

Mémoire de Thèse

Visualisation et Interactions avec une colonie d'abeilles virtuelle :

Simulation, pédagogie et complexité.

Thomas Alves

Dirigé par Thierry Duval et Vincent Rodin
Encadré par Jérémy Rivière



Contents

1	Introduction	3
2	État de l'Art: Insectes Sociaux et Simulation Multi-Agents	4
2.1	Complexité et Colonie d'abeilles, Biologie et Système complexe	4
2.1.1	Complexité	4
2.1.2	Auto-Organisation de la Colonie	4
2.1.3	Pheromones et Physiologie	4
2.2	Modèles existants de répartition des tâches	4
2.2.1	Foraging For Work	5
2.2.2	Modèles à Seuils	5
2.2.3	BDI ?	5
3	Proposition : Prise de Décision et Interruption	6
3.1	Modélisation des tâches : Exécution du comportement	6
3.1.1	Actions, Activités et Tâches	6
3.1.2	Subsorption Hiérarchique et Exécution	6
3.2	Sélection : Modèle à Seuil	8
3.3	Interruption : Motivation	8
3.3.1	Etat de l'art rapide sur la Motivation	8
3.3.2	Action Démotivante et Tâche Motivée	8
3.4	Définir un Agent	9
3.5	Conclusion	9
3.5.1	Récapitulatif de notre proposition de modèle de prise de décision	9
3.5.2	Utilisation en Swarm Robotics	9
4	Modélisation et Implémentation pour une Colonie d'Abeilles	11
4.1	Description du modèle adapté	11
4.2	Description de l'implémentation	11
4.3	Calibration	11
4.3.1	Calibration rapide - Accélération	11
4.3.2	Calibration biologique - Données directes et indirectes	11
4.4	Evaluation Expe resultats et conclusion	12

5	Evaluation de l'Implémentation	13
5.1	Hypothèses et Expérimentations	13
5.2	Resultats	13
5.3	Interprétation et Perspectives d'Améliorations	13
6	Etat de l'art: Simulation Multi-Agents et Environnements Immersifs	14
6.1	SMA : Recréer et comprendre des systemes complexes existants par l'interaction	14
6.2	Manipuler et observer ces systèmes complexes	14
6.3	DataViz	14
6.4	Interaction / visu immersive pour comprendre	14
6.5	Interaction tangible	14
7	Proposition visu interaction	16
7.1	Interaction Immersive avec Manettes	16
7.2	Interaction Immersice ET tangible	16
7.3	Visualisation : Graph3D sur l'état interne de la colonie	16
7.4	Résultats/Évaluation Visualisation Interactive proposée . . .	16
8	Conclusions	17
8.1	Discussions	17
8.2	Perspectives	17
8.3	Conclusion	17

Chapter 1

Introduction

Orga Abeille - Visu Interagir avec - Simuler - Modeliser - (pk existant pas bon)proposition Modele motivation (orga++) - Eval Ccl (motivation mieux)
- Enfin intéragir et visu

- Colonie d'abeille en tant que système complexe. Beaucoup de recherche sur les butineuses mais moins sur l'intérieur de la ruche.
- Multi agent car concentration sur les individus et leurs interactions
- Comprendre l'auto organisation interne par la modélisation SMA vs Équation différentielles - de l'importance des contacts individuels.
- Transmettre et faciliter l'apprentissage avec l'environnement immersif.

Chapter 2

État de l'Art: Insectes Sociaux et Simulation Multi-Agents

2.1 Complexité et Colonie d'abeilles, Biologie et Système complexe

Synthèse des connaissances, notamment de notre visite à Avignon et ses 43 degrés à l'ombre.

2.1.1 Complexité

2.1.2 Auto-Organisation de la Colonie

Differentes tâches à effectuer sans control central. Perception locale

2.1.3 Pheromones et Physiologie

Mécanismes de l'auto organisation

Rôles et fonctions physiologiques associées.

Différents rôles de différentes phéromones.

Modélisation estimée hypothèse :

Cas classique de la vie d'une abeille.

Quels changements provoquent quelles réactions dans le métabolisme de l'abeille, et donc dans la répartition du travail.

2.2 Modèles existants de répartition des tâches

Etat de l'art de l'article PAAMS

2.2.1 Foraging For Work

2.2.2 Modèles à Seuils

2.2.3 BDI ?

Conclusion

Chapter 3

Proposition : Prise de Décision et Interruption

Description du modèle d'interruption PAAMS

3.1 Modélisation des tâches : Exécution du comportement

3.1.1 Actions, Activités et Tâches

Afin de modéliser nos tâches, nous allons utiliser 3 concepts, actions, activités et tâches. Une action est définie comme une interaction avec l'environnement extérieur, d'une durée déterminée et courte (pas plus de quelques pas de temps). Elle n'est donc pas forcément élémentaire, mais doit s'en approcher. Chaque action possède une condition d'activation.

Ensuite, une activité est une ensemble d'actions et/ou d'autres activités. Une activité possède aussi sa propre condition d'activation. Indirectement, tout ce qu'elle contient partage alors sa condition d'activation, ce qui nous permet de factoriser cette condition et d'alléger notre écriture, tout en permettant des comportements complexes.

Pour finir, une tâche est l'ensemble des activités et actions qui concernent un comportement. On peut donc voir une tâche comme l'activité racine, un peu à la manière d'un système de fichier: les activités sont des dossiers et contiennent d'autre dossier, ou des fichiers que sont les actions.

3.1.2 Subsumption Hiérarchique et Exécution

Une architecture de subsumption permet de hiérarchiser différents comportement entre eux, afin d'obtenir un comportement complexe. Dans un ordre défini, la subsumption interroge tour à tour la condition d'activation de ses différents blocs comportement, et exécute le premier dont la condition est

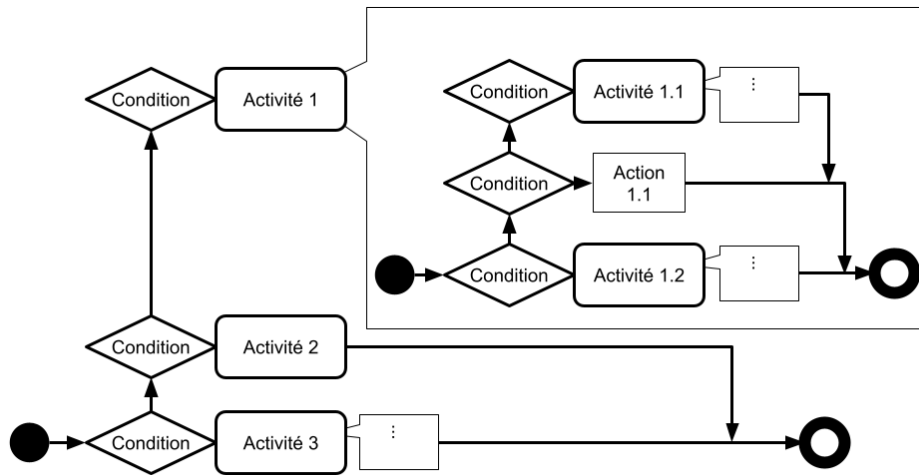


Figure 3.1: Modélisation d'une tâche.

valide. Par exemple, modéliser le comportement d'un mouton peut se faire en deux blocs. Un premier bloc "Brouter", toujours valide. Au dessus de celui ci, donc avec une priorité plus importante, un autre bloc "Fuir", qui s'active dès que le mouton a perçu un prédateur dans un temps déterminé. Ainsi, tant qu'aucun grand méchant loup n'est en vue, le mouton va brouter paisiblement. Dès qu'il en verra un, alors il pourra fuir.

Afin de respecter l'aspect quasi-élémentaire des actions, l'action de "Fuir" sera réaliser une multitude de fois. Ainsi, une seule exécution de Fuir ne fera faire au mouton qu'un pas, il va y faire appel plusieurs fois avant de considérer avoir semé le loup.

Une subsomption hiérarchique ajoute à cette structure simple, le fait que chaque bloc comportement puisse être une autre architecture de subsomption. Cette légère modification apporte une grande modularité dans la conception de ces architectures, et permet de modéliser des comportements plus complexes sans la lourdeur des subsomption classiques.

Ce que j'ai donc appelé "bloc comportement" des subsomptions correspond à nos actions et activités. Les blocs qui contiennent une autre subsomption sont appelés activités, et ceux qui contiennent du comportement sont des actions. Ensuite, la subsomption en elle même est alors une tâche, vous trouverez Figure 3.1 une représentation graphique de subsomption hiérarchique contenant nos concepts définis plus tôt.

Ainsi, pour qu'un agent puisse exécuter une tâche, il interroge l'activité racine puis va récursivement interroger ses composants. Chaque activité ou action interrogée va ainsi vérifier sa condition d'activation. Une activité dont

l'activation est valide va alors continuer d'interroger ses composantes. On a donc une recherche en profondeur, qui s'arrête dès qu'une action interrogée voit sa condition d'activation validée. Cette action est alors remontée à l'agent, qui pourra alors l'exécuter pendant toute sa durée. Ensuite, une fois l'action terminée, tout ce processus recommence afin de pouvoir récupérer une nouvelle action à exécuter.

Pour reprendre l'exemple du mouton, nous pouvons complexifier son comportement, en transformant son action "Fuir", en une activité "Fuir" contenant deux actions. L'action prioritaire "Esquiver" a pour condition le fait de voir le loup droit devant, et consiste à fuir mais en tournant, afin d'éviter le loup. Ensuite, la deuxième action, "Pleine Puissance" est l'action par défaut, sans condition, qui consiste à courir tout droit tant que le loup n'a pas été vu depuis un certain temps.

La condition d'activation de l'activité "Fuir" définie plus tôt, le fait d'avoir vu un prédateur dans les dernières secondes/minutes, est alors nécessaire à l'activation des deux actions "Esquiver" et "Pleine Puissance" sans que nous ayons à les réécrire explicitement.

3.2 Sélection : Modèle à Seuil

Agent toujours à jours des stimuli internes / externes
Executer la bonne tâche pour la bonne durée

évaluation systématique
Un seuil pour chaque tâche
Comparer les scores
Quand sélectionner une tâche

3.3 Interruption : Motivation

3.3.1 Etat de l'art rapide sur la Motivation

Les deux types de motivation qu'on a pu croiser dans la littérature. La motivation comme stimulus source de l'action (Drogoul), et la motivation comme perception interne (Flow), guide de l'action. On utilise les deux pour deux cas différents.

3.3.2 Action Démotivante et Tâche Motivée

On se base sur l'idée du Flow. Nos agents vont essayer d'estimer leur apport à la société dans leur tâche actuelle. Ainsi, lorsqu'un agent verra qu'il n'arrive pas à réaliser sa tâche, il sera dans l'état de malaise décrit pas le

Flow et cherchera de plus en plus à changer de tâche. Nous cherchons alors à mesurer l'efficacité de chaque agent dans sa tâche, afin de pouvoir détecter lorsqu'il arrive à la réaliser, mais surtout lorsqu'il n'y arrive pas. Il suffit alors de déterminer dans le comportement de l'agent, quelles sont les actions réalisées lorsque celui-ci n'arrive pas à réussir sa tâche. Par exemple, un robot ayant pour tâche de récolter des ressources aura dans cet algorithme une séquence de déplacement aléatoire lorsqu'aucune ressource n'est en vue. Répéter en boucle ce déplacement aléatoire, cette action, est un signe que ce robot n'arrive pas à correctement réaliser sa tâche, il devrait donc essayer de faire autre chose.

Nous venons alors ajouter aux Actions définies plus tôt le fait de pouvoir être "Démotivante". Lors de l'exécution d'une action "Démotivante", un agent va baisser sa motivation interne d'un montant défini. Le but est d'augmenter les chances qu'il abandonne cette tâche au profit d'une autre, lors du processus de sélection que nous allons détailler plus tard. L'action de déplacement aléatoire du robot que nous venons de citer est un bon exemple d'Action démotivante. Nous pouvons alors augmenter notre concept de tâche, en disant qu'une tâche est "motivée" lorsqu'elle contient au moins une action démotivante.

3.4 Définir un Agent

Définition d'un agent

3.5 Conclusion

3.5.1 Récapitulatif de notre proposition de modèle de prise de décision

Récapitulatif, exemple et conclusion sur le modèle en entier. L'exemple avancé dans l'article PAAMS pourrait être poussé un peu grâce au travail du stagiaire et à des nouveaux trucs que j'ai compris pendant la paramétrisation complexe du bousin.

3.5.2 Utilisation en Swarm Robotics

Les systèmes multi-agents sont souvent utilisés dans la mise en place d'essaims de robots, "Swarm Robotics". Ces essaims consistent en une grande quantité (dizaines) de robots simples, amenés à exécuter des tâches complexes. A l'image de la capacité de fourragement des fourmis est souvent utilisée comme cas d'application, nous avons donc construit notre exemple dans ce contexte. Un ensemble de robots va devoir ramener des ressources à leur base commune. Les ressources sont éparpillées dans l'environnement et doivent être traitées par un robot avant d'être déplacées jusqu'à la base. En plus de

cette activité de collecte, les robots vont devoir assurer une surveillance de la base, en patrouillant autour.

Conclusion

Chapter 4

Modélisation et Implémentation pour une Colonie d'Abeilles

4.1 Description du modèle adapté

Application du modèle théorique à notre simulation de colonie d'abeilles virtuelle. Simplification gigantesque de la biologie de l'abeille, ainsi que des différentes tâches au sein de la colonie. On veut de la répartition des tâches dans le plus simple des contextes.

4.2 Description de l'implémentation

L'implémentation du modèle de prise de décision décrits dans la partie d'avant, et modèle biologique de l'abeille adulte avec le système à seuil.

4.3 Calibration

4.3.1 Calibration rapide - Accélération

Dans sa version PAAMS avec la biologie accélérée Obtenir des résultats dans trop de temps de simulation. Le but était de vérifier la répartition des tâches à l'aide du modèle, pas encore de chercher une validité biologique.

4.3.2 Calibration biologique - Données directes et indirectes

En ajustant les paramètres connus grâce à la biologie, comme les durées de vie ou la durée et fréquence des nourrissements par exemple. Paramétrisation des paramètres indirects, ou émergents, comme par exemple le fameux "UNE

nourrice s'occupe de DEUX larves", qui n'est codé nul part mais qu'on doit retrouver.

4.4 Evaluation Expe resultats et conclusion

Ce qui marche, ce que ça montre et ce qui ne marche pas, et ce que ça peut vouloir dire.

Conclusion

Chapter 5

Evaluation de l'Implémentation

5.1 Hypothèses et Expérimentations

5.2 Resultats

5.3 Interprétation et Perspectives d'Améliorations

Conclusion

Chapter 6

Etat de l'art: Simulation Multi-Agents et Environnements Immersifs

6.1 SMA : Recréer et comprendre des systemes complexes existants par l'interaction

Comprendre les mécanismes, créer un modèle puis évaluer l'impact des différents paramètres sur l'évolution du comportement du système.

6.2 Manipuler et observer ces systèmes complexes

Systèmes en général non immersif, on s'arrête à la RA, et [de ce que j'ai vu], sans utilisation d'interacteurs tangibles. Comment mieux comprendre un système complexe : Environnement immersif et interacteurs tangibles. Ruche, cadre et connaissances apicoles.

6.3 DataViz

Comment visualiser de grande quantité de données, les données micro.

6.4 Interaction / visu immersive pour comprendre

Environnement immersif

6.5 Interaction tangible

Interacteurs tangibles

Conclusion

Chapter 7

Proposition visu interaction

Notre proposition

7.1 Interaction Immersive avec Manettes

Manipulation des cadres avec les manettes

7.2 Interaction Immersice ET tangible

L'apport et les contraintes du tangible (qu'il faut encore définir) pour la manipulation des cadres

7.3 Visualisation : Graph3D sur l'état interne de la colonie

Le graph3D et les information qu'il apporte

7.4 Résultats/Évaluation Visualisation Interactive proposée

Conclusion

Chapter 8

Conclusions

Comment l'ensemble se comporte et avis critique sur la totalité. Notamment le modèle multi agent simplifié qui apporte une quantité gigantesque de biais.

8.1 Discussions

8.2 Perspectives

Tout est possible, pousser le modèle de l'abeille de la simulation, pousser le domaine tangible avec peut être un cadre manette (un cadre avec un ou deux boutons?)

8.3 Conclusion

C'était cool

Rideau

clapclapclapclapclapclapclapclapclapclapclap