foule lors d'une évacuation :

- les leaders expérimentés (ou formés).
- les leaders non expérimentés.
- les non leaders.

Les agents leaders peuvent être expérimentés ou non, c'est-à-dire, qu'ils peuvent avoir une connaissance complète de l'environnement ou pas.

Nous avons vu au début de ce paragraphe le cas du leader expérimenté. Un leader non expérimenté est un agent qui peut aider les autres durant le processus d'évacuation en cherchant de nouveaux chemins, tout en faisant évoluer sa carte mentale de l'environnement à travers sa propre expérience ou via la communication. Les agents non leaders modélisent les personnes qui perdent leurs moyens lors des situations d'urgences où ils sont incapables de prendre la bonne décision [McG70] et préfèrent suivre les autres. Contrairement aux agents leaders qui représentent les personnes qui gardent leur calme (maîtrisent leur stress) lors des situations d'évacuation d'urgences.

La communication entre agents est primordiale pour obtenir une évacuation rapide et efficace d'un bâtiment lors d'une situation d'urgence. Dans [PB06], les auteurs ont réalisé deux expérimentations, dans lesquelles ils ont simulé deux foules en utilisant pour chaque foule un degré de communication différent. Le degré de communication élevé réfère à un nombre important d'agents

6. Multi-Agent Communication for Evacuation Simulation

leaders et expérimentés et le degré faible renvoie à la situation inverse. Lorsque un degré de communication est grand, les agents ont plus de chance d'avoir des informations venues des autres agents à propos de certains chemins qui ne mènent pas à des sorties, et donc l'agent peut éliminer ces chemins de sa carte mentale, et il gagner ainsi du temps pour trouver le chemin le plus efficace pour sortir. Ou tout simplement, les agents expérimentés lui communiqueront directement un chemin qui mène vers une sortie. Les résultats présentés montrent une différence dans la durée d'évacuation (beaucoup plus courte lorsque le degré de communication est élevé). Cela paraît logique, puisque les agents expérimentés ont une carte mentale exacte du plan du bâtiment, et ils peuvent ainsi aider les autres agents.

Ils affirment que le nombre petit ou grand d'agents leaders mais non expérimentés (qui ne connaissent pas le plan du bâtiment) n'influe pas beaucoup sur le temps d'évacuation. Naturellement le temps d'évacuation sera plus grand que celui mis par des agents leaders expérimentés. Les agents leaders non expérimentés vont prendre leurs propres initiatives (décisions) et vont faire évoluer leur carte mentale du plan du bâtiment au fur et à mesure qu'ils explorent ce dernier à la recherche d'un chemin pour sortir ; s'ils tombent sur un chemin bloqué, alors ils reviennent sur leurs pas, et ils éliminent ce chemin de leur carte mentale. Ces agents vont aussi aider les agents qui sont ni leaders ni expérimentés, et qui ne peuvent pas prendre leur propre décision. Dans ce cas [PB06], ils ont observé d'après les expérimentations que d'utiliser un taux relativement petit de leaders non expérimentés contre un plus grand nombre d'agents non leaders et non expérimentés aura pour effet la formation de groupes avec une taille importante, constitués d'agents non leaders et guidés par un agent leader. Contrairement, au cas où il y a beaucoup d'agents leaders et peu d'agents non leaders (figure 4.6).

MACES se compose de deux niveaux : un haut niveau et un bas niveau.

1. Le haut niveau de MACES comporte trois étapes principales :
 - communication et partage des informations.
 - mettre à jour la carte mentale.
 - choisir le chemin le plus court.

La première étape correspond à la communication à l'intérieur des pièces à propos par exemple d'un chemin bloqué qui ne mène nulle part, ou encore un autre qui est bloqué par le feu. Dans la deuxième étape, l'agent met à jour sa carte mentale grâce aux informations collectées soit à travers la communication avec les autres agents ou à travers son exploration de l'environnement. Dans la troisième étape, selon le type d'agent, ce dernier va choisir le chemin le plus court à emprunter en prenant en compte l'état de sa carte mentale. L'agent réitère l'action

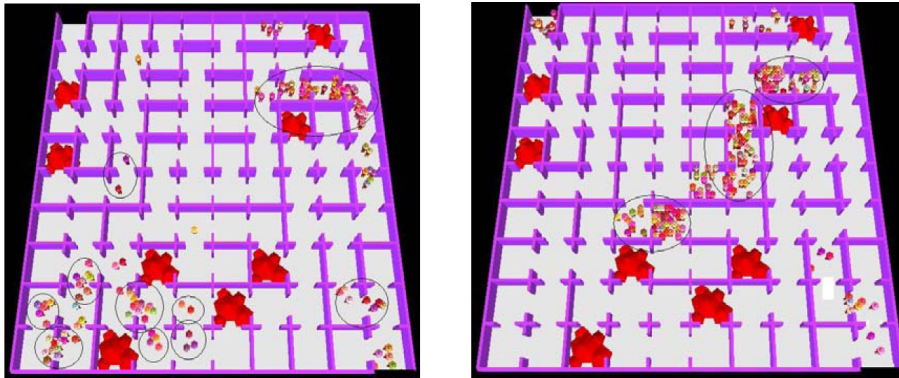


FIGURE 4.6 – Exemple d'évacuation de foule avec un grand contre un petit nombre de leaders [PB06]

toutes les fois qu'il tombe sur un chemin bloqué.

2. Le bas niveau correspond au mouvement local à l'intérieur des pièces, lequel, est basé sur le modèle de Helbing [HD00]. Il correspond au processus de déplacement d'un point attracteur (points d'intérêt) à un autre. Chaque point est placé à l'extérieur des portes correspondantes aux pièces (chambres), et chaque pièce comporte plusieurs portes (donc un point d'intérêt à chaque porte). A chaque fois qu'un agent accède à un point, alors il récupère de ce point des informations concernant la vitesse et la direction à prendre pour arriver au prochain point. Ils utilisent les forces de répulsions appliquées aux obstacles statiques afin d'éviter les collisions.

Pour obtenir un comportement humain plus réaliste, les auteurs ont défini un modèle nommé PMFserv, dont l'objet est de tenir compte d'éléments psychologiques.

PMFserv⁷ est un système qui permet d'implémenter des modèles de comportements basés sur les sources de motivations [SJCO06][SBOC06] ; les émotions.

L'information est traitée sur plusieurs couches dans la mémoire de travail de telle sorte que chaque couche dépend de la couche en dessous pour satisfaire un cycle de décision pour un agent (figure 4.7). En observant le schéma qui représente le cycle de décision à partir du bas, on trouve tout d'abord les données physiologiques, qui représentent des mesures correspondantes à des émotions (l'excitation, la fatigue, la faim, la soif, ...), lesquelles, sont combinées pour définir les niveaux d'une série de réservoirs de stress, qui vont déterminer le

⁷. performance moderator functions

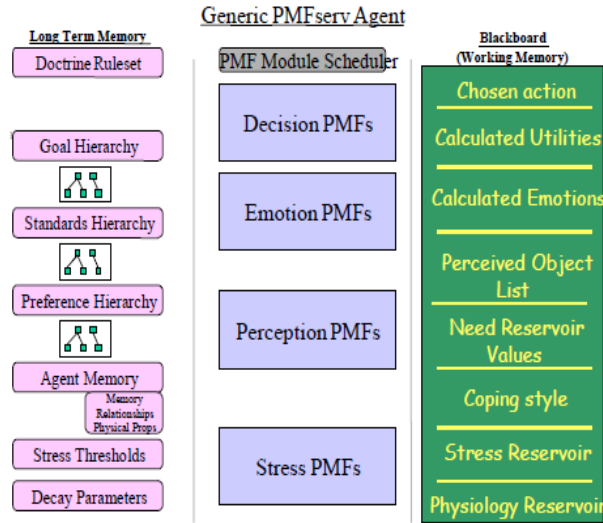


FIGURE 4.7 – Architecture de PMFserv [PB05].

style d'adaptation de chaque agent. En utilisant la couche précédente, chaque agent dans le système exécute ses règles de perception pour déterminer les objets qui sont actuellement perceptibles. L'utilité émotionnelle d'une action est déterminée en utilisant le modèle émotionnel OCC⁸ [OA88] qui génère une valeur pour 11 types d'émotions. En additionnant ces émotions, ils obtiennent une valeur d'utilité pour l'action considérée. Ce processus sera répété pour chaque action. L'action avec la plus grande valeur d'utilité sera choisie et exécutée.

Dans [PB05], est définie une architecture combinant le modèle psychologique (PMFserv) avec le système de simulation de foules (MACES). MACES produit et fournit les corps et les actions des agents, tandis que PMFserv fournit les motivations, le stress, le style d'adaptation, les émotions, la personnalité et les décisions de l'agent. En d'autres termes, PMFserv met en œuvre le processus de décision alors que MACES gère les processus de planification de mouvement (mouvement locaux et l'orientation), ainsi que l'exécution des actions résultantes. La figure 4.8 met en évidence la boucle d'interaction principale. Cependant les résultats incluent seulement des comportements relatifs à une seule situation spécifique (c'est à dire, un seul contexte « évacuations »).

8. Ortony, Clore, Collins

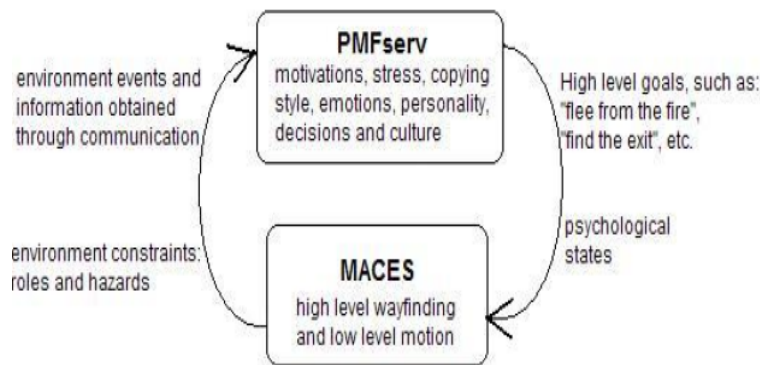


FIGURE 4.8 – La boucle d’interaction principal [PB05].

4.4 Système HiDAC

Le système HiDAC « High-Density Autonomous Crowds » [PG06] a pour objet de simuler les comportements correspondants au mouvement local dans les larges foules (qui consiste à trouver son chemin) tout en considérant la densité ainsi que la personnalité des individus de la foule. Pour ce faire, il se fonde sur une combinaison de facteurs psychologiques (impatience, panique) afin de fournir une variété dans les comportements des agents, ce qui leur permet de voir les conséquences de certains comportements sur le comportement émergent de la foule (par exemple, un agent sujet à la panique et qui a un degré d’impatience important, peut en se déplaçant faire tomber ou pousser d’autres agents et ainsi provoquer un blocage).

Dans HiDAC, chaque agent est doté d’une perception et il est capable de réagir aux objets statiques et dynamiques (par exemple, d’autres agents) qui se trouvent dans son espace.

Chaque agent en mouvement dans HiDAC perçoit son entourage en se basant sur les informations collectées dans sa zone. Pour ce faire, les auteurs ont défini une zone d’influence représentée par un rectangle (connue aussi sous nom de région ou espace d’influence). De cette manière, lorsqu’une entité statique ou dynamique apparaît dans cette zone (par exemple, un autre agent ou un obstacle), alors plusieurs informations concernant cette entité (distance aux obstacles,— direction de l’autre obstacle par rapport à lui,— vitesse,— densité de la foule) seront prises en considération afin de modifier la trajectoire du mouvement pour éviter une collision.

De cette manière ils minimisent les coûts de calcul concernant la prise de décision, en réagissant seulement aux entités se trouvant dans leur espace.

Les comportements des agents sont calculés sur deux niveaux (figure 4.9) :

le haut niveau (la navigation, la communication, et la prise de décision) et le bas niveau (La perception, le mouvement de l'agent, la locomotion).

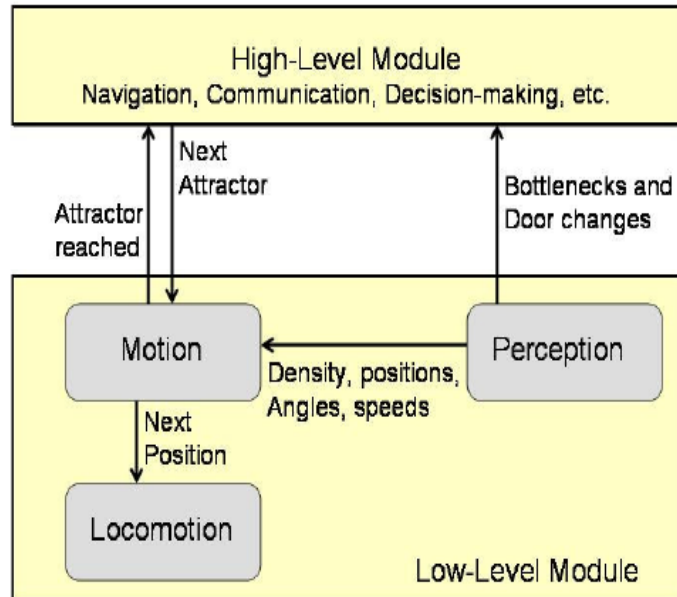


FIGURE 4.9 – Les deux niveaux du système HiDAC [PG06].

Le module du haut niveau permet de prendre des décisions concernant le prochain endroit à visiter en se basant sur les informations qu'il reçoit via la communication concernant l'état de l'environnement (un chemin à prendre qui mène à la destination), ou encore de quelconques informations collectées par l'agent pendant sa navigation dans l'environnement, tel que la largeur des chemins. Une fois que le prochain endroit a été choisi, alors le haut niveau envoie cette information au bas niveau pour que ce dernier effectue les mouvements nécessaires afin de l'atteindre.

Le sous-module du mouvement se base sur le sous-module de perception, lequel lui fournit des informations concernant des éventuels obstacles se trouvant entre l'agent et la prochaine position à atteindre (l'angle, la vitesse des obstacles dynamiques, ou encore la densité de la foule devant l'agent, ...). Une fois la vitesse ainsi que la prochaine position de l'agent établies par le sous-module de mouvement, ce dernier envoie ces informations au sous-module de locomotion pour exécuter les mouvements adéquats des pieds.

Quand le module de bas niveau atteint la position voulue, il questionne à nouveau le module du haut niveau à propos de la prochaine position.

Le mouvement local d'un agent dans une pièce est basé sur le modèle de Helbing [HD00], lequel, permet de produire des mouvements réalistes en ce

qui concerne les courbures de trajectoires au niveau des portes ou lors de l'évitement de collision.

Afin d'éviter l'effet d'une armée de robots, les auteurs dans [PG06] ont défini deux types d'agents ; celui qui est respectueux (lors d'une panique, il peut laisser la priorité à d'autres agents afin de ne pas créer un désordre dans son environnement), et celui qu'il ne l'est pas (qui peut forcer son chemin, et provoquer des poussés sur d'autres agents, ce qui peut provoquer des blocages). Pour traiter le problème dû au blocage, les auteurs ont intégré un facteur *d'impatience* ainsi que des *règles d'arrêts*, afin de créer une certaine priorité entre les agents lors des passages (par exemple, un jeune qui a un pourcentage faible pour l'impatience laissera passer une femme enceinte avant lui).

Comme pour MACES, HiDAC est destiné seulement à traiter les situations d'évacuations dont les seuls comportements sont des déplacements. En plus, il est difficilement imaginable de pouvoir appliquer ce modèle aux foules à haute densité, car non seulement chaque agent doit faire appel à un processus de décision dans lequel plusieurs sous-modules coopèrent afin de permettre à l'agent de changer de position. Mais encore, lors du processus de l'évitement de collision, plusieurs paramètres doivent être pris en considération, lesquelles permettent de modifier la trajectoire (suivant une courbure via le modèle de Helbing) afin d'éviter la collision.

4.5 Le système de classeurs (LCS)

Introduits par John Holland en 1976, le système de classeurs (LCS⁹) est basé sur des règles, ainsi qu'un algorithme génétique, lequel permet de faire évoluer les classeurs en modifiant ou en créant de nouvelles règles [BBMH08]. Les connaissances d'un agent sont représentées par un ensemble de classeurs. Chaque classeur est formé de deux parties (figure 4.10) : une règle de type « SI CONDITION ALORS ACTION », ainsi qu'une force indiquant la performance de ce classeur et représentant une récompense obtenue pour l'exécution d'une action. Elle permet aussi d'arbitrer l'activation de plusieurs classeurs en même temps, afin d'exécuter une seule action à la fois.



FIGURE 4.10 – Classeur de Holland.

9. Learning Classifier System

La partie condition de la règle représente une forme générale qui correspond à un genre de situations pouvant surgir dans l'environnement (par exemple, un feu qui correspond à une situation particulière d'une urgence). La partie condition est codée en utilisant l'alphabet 1, 0, #, où # est considéré comme une variable (représentée par 1 ou 0) permettant d'associer la condition d'une règle à plusieurs situations particulières correspondantes à l'état actuel de l'environnement. (par exemple, si on a une condition avec le code suivant ;00#1, alors elle peut être satisfaite par deux situations représentées par les codes ; 0001 et 0011).

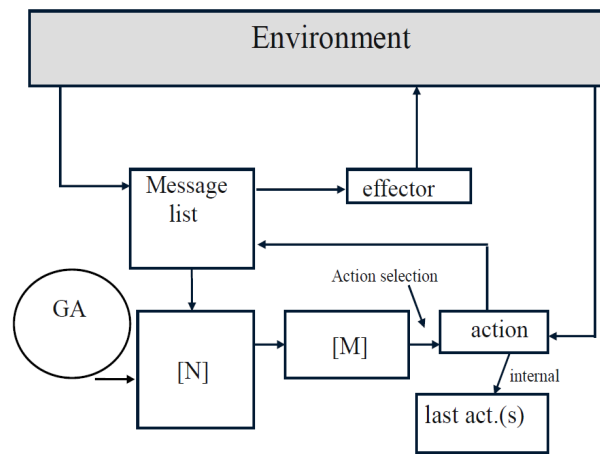


FIGURE 4.11 – Schéma de fonctionnement de LCS [BBMH08].

Le système reçoit un message de l'environnement sous forme d'un code en utilisant l'alphabet 1, 0 et exécute l'action correspondante en générant un code sous la même forme.

L'agent perçoit l'évolution de l'état de son environnement via des capteurs, lesquels, fournissent une liste de messages qui seront envoyés à la base de connaissance de l'agent représentée par un ensemble de classeurs (figure 4.11). Le processus de sélection (correspondant à un algorithme génétique) se charge de sélectionner et d'activer les classeurs dont la condition est satisfaite. Si plusieurs classeurs sont activés en même temps, alors dans ce cas, le classeur ayant la force la plus grande sera choisi (voir l'exemple de la figure 4.12).

Une fois le classeur choisi, il envoie le code de l'action à la liste des messages, laquelle transmet aux effecteurs qui se chargent de faire exécuter l'action à l'agent. Une fois l'action exécutée, le processus met à jour la force des classeurs selon l'utilité de chacun d'eux (augmenter celle des classeurs utiles et diminuer celle des moins utiles). La mise à jour régulièrement de la force ainsi

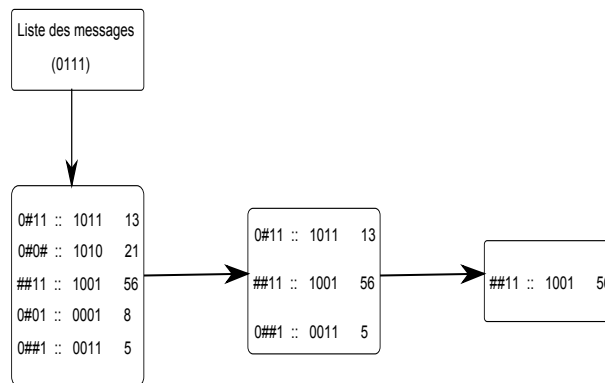


FIGURE 4.12 – Un exemple avec LCS.

que la possibilité de remplacer les règles existantes (inutiles) par des nouvelles règles plus performantes permet l'évolution et l'apprentissage.

Des améliorations des LCS ont été proposées par Wilson ; la première ZCS¹⁰ [Wil94] qui permet la génération de nouvelles règles en fonction des situations, la deuxième XCS¹¹ [Wil95] permet d'augmenter la précision d'évolution des classeurs.

4.6 Le modèle ViCrowd

Introduit par Musse et al.[MT01], c'est un modèle multi-couches qui a pour objet de doter des foules avec différents niveaux d'autonomie afin d'avoir des comportements hiérarchiques de foules.

Dans ViCrowd, les auteurs définissent la foule comme un ensemble de groupe d'agents, où la perception est associée au chef du groupe, c'est-à-dire que les informations concernant la perception seront stockées dans une mémoire définie pour chaque groupe et traitée seulement par le chef.

L'information traitée par les entités du modèle (groupes, agents) concernent :

- leurs connaissances, qui représentent les informations sur l'environnement virtuel où la foule se trouve (par exemple, les endroits où se trouvent les points d'intérêts, où encore les actions à appliquer dans certains endroits).
- les intentions, qui correspondent aux buts des entités (par exemple, aller à la banque).

10. Zeroth level Classifier System.

11. eXtended Classifier Systems.

- les statuts, qui représentent les états interne des entités (par exemple, les états émotionnels).

Deux approches ont été définies [MT97] ; tout d'abord, on a les points d'intérêts (PI) qui représentent les buts des mouvements de la foule dans ViCrowd. Ensuite, on trouve les points d'actions (PA), où des actions peuvent se produire.

Le déplacement des entités de la simulation se fait en suivant une courbe de Bézier. Cette méthode permet de gérer les collisions entre les entités et les obstacles en déformant la trajectoire de l'entité via les points de contrôles (figure 4.13).

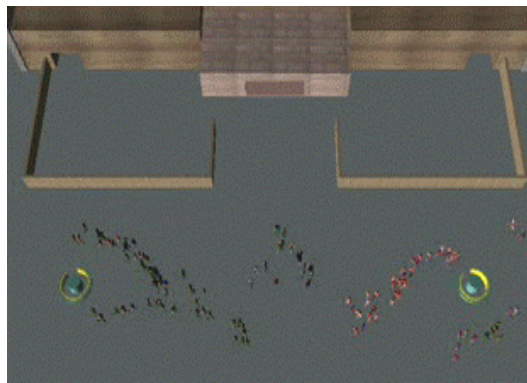


FIGURE 4.13 – Groupes évitant la collision avec des obstacles [MT97].

4.6.1 Types de comportements dans ViCrowd

Les auteurs ont distingués trois types de comportements [MT01] :

1. le comportement programmé, qui est scripté permettant de spécifier à l'avance les séquences d'actions des agents.
2. Le deuxième est le comportement réactif, qui permet aux entités simulées de réagir à diverses situations de types événement / réaction. Où les événements représentent les incidents causant des réactions.
3. Le troisième est le comportement guidé, qui est fourni par l'utilisateur via un module permettant d'interpréter le comportement en utilisant le langage naturel.

4.6.2 La structure des groupes dans ViCrowd

Chaque groupe a un leader, et chaque leader peut communiquer avec les autres leaders des autres groupes. Ces mêmes leaders peuvent changer de but ainsi que de groupe selon leurs intentions en se basant sur des aspects sociologiques [MBCT98]. Un agent peut être programmé au début pour qu'il soit un leader, et peut aussi en devenir un.

Le groupement dans ViCrowd peut être spécifié au début, comme il peut émerger dû au placement des individus (agents). On trouve aussi la notion de densité, qui exprime la relation entre le nombre d'agents et la taille de l'espace dans lequel ils se trouvent. Ils affirment que l'espace personnel d'un agent est inférieur ou égal à un mètre carré.

4.7 Modèle de comportements scalables

Introduit par Mankyu Sung et al.[SGC04], c'est un système de simulation de foule, où les agents sont définis par des comportements de base. Selon la situation, chaque acteur répond par une composition de ces comportements de base selon un système probabiliste.

Ce système est composé de deux niveaux (figure 4.14) :

- un haut niveau (pour le contrôle direct des agents de la foule).
- un bas niveau (pour les calculs probabilistes afin d'obtenir les actions à effectuer, selon l'environnement local de chaque agent).

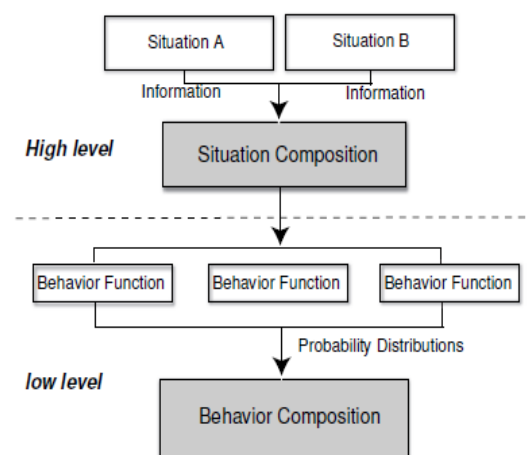


FIGURE 4.14 – Architecture d'agent à deux niveaux.

Leur approche est basée sur une vue global et non pas sur le comportement individuel¹².

Pas besoin d'avoir plusieurs situations en même temps, par exemple, lorsque l'agent arrive dans une situation particulière, il l'ajoute à ses comportements de base, et une fois qu'il a fini avec cette situation, alors il l'enlève (par exemple, un agent devant la machine à billets, lorsqu'il a acheté un billet, alors il abandonne cette situation). Une situation spécifique prend en compte un nombre restreint d'agents, c'est-à-dire que les agents seront concernés seulement par les situations dans lesquelles ils se trouvent.

Les auteurs utilisent les probabilités dans les choix des actions des agents de la foule (figure 4.15), ils supposent que les individus de la foule ne sont pas généralement connus, donc les comportements sélectionnés seront aléatoires selon la situation auxquelles ils sont soumis.

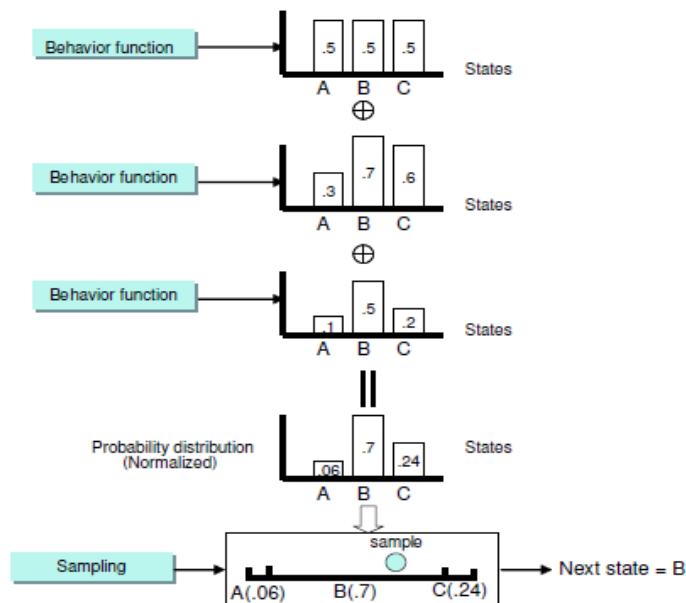


FIGURE 4.15 – Les fonctions de comportements calcules la distribution des probabilités à l'entrée de chaque état. L'état suivant est sélectionné par échantillonnage sur la distribution de la probabilité composée [SGC04].

Des fonctions de comportements sont défini par des distributions de probabilités pour chaque comportement simple. Une combinaison des fonctions simples est effectuée pour obtenir des comportements globaux plus complexes.

12. Les auteurs suppose que : « Quand nous regardons une foule, nous nous inquiétons seulement de ce qui se produit, pas qui la fait ».

La variation dans les comportements adoptés est satisfaite en permettant une composition aléatoire des comportements pour chaque agent, et cela pour la même situation.

L'action pour éviter les collisions est réalisée par une fonction de comportement, qui pénalise les états pouvant causer une collision en leur donnant la probabilité la plus basse.

La perception ainsi que l'algorithme de détection de collision utilisé dans ce système est basé sur une subdivision spatiale simple (grille de cellules uniformes).

Le principe du mécanisme probabiliste ici est de choisir un prochain état donné par l'état actuel. En d'autres termes, c'est au niveau de l'état actuel que la transition vers un nouvel état sera déterminé. Les fonctions de comportements jugent les transitions potentielles des états avec leurs propres règles indépendamment, et renvoient les probabilités.

4.8 Le système MAGS

MAGS¹³ est un simulateur multi-agents développé au sein du laboratoire informatique de l'université de Laval [MCP⁺03], qui a pour objet de simuler le comportement d'un grand nombre d'agents dans un environnement géographique virtuel urbain, où chaque agent possède un comportement qui lui est propre qui dépend de son profil (un profil ici correspond à un rôle, lequel est représenté par un arbre d'objectifs qu'il doit réaliser).

Ce système comporte cinq types d'agents ;

1. Les agents mobiles qui ont la capacité de se mouvoir dans l'environnement (par exemple, les voitures ou les humains, ou encore les animaux.).
2. Les objets qui n'ont pas la capacité de se mouvoir (des objets statiques), tels que les chaises ou les bancs.
3. Les groupes, qui peuvent être mobiles ou objets, et qui représentent un ensemble d'agents partagent des caractéristiques, des informations, ou des buts communs. Les membres d'un groupe peuvent avoir des liens qui peuvent être exprimés en une hiérarchie. Par exemple, une équipe de football qui représentent un groupe mobile, ou encore des restaurants qui représente un groupe objet.
4. Contrairement aux groupes, un *cluster* peut réunir un ensemble d'agents qui ne se connaissent pas autour d'un point d'intérêt, tel qu'une file d'attente devant un guichet au cinéma.

13. Multi-Agent GeoSimulation

5. Un agent scénario (scripté) permet de simuler les comportements spécifiés au préalable par l'utilisateur et correspondants à des événements particuliers. C'est-à-dire que ce type d'agents ne sera pas concerné par le processus de décision.

Le comportement sélectionné se fait via une règle et des paramètres propres à l'agent. Ainsi, lorsqu'une règle d'activation d'un objectif se déclenche, les actions correspondantes à la règle seront exécutées. Chaque règle est composée de pré-conditions en relation avec l'état de l'agent, de ses objectifs, et de son environnement.

Chaque objectif possède une priorité qui permet de déterminer l'objectif le plus important pour un agent à cet instant. En d'autres termes, un agent choisira d'exécuter les comportements les plus adéquats afin d'accomplir son objectif le plus prioritaire.

La perception de chaque agent dans MAGS est basée sur des cartes 2D (figure 4.16) obtenu à partir d'images réelles ou en 3D. Ils ont développé plu-

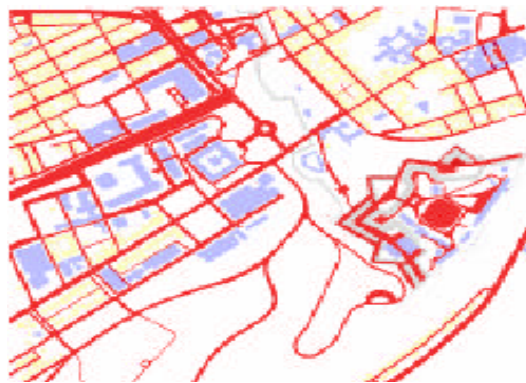


FIGURE 4.16 – Plan en 2D de la ville de Québec [MCP⁺03].

sieurs mécanismes qui permettent aux agents de profiter de plusieurs modes de perception pour récupérer des informations concernant leur environnement. Ces informations nécessaires à la navigation des agents concernent : les caractéristiques géographiques et topologiques du terrain, les voisinages, les zones dynamiques avec des propriétés spéciales, les messages entre les agents, ou encore les événements spéciaux, tel que *le feu*.

4.9 Synthèse

Dans le paragraphe ci-dessous nous présentons les tendances de chacun des modèles présenté dans ce chapitre envers les critères de réalisme (cha-

pitre 2) qui caractérises le plus ces capacités.

4.9.1 Smart objects

	type de contrôle : pré-défini et interactif	Type de réalisme : microscopique	type de foule : Collection d'agents
Smart objects	-L'utilisateur peut définir et insérer des objets dynamiquement.- La liste des comportements à exécuter est prédéfinie dans l'objet.	-L'agent peut interagir avec des objets très complexes.	-Gère l'interaction des agents avec les objets dans leur environnement.

TABLE 4.1 – Smart objects

Les smart objects [KT98] ont comme principal objectif de modéliser l'interaction entre les collections d'agents et les objets pouvant être rencontrés dans une simulation de l'environnement urbain d'une ville virtuelle. Le réalisme microscopique ici est satisfait du fait que, chaque agent a la capacité d'interagir avec les différents objets de son environnement, que ce soit des objets simples ou complexes (tel que, prendre l'ascenseur). Étant donné que chaque objet de cet environnement contient les informations nécessaires à l'interaction (tel que, l'information topologique concernant la forme de l'objet, ou les différents comportements à adopter par l'agent vis-à-vis de cet objet). Cela offre à l'utilisateur un contrôle interactif. Par exemple, l'ajout d'un nouvel objet au cours de la simulation via une application, tel que SOMOD qui permet de créer et de charger ce type d'objet. Ces objets vont influencer les agents, en leurs imposant les comportements à adopter. Les smart objects permettent aussi un contrôle prédéfini, où lors de l'interaction de l'agent avec un smart object, ce dernier impose une séquence d'actions sous forme de script à l'agent (par exemple, lorsqu'un agent interagit avec une porte, tout d'abord il va tourner la poignée vers le bas, ensuite il va pousser la porte dans le sens de l'ouverture, et après dans le sens de la fermeture).

Cependant, les smart objects ne tiennent pas compte de la variabilité entre les agents (ici tous les agents sont égaux et se comportent de la même manière avec le même objet), ou des interactions entre les agents et événements

qui peuvent surgir dans l'environnement, ou encore l'adaptabilité des agents durant ces événements (tel que les événements d'urgences).

4.9.2 MACES

	Variabilité	Type de communication : local	Type de réalisme : microscopique	Type de foule : Foule homogènes et Groupes
MACES	-Définition des catégories d'agents (leader et non leader).	-Communication entre agents afin de trouver la sortie.	-Satisfait via la communication ainsi que le système d'évitement de collision.	-Tous les agents de la foule sont soumis à une seule situation qui est l'évacuation -Formation de groupes constitués d'agents non leader et d'un leader.

TABLE 4.2 – MACES

Contrairement aux smart objects, dans MACES [PB05], ils gèrent les foules homogènes, où tous les membres de la foule sont soumis à une même situation, qui est l'évacuation vers les sorties. Ils considèrent aussi la variabilité entre les agents, en se basant sur des aspects de formations et de personnalités, où on peut trouver les agents leaders, qui peuvent être formés (expérimenté) ou non¹⁴, et les agents non leaders, qui perdent leurs moyens lors des situations d'urgences¹⁵. Cela mène à la formation des groupes constitués d'agents non leaders (incapable de prendre une décision) qui ne font que suivre un leader. Les communications locales entre les agents permettent de satisfaire un réalisme microscopique dans la simulation. Elles permettent aussi aux agents de développer leur carte mentale de l'environnement afin de trouver une sortie. Étant destiné à un cas spécifique qui est l'évacuation d'un bâtiment lors

14. c'est-à-dire qu'ils peuvent avoir une connaissance complète de l'environnement ou pas.

15. ils sont incapables de prendre la bonne décision.

d'une urgence, ce modèle ne considère pas l'adaptabilité du modèle pour un cas plus général (autre que l'évacuation.). Par exemple, des agents faisant des courses dans un supermarché, lequel nécessite d'autres caractères plus spécifiques aux agents par apport à cet environnement.

4.9.3 HiDAC

	Variabilité	Type de communication : local	Type de réalisme : microscopique	Type de foule : Foule homogènes
HiDAC	- Utilisation des traits de personnalités (patient et impatient).	-Communication entre les agents concernant le chemin à prendre.	-Via la communication entre les agents. -Utilisation des courbures dans les portes. -Traite le problème dû aux blocages.	Tous les agents de la foule sont soumis à une seule situation qui est l'évacuation.

TABLE 4.3 – HiDAC

Le système HiDAC¹⁶ [PG06] qui est destiné à la simulation des mouvements locaux dans les foules homogènes lors de l'évacuation. Il se base sur la perception et la communication pour la prise de décision. Il gère le mouvement local de l'agent dans les espaces communs à l'intérieur des pièces en se basant sur le modèle de helbing [HD00] afin de satisfaire un réalisme microscopique. Les auteurs ont intégré le facteur « impatience » afin d'obtenir deux types d'agents (patient, et pas patient), ce qui leur permet de faire varier un peu les comportements entre les agents.

Par contre, comme pour MACES, l'utilisation de ce type de modèle se limite à l'évacuation dans un bâtiment (un seul contexte correspondant au déplacement vers la sortie), où les seuls comportements sont ceux correspondants aux mouvements.

4.9.4 Le système de classeurs (LCS)

(LCS)¹⁷ utilise un algorithme évolutionniste à base de règles, où le but est de faire apprendre à des collections d'agents la réalisation des tâches en inter-

16. High-Density Autonomous Crowds

17. Learning Classifier Systems

	Adaptabilité	Type de réalisme : microscopique	Type de foule : Collection d'agents
Les systèmes de classeurs	-Utilise un algorithme évolutionniste. -L'évolution des classeurs permet l'apprentissage.	-Chaque agent perçoit son entourage via des capteurs ce qui lui permet de réagir en activant les classeurs correspondants.	-Chaque agent a sa propre histoire individuelle.

TABLE 4.4 – LCS

agissant avec leur environnement, tout en essayant d'obtenir le maximum de récompenses pour ces actions. Dans (LCS), chaque agent reçoit les informations via la perception ce qui lui permet de réagir aux différentes situations par l'activation des classeurs correspondants. Cela permet de satisfaire un réalisme microscopique. (LCS) permet l'adaptabilité des agents grâce à leur capacité d'apprentissage via laquelle chaque agent peut s'adapter à des nouvelles situations pour exécuter des tâches. Cependant, (LCS) est basé sur des algorithmes génétiques, lesquelles permettent l'évolution et l'apprentissage à travers la mise à jour régulièrement de la force qui va servir à arbitrer l'activation de plusieurs classeurs en même temps pour exécuter une action. En d'autres termes, le système de classeurs (LCS) ne peut être utilisé pour les larges foules. De plus, il ne considère pas la différence en terme de caractères (personnalité) entre les agents de la simulation.

4.9.5 Le modèle ViCrowd

	Type de contrôle : pré-défini et interactif	Type de réalisme : microscopique	Type de foule : Collection d'agents et groupes
ViCrowd	-Comportements scriptés.- Comportements guidés.	-Communication et déplacement suivant une courbe de bezier.	-Chaque agent a un but et un rôle -La formation de groupements d'agents ayant les mêmes intentions.

TABLE 4.5 – ViCrowd

Le modèle ViCrowd [MT01] simule une collection d'agents, où certains de ces agents peuvent se grouper (ceux qui ont les mêmes intentions ou buts). Il fournit à l'utilisateur différents niveaux pour contrôler les entités de la foule, que ce soit en programmant les comportements de ces derniers d'une manière scriptée, ou encore permettre aux entités simulées de réagir à diverses situations de types événement / réaction. Ils peuvent aussi être guidés dans l'environnement via des commandes écrites en langage naturel. La trajectoire de déplacement pour chaque agent est basée sur une courbe de bezier, ce qui permet d'obtenir un réalisme microscopique lors de l'évitement de collisions en modifiant juste les points de contrôles. Chaque agent dans ViCrowd a un rôle et des intentions basées sur des aspects sociologiques, cela implique des capacités cognitives, ce qui est généralement destiné à des simulations spécifiques et très développées en termes de réalisme microscopique.

4.9.6 Modèle de comportements scalable

	Variabilité	Type de réalisme : macroscopique	Type de contrôle : interactif	Type de foule : homogènes
Modèle de comportements scalable	-Se base sur un système probabiliste pour varier les comportements.	-Les agents de la foule dans une même situation peuvent avoir des comportements différents sélectionnés aléatoirement.	-Fournit un moyen pour ajouter des objets.	-Les foules sont soumises à des situations communes.

TABLE 4.6 – Modèle de comportements scalable

Ce modèle traite les foules homogènes en soumettant ces dernières à des situations dans un environnement virtuel. Ils ne considèrent que les situations simples à court terme, tout en ignorant la différence en termes de personnalité entre les agents. Le but est de satisfaire un réalisme macroscopique en ne s'intéressant qu'aux statistiques globales de la foule : par exemple, la direction de déplacement, ou encore l'agitation apparente de ses membres. L'un

des critères les plus importants traité et sur lequel se base le modèle de comportements scalable est la variabilité entre les agents lors de l'adoption des comportements dans une même situation. En effet, en se basant sur un modèle probabiliste, le modèle offre une certaine variation entre les membres de la foule en permettant à ces derniers d'adopter aléatoirement un certain nombre de comportements. Cela assure un certain réalisme visuel, qui permet d'éviter l'effet d'une armée de robots. Cependant, ce réalisme n'est pas significatif dans le sens où il ne considère pas les personnalités des agents qui constituent la foule. Dans ce cas, l'agent ne va pas choisir les comportements les plus adéquats selon son caractère vis-à-vis de la situation courante, mais plutôt, aléatoirement suivant un schéma probabiliste. Pour contrôler la foule, le modèle adopte une interface de peinture qui permet à l'utilisateur d'indiquer une situation particulière¹⁸ en la dessinant directement sur l'environnement.

4.9.7 Système MAGS

	Adaptabilité	Type de réalisme : local	Type de contrôle : prédéfini	Type de foule : collection d'agents
Système MAGS	-La décision tient compte de l'état de l'environnement local de l'agent.	-Basé sur les objectifs, chaque agent possède un objectif qu'il doit satisfaire.	-Utilisent des automates pour structurer des objectifs.	-Les agents simulés sont isolés. Cependant, des interactions entre eux peuvent se produire.

TABLE 4.7 – MAGS

Le système MAGS s'inspire du modèle ViCrowd, il permet de simuler les collections d'agents, où chacun de ces agents possède un profil et un objectif. Chaque agent réagit selon l'état courant de son environnement local, ce qui lui permet d'adopter les comportements adéquats selon cette situation. De cette manière, MAGS assure un certain réalisme microscopique, du fait qu'il permette à l'agent de s'adapter aux événements qu'il peut rencontrer, tel que l'évitement de la collision. Cependant, un profil ne représente pas la personnalité propre à chaque agent (par exemple, son caractère). En effet, deux agents ayant le même profil et les mêmes objectifs peuvent ne pas avoir

18. Cette situation représente un objet sur lequel les agents vont réagir aléatoirement.

les mêmes comportements s'ils ont deux personnalités (caractères) différentes. Il permet un contrôle prédéfini en fournissant une structure concernant une liste des objectifs, où le choix d'un objectif se fait via une priorité, laquelle est définie selon l'état de son environnement local. Vu l'ensemble de contraintes posées afin de définir cette priorité (temporelle, ou basée sur des filtres de perceptions), le calcul de cette priorité devient très coûteux, et peut ralentir le système dans le cas des situations très complexes.

4.10 Conclusion

Chaque modèle présenté dans ce chapitre possède des caractéristiques propres, lesquelles permettent de satisfaire un ou plusieurs critères (chapitre 2). On constate lors de la présentation des différents modèles que, si on augmente les détails au niveau du processus de décision, cela implique une quantité de calculs importante, qu'il est difficile à modéliser et à appliquer à une foule. Si on considère un niveau de détail faible, c'est-à-dire, sans tenir compte, par exemple de la personnalité des agents, cela donne des résultats peu convaincants. Dans le chapitre suivant on va présenter le modèle issu de cette thèse, lequel doit être assez souple avec un minimum d'éléments, afin qu'il soit facile à mettre en œuvre. Il doit être aussi général pour qu'il puisse satisfaire aux différents critères de réalisme et simuler les différents types de foules (chapitre 2).

Chapitre 5

Présentation du modèle

Sommaire

5.1	Introduction	69
5.2	Composants principaux du modèle	70
5.2.1	Agent et attributs de caractère	71
5.2.2	Contexte	72
5.2.3	Environnement	84
5.3	Fonctionnement du modèle	86
5.3.1	Calcul de l'ensemble des contextes	86
5.3.2	Calcul de la combinaison des contextes	86
5.3.3	Calcul des valeurs des attributs de caractère	88
5.3.4	Activation des comportements	88
5.3.5	Exécution des comportements	88
5.4	Expérimentations	90
5.4.1	Deuxième expérimentation : soumission à plusieurs contextes	94
5.4.2	Troisième expérimentation : contexte propagateur	99
5.5	Conclusion	102

5.1 Introduction

Comme nous l'avons vu ci-dessus, soit les modèles d'animation comportementale de foules existants sont dédiés à un domaine spécifique, soit ils sont conceptuellement très riches voire complexes. De plus, sauf d'une manière

ad-hoc, ils ne peuvent pas combiner les exigences du réalisme microscopique et macroscopique.

Ce chapitre a pour objet de présenter les bases conceptuelles d'un modèle simple et général, flexible qui peut être aisément paramétré ou enrichi.

L'idée principale qui a présidé à la réalisation du modèle est qu'un comportement n'est jamais fait sans raison. Il est toujours produit dans le cadre d'une situation donnée, issue éventuellement d'un événement, que nous appellerons contexte. Autrement dit, un contexte tend à faire prendre aux agents sous son influence un comportement spécifique.

Les comportements étant liés aux contextes, il est alors possible, comme dans le modèle des smart-objects, de voir ces derniers comme des réservoirs de comportements. L'agent est donc en soi dénué de tout comportement initial et « hérite » dynamiquement de ceux des contextes. Bien que sa réalisation n'ait pas été conçue ainsi, il est alors possible de voir le modèle présenté dans ces pages comme une généralisation du modèle des smart-objects.

Contrairement aux spécifications classiques de l'animation comportementale, les agents ne sont alors pas du tout autonomes, puisque pilotés par les contextes — ce qui favorise l'obtention d'un réalisme global. Le second principe du modèle est alors de donner une « personnalité » aux agents (sous forme d'attributs de caractère), qui leur permet de résister plus ou moins au contexte. Des agents différents peuvent alors se comporter différemment dans un même contexte, ce qui leur donne une apparente autonomie et satisfait à certaines contraintes du réalisme microscopique.

Le contexte est doté d'une force évolutive (par exemple décroissante quand il représente un événement) qui permet d'annihiler plus ou moins la « personnalité » des agents. Il est alors possible de faire évoluer dynamiquement le degré de contrôle des agents, ce qui privilégie au cours d'une même animation le réalisme microscopique ou macroscopique.

Ce chapitre sera consacré à la présentation plus en détails des différents composants du modèle avec des exemples qui illustrent son fonctionnement, ainsi que différentes démonstrations qui permettront d'illustrer les limites du champ d'application du modèle.

5.2 Composants principaux du modèle

Notre modèle est basé sur deux concepts principaux : celui de contexte et d'attribut de caractère. Les contextes fournissent des comportements et influencent les agents qui leur sont soumis pour les conduire à adopter ces comportements (ils tendent à contrôler les agents). Les attributs de caractère sont des attributs des agents qui leur donnent une sorte de « personnalité » ;

cette personnalité aide ces agents à résister aux contextes. L'idée est que deux agents qui diffèrent sur leur « personnalité » puissent avoir des comportements différents dans le même contexte. Dans les pages suivantes, le modèle sera principalement illustré dans le cadre de l'animation comportementale des spectateurs d'un événement sportif (un match de football).

5.2.1 Agent et attributs de caractère

L'idée principale est qu'un agent n'a au départ aucune capacité ; aucune méthode représentant un comportement n'est définie dans sa description. Il trouve les « moyens de faire des choses » dans les contextes auxquels il est soumis¹.

Le rôle principal d'un agent est d'être spécifique, c'est-à-dire potentiellement différent des autres agents. Sa description contient des attributs habituels (comme la *position*) et des attributs particuliers que nous appellerons des attributs de caractère² (par exemple, *agressivité*, *panique*, *sensibilité à la publicité*...). Lorsqu'un agent est créé, une *tendance* est associée à chaque attribut de caractère de l'agent (figure 5.1).

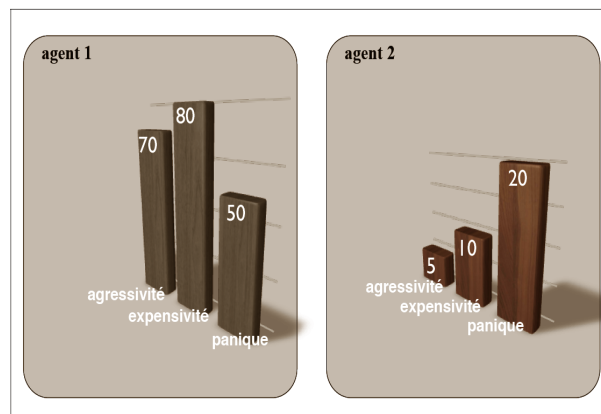


FIGURE 5.1 – Les tendances de l'agent 1 et 2 pour les trois attributs de caractère.

1. L'idée qui préside à ce choix est que le fait d'effectuer une action n'a de sens que si une situation dans laquelle l'agent est plongé le justifie. Cette situation peut-être un but, un problème à résoudre, un état interne de l'agent, un environnement... Toutes choses qui seront représentées dans le modèle par des contextes.

2. Les attributs de caractère peuvent être vus comme une abstraction des états émotionnels ([MT97], [MGT99], [PB06], [SMGT99]), ou des attributs de la personnalité [SSN08], ou des états mentaux [Nyg07]. La notion d'attribut de caractère est plus générale, avec moins de références à la psychologie.

Une *tendance* est une valeur numérique prédéfinie entre 0 et 100 qui représente le potentiel de cet agent pour l'attribut de caractère correspondant (potentiel d'agressivité, potentiel de panique, sensibilité potentielle à la publicité...). Pendant l'animation, pour un agent A , la valeur réelle v d'un attribut de caractère a peut être différente de la tendance t . Le rôle de t est d'être « la valeur de référence » de a dans le cadre de l'agent A ; v vaut t si A n'est soumis à aucune influence (voir plus loin).

5.2.2 Contexte

Un contexte C est essentiellement un réservoir de *comportements conditionnels*, c'est-à-dire des *règles de comportement* dont les prémisses sont des conditions sur les valeurs des attributs (attributs de caractère ou autres attributs) et dont les conclusions sont des descriptions de comportements³. Ces règles doivent être appliquées par les agents qui sont soumis à C si leurs caractéristiques respectent les conditions.

La description des contextes contient aussi un jeu d'attributs de caractères et leurs *influences* associées dans le contexte — qui sont similaires aux tendances des agents. Une influence i pour un attribut de caractère a dans un contexte C représente la valeur que C tend à donner pour a aux agents qui lui sont soumis. La valeur de a pour un agent donné A est le fruit de la combinaison entre la tendance de A pour a et les influences de chaque contexte auxquels A est soumis (voir paragraphe 4).

Nous distinguerons les *contextes primitifs*, qui sont donnés par le concepteur de l'animation courante, et les *combinaisons de contextes* qui sont calculées à partir des contextes primitifs. Une force globale f_C est attribuée à chaque contexte primitif C . C'est une valeur numérique entre 0 et 100 qui mesure la prégnance de C relativement aux autres contextes. Chaque influence et règle de comportement de C sont pondérées par f_C . Nous verrons plus loin que, dans une combinaison de contextes, la force ne sera plus globale mais spécifique à chaque attribut de caractère et chaque règle de comportement.

3. Quand il est introduit dans un modèle, le rôle de la notion de contexte est en général d'exprimer le fait qu'un agent est immergé dans une situation particulière, qui pourrait induire une modification de son comportement (voir par exemple [Hu06], [HE06]). C'est aussi le point de vue qui est développé ici mais le rôle technique du contexte dans notre modèle (être un réservoir de comportement) diffère singulièrement de celui qu'il joue dans les autres modèles.

1. Contexte privé

Un champ de la description de chaque agent est dédié au stockage de la liste des contextes primitifs *privés* de cet agent. Un agent est soumis à l'ensemble des contextes privés qui figurent dans sa description (figure 5.2).

Les contextes privés peuvent être utilisés pour stocker les comportements de base des agents, comme *marcher* — ou *faire ses courses* dans le cadre d'un supermarché. Un contexte privé partagé par plusieurs agents peut aussi être utilisé pour modéliser des groupes (chaque membre du groupe référence le contexte qui est un « point de rencontre » pour tous). Des informations partagées, la liste des membres, ainsi que des comportements de communication peuvent être stockés dans des contextes privés partagés.

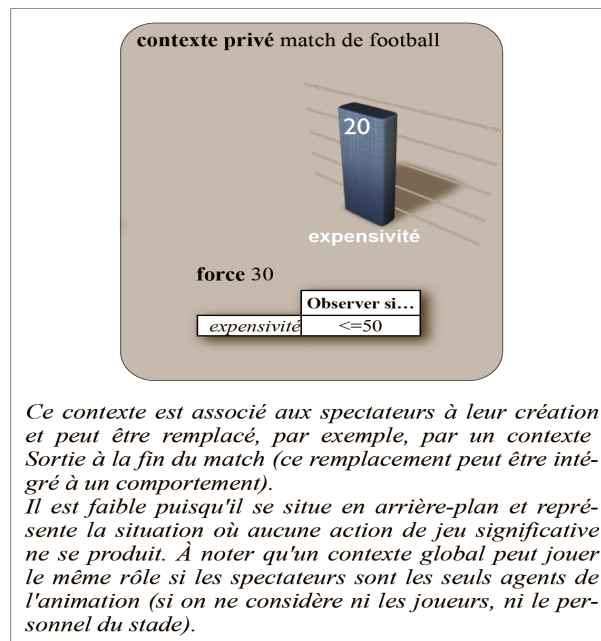


FIGURE 5.2 – Un contexte privé associé aux spectateurs.

2. Contexte global

Les contextes de cette catégorie sont perceptibles partout, par tous les agents (figure 5.3), ce qui signifie que tous les agents leurs sont soumis. Techniquement, la liste des contextes actifs de cette sorte est stockée

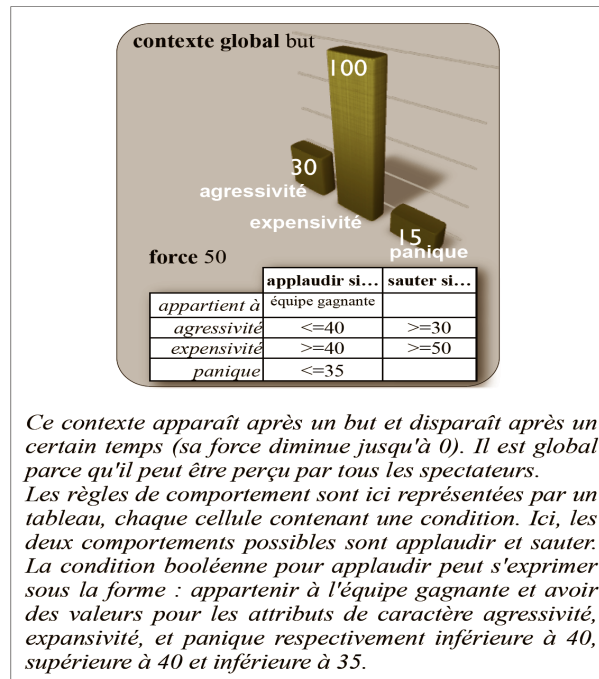


FIGURE 5.3 – Un contexte global dans un match de football.

dans l'environnement, dans un endroit accessible par tous les agents. Les contextes globaux peuvent servir à modéliser les contextes qui succèdent à des événements globaux et qui induisent des comportements spécifiques. Par exemple, dans le cadre d'un match de football, des contextes globaux seront associés aux actions « fortes » du jeu : *buts*, *coups francs*, *penalties*, qui conduisent les spectateurs à *applaudir*, *sauter*,...

Exemple d'application « match de football »

Nous avons placé des agents dans l'environnement d'un match de football [SSN09]. Dans l'application, ces agents jouent le rôle des spectateurs qui soutiennent une des deux équipes.

- Définition des contextes globaux

L'environnement « stade de football » peut contenir plusieurs contextes différents. Nous nous sommes limités à six contextes : *situation normale*, *but*, *coup-franc*, *sortir du stade*, *fête*, et *feu*.

- Définition des attributs de caractère

Nous avons défini trois attributs de caractère : L'*agressivité* qui augmente après une défaite, un but marqué par l'équipe soutenue, ou en raison de l'*agressivité* des voisins, ce qui peut déclencher des incidents (les événements qui induisent des contextes localisés). L'*expansivité* est liée à un but ou à une victoire. Ses comportements associés sont l'expression d'un plaisir. Enfin il y a la *panique* qui est un attribut fondamental que nous trouvons également dans d'autres environnements. Son expression est liée à l'occurrence d'une situation d'urgence.

- Définition des comportements

À chaque contexte défini, nous assignons un ou plusieurs comportements (figure 5.4).

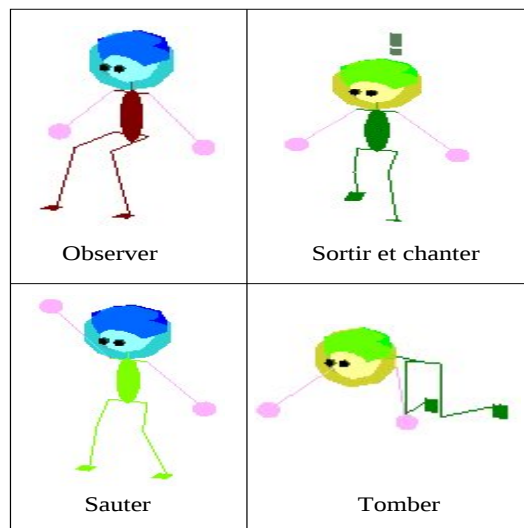


FIGURE 5.4 – Quelques comportements possibles de spectateurs.

Nous avons respectivement : *observer* et *manger* pour le contexte *situation normale*, *applaudir* et *sauter* pour le contexte *but*, *jeter des objets* pour le contexte *coup franc*, *sortir* pour le contexte *sortir du stade*, *chanter* et *danser* pour le contexte *fête*, et finalement *fuir* et *marcher vers la sortie* pour le contexte *feu*.

Nous avons aléatoirement généré une population diverse d'agents, avec des « personnalités » spécifiques et l'avons soumise à plusieurs contextes, du plus faible (*situation normale*) aux plus forte (*but* et *feu*). Nous avons

noté la variété des comportements résultants adoptés par les spectateurs (figure 5.5).

Nous avons obtenu un degré élevé de variété dans les comportements des agents soumis à un contexte faible, comparé aux agents qui sont soumis à un contexte fort (voir [SSN09]).

Algorithme de combinaison d'un contexte privé avec des contextes globaux

```

if new global context in environnement then
    list of context = list of context + new global context
    select influences corresponding to biggest values of product ( $i_{C,a} \times f_C$ )
    calculate new values for tendencies
    activate behaviors rules
while time of simulation after submission < time of new global context then
    if time of simulation after submission <= time of stabilization then
        activate behaviors rules
else
    Reduce the force of global context
    new values for tendencies
    activate behaviors rules
The corresponding global context out of the list of context

```

Si les agents sont soumis à un nouveau contexte global, ce dernier sera ajouté à la liste des contextes de chaque agent de la foule. Celle-ci contient tous les contextes auxquels l'agent est soumis. Une fois le contexte ajouté, une nouvelle composition de contextes résulte, dont l'influence pour chaque attribut de caractère est la plus grande valeur parmi les produits $(i_{C,a}, f_C)$ ⁴, et les règles de comportement sont celles de tous les contexte composants réunies. Une fois le contexte global déclenché, la force de ce dernier va rester stable pendant un certain temps. C'est-à-dire, que la valeur $V_{a,C,A}$ d'un attribut de caractère a pour un agent A dans le contexte C pour chaque attribut de caractère ne changera pas durant la période où la force sera stable. Ce qui signifie que l'attitude à adopter par l'agent durant cette période sera la même. Après cette période, la force du contexte global commence à diminuer sur une période fixée par l'utilisateur lors de la définition de ce contexte global. Durant cette période, l'attitude de chaque agent peut changer selon la nouvelle force de ce contexte jusqu'à ce que cette dernière soit nulle.

4. influence du contexte C pour l'attribut a , force de C .

Dans le contexte
"Situation normale"



Dans le contexte
"But"



Dans le contexte
"Feu"



FIGURE 5.5 – Foule de spectateurs dans un match de football soumise à des contextes globaux Situation normale, But, Feu.

3. Contextes localisés

Un contexte de ce type n'est accessible que dans une zone précise de l'environnement, dite *zone d'influence* — qui peut évoluer au cours de l'animation. Un agent est soumis à un contexte localisé s'il entre dans sa zone d'influence. Il peut être simultanément soumis à plusieurs contextes localisés (et à d'autres contextes) s'il se situe dans l'intersection de leurs zones d'influence (voir figure 5.6).

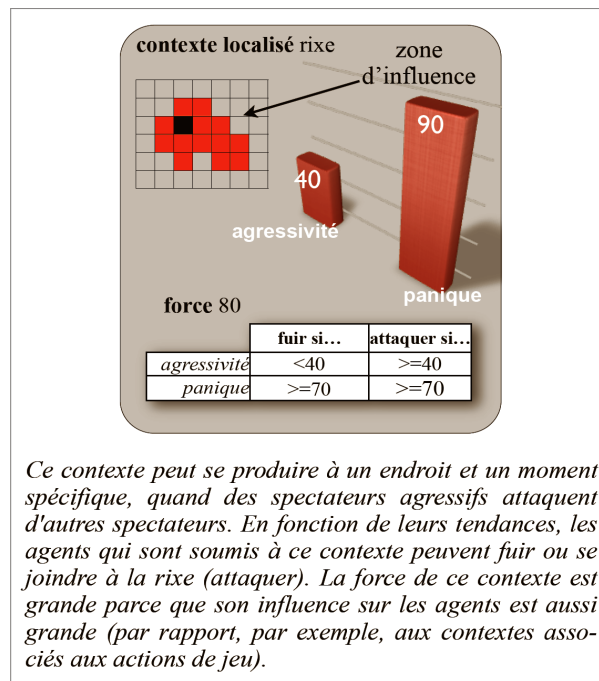


FIGURE 5.6 – Un contexte localisé dans le match de football.

Les contextes de cette sorte peuvent être utilisés par un agent pour percevoir son voisinage et pour y réagir. Par exemple, pour éviter les collisions, chaque agent a_1 peut être accompagné par un contexte localisé C_{a_1} qui se déplace avec lui (voir figure 5.7). La zone d'influence de C_{a_1} entoure a_1 ⁵. Quand il atteint cette zone, un autre agent a_2 peut ainsi « percevoir » qu'il est proche de quelque chose et peut avoir des informations sur cette « chose » en obtenant les propriétés de C_{a_1} . De plus, des comportements sont stockés dans C_{a_1} ; ils expriment comment se comporter en présence de a_1 . Pour éviter les collisions, C_{a_1} doit conte-

5. Chaque zone d'influence peut avoir sa propre forme, qui est définie par le concepteur du système.

nir les comportements qui seront adoptés par a_2 et tout autre agent qui sera soumis à C_{a_1} . Tout se passe comme si a_1 contrôlait⁶ le comportement des agents de son entourage. Une conséquence de cette technique est que le comportement d'évitement d'un même agent peut varier en fonction du type d'obstacle qu'il rencontre. Par exemple, un contexte localisé peut être associé à un groupe de plusieurs agents (des amis ou une famille). Ce contexte contient une stratégie spécifique pour éviter les collisions, qui prend en compte la nature particulière des groupes. Chaque agent qui rencontrera le groupe, — ou plutôt son contexte —, sera capable de suivre cette stratégie — s'il satisfait aux pré-requis de la règle de comportement associée.

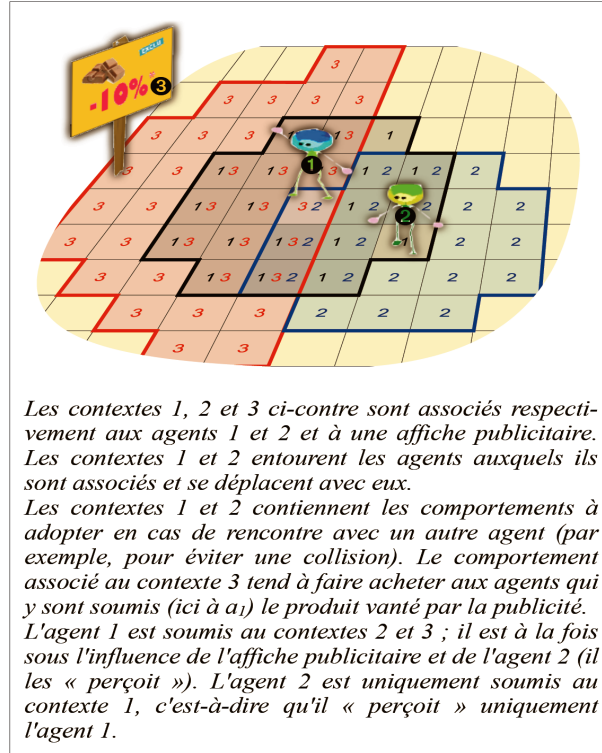


FIGURE 5.7 – Trois contextes localisés.

6. Dans une certaine mesure. Certains agents soumis à C_{a_i} peuvent « résister » plus que d'autres, en fonction de leurs tendances.

Algorithme de combinaison d'un contexte privé avec des contextes localisés

```
if new location in new localised context then
    list of context = list of context + new localised context
    select influences corresponding to biggest values of product ( $i_{C,a} \times f_C$ )
    calculate new values for tendencies
    activate behaviors rules
while location in new localised context then
    activate behaviors rules
The corresponding localised context out of the list of context
```

Un agent en mouvement peut rencontrer un contexte localisé (par exemple, d'autres agents, ou un secteur de l'environnement dans lequel il y'a une rixe.). Dans ce cas, l'agent sera soumis à ce contexte tant qu'il se trouvera dans ce secteur. Après l'ajout de ce contexte localisé à la liste des contextes de l'agent. Le calcul des nouvelles tendances de l'agent se fait après la sélection des valeurs d'influences sur les attributs de caractère⁷ qui correspondent aux plus grandes valeurs du produit ($i_{C,a}, f_C$). Ensuite, les règles de comportements qui seront satisfaites vont permettre l'exécution des comportements correspondants. L'agent adoptera la même attitude durant la période où il sera soumis à ce contexte. Une fois que l'agent est en dehors de la zone d'influence de ce contexte localisé, il n'est plus influencé par de ce dernier.

Une application a été réalisée dans laquelle une entité⁸ sera soumise à plusieurs contextes localisés⁹. Au début, l'entité (ayant la forme d'un cube en noir) se trouve dans une zone neutre ayant juste un contexte privé (figure 5.8 a). Une fois que l'entité rentre dans une zone d'influence de l'un des contextes localisés, alors le cube se transforme en une sphère. Le fait que cette sphère change de couleur et prenne la même couleur que le contexte en question signifie que l'entité est influencée par ce contexte et sera soumise à ses contraintes (figure 5.8 c). Sinon, le fait que l'entité change de forme mais garde sa couleur d'origine veut dire que l'entité se trouve bien dans la zone d'influence du contexte, mais résiste à son influence (figure 5.8 b), l'entité *ignore* alors les comportements de ce contexte (elle n'adopte pas sa couleur). Cela signifie que les prémisses des règles de comportement du contexte ne sont pas sa-

7. Correspondants aux contextes auquel l'agent est soumis

8. représentée par un cube noir

9. lesquels sont représentés par des sphères de différentes couleurs

tisfaites et, donc, que la règle n'est pas appliquée.

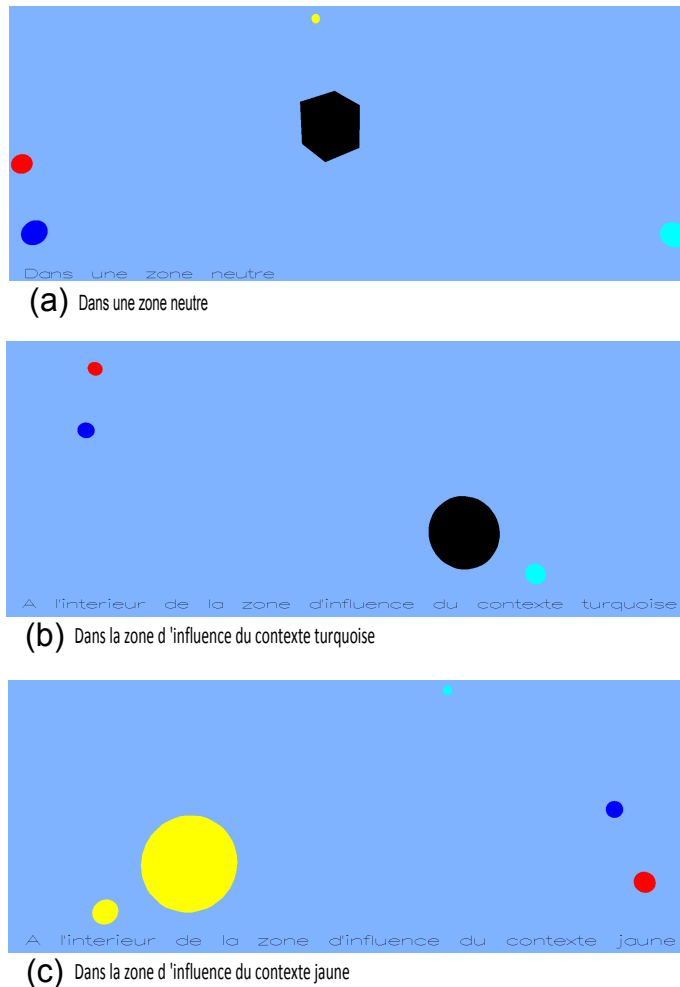


FIGURE 5.8 – Soumission d'une entité à des contextes localisés.

Dans la figure 5.9, on a placé plusieurs contextes dans la même zone. Comme on l'a vu précédemment, à chaque contexte est affectée une force. Avant la combinaison des trois contextes, on a soumis l'entité à chacun d'eux (figure 5.9 a b c). A chaque fois l'entité est influencée par le contexte correspondant (changement de forme et de couleur). En combinant les trois contextes, une fois que l'entité est dans la zone d'influence du contexte résultant de cette combinaison, elle adopte les comportements du contexte le plus prégnant (dans notre cas, elle va prendre la couleur jaune correspondante au contexte jaune) (figure 5.9 d).

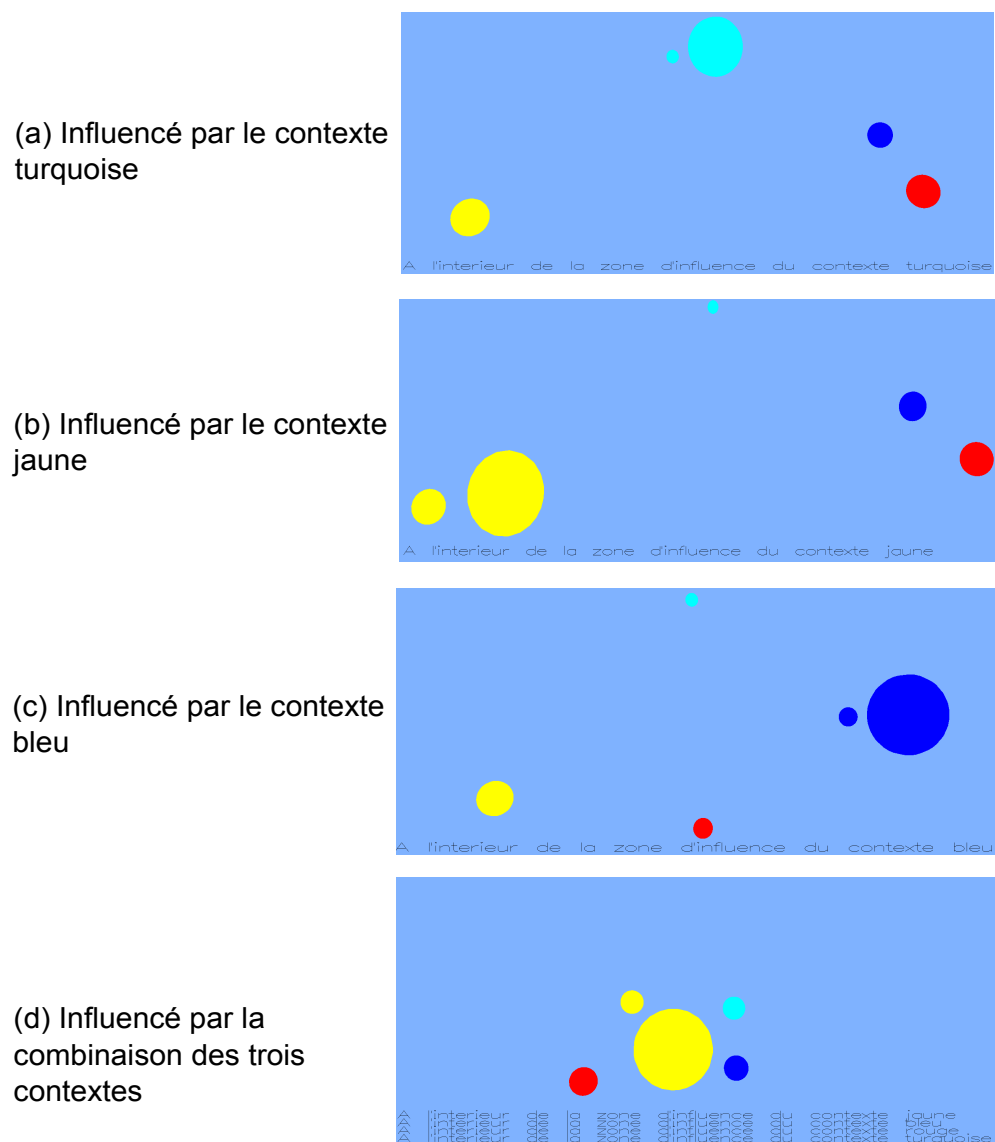


FIGURE 5.9 – Soumission d'une entité à une combinaison de contextes localisés.

4. Contexte propagateur

Sa particularité est de se propager à chaque agent qui entre dans sa zone d'influence, lequel devient ainsi lui-même un contexte propagateur pour les autres agents. Cela permet de modéliser la situation où un agent perçoit un événement localisé et communique cette information à son entourage — qui perçoit ainsi indirectement l'événement (exemple : la Ola dans un match de foot ou certains effets de panique collective). Ce type de contexte est un contexte localisé qui contient un comportement principal (par exemple, un comportement de fuite dans le cas d'un incident grave), ainsi qu'un comportement de propagation. Lorsqu'un agent A est soumis à un contexte propagateur C_p , suivant ses tendances il applique ou non le comportement principal et/ou le comportement de propagation. Ce dernier consiste à cloner C_p pour en faire un nouveau contexte localisé associé à A, lequel devient alors lui-même contexte propagateur. Au cours du clonage, il est possible de réduire la force de C_p pour que la propagation ait un terme.

Algorithme de combinaison d'un contexte privé avec un contexte propagateur

```

if new location in new localised context then
    list of context = list of context + new localised context
if new localised context contains behavior propagate then
    localised context == propagate context
    select influences corresponding to biggest values of product ( $i_{C,a} \times f_C$ )
    calculate new values of tendencies
    activate behaviors
if behavior rule corresponding to propagate is activated then
    create a clone of propagate context

```

Lors de la définition d'un contexte propagateur, en plus des règles de comportements qu'il peut contenir, en trouve aussi la règle de comportement qui correspond au comportement *propager*. De ce fait, lorsque l'agent ajoute à sa liste de contextes un nouveau contexte localisé qu'il vient de rencontrer, et que ce dernier contient le comportement *propager*, alors ce contexte est considéré comme un contexte *propagateur*. C'est-à-dire, que l'agent va créer un clone de ce contexte qui va se déplacer avec lui pendant de la simulation. En plus des comportements que l'agent va adopter, si le comportement *propager* est activé, l'agent

est susceptible de prévenir les autres agents d'un événement qui s'est produit (par exemple, un feu qui s'est déclenché).

5.2.3 Environnement

Comme cela a été évoqué plus haut, l'environnement d'une animation est représenté en interne par un objet dont les propriétés principales sont :

1. La liste des contextes de l'animation courante (à un moment donné).
2. Une grille [Wol83][LMM03][TLCC01] dont les cellules contiennent une liste de références (éventuellement vide) à des contextes localisés.

L'acquisition de l'information dans notre modèle

Comme on l'a vu au chapitre 2, dans toute simulation d'agents virtuels, l'acquisition de l'information joue un rôle très important pour le réalisme. Cela s'applique aussi au sein de notre modèle. Une bonne acquisition de l'information nécessite une représentation de l'environnement dont le but est de permettre une bonne perception de l'environnement nécessaire à la communication entre les agents et à la prise de décision.

1. Représentation de l'environnement

La représentation de l'environnement représente un point essentiel sur lequel l'acquisition de l'information se base dans notre modèle. Comme on l'a vu précédemment, il existe plusieurs méthodes pour représenter l'environnement. Dans notre modèle, on utilise une représentation de l'environnement basée sur une grille de cellules uniformes, où chaque cellule contient des informations concernant l'état de l'environnement (contextes globaux et localisés). Les contextes localisés peuvent représenter un événement ou encore les informations concernant des éventuelles entités statiques (par exemple, les murs) ou dynamiques (des agents). Ces informations permettent aux agents d'éviter les collisions. Nous avons choisi une grille de cellules uniformes pour des raisons de simplicité mais de nombreux autres cadres de représentation sont possibles, qui resteront cependant dans le cadre des environnements informés. Dans le cadre posé ci-dessus, chaque entité de l'environnement sera représentée par un ensemble de cellules, où chacune d'elles contient deux états : le premier étant inaccessible, c'est-à-dire que la cellule en question représente une partie d'une entité (statique ou dynamique), le deuxième état étant accessible, c'est-à-dire qu'il est susceptible d'être occupé par une entité ou une partie d'une entité.

Cependant, un moyen de minimiser la perte d'espace lors de la représentation des environnements ayant une topologie complexe est d'utiliser une grille de cellules non uniformes (figure 5.10).

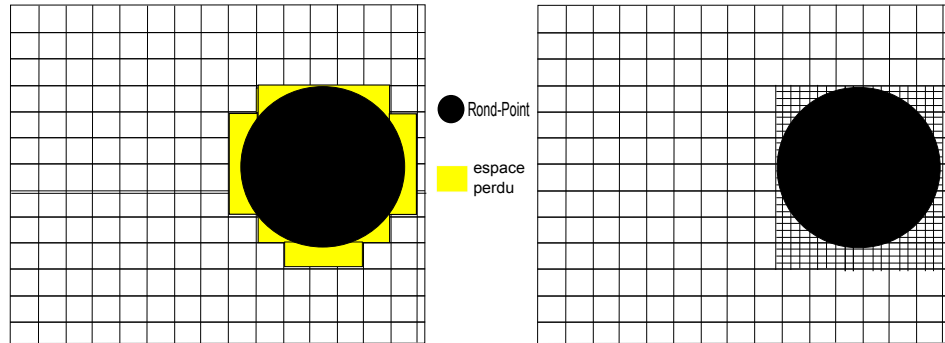


FIGURE 5.10 – (À gauche) avec une grille de cellules uniformes. (À droite) avec une grille de cellules non uniformes.

2. La perception

La perception dans notre modèle est représentée via une approche zonale. Cette dernière est satisfaite en utilisant un espace d'influence pour chaque agent (figure 5.11). Cet espace d'influence est représenté par plusieurs cellules uniformes de la grille. Chaque agent est situé au centre de cet espace et va réagir aux événements et aux autres entités (qui correspondent à des contextes localisés) qui entrent dans son espace. Ces entités ont elles mêmes leurs propres espaces d'influences. L'agent va réagir différemment selon chaque situation. Par exemple, si le contexte localisé correspond à une entité statique ou encore dynamique (d'autres agents ou des véhicules) alors l'agent va réagir à ce type de contextes localisés en activant les comportements correspondant à chaque cas (généralement le comportement pour éviter la collision). Un autre cas de contextes localisés qui peuvent entrer dans l'espace d'influence de l'entité concerne les événements locaux discrets que peuvent rencontrer les agents. Par exemple, des rixes, ou encore des situations d'urgences, par exemple un feu. À la différence avec le premier cas de contextes, ils sont susceptibles d'influencer l'état interne de l'agent de tel sorte qu'ils puissent faire dévier l'agent de son but initial (par exemple, dans le cas d'un agent se baladant dans un magasin à la recherche d'un article. Si un feu se déclenche, alors il va abandonner l'article pour se diriger vers une sortie).

La perception dans notre modèle peut être vue aussi comme un environnement informé, dans lequel, chaque agent va récupérer l'information directement à l'intérieur de son espace d'influence.

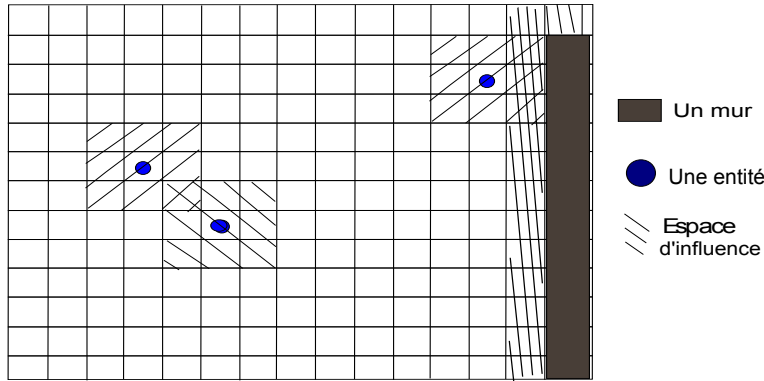


FIGURE 5.11 – Des entités avec leurs zones d'influences.

5.3 Fonctionnement du modèle

À la création d'un agent A , sa « personnalité » est définie en donnant des valeurs de tendances pour les attributs de caractère de sa description. Au cours de sa vie virtuelle, et particulièrement au cours de ses déplacements, A rencontre des contextes (en général plusieurs à la fois). Il est alors sous l'influence de ces contextes ou *soumis* à eux. À chaque instant l'ensemble des contextes auxquels est soumis A forme l'ensemble $L_C(A)$. À cet instant, les opérations effectuées sur A sont les suivantes :

5.3.1 Calcul de l'ensemble des contextes

$L_C(A)$ est obtenu en effectuant l'union de l'ensemble des contextes privés de la description de A avec l'ensemble des contextes globaux que A trouve dans la description de l'environnement, ainsi qu'avec l'ensemble des contextes localisés qui sont référencés par la cellule dans laquelle A est positionné ¹⁰.

5.3.2 Calcul de la combinaison des contextes

Rappelons que chaque contexte de $L_C(A)$ contient un ensemble de règles de comportement (figure 5.12) et peut contenir des influences relatives aux

¹⁰. Après l'étape initiale, il suffira évidemment de remettre à jour cet ensemble sans le recalculer complètement.

attributs de caractère de A ¹¹. Il est possible d'obtenir une *combinaison de contextes* U_C en combinant les éléments de $L_C(A)$.

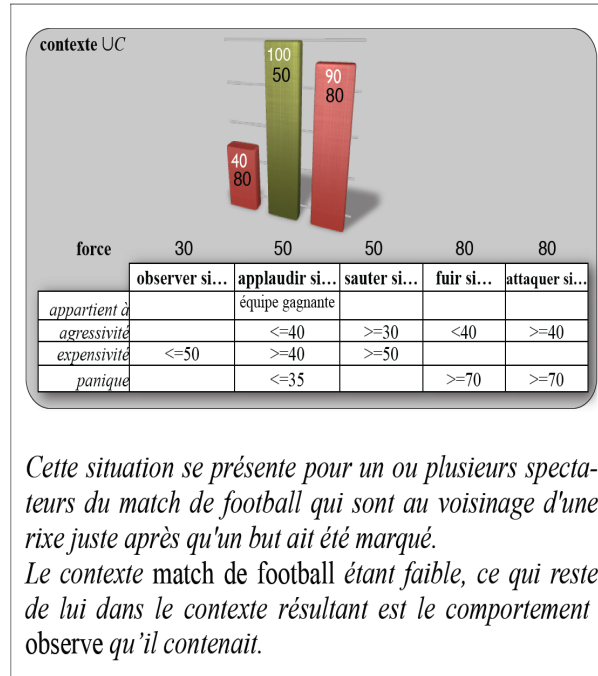


FIGURE 5.12 – *Combinaison des trois contextes précédents.* Contexte global « But », contexte localisé « rixe », et contexte privé « match de football ».

Cette situation (combinaison des trois contextes dans la figure 5.12) se présente pour un ou plusieurs spectateurs du match de football qui sont au voisinage d'une rixe juste après qu'un but ait été marqué.

Le contexte *match de football* étant faible, ce qui reste de lui dans le contexte résultant est le comportement *observe* qu'il contenait.

1. U_C accumule l'ensemble des règles de comportements et des attributs de caractère de $L_C(A)$.
2. Quand une règle de comportement est ajoutée à U_C , elle est associée à la force de son contexte d'origine. De ce fait, dans le contexte résultant, chaque règle de comportement est pondérée spécifiquement.
3. L'influence de U_C pour chaque attribut de caractère est la plus forte des influences parmi celles des contextes d'origine qui possèdent cet attribut.

11. L'ensemble des attributs de caractère référencés par un contexte peut différer d'un contexte à un autre et ne pas recouvrir ceux de A . Ses éléments sont les attributs sur lesquels le contexte concerné est susceptible d'influer.

Plus précisément, le couple $(i_{C,a}, f_C)$ (influence de C pour l'attribut a , force de C) dont le produit est maximal est conservé dans U_C . Chaque influence de U_C est spécifiquement pondérée par la force du contexte dont elle vient.

5.3.3 Calcul des valeurs des attributs de caractère

La situation qui suit les étapes précédente revient à soumettre les agents concernés à un seul contexte C : la combinaison des contextes accessibles. La valeur $V_{a,C,A}$ d'un attribut de caractère a pour un agent A dans le contexte C dépend à la fois de la tendance $t_{A,a}$ de A pour a et de l'influence $i_{C,a}$ de C pour a . Il dépend aussi de la force $f_{C,a}$ de C pour l'attribut a . L'idée est que, si C est un contexte « fort » pour l'attribut a , $V_{a,C,A}$ dépend plus de C que de A ; à la limite, si $f_{C,a}=100$, $V_{a,C,A} = i_{C,a}$. A contrario, quand C est faible, $t_{A,a}$ représente la « plus grande partie de » $V_{a,C,A}$; à la limite, si $f_{C,a}=0$, $V_{a,C,A} = t_{A,a}$. La formule suivante reflète simplement cette idée :

$$V_{a,C,A} = \frac{(100 - f_{C,a}) \times t_{A,a}}{100} + \frac{f_{C,a} \times i_{C,a}}{100} \quad (5.1)$$

Pour un agent donné, elle doit être appliquée pour chaque attribut de caractère de A (figure 5.13).

5.3.4 Activation des comportements

Les comportements activés chez les agents sont ceux dont la condition booléenne de la règle de comportement est satisfaite¹². Tant que subsistent des comportements antagonistes, ils ne sont pas exécutés figure 5.13.

5.3.5 Exécution des comportements

L'ensemble des comportements activés peut comporter des éléments incompatibles ou antagonistes, c'est-à-dire qu'ils ne peuvent pas être appliqués ensemble parce qu'ils partagent des ressources non partageables ou que leur concomitance induit des incohérences de comportement (par exemple, freiner et accélérer en même temps). Il est alors nécessaire d'arbitrer entre les comportements incompatibles. Pour ce faire, rappelons que les comportements

12. Jusqu'ici, nous n'avons pas encore mis l'accent sur la représentation des comportements, qui n'est pas au cœur de notre étude. Dans les expériences déjà réalisées, ils sont représentés par des programmes spécifiques. Dans l'état final de la plateforme, nous utiliserons un modèle générique de comportement, comme HPTS (voir par exemple [Don01]).

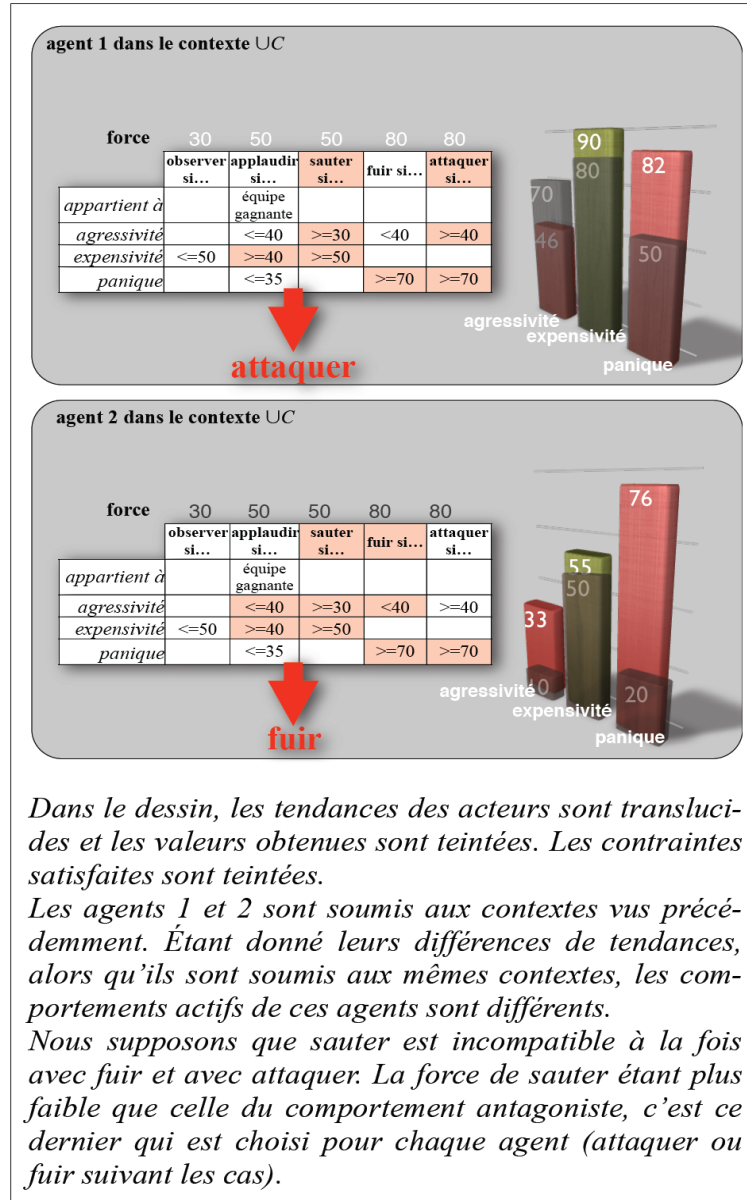


FIGURE 5.13 – Effets des contextes sur les agents 1 et 2.

sont affectés d'une force issue de leur contexte d'origine. C'est le comportement le plus « fort » parmi les comportements antagonistes qui sera choisi et exécuté.

5.4 Expérimentations

Nous avons réalisé plusieurs expérimentations pour illustrer les différentes caractéristiques du modèle, et notamment sa flexibilité et sa capacité de représentation. Pour cela, nous avons soumis des agents à des situations ¹³ « fête et manifestation » que nous avons définis aux préalables.

Nous avons dotés des agents de l'attribut de caractère expressivité en tirant au hasard leur tendances personnelles pour cet attribut (entre 0 et 100). Chacun est donc doté de sa tendance propre.

La description des deux contextes fête et manifestation contient aussi l'attribut expressivité à laquelle est affectée une influence de 80 pour le premier contexte, et 85 pour le deuxième. Ces valeurs représentent l'influence des deux contextes sur les agents pour cet attribut. Les deux contextes fête et manifestation sont dotés respectivement des forces 75 et 80 lesquelles n'ont pas d'utilité quand les contextes ne sont pas en concurrence avec d'autres contextes. Ils contiennent aussi des comportements ; leverLeBrasgauche pour le contexte fête qui se déclenche pour les agents dont la valeur pour l'attribut expressivité en contexte ¹⁴ dépasse le seuil : 60 (les agents qui l'adoptent lèvent bras gauche). Et crier pour le contexte manifestation adopté par les agents qui dépasse le seuil : 70.

1. Nous avons placé un groupe d'agents dans une zone de l'espace dénuée de contexte (en blanc), c'est-à-dire, sans influence sur les agents (figure 5.14). Au début, on a appliqué sur l'environnement un contexte global qui va influencer tous les agents qui se trouvent dans cet environnement. Dans un premier temps, on ne va pas tenir compte des attributs de caractère de chaque agent ; dans ce cas, en soumettant l'environnement au contexte global fête, tous les agents de cet environnement vont adopter le même comportement (dans notre cas leverLeBrasgauche) (figure 5.15 a). Même effet en soumettant cette fois-ci le contexte global manifestation ; dans ce cas, tous les agents vont adopter le comportement crier (représenté dans notre application par un point d'exclamation)(figure 5.15 b). Cependant en tenant compte des attributs de caractère pour chaque agent, cela permet de prendre en considération le

13. représentées par des contextes

14. La valeur qui est calculée à partir de la tendance de l'agent et de l'influence du contexte modulé par sa force.

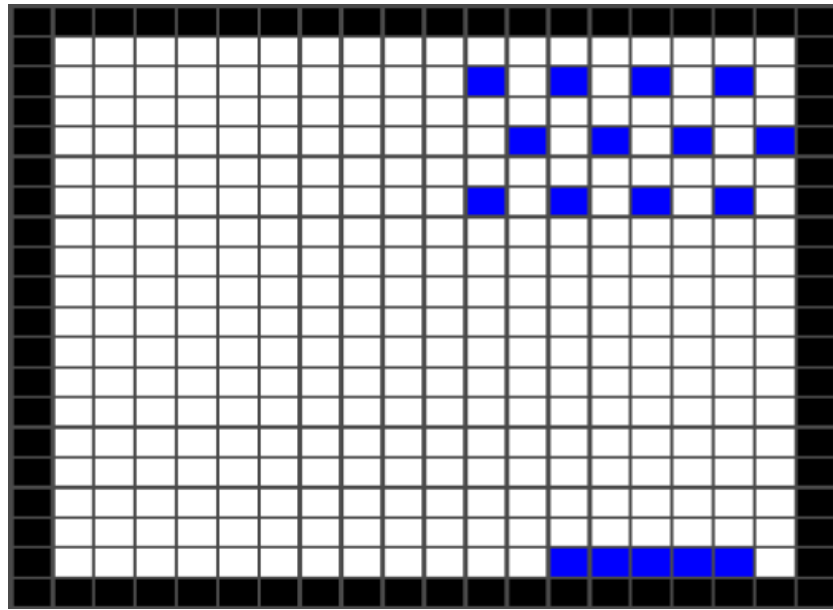


FIGURE 5.14 – Un groupe d’agents en bleu dans une zone neutre en blanc.

caractère de chacun d’eux envers l’influence de la situation globale à laquelle ils sont soumis. Cela implique que les comportements des agents peuvent varier selon l’état interne actuel de chacun. Dans notre cas, lors de la soumission au contexte global fête, certains agents vont lever le bras gauche alors que d’autres non (figure 5.16).

2. Dans ce deuxième cas, nous avons placé dans l’environnement plusieurs zones, lesquelles, correspondent à des contextes localisés (fête ou manifestation). Les zones d’influences correspondantes au contexte localisé fête sont représentées en rouge, et ceux du contexte localisé manifestation en vert. Le reste de l’espace est une zone neutre (Voir figure 5.17). Les agents sont aussi soumis à un contexte global comportant un comportement de déplacement inconditionnel vers la zone neutre ; ils sont donc tous soumis à ce déplacement dès que ce contexte apparaît.

Si les attributs de caractère ne sont pas pris en considération, alors, lors de leur entrée dans la zone d’influence de fête, tous les agents adoptent le comportement leverLeBrasgauche (figure 5.18). La foule est totalement homogène ce qui, dans la plupart des cas, n’est pas réaliste.

Par contre, lorsque les attributs de caractère sont pris en compte, les agents dont la tendance à l’expressivité est la plus forte vont être influencés par le contexte correspondant, et adopteront le comportement leverLeBrasgauche, et les autres non (figure 5.19).

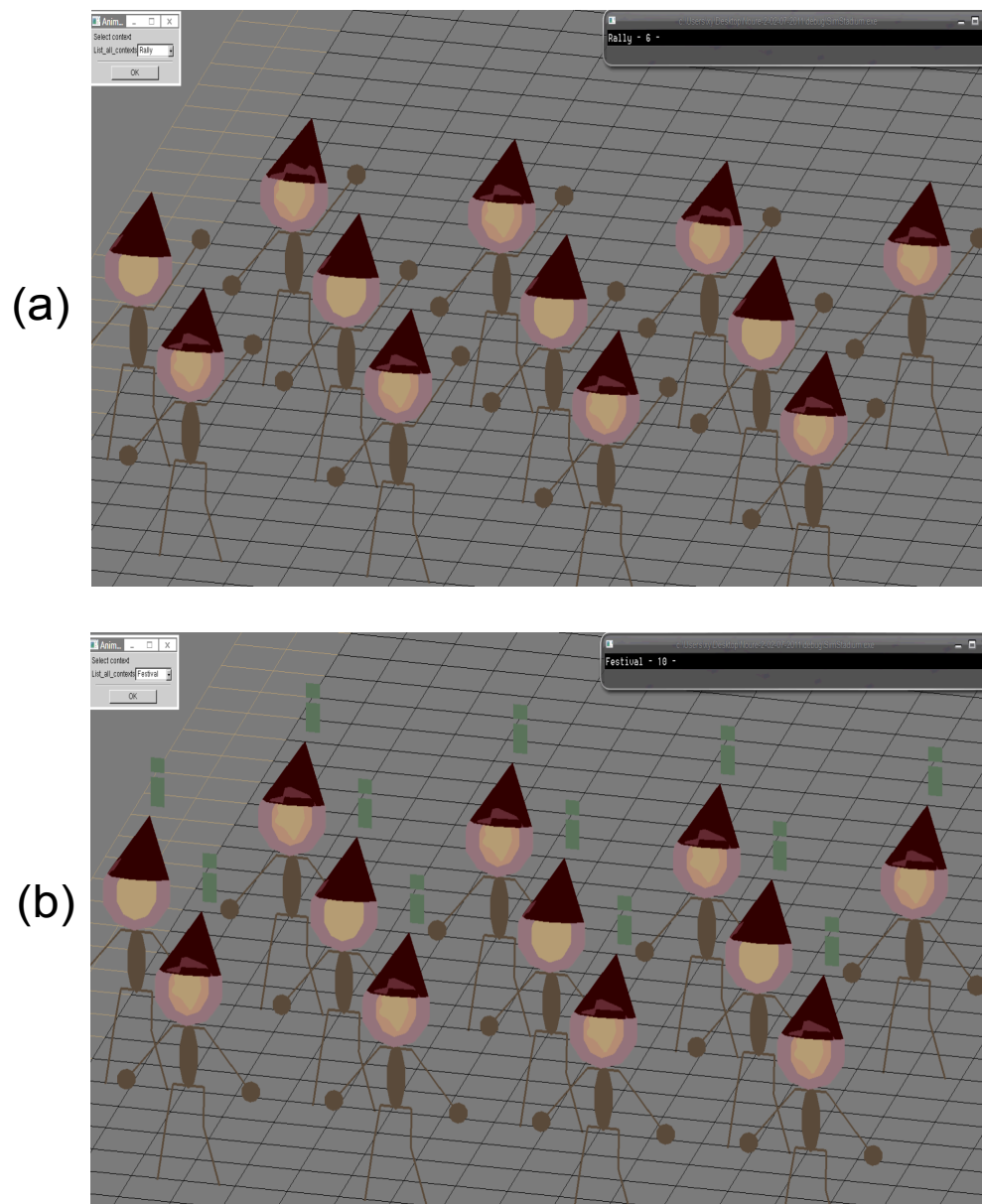


FIGURE 5.15 – Application d'un contexte global « fête (a) » et un autre « manifestation (b) » sans tenir compte des attributs de caractère.

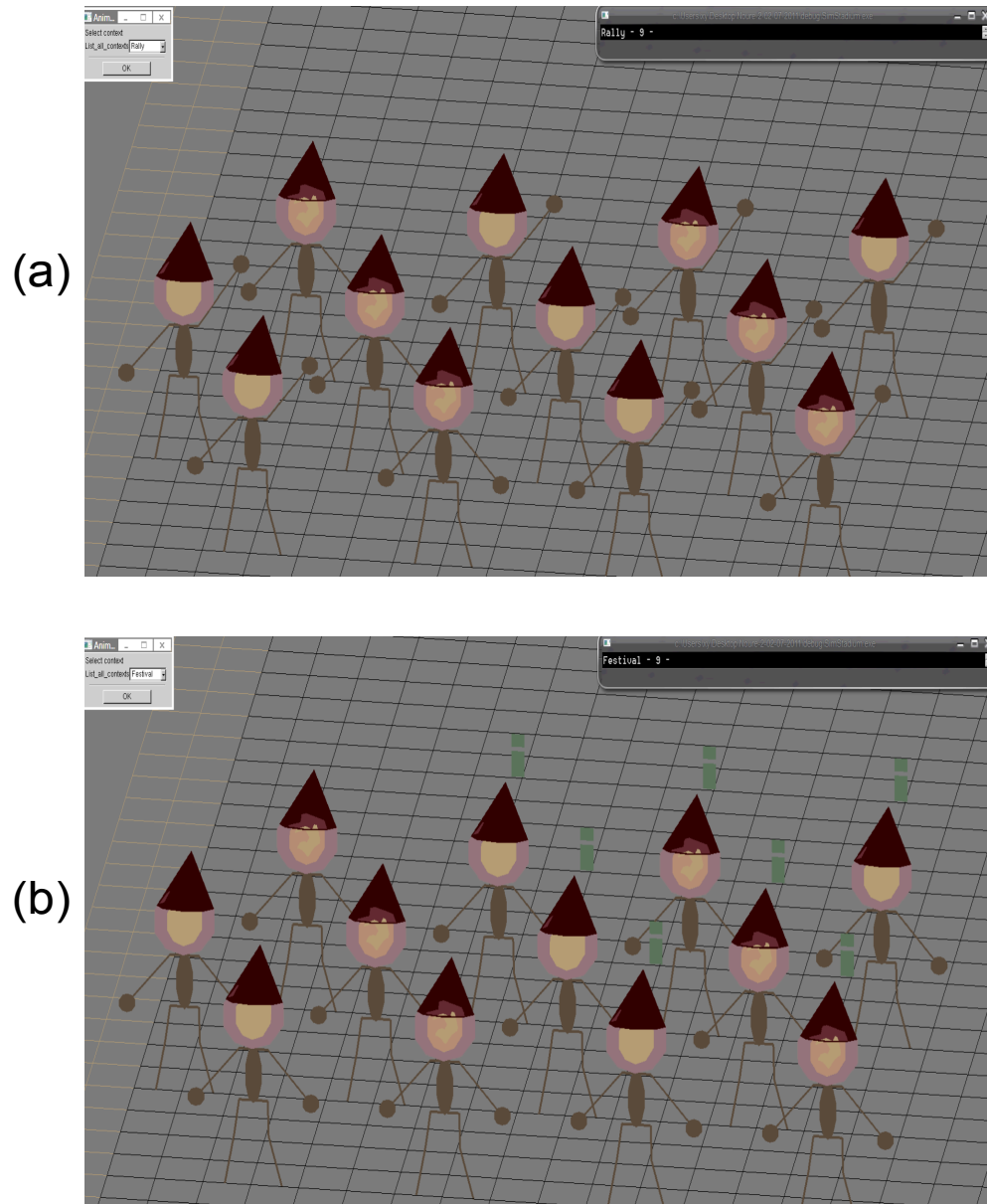


FIGURE 5.16 – Application d’un contexte global « fête (a) » et un autre « manifestation (b) » en tenant compte des attributs de caractère.

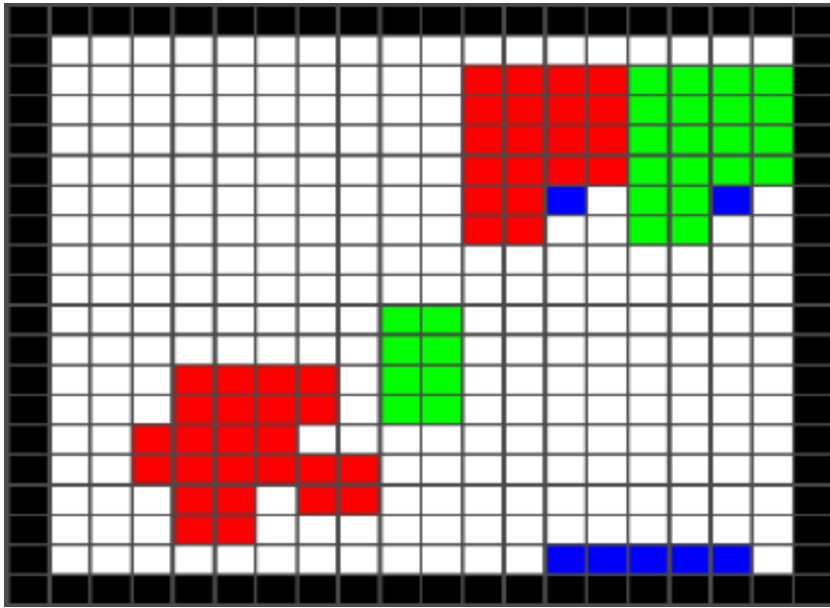


FIGURE 5.17 – Un contexte localisé « fête » en rouge, un autre « manifestation » en vert et une zone neutre en blanc.

Cette expérimentation illustre comment les attributs de caractère et les contextes permettent de donner à une foule un caractère homogène pour certains comportements (ici le déplacement) et hétérogène pour d'autres (ici les bras levés). En jouant sur la force des contextes, il est d'ailleurs possible de jouer dynamiquement sur cette homogénéité ; un contexte « fort » impose un comportement homogène, un contexte « faible » n'impose aucun comportement et un contexte intermédiaire impose ses comportements à certains agents et pas à d'autres.

5.4.1 Deuxième expérimentation : soumission à plusieurs contextes

1. Comme on l'a vu précédemment, lors de la combinaison de plusieurs contextes, un nouveau contexte résultant de cette combinaison sera considéré. Ce nouveau contexte va influencer les agents de l'environnement et cela en tenant compte du facteur temps de chaque contexte. Durant l'écoulement du temps, l'intensité de l'influence du contexte résultant diminuera, ce qui permet à l'état interne de chaque agent d'évoluer (changer) avec le temps.

Dans notre application, nous avons combiné les deux contextes fête

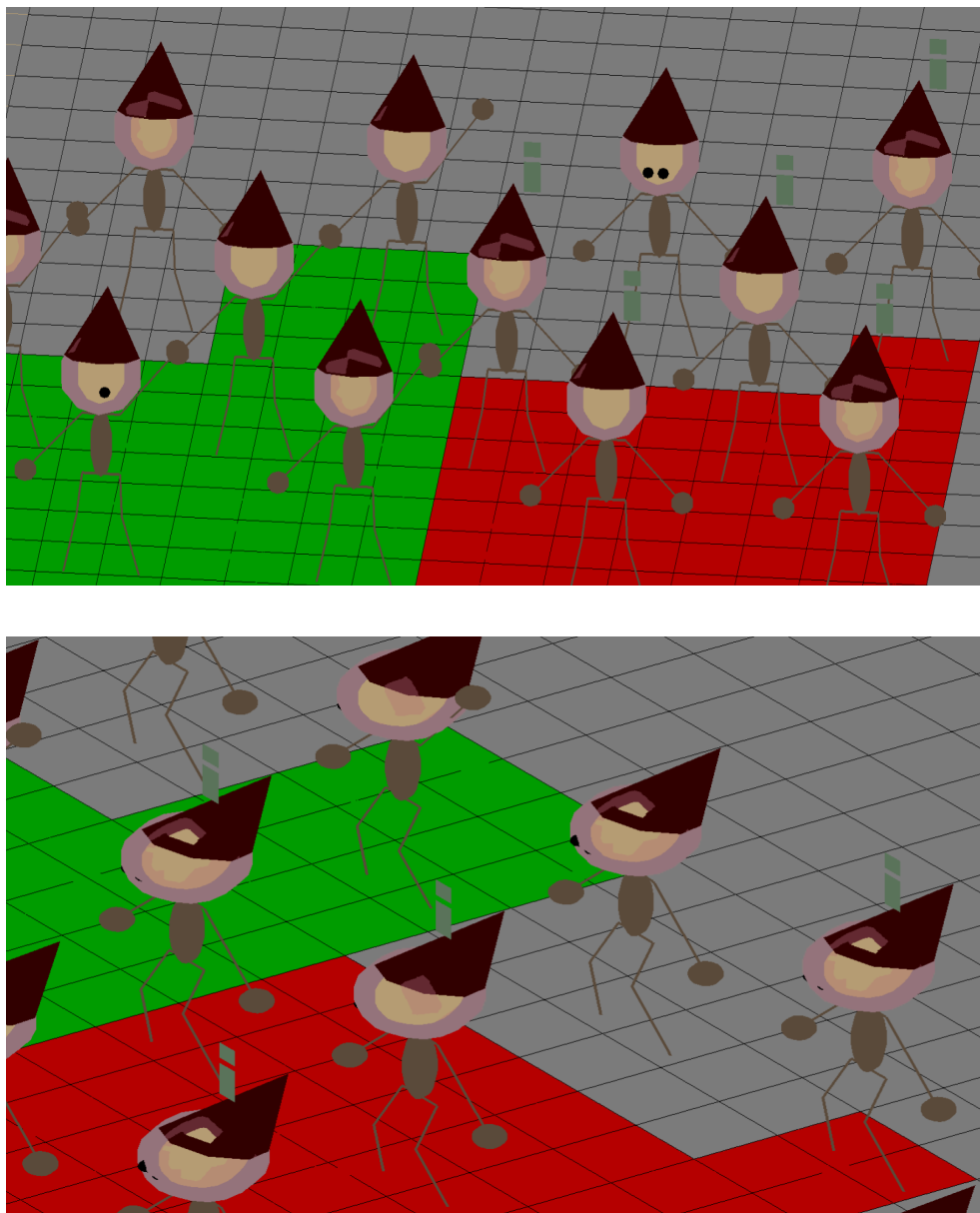


FIGURE 5.18 – Sans les attributs de caractère.

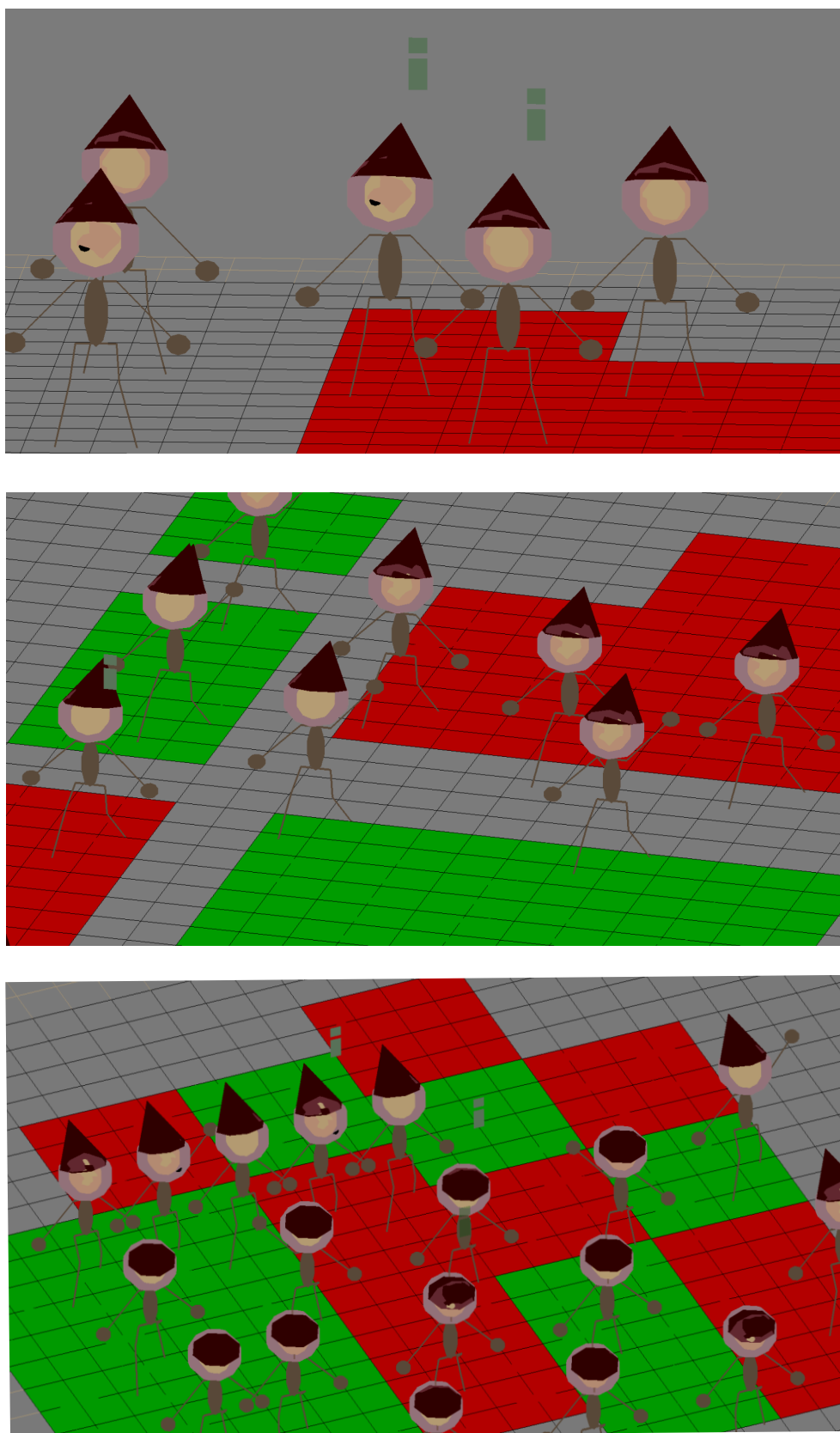


FIGURE 5.19 – Avec les attributs de caractère.

et manifestation ; au début, certains agents vont être complètement influencés par le contexte résultant de cette combinaison et adopter les comportements correspondants à cette union (leverLeBrasgauche et crier). Ensuite, avec le temps, certains agents ne vont plus être totalement influencés par cette combinaison et vont par cela abandonner certains comportements, jusqu'à revenir à leur état initial (c'est-à-dire, qu'ils ne seront plus influencés par aucun des contextes de la combinaison). La figure 5.20 a montre l'état des agents de l'environnement qui sont soumis à une combinaison de contextes globaux (fête et manifestation). Dans ce premier cas, on ne tiens pas compte des attributs de caractère. Tous les agents adopteront les deux comportements leverLeBrasgauche et crier.

Dans la figure 5.20 b, où contrairement à la figure 5.20 a, on tiens compte ici des attributs de caractère pour chaque agent. Dans ce cas, certains agents adoptent les deux comportements alors que d'autres adoptent l'un des deux, ou encore aucun.

L'environnement virtuel comprend ici deux contextes qui se chevauchent : le contexte fête déjà vu (toujours en rouge, auquel on a associé le comportement leverLeBrasgauche), et un contexte manifestation (en vert) . La zone de chevauchement est représentée en noir (figure 5.21).

2. Les agents sont dotés d'un nouvel attribut de caractère gaieté dont les valeurs sont aussi tirées au hasard. On considère que le contexte fête a pour influence 80 sur la gaieté et n'a pas d'influence sur l'expressivité, alors que le contexte manifestation a pour influence 85 sur l'expressivité et aucune sur la gaieté. Le contexte manifestation a pour comportement associé le comportement crier — qui est caractérisé par un point d'exclamation qui surmonte l'agent qui l'adopte. Un agent adopte ce comportement quand sa valeur pour l'expressivité en contexte dépasse : 50. Les deux contextes se chevauchent (zone noire dans la figure 5.22). Les agents figurent au départ dans la zone d'influence d'un seul contexte (manifestation ou fête), mais ils sont ici aussi soumis à un contexte global de déplacement inconditionnel qui les mène dans la zone d'influence du contexte fête. La partie intéressante de l'expérimentation se situe dans la zone de chevauchement. Suivant leurs tendances, les agents adoptent l'un ou l'autre des comportements ou les deux à la fois. Ils « résistent » plus ou moins à un contexte ou à l'autre (figure 5.22). La combinaison de contextes différents permet ainsi d'obtenir des comportements très différents d'un agent à l'autre.

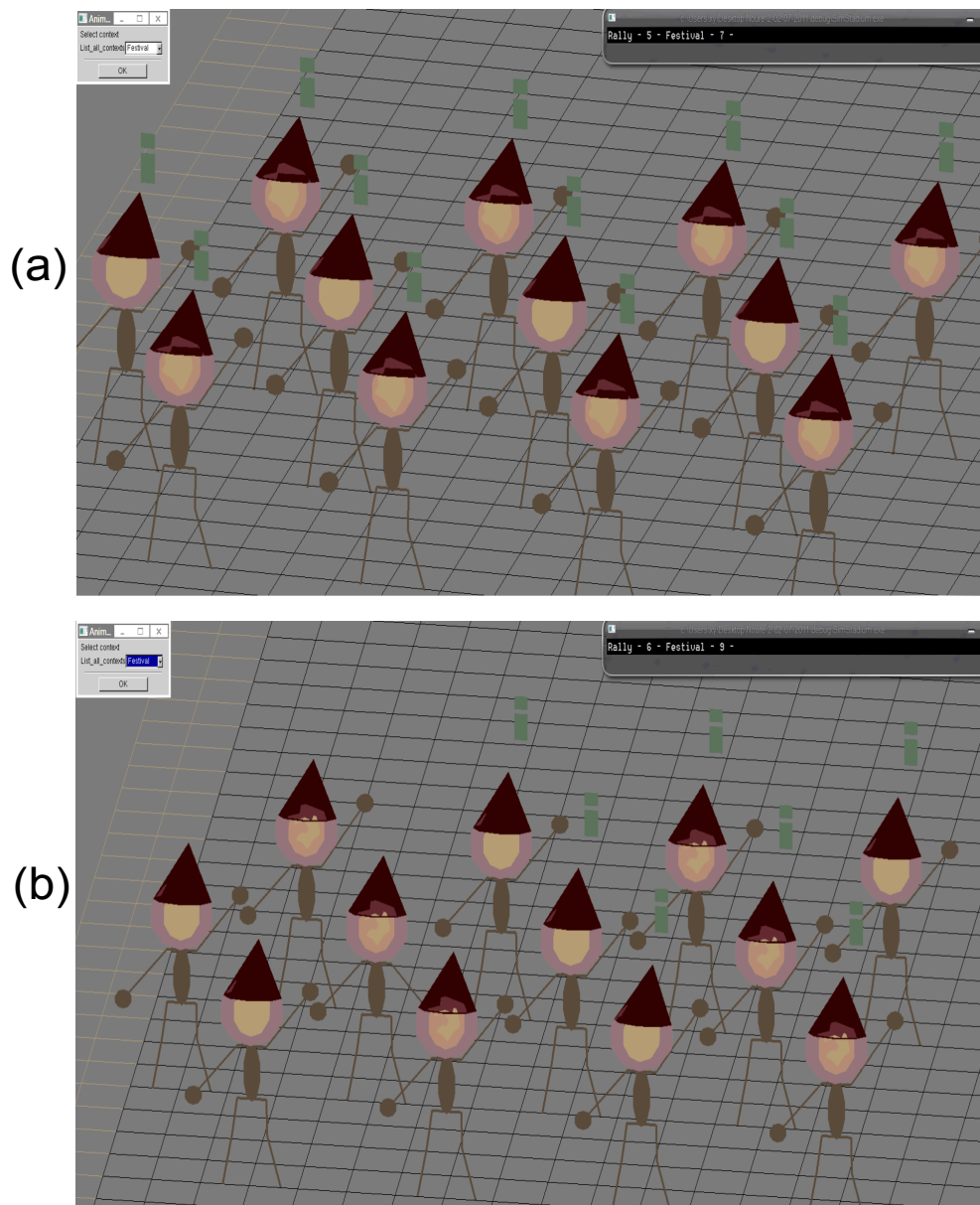


FIGURE 5.20 – (a) Composition de deux contextes globaux « fête (a) » et « manifestation (b) » sans tenir compte des attributs de caractère. (b) Composition de deux contextes globaux « fête (a) » et « manifestation (b) » en tenant compte des attributs de caractère.

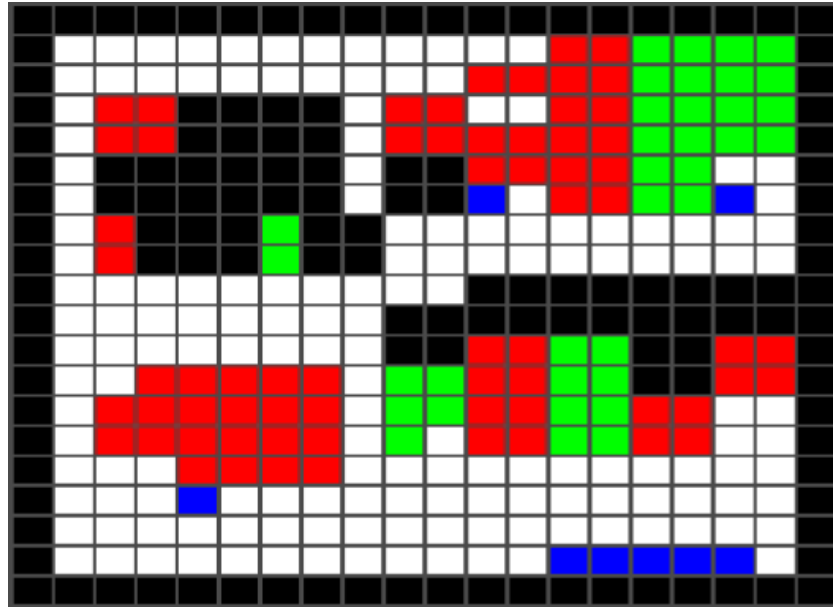


FIGURE 5.21 – Zone de chevauchement en noir de deux contextes localisés.

5.4.2 Troisième expérimentation : contexte propagateur

Le troisième cadre d'expérimentation contient à nouveau le contexte global de déplacement inconditionnel qui, ici, sera aléatoire. Il comporte en plus un autre contexte incendie, qui est un contexte propagateur, représenté par sa zone d'influence ponctuelle en rouge dans la figure 5.23. Comme chaque contexte, il est caractérisé par sa force : 90 et son influence sur les attributs de caractère (dans notre cas ; panique et agressivité), ainsi que par un ensemble de comportements ; pour notre expérience nous avons choisi le comportement leverLesBras (lequel est associé à la règle de déclenchement « si $\text{agressivité} > 70$ alors leverLesBras »)- en plus du comportement propagé nécessaire pour les contextes propagateurs. Suivant leurs tendances, les agents peuvent ou non lever les bras. Par contre, pour des soucis de simplicité de la démonstration, nous avons convenu ici que le comportement de propagation était inconditionnel (adopté d'office par tous les agents qui rencontrent un contexte propagateur). Dans la figure 5.23, le centre de la zone d'influence d'un agent qui devient propagateur est coloré en rouge. Au fur et à mesure que les agents se déplacent ; les clones du contexte incendie sont associés aux agents qui rencontrent, soit le contexte propagateur d'origine, soit un clone associé à un autre agent.

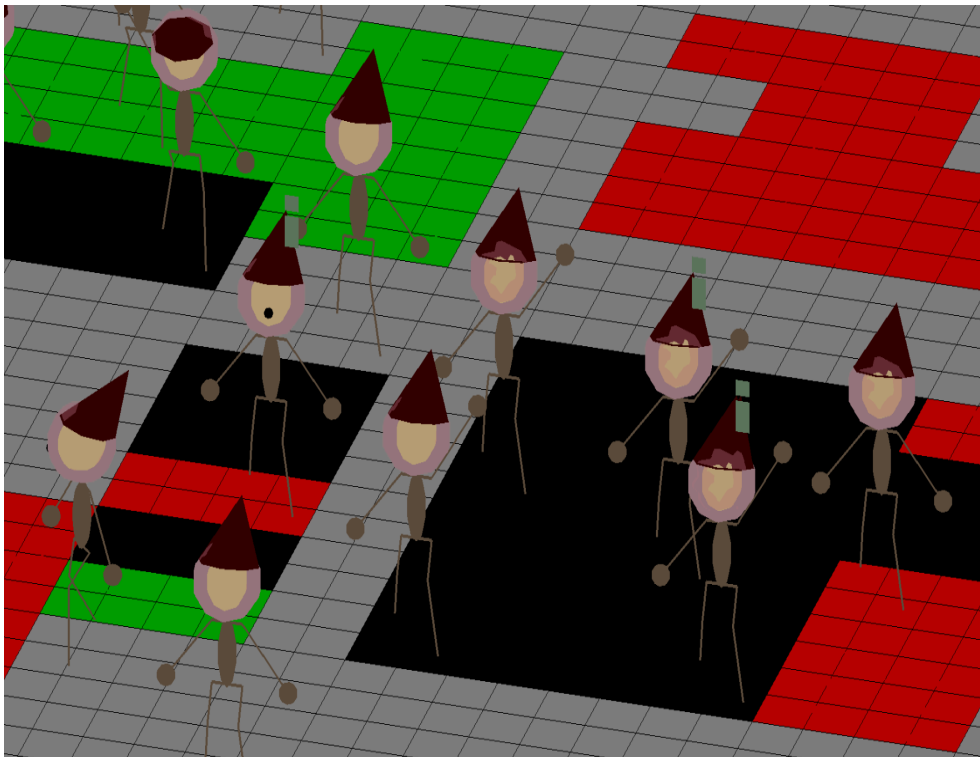


FIGURE 5.22 – Composition de deux contextes localisés « fête » et « manifestation » en tenant compte des attributs de caractère.

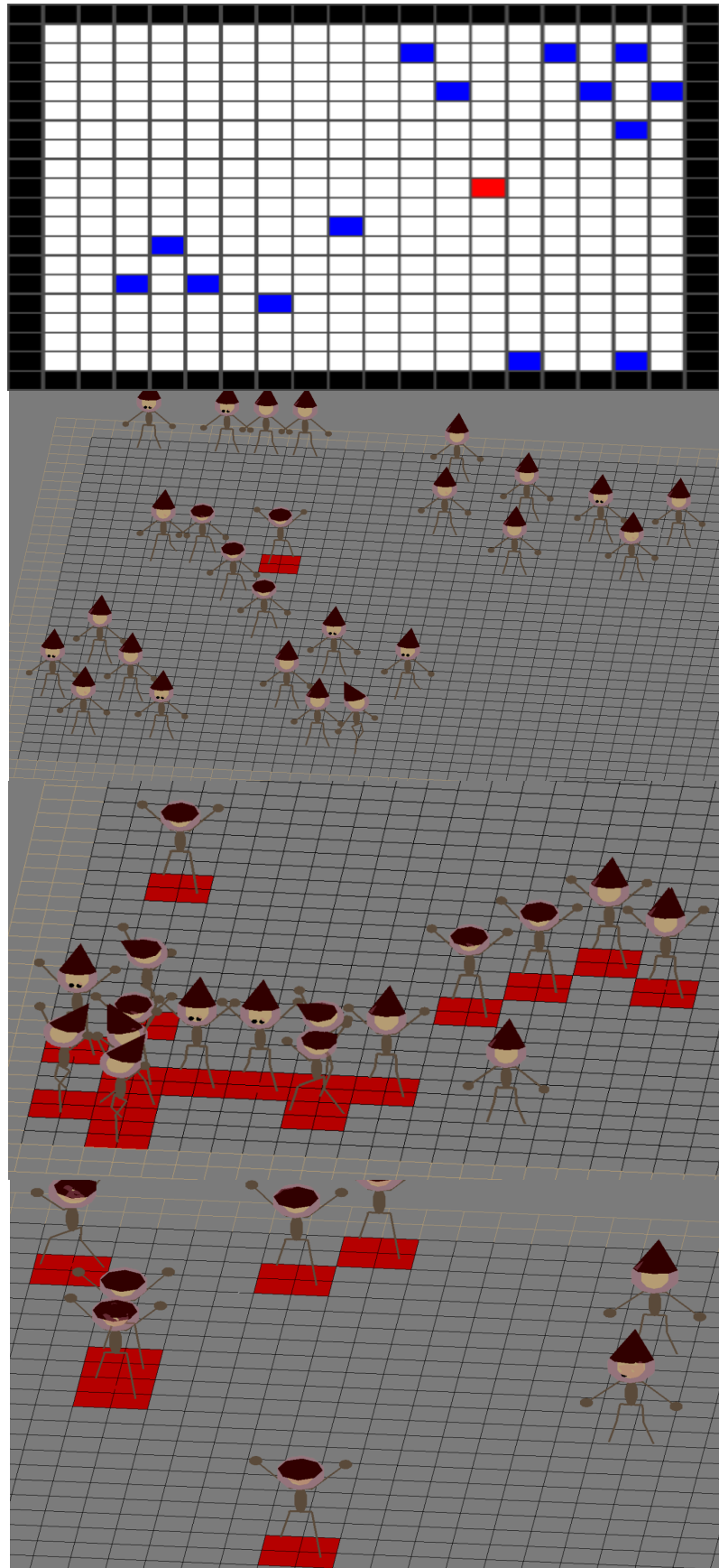


FIGURE 5.23 – Insertion d'un contexte propagateur dans une zone et la propagation de ce dernier.

5.5 Conclusion

Le modèle permet de doter les applications qui en sont issues des fonctionnalités suivantes :

1. Modulation du **contrôle** sur les agents en utilisant la force des contextes (ce contrôle peut même être dynamique). Cela va dans le sens du réalisme macroscopique de l'application ; la modulation du contrôle permet d'éviter ou de résoudre des situations de blocage et de faire intervenir des événements très prégnants, qui s'imposent à tous (situations d'urgence par exemple).
2. **Variabilité** des comportements des agents dans un contexte donné. . . grâce à leurs tendances et aux contextes spécifiques auxquels ils sont soumis. Les agents ne seront pas vraiment autonomes (réalisme microscopique) car cette propriété est atteinte seulement avec des capacités d'évolution mais, au sein d'une foule, la notion de diversité se substitue sans difficulté au concept d'autonomie.
3. **L'adaptabilité** du modèle à de nouvelles situations. Pour faire prendre en compte une nouvelle situation dans une animation donnée, il suffit de décrire le contexte qui lui est associé sans modifier les agents — sauf s'il est nécessaire d'ajouter des attributs de caractère. Dès son ajout à l'animation, un contexte peut ainsi opérer sur les agents (si ce contexte est assez prégnant pour pouvoir influencer l'agent en question).
4. **La communication** dans notre modèle est assuré par le biais des contextes propagateurs. Comme on l'a vu au-dessus, les contextes propagateurs permettent la circulation de l'information entre les agents. Par exemple, dans le cas des situations d'urgences, tel qu'un feu (figure 5.24), un agent qui va découvrir le feu va être soumis à ce contexte et adopter éventuellement ces comportements. Si la règle de déclenchement correspondante au comportement *propager* est déclenchée, alors dans ce cas, l'agent en question va prévenir les autres agents qu'il peut rencontrer.

Dans les applications que nous avons réalisées, chaque agent soumis à un contexte propageur, voit sa case prendre la couleur rouge.

Nous considérons donc que le modèle peut traiter d'une manière assez efficace la relation entre réalisme macroscopique et microscopique. Nous pouvons aussi représenter les différents types de foules distingués plus haut (chapitre 2) :

1. Les **foules homogènes** peuvent être modélisées par des contextes globaux qui contiennent les comportements communs à tous les agents. Aucun attribut de caractère n'est nécessaire (cela signifie que les agents

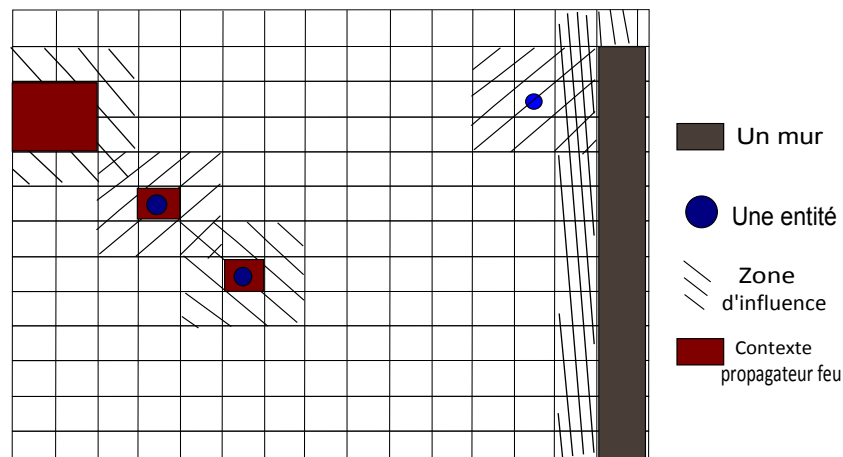


FIGURE 5.24 – Communications entre agents à propos du contexte propagateur « feu ».

n'ont aucune tendance et le contexte n'a aucune influence). Naturellement, il est intéressant de dépasser ce cadre trop restrictif. Une amélioration possible pourrait être de définir des attributs de caractère pour que de légères variations de comportement puissent se produire au sein de la foule. Par exemple, une tendance à la réactivité pourrait être donnée aux agents pour leur donner un comportement plus ou moins réactif en cas de modification de direction ou de vitesse. Parmi de nombreuses autres possibilités, il est également possible de définir des contextes localisés dont le rôle est de contenir seulement des influences ; par exemple, des contextes qui tendent à donner une réactivité plus ou moins grande. Les individus de la foule qui sont soumis à ces contextes (qui atteignent leur zone d'influence), tendront à être plus réactifs.

2. Les **groupes** nécessitent une distribution des rôles et un système de communication. Considérons le cas typique d'une armée. La représentation d'un tel groupe avec notre modèle peut être fondée sur les éléments suivants :
 - (a) Un contexte privé par rôle dans le groupe (ici, par type de soldat : commandement, éclaireur, fantassin, radio,...), qui est affecté à tous les agents qui jouent ce rôle. Le rôle sera techniquement représenté par les comportements du contexte.
 - (b) Un contexte privé partagé par tous les membres du groupe, qui représente le groupe lui-même (ici l'armée) et permet la communication entre les militaires virtuels.
3. Les **collections** fonctionnent comme des agents isolés de plusieurs na-

tures qui sont arbitrairement regroupés. Il n'y a pas de contrainte sur le comportement émergent de la foule (pas de scénario). Savoir modéliser les collections de tous types revient à savoir modéliser tous les types d'agents isolés, y compris, comme dans [MGT99], ceux qui ont des hauts niveaux de cognition. Ces niveaux de cognition nécessitent en général que chaque agent soit doté de capacités d'évolutions, d'un système de croyance et de facultés de communication — alors que notre modèle suppose que l'agent n'est qu'une « personnalité » pilotée par des contextes. Cependant, il est possible d'associer à chaque agent des contextes privés spécifiques, donc des comportements spécifiques. Ces comportements peuvent être entre autres des mécanismes d'apprentissage. Il est aussi possible de stocker de l'information dans ces contextes, qui peut avoir le degré de sophistication souhaité. Cependant, les spécificités du modèle n'ont aucun intérêt dans ce cas ; elles ne facilitent pas la modélisation. Notre objectif est de rester dans le cadre d'une foule d'agents où le comportement collectif est le plus important. Dans ce dernier cadre, nous avons vu que notre modèle était bien adapté.

Chapitre 6

Conclusion et perspectives

Dans ce document, nous avons décrit un modèle d'animation comportementale de foules général qui a donné lieu à un certain nombre d'expérimentations. Nous nous sommes pas intéressés ici aux aspects graphiques de l'animation mais à la construction dynamique de sa structure comportementale, c'est-à-dire à la manière dont chaque agent de la scène va adopter un comportement spécifique (plutôt que d'autres) dans une situation donnée.

Apports du modèle

Flexibilité

Le modèle a une grande unité, une faible dispersion conceptuelle, qui rend sa mise en application aisée. Il semble pouvoir représenter de nombreuses situations d'une manière simple et efficace (partie expérimentation du chapitre 5). Pour enrichir une animation en construction avec de nouvelles situations, il n'est pas besoin de la refondre complètement ni de modifier les agents. Il suffit d'y intégrer de nouveaux contextes (il est envisageable d'effectuer cette opération dynamiquement).

De plus, le chapitre 5 montre comment représenter les différentes catégories de foules (foule homogène, collection d'agents, groupes) dont la combinaison nous paraissait pouvoir constituer la plupart des foules virtuelles possibles (chapitre 2).

En dernier lieu, étant donné sa simplicité, le modèle lui-même peut être enrichi sans perdre de son unité.

Contrôle versus autonomie

En utilisant d'une part l'influence des contextes — pondérée par leur force, dynamiquement modifiable — et d'autre part les tendances des agents — qui représentent leur « personnalité » —, il est possible de faire évoluer au cours d'une même animation le contrôle sur les agents. Suivant les circonstances, un agent peut alors être quasi autonome ou fortement contrôlé. C'est une caractéristique de notre modèle qui n'est pas traitée, au moins d'une manière aussi simple, dans les autres études.

Ces mêmes caractéristiques permettent d'interagir avec la foule d'une manière flexible et efficace en la plongeant dans des contextes successifs ou simultanés qui correspondent au scénario de l'animation ou à l'action d'avatars.

Circulation de l'information

La perception des agents est assurée par les contextes auxquels ils sont soumis. Les contextes localisés peuvent aussi bien représenter des pôles attracteurs (tableaux dans un musée) que d'autres agents à éviter (un contexte localisé est associé à chaque obstacle, statiques ou mobiles). Les contextes propagateurs assurent aussi la communication d'informations entre agents. Ce sont des contextes privés localisés qui essaient d'un agent à un autre en perdant ou non leur capacité de diffusion.

Le traitement de ces différents types d'interaction, plus ou moins complexes, est assuré toujours par les seuls quelques éléments de base du modèle.

Limites

Situations émergentes

L'évolution d'une animation peut engendrer des situations problématiques (interblocages par exemple), qui peuvent nécessiter que les agents adoptent des comportements spécifiques (retour en arrière par exemple). Dans le cadre du modèle actuel, ces comportements doivent être supportés par un contexte créé pour l'occasion. Or, les agents ne peuvent pas a priori créer un tel contexte car il faudrait qu'ils soient dotés d'une vision globale pour constater la situation problématique. Dans l'état actuel du modèle, un contexte ne peut pas être créé par le système lui-même. Seule une intervention d'un utilisateur via un avatar peut actuellement résoudre le problème.

A noter cependant que, dans tous les modèles où l'autonomie des agents est privilégiée, il est aussi malaisé de traiter des problèmes de cet ordre.

Sophistication des agents

Nous avons vu que le modèle décrit dans ce document permet de représenter la plupart des comportements au sein d'une foule, mais les agents n'y sont pas vraiment autonomes. C'est la variété des comportements, provenant des contextes auxquels ils sont soumis et de leurs tendances¹ qui donne l'illusion de l'autonomie. On peut se poser la question des limites de cette « illusion », c'est-à-dire si notre modèle pourrait permettre de modéliser des agents d'un plus haut niveau de cognition (hors de l'animation comportementale de foule par exemple) en conservant ces principes. Dans ce cadre, les limites de notre modèle sont rapidement atteintes. Les hauts niveaux de cognition requièrent des niveaux de contrôles cognitifs, rationnels ou sociaux. Ces fonctionnalités cognitives nécessitent des processus d'apprentissage, des systèmes de communication et une connaissance du monde (un système de croyances) propre à chaque agent. Rien n'interdit de doter des contextes (privés en l'occurrence) de comportements de haut niveau, comme des mécanismes d'apprentissage. Il est aussi possible de stocker de l'information dans ces contextes ou les agents eux-mêmes, qui peut avoir le degré de sophistication souhaité. Cependant, les spécificités du modèle n'ont aucun intérêt dans ce cas ; elles ne facilitent pas la modélisation.

L'idée reste donc bien que nous nous situons dans le cadre d'une foule d'agents et que le comportement collectif est le plus important. Dans ce dernier cadre, nous avons vu que notre modèle était bien adapté.

Perspectives

Actuellement, le modèle a été instancié en plusieurs applications qui ont permis de réaliser les expérimentations du chapitre 5. Le prolongement naturel de ces travaux est de réaliser une plateforme abstraite, dotée d'un langage spécifique, qui puisse constituer un générateur d'applications efficace. Cette plateforme devra intégrer aussi les aspects graphiques qui n'ont pas fait l'objet de notre étude et constituent une perspective de développements importante.

1. valeurs de référence des attributs de caractères propres à chaque agent.

Table des figures

2.1	Collection d'agent [MGT99].	9
2.2	Groupe (édité par Electronic Arts).	10
2.3	Règles locales de Reynolds [Rey87].	15
2.4	Capture d'écran d'une scène de protestation développé en utilisant PMFServ et OpenSteer [PB05].	16
2.5	Le contrôle interactif.	19
2.6	La communication au niveau local entre les agents.	20
3.1	Les informations collectées	27
3.2	Le modèle de l'espace d'influence rectangulaire [PAB07][PG06].	29
3.3	Modèle de tableau noir [EL80].	31
3.4	Représentation en utilisant une grille de cellules	33
3.5	La figure de gauche montre la trajectoire des déplacements observées et les potentiels attribués. La figure de droite montre des forces résultantes des champs de potentiels [TBS04].	34
3.6	Capture d'écran d'une application interactive où un humain virtuel est capable de marcher à n'importe quel endroit sélectionné sans entrer en collision avec des boîtes à l'intérieur de la salle. Les boîtes sont également capables de changer de position. En bas à droite sa représentation correspondante par triangulation de delaunay. [KBT03].	34
3.7	Les trois niveaux hiérarchiques « géométrique, sémantique, application » [JXM ⁺ 09]	35
3.8	La force répulsive du chien est plus élevé, ce qui oblige le groupe à choisir un autre chemin [BLA03]	37
3.9	A gauche : Une carte de cheminement probabiliste. A droite : Une série de points de passages qui relies deux positions. [SKG05]	37
3.10	Optimisation visuelle de la trajectoire [LD04]	37
4.1	Interaction avec un smart object <i>casier</i> [KT02].	42
4.2	Une application basée sur du texte [KT98].	43

4.3	Un agent (voleur) interagissant avec des smart objects dans un laboratoire. [KMCT00].	44
4.4	Interaction avec un smart object « toilette » [SKT01].	45
4.5	(à gauche) Un état machine simple pour une fonctionnalité de l'ascenseur. (à droite) Un état machine plus complexe pour l'ascenseur, où des états intermédiaires sont considérés [KT02].	45
4.6	Exemple d'évacuation de foule avec un grand contre un petit nombre de leaders [PB06]	50
4.7	Architecture de PMFserv [PB05].	51
4.8	La boucle d'interaction principal [PB05].	52
4.9	Les deux niveaux du système HiDAC [PG06].	53
4.10	Classeur de Holland.	54
4.11	Schéma de fonctionnement de LCS [BBMH08].	55
4.12	Un exemple avec LCS.	56
4.13	Groupes évitant la collision avec des obstacles [MT97].	57
4.14	Architecture d'agent à deux niveaux.	58
4.15	Les fonctions de comportements calcules la distribution des probabilités à l'entrée de chaque état. L'état suivant est sélectionné par échantillonnage sur la distribution de la probabilité composée [SGC04].	59
4.16	Plan en 2D de la ville de Québec [MCP ⁺ 03].	61
5.1	Les tendances de l'agent 1 et 2 pour les trois attributs de caractère.	71
5.2	Un contexte privé associé aux spectateurs.	73
5.3	Un contexte global dans un match de football.	74
5.4	Quelques comportements possibles de spectateurs.	75
5.5	<i>Foule de spectateurs dans un match de football soumise à des contextes globaux</i> Situation normale, But, Feu.	77
5.6	Un contexte localisé dans le match de football.	78
5.7	Trois contextes localisés.	79
5.8	Soumission d'une entité à des contextes localisés.	81
5.9	Soumission d'une entité à une combinaison de contextes localisés.	82
5.10	(À gauche) avec une grille de cellules uniformes. (À droite) avec une grille de cellules non uniformes.	85
5.11	Des entités avec leurs zones d'influences.	86
5.12	<i>Combinaison des trois contextes précédents.</i> Contexte global « But », contexte localisé « rixe », et contexte privé « match de football ».	87
5.13	Effets des contextes sur les agents 1 et 2.	89
5.14	Un groupe d'agents en bleu dans une zone neutre en blanc. . .	91

5.15	Application d'un contexte global « fête (a) »et un autre « manifestation (b) »sans tenir compte des attributs de caractère. .	92
5.16	Application d'un contexte global « fête (a) »et un autre « manifestation (b) »en tenant compte des attributs de caractère. .	93
5.17	Un contexte localisé « fête »en rouge, un autre « manifestation »en vert et une zone neutre en blanc.	94
5.18	Sans les attributs de caractère.	95
5.19	Avec les attributs de caractère.	96
5.20	(a) Composition de deux contextes globaux « fête (a) »et « manifestation (b) »sans tenir compte des attributs de caractère. (b) Composition de deux contextes globaux « fête (a) »et « manifestation (b) »en tenant compte des attributs de caractère. . .	98
5.21	Zone de chevauchement en noir de deux contextes localisés. . .	99
5.22	Composition de deux contextes localisés « fête »et « manifestation »en tenant compte des attributs de caractère.	100
5.23	Insertion d'un contexte propagateur dans une zone et la propagation de ce dernier.	101
5.24	Communications entre agents à propos du contexte propagateur « feu ».	103

Liste des tableaux

2.1	Modèles analysés dans [Par07].	13
4.1	Smart objects	62
4.2	MACES	63
4.3	HiDAC	64
4.4	LCS	65
4.5	ViCrowd	65
4.6	Modèle de comportements scalable	66
4.7	MAGS	67

Bibliographie

- [ACT05] T. Abaci, J. Cíger, and D. Thalmann. Planning with Smart Objects. In *WSCG (Short Papers)*, pages 25–28, 2005.
- [AMC03] Matt Anderson, Eric McDaniel, and Stephen Chenney. Constrained animation of flocks. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA)*, pages 286–297, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [BBMH08] Larry Bull, Ester Bernadó-Mansilla, and John H. Holmes. Learning classifier systems in data mining : An introduction. In *Learning Classifier Systems in Data Mining*, pages 1–15. 2008.
- [BLA03] O. Burchan Bayazit, Jyh-Ming Lien, and Nancy M. Amato. Better group behaviors in complex environments using global roadmaps. In *Proceedings of the eighth international conference on Artificial life*, pages 362–370, Cambridge, MA, USA, 2003. MIT Press.
- [BMMV04] Stefania Bandini, Sara Manzoni, S. Manzoni, and Giuseppe Vizzari. Situated cellular agents for crowd simulation and visualization, 2004.
- [BY98] J.-D. Boissonnat and M. Yvinec. Algorithmic geometry. *Cambridge University Press*, 1998.
- [Che04] Stephen Chenney. Flow tiles. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA)*, pages 233–242, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.
- [CL83] Daniel D. Corkill and Victor R. Lesser. The use of meta-level control for coordination in a distributed problem solving network. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 748–756. IEEE Computer Society Press, 1983.

- [DDLT02] Frédéric Devillers, Stéphane Donikian, Fabrice Lamarche, and Jean-François Taille. A programming environment for behavioural animation. *Journal of Visualization and Computer Animation*, 13(5) :263–274, 2002.
- [DF92] Alexis Drogoul and Jacques Ferber. From tom thumb to the dockers : Some experiments with foraging robots. In *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 451–459. MIT Press, 1992.
- [DHV00] I. Farkas D. Helbing and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803) :487–490, 2000.
- [Don01] Stéphane Donikian. Hpts : a behaviour modelling language for autonomous agents. In *AGENTS '01 : Proceedings of the fifth international conference on Autonomous agents*, pages 401–408, New York, NY, USA, 2001. ACM Press.
- [Don04] Stéphane Donikian. *Modélisation, contrôle et animation d'agents virtuels autonomes évoluant dans des environnements informés et structurés*. Hdr, L'Université de Rennes 1, 2004.
- [DR95] Stéphane Donikian and Eric Rutten. Reactivity, concurrency, data-flow and hierarchical preemption for behavioural animation. In R.C. Veltkamp (E.H. Blake). Springer-Verlag, editor, *Programming Paradigms in Graphics'95*, 1995.
- [EHN94] Kutluhan Erol, James Hendler, and Dana S. Nau. Htn planning : Complexity and expressivity. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pages 1123–1128. AAAI Press, 1994.
- [EL80] Lee D. Erman and Victor R. Lesser. The hearsay-ii speech understanding system : Integrating knowledge to resolve uncertainty. *Computing Surveys*, 12(2) :213–253, 1980.
- [FTT99] John Funge, Xiaoyuan Tu, and Demetri Terzopoulos. Cognitive modeling : knowledge, reasoning and planning for intelligent characters. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques, SIGGRAPH '99*, pages 29–38, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [GGL⁺98] H. H. González-Baños, L. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, R. Motwani, and C. Tomasi. Motion planning with visibility constraints : Building autonomous observers. In Y. Shirai and S. Hirose, editors, *Proceedings Eighth International Sympo-*

- sium on Robotics Research*, pages 95–101. Springer-Verlag, Berlin, 1998.
- [HD00] Vicsek T. Helbing D., Farkas I. Simulating dynamical features of escape panic. *Nature*, 407(407) :487–490, 2000.
- [HE06] Xiaolin Hu and Donald H. Edwards. Context-dependent structure control for adaptive behavior selection. In *Workshop on Bio-inspired Cooperative and Adaptive Behaviours in Robots, in cooperation with the ninth international conference on the simulation of adaptive behavior (sab'06)*, 2006.
- [Hen71] L. F. Henderson. The Statistics of Crowd Fluids. *Nature*, 229(5284) :381–383, February 1971.
- [HMF01] Dirk Helbing, P Molnar, I J Farkas, and K Bolay. Self-organizing pedestrian movement. *Environment and Planning B Planning and Design*, 28(3) :361–383, 2001.
- [Hu06] Xiaolin Hu. Context-dependent adaptability in crowd behavior simulation. In *The 2006 IEEE International Conference on Information Reuse and Integration (IRI'06)*, pages 214–219, 2006.
- [J95] Ferber. J. *Les systèmes multi-agents. Vers une intelligence collective*. InterEditions, Paris, intereditions edition, 1995.
- [JNR98] Wooldridge M Jennings N. R. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, pages 7–38, 1998.
- [JXM⁺09] Hao Jiang, Wenbin Xu, Tianlu Mao, Chunpeng Li, Shihong Xia, and Zhaoqi Wang. A semantic environment model for crowd simulation in multilayered complex environment. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology, VRST '09*, pages 191–198, New York, NY, USA, 2009. ACM.
- [Kal04] Marcelo Kallmann. Interaction with 3-d objects. In Nadia Magnenat-Thalmann and Daniel Thalmann, editors, *Handbook of Virtual Humans*, pages 303–322. John Wiley & Sons, first edition, 2004.
- [KBT03] M Kallmann, H Bieri, and D Thalmann. Fully Dynamic Constrained Delaunay Triangulations. *Geometric Modelling for Scientific Visualization*, G. Brunnnett, B. Hamann, H. Mueller (Eds.), Springer-Verlag, 2003.
- [Kha86] O Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Rob. Res.*, 5(1) :90–98, April 1986.

- [KMCT00] M. Kallmann, J. S. Monzani, A. Caicedo, and D. Thalmann. ACE : a platform for the real time simulation of virtual human agents. Computer Animation and Simulation 2000. Proceedings of the Eleventh Eurographics Workshop. Eurographics. Springer-Verlag/Wien, 2000. EPFL Comput. Graphics Lab., LIG, Lausanne, Switzerland.
- [KS03] Namazi A. Nishinari K. Kirchner, A. and A Schadschneider. Role of conflicts in the floor field cellular automaton model for pedestrian dynamics. In *2nd International Conference on Pedestrians and Evacuation Dynamics*, pages 51–62, 2003.
- [KT98] Marcelo Kallmann and Daniel Thalmann. Modeling objects for interaction tasks. In *Proc. Eurographics Workshop on Animation and Simulation*, pages 73–86, 1998.
- [KT02] M. Kallmann and D. Thalmann. Modeling behaviors of interactive objects for real time virtual environments. *Journal of Visual Languages and Computing*, 13(2) :177–195, 2002.
- [LD02] Fabrice Lamarche and Stéphane Donikian. Automatic orchestration of behaviours through the management of resources and priority levels. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems : part 3*, AAMAS '02, pages 1309–1316, New York, NY, USA, 2002. ACM.
- [LD04] Fabrice Lamarche and Stéphane Donikian. Crowd of virtual humans : a new approach for real time navigation in complex and structured environments. *Computer Graphics Forum*, 23(3) :509–518, 2004.
- [LGN⁺09] Ming C. Lin, Stephen Guy, Rahul Narain, Jason Sewall, Sachin Patil, Jatin Chhugani, Abhinav Golas, Jur Berg, Sean Curtis, David Wilkie, Paul Merrell, Changkyu Kim, Nadathur Satish, Pradeep Dubey, and Dinesh Manocha. Interactive modeling, simulation and control of large-scale crowds and traffic. In *Proceedings of the 2nd International Workshop on Motion in Games*, MIG '09, pages 94–103, Berlin, Heidelberg, 2009. Springer-Verlag.
- [LMM03] Celine Loscos, David Marchal, and Alexandre Meyer. Intuitive crowd behaviour in dense urban environments using local laws. In *Proceedings of the Theory and Practice of Computer Graphics 2003*, TPCG '03, pages 122–129. IEEE Computer Society, 2003.
- [LRDG90] Jed Lengyel, Mark Reichert, Bruce R. Donald, and Donald P. Greenberg. Real-time robot motion planning using rasterizing

- computer graphics hardware. In *Proc. SIGGRAPH*, pages 327–335, 1990.
- [MB75] S. J. Melinek and S. Booth. An analysis of evacuation times from buildings. In *CIB Symposium on the control of smoke movement in building fires*. Building research establishment, Fire research station., Watford, U.K., 1975.
- [MBCT98] Soraia R. Musse, Christian Babski, Tolga Capin, and Daniel Thalmann. Crowd modelling in collaborative virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology, VRST '98*, pages 115–123, New York, NY, USA, 1998. ACM.
- [McG70] Joseph E. McGrath. A conceptual formulation for research on stress. *Social and Psychological factors in stress*, 3 :1–13, 1970.
- [MCP⁺03] Bernard Moulin, Walid Chaker, Jimmy Perron, Patrick Pelletier, Jimmy Hogan, and Edouard Gbei. Mags project : Multi-agent geosimulation and crowd simulation. In *COSIT*, pages 151–168, 2003.
- [MGT99] Soraia R. Musse, F Garat, and D Thalmann. Guiding and Interacting with Virtual Crowds in Real-time. In *Proc. Eurographics Workshop on Animation and Simulation '99 (CAS '99)*, Springer, Wien, pages 23–34, 1999.
- [MT97] Soraia R. Musse and D. Thalmann. A model of human crowd behavior : Group inter-relationship and collision detection analysis. In *Proc. Workshop of Computer Animation and Simulation of Eurographics'97*, pages 39–51, 1997.
- [MT01] Soraia R. Musse and Daniel Thalmann. Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, 7(2) :152–164, 2001.
- [MYMT07] Fiorenzo Morini, Barbara Yersin, Jonathan Maïm, and Daniel Thalmann. Real-Time Scalable Motion Planning for Crowds. In *Cyberworlds International Conference*, pages 144–151, 2007.
- [Nii86] H. Penny Nii. Blackboard systems, part one : The blackboard model of problem solving and the evolution of blackboard architectures. *AI Magazine*, 7(2) :38–53, 1986.
- [Nil93] Nils J. Nilsson. *Principles of Artificial Intelligence*. Morgan Kaufmann Publishers, January 1993.
- [NT96] Hansrudi Noser and Daniel Thalmann. The animation of autonomous actors based on production rules. In *Proceedings Computer Animation'96*, pages 47–57. Society Press, 1996.

- [Nyg07] Martin Nygren. Simulation of human behaviour in stressful crowd situations. Master's thesis, Royal Institute of Technology School of Computer Science and Communication, SE-100 44 Stockholm, Sweden, 2007.
- [OA88] Collins A. Ortony A., Clore G.L. The cognitive structure of emotions. *Cambridge University Press*, Cambridge, 1988.
- [OD02] Cassell J. Vilhjalmsen H. Dobbyn S. Peters C. Leeson W. Giang T. O'sullivan, C. and J. Dingliana. Crowd and group simulation with levels of detail for geometry, motion and behavior. In *Third Irish Workshop on Computer Graphics*, pages 15–20, 2002.
- [OPOD10] Jan Ondřej, Julien Pettré, Anne-Hélène Olivier, and Stéphane Donikian. A synthetic-vision based steering approach for crowd simulation. *ACM Trans. Graph.*, 29(4) :1–9, July 2010.
- [PAB07] N. Pelechano, J. M. Allbeck, and N. I. Badler. Controlling individual agents in high-density crowd simulation. In *SCA '07 : Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 99–108, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [Par07] Sébastien Paris. *Caractérisation des niveaux de services et modélisation des circulations de personnes dans les lieux d'échanges*. Phd thesis, Université de Renne, 2007.
- [PB05] O'Brien K. Silverman B. Pelechano, N. and N. Badler. Crowd simulation incorporating agent psychological models, roles and communication. In *First International Workshop on Crowd Simulation. (V-CROWDS '05)*, pages 21–30, 2005.
- [PB06] Nuria Pelechano and Norman I. Badler. Modeling crowd and trained leader behavior during building evacuation. *IEEE Computer Graphics and Applications*, 26(6) :80–86, 2006.
- [PDMO03] Christopher Peters, Simon Dobbyn, Brian Macnamee, and Carol O'Sullivan. Smart Objects for Attentive Agents, 2003.
- [Pel03] Patrick Pelletier. *simulation de comportements de foule*. PhD thesis, université laval, québec, université laval, québec, septembre 2003.
- [PG06] Nuria Pelechano Gomez. *Modeling realistic high density autonomous agent crowd movement : social forces, communication, roles and psychological influences*. PhD thesis, Philadelphia, PA, USA, 2006. Adviser-Badler, Norman I.

-
- [PM78] V.M. Predtechenskii and Milinskiĭ. *Planning for foot traffic flow in buildings*. TT. Published for the National Bureau of Standards, U.S. Dept. of Commerce, and the National Science Foundation, Washington, D.C. by Amerind Pub. Co., 1978.
- [PoBR75] J.L. Pauls and National Research Council Canada. Division of Building Research. *Evacuation and other fire safety measures in high-rise buildings*. Research paper ... of the Division of Building Research. Division of Building Research, National Research Council, 1975.
- [Rey87] Craig W. Reynolds. Flocks, herds, and schools : A distributed behavioral model. In *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, New York, NY, USA, 1987. ACM.
- [Rey99] Craig Reynolds. Steering behaviors for autonomous characters. pages 763–782, 1999.
- [RG91] Anand S. Rao and Michael P. Georgeff. Modeling rational agents within a bdi-architecture. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484. Morgan Kaufmann publishers Inc. : San Mateo, CA, USA, 1991.
- [RKBB94] Barry D. Reich, Hyeonseok Ko, Welton Becket, and Norman I. Badler. Terrain reasoning for human locomotion. In *Proceedings of Computer Animation '94*, pages 996–1005. IEEE Computer Society Press, 1994.
- [SBOC06] Barry G. Silverman, Gnana Bharathy, Kevin O'Brien, and Jason Cornwell. Human behavior models for agents in simulators and games : Part ii : Gamebot engineering with pmfserv. *Presence*, 15(2) :163–185, 2006.
- [Sep07] Cyril Septseault. *Représentation d'environnements virtuels informés et de leur dynamique par un personnage autonome en vue d'une crédibilité comportementale*. PhD thesis, Université de Brest, 11 décembre 2007.
- [SFG⁺05] Mamadou Seck, Claudia Frydman, Norbert Giambiasi, Université Paul, Cézanne Aix Marseille, Tuncer I. Ören, and Levent Yilmaz. Use of a dynamic personality filter in discrete event simulation of human behavior under stress and fatigue. In *1st International Conference on Augmented Cognition*, 2005.

- [SGC04] Mankyu Sung, Michael Gleicher, and Stephen Chenney. Scalable behaviors for crowd simulation. *Comput. Graph. Forum*, 23(3) :519–528, 2004.
- [SJCO06] Barry G. Silverman, Michael Johns, Jason Cornwell, and Kevin O'Brien. Human behavior models for agents in simulators and games : Part i : Enabling science with pmfserv. *Presence*, 15(2) :139–162, 2006.
- [SKG05] Mankyu Sung, Lucas Kovar, and Michael Gleicher. Fast and accurate goal-directed motion synthesis for crowds. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '05, pages 291–300, New York, NY, USA, 2005. ACM.
- [SKT01] Etienne De Sevin, Marcelo Kallmann, and Daniel Thalmann. Towards real time virtual human life simulations. In *Computer Graphics International Symposium*, pages 31–37. IEEE Computer Society, 2001.
- [SMGT99] Elsa Schweiss, Soraia Raupp Musse, Fabien Garat, and Daniel Thalmann. An architecture to guide crowds using a rule-based behavior system. In *Proceedings of Autonomous Agents' 99, Whashington*, pages 334–335. ACM Press, 1999.
- [SSN08] Hakim Soussi, Joël Savelli, and Marc Neveu. Vers une architecture générique pour la modélisation des comportements dans les grandes foules. In *Actes des 21èmes Journées de l'AFIG'08*, pages 133 – 141, Toulouse, FRANCE, November 19-22 2008. IRIT Presse.
- [SSN09] Hakim Soussi, Joel Savelli, and Marc Neveu. A platform for the behavioral animation of crowds. In *International summer computer simulation conference SCSC'2009*, 2009.
- [ST05] W. Shao and D. Terzopoulos. Autonomous pedestrians. In *Proceedings of ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 19–28, 2005.
- [TBS04] C. Thureau, C. Bauckhage, and G. Sagerer. Learning human-like movement behavior for computer games. In *Proc. Int. Conf. on the Simulation of Adaptive Behavior*, pages 315–323. MIT Press, 2004.
- [TCP06] Adrien Treuille, Seth Cooper, and Zoran Popovic'. Continuum crowds. *ACM Trans. Graph*, 25 :1160–1168, 2006.

-
- [TLCC01] Franco Tecchia, Céline Loscos, Ruth Conroy, and Yiorgos Chrysanthou. Agent behaviour simulator (abs) : A platform for urban behaviour development. In *Proceedings of ACM/EG Games Technology Conference (GTEC)*, pages 17–21, 2001.
- [Tog55] K. Togawa. Study on fire escapes basing on the observation of multitude currents. Report of the building research institute, Ministry of Construction, Japan, 1955.
- [TTK88] K. Takahashi, T. Tanaka, and S. Kose. An evacuation model for use in fire safety design of buildings. In *Fire Safety Science*, pages 551–560, Hemisphere Publishing Company, 1988.
- [UCT04] Branislav Ulicny, Pablo de Heras Ciechowski, and Daniel Thalmann. Crowdbush : interactive authoring of real-time crowd scenes. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '04, pages 243–252, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.
- [Vas04] G. Vassalos. Research project 490 phase 1 : the effects of ship motion on the evacuation process, critical review of data available as input to evacuation simulation tools. *Universities of Glasgow and Strathclyde*, 2004.
- [Wig96] J.S Wiggins. *The Five-Factor Model of Personality : Theoretical Perspectives*. The Guilford Press, New York, 1996.
- [Wil94] Stewart W Wilson. Zcs : A zeroth level classifier system. *Evolutionary Computation*, 2(1) :1–18, 1994.
- [Wil95] Stewart W. Wilson. Classifier fitness based on accuracy. *Evol. Comput.*, 3(2) :149–175, June 1995.
- [Wol83] S. Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics (1983)*, 55(3) :601–644, 1983.
- [WTML07] Kai Yip Wong, Mary-Ann Thyvetil, Andriana Machaira, and Céline Loscos. Density distribution and local behaviour for realistic crowd simulation. *IADIS International Computer Graphics and Visualization*, 2007.