

RAPPORT DE MASTER

présenté par

Robin Lamarche-Perrin

Master 2 Recherche *sp.* Informatique – AIW *option*
Université Joseph Fourier – UFR IMA Grenoble
Institut Polytechnique de Grenoble – Ensimag

Équipes MESCAL & MAGMA
Laboratoire d'Informatique de Grenoble

Observation macroscopique pour l'analyse de systèmes multi-agents à grande échelle

Soutenu le 23 juin 2010
devant les membres du jury

<i>Encadrants</i>	Jean-Marc Vincent Yves Demazeau
-------------------	------------------------------------

<i>Membres permanents</i>	Marie-Christine Rousset Gilles Sérasset Christine Verdier
---------------------------	---

<i>Expert extérieur</i>	Philippe Jorrand
-------------------------	------------------

<i>Responsables de l'option</i>	Jérôme Euzenat Éric Gaussier
---------------------------------	---------------------------------

Remerciements

Je tiens à remercier mes deux encadrants, Jean-Marc et Yves, pour avoir été mes guides tout au long de cette expédition au cœur des systèmes multi-agents; Laurent Lacomme pour avoir assisté à (presque) toutes nos avancées et pour avoir donné de bons conseils aux bons moments; Valérie Camps qui, même si ça n'a pas été jusqu'au bout, a tout de même accepté de m'accompagner. Je remercie également Lucas Mello Schnorr et Hussein Joumaa pour avoir préparé le terrain de cette exploration.

Je remercie bien évidemment Régis pour son amitié éloquente en cette année de labeur, Max Kistler pour l'encadrement qu'il veut bien m'accorder en philosophie des sciences et Denis Perrin pour chacun de ses cours.

Table des matières

Remerciements	3
Table des matières	5
1 Introduction	7
1.1 Problématique	7
1.1.1 Systèmes multi-agents	7
1.1.2 Distribution des problèmes et émergence	8
1.1.3 Comportement des agents	8
1.1.4 Dynamique et structure causale	8
1.1.5 Analyse et grande échelle	9
1.1.6 Interdisciplinarité	9
1.2 Contenu du rapport	10
2 Analyse du problème	11
2.1 État de l'art	11
2.1.1 Validation et preuve	11
2.1.2 Analyse de traces	12
2.1.3 Simulation	13
2.1.4 Analyse des données	14
2.2 Positionnement	14
2.2.1 Les trois étapes de l'analyse	14
2.2.2 Les deux paradigmes de l'observation	16
2.2.3 Un changement de paradigme	17
2.2.4 Justifications	18
3 Modélisation	19
3.1 Modèle de système distribué	19
3.1.1 Origines du modèle	19
3.1.2 Pourquoi utiliser ce modèle?	20
3.1.3 Système distribué	21
3.1.4 Processus et temps local	21
3.1.5 Messages et temps global	23
3.2 Méta-modèle de système multi-agents	26
3.2.1 A comme Agents	27
3.2.2 E comme Environnements	28
3.2.3 I comme Interactions	28
3.2.4 O comme Organisation	29

3.3	Modèle d'observation macroscopique	29
3.3.1	Modèle générique	30
3.3.2	Observation spatiale	30
3.3.3	Observation temporelle	34
3.4	Modèle de sonde macroscopique	37
3.4.1	Sonde spatiale	37
3.4.2	Sonde temporelle	38
3.4.3	Composition de sondes	38
4	Expérimentations et évaluation	39
4.1	Présentation	39
4.1.1	Protocole d'évaluation	39
4.1.2	Critères d'évaluation	39
4.2	Plate-forme d'expérimentation	40
4.2.1	Contribution	40
4.2.2	<i>MASON</i>	41
4.3	Applications	41
4.3.1	Grille informatique	41
4.3.2	Colonie de fourmis	45
5	Bilan et perspectives	49
5.1	Bilan	49
5.2	Perspectives	50
5.2.1	A propos du modèle	50
5.2.2	A propos des sondes	51
5.2.3	A propos de la visualisation	52
	Bibliographie	55
A	Modèle standard de temps	59
B	Sujet de stage 2A ingénieur	61

Chapitre 1

Introduction

1.1 Problématique

Ce projet s’inscrit dans le cadre d’une collaboration entre deux équipes du Laboratoire d’Informatique de Grenoble. L’équipe MAGMA s’intéresse en toute généralité au domaine des systèmes multi-agents et l’équipe MESCAL (pour *Middleware Efficiently SCALable*) à l’exploitation d’architectures distribuées à grande échelle. Cette collaboration a commencé en 2007 par le co-encadrement de la thèse d’Hussein Joumaa qui a conduit au développement de *MAS-Pajé* [Joumaa et al., 2009]. Cet environnement de visualisation pour l’analyse de l’exécution de SMA de grande taille est présenté dans la section 2.1.2.

Ce projet s’intéresse à l’*analyse des systèmes multi-agents à grande échelle*. Cette introduction a pour but d’introduire et de préciser les termes utilisés dans ce rapport et d’en énoncer la problématique spécifique.

1.1.1 Systèmes multi-agents

Un *agent* est « une entité autonome, réelle ou abstraite, qui est capable d’agir sur elle-même et sur son environnement, qui, dans un univers multi-agent, peut communiquer avec d’autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents. » [Ferber, 1995] Un *système multi-agents* (SMA) est alors « un système distribué composé d’un ensemble d’agents. » [Chaib-Draa et al., 2001]

La littérature concernant les SMA est large et variée. Pour ne pas grossir l’introduction de ce rapport, les différentes typologies de systèmes ne sont pas présentées ici. Le lecteur pourra se référer à [Wooldridge, 2009] et [Weiss, 1999] pour une plus longue introduction au domaine des SMA.

Les SMA sont utilisés dans de nombreux domaines scientifiques pour l’analyse, la conception et l’implantation de systèmes informatiques complexes. Contrairement aux systèmes développés par les approches classiques de l’intelligence artificielle, les SMA bénéficient d’une résolution distribuée et concurrente des problèmes. On dit que la solution *émerge* de la coopération entre les agents, de leur coordination et de leurs négociations.

L’implémentation des SMA est réalisée au niveau des agents et de leurs comportements. Dans la mesure où les techniques de preuve à l’œuvre dans le génie logiciel ne sont pas adéquates pour décrire *a priori* les phénomènes émergents (*cf.* section 2.1.1), le concepteur doit être capable d’ajuster les modèles d’agents et leurs paramètres pour engendrer la solution souhaitée. Cette

technique de conception/développement en boucle de type VOYELLE (*cf.* [Demazeau, 2001]) nécessite de détecter les phénomènes émergents et d'en comprendre les causes. L'objectif *à long terme* de ce projet est de répondre à ce besoin (*cf.* perspectives méthodologiques de la section 5.2.3).

1.1.2 Distribution des problèmes et émergence

« Un SMA est généralement caractérisé ainsi :

- chaque agent a des informations ou des capacités de résolution de problèmes limitées (ainsi, chaque agent a un point de vue partiel),
- il n'y a aucun contrôle global du système multi-agent,
- les données sont décentralisées,
- le calcul est asynchrone. » [Chaib-Draa et al., 2001]

Les SMA permettent ainsi de résoudre des problèmes naturellement distribués. Cette résolution par émergence induit une notion de *spatialité* : l'espace d'un SMA est l'ensemble de ses agents. Dans le cas d'un système informatique concret, cet espace est supporté par l'environnement d'exécution des agents.

Au *niveau local* se situent les comportements des agents, les informations partielles dont ils disposent, les interactions, les processus de coordination, de synchronisation, *etc.* Au *niveau global* se situe le comportement du système, pris comme un tout. C'est à ce niveau qu'est effectuée la résolution des problèmes. Elle émerge de l'activité distribuée des agents.

C'est l'étude des phénomènes émergents, et non celle des agents, qui constitue l'objet de ce projet.

1.1.3 Comportement des agents

Afin d'étendre la méthode d'analyse proposée par ce projet à l'ensemble des SMA, aucune supposition n'est faite quant à l'architecture réelle des agents. Cette architecture, constituant la *description interne* de l'agent (*cf.* [Demazeau and Müller, 1991]), est inconnue *a priori*. **La connaissance des agents est donc limitée à leur description externe** (*i.e.* leur comportement observable).

Cette approche par *boîte grise* [Viroli and Omicini, 2002] est fondée sur le partitionnement des caractéristiques internes de l'agent en deux parties :

- La *partie blanche*, entièrement modélisées, qui est responsable des interactions avec l'environnement et les autres agents.
- La *partie noire*, inconnue, qui représente le processus de délibération interne de l'agent et qui affecte la *partie blanche*.

1.1.4 Dynamique et structure causale

Les SMA auxquels nous nous intéressons sont des systèmes ouverts susceptibles d'évoluer rapidement. Leur taille et leur structure évoluent et s'adaptent en fonction du problème auquel ils sont globalement confrontés. La dynamique des SMA induit une notion de *temporalité*. Celle-ci est partiellement supportée par le temps de l'exécution des agents.

Le fonctionnement des SMA repose donc sur deux dimensions : l'*espace* et le *temps*. La structure de leur exécution met en jeu des *relations de causalités dimensionnelles*.

- La *causalité spatiale* met en relation le comportement d'un agent avec le comportement d'un autre agent.

- La *causalité temporelle* met en relation un comportement apparaissant à un instant donné avec un second comportement qui apparaît plus tard.

Ces deux types de causalité dépendent l’une de l’autre et co-définissent la structure de l’exécution des SMA. Cette structure est le support de l’émergence de phénomènes globaux et de leur évolution au cours du temps.

Ce projet s’intéresse plus particulièrement à l’analyse de la structure causale de l’exécution pour étudier la dynamique des phénomènes émergents.

1.1.5 Analyse et grande échelle

L’analyse, au sens de « décomposition », est une « opération intellectuelle consistant à décomposer un [objet d’étude] en ses éléments essentiels, afin d’en saisir les rapports et de donner un schéma de l’ensemble » [Robert et al., 2002]. Analyser l’exécution d’un SMA, c’est donner les causes essentielles des comportements émergents, c’est schématiser les structures causales qui conduisent à l’émergence, c’est enfin les exposer de manière simple et intelligible. Il existe plusieurs techniques pour réaliser une telle analyse. Elles sont présentées dans la section 2.1.

L’échelle d’un SMA peut être envisagée selon les deux dimensions de sa dynamique : l’*espace* et le *temps*. Ce projet propose de faire l’analyse de SMA à grande échelle. Il s’intéresse donc à l’exécution sur de longues périodes de temps de SMA de très grande taille (les équipes MESCAL et MAGMA visent à long terme le million d’agents).

Pour satisfaire ce point important, les techniques d’analyse proposées doivent supporter le *passage à l’échelle*. Cela sans que leur coût (en terme d’implantation ou de temps de calcul) n’explose. **Ce projet a pour objectif le passage à l’échelle de l’analyse causale des phénomènes émergents.**

1.1.6 Interdisciplinarité

Du fait de leur ouverture conceptuelle, les SMA sont à l’intersection de nombreuses disciplines scientifiques. Outre les disciplines informatiques (intelligence artificielle, vie artificielle, informatique répartie et génie logiciel), ils s’inspirent de sciences connexes comme la sociologie, l’économie, la psychologie sociale et les sciences cognitives [Chaib-Draa et al., 2001]. Le principe même d’émergence est issu de concepts longuement étudiés en philosophie et, plus tard, dans les disciplines citées précédemment. On pourra trouver une description de cette ouverture disciplinaire dans [M.R. Jean et al., 1997] et [O’Connor and Wong, 2006].

Ce projet de Master 2 Recherche est effectué en parallèle à un projet de Master 1 de philosophie intitulé « Philosophie, Langages et Cognition ». Le projet de mémoire préparé dans le cadre de ce Master porte sur une étude comparative du principe d’émergence tel qu’il est conjointement envisagé par les sciences cognitives et par l’intelligence artificielle. Ce projet, encadré par Max Kistler (directeur du département philosophie de l’Université Pierre Mendès France à Grenoble), sera soutenu en septembre 2010.

1.2 Contenu du rapport

Le chapitre 2 fait l'état de l'art des différentes approches pour l'analyse de l'émergence dans les SMA. Il discute de leur efficacité pour résoudre le problème présenté dans cette partie. Il présente et défend également l'approche spécifique proposée par ce projet. Celle-ci est fondée sur la notion d'*observation macroscopique*.

Le chapitre 3 présente un méta-modèle de SMA généralisé à partir d'un modèle de système distribué. Il présente sur cette base un modèle d'observation macroscopique appliqué aux SMA et donne des exemples de sondes effectuant ce genre d'observation.

Le chapitre 4 présente les expérimentations que nous avons réalisées pour évaluer l'approche défendue dans le chapitre 2. Elles consistent à simuler des observations macroscopiques de SMA et à montrer leur efficacité pour l'analyse de leur exécution et la détection de phénomènes émergents.

Le chapitre 5 fait un bilan du projet et ouvre des perspectives quant à la conception de sondes macroscopiques pour l'analyse des SMA.

Chapitre 2

Analyse du problème

Ce chapitre est consacré au positionnement du projet vis-à-vis des travaux effectués dans le domaine. La section 2.1 présente de manière synthétique quelques-uns de ces travaux et discute de leurs limitations quant à la résolution du problème posé. La section 2.2 présente et justifie l'approche proposée par ce projet pour dépasser ces limites.

2.1 État de l'art

Cette section présente les différents outils disponibles actuellement pour l'analyse des systèmes multi-agents (SMA). Leur efficacité pour résoudre le problème formulé dans la partie précédente est discutée en fonction de la notion fondamentale d'*observation*. L'observation est un « procédé scientifique d'investigation », une « constatation attentive des phénomènes tels qu'ils se produisent, sans volonté de les modifier » [Robert et al., 2002]. L'observation se distingue ainsi de l'*expérimentation* qui agit sur le système afin d'en étudier la réaction. Elle est, dans la plupart des cas, nécessaire à l'analyse.

2.1.1 Validation et preuve

Présentation

Il est communément admis que les techniques de validation formelle (analyse et vérification intelligente d'un logiciel) et de preuve (correction logique d'un logiciel) à l'œuvre dans le domaine du génie logiciel ne sont pas adéquates pour l'analyse des applications multi-agents. Ceci en raison « des impossibilités de produire *manuellement* des tests ou des scénarios ayant une couverture suffisante, surtout dans des environnements rapidement changeants », dans le cas des techniques de validation, et « des impossibilités de spécifier *formellement* les comportements qui résultent d'interactions probabilistes, dans des environnements non prévus, entre des agents logiciels » [Drogoul et al., 2004].

Il est alors nécessaire de proposer des alternatives à ces méthodes d'analyse qui se fondent sur une connaissance exacte des caractéristiques de l'application. D'autant plus que, dans le cadre du problème exposé, ces méthodes sont incompatibles avec l'approche par *boîte grise* selon laquelle la *description interne* des agents est inconnue (*cf.* section 1.1.3).

Discussion

Les méthodes formelles d'analyse sont des méthodes *a priori* en ce sens qu'elles sont indépendantes de l'observation des SMA en cours d'exécution. Puisqu'une telle analyse logique est

contraire aux principes d'imprévisibilité, d'évolution et d'adaptation des SMA, il est nécessaire de recourir à des méthodes d'analyse empiriques. De telles méthodes *a posteriori* (*i.e.* qui dépendent de l'observation des systèmes) sont présentées dans les sections suivantes.

2.1.2 Analyse de traces

Présentation

Les techniques de *monitoring* permettent la surveillance et la mesure de l'activité des systèmes informatiques. Elles utilisent des sondes pour observer les nœuds du système et pour analyser leur activité au cours de l'exécution (*on-line monitoring*). Les contraintes d'interactivité exigées par ces techniques, pour implémenter un contrôle efficace du système observé, limitent la quantité d'analyse que l'on peut effectuer « à la volée ». Dans le cas de systèmes de très grande taille, ces contraintes sont relâchées par la mise en place d'une analyse non-interactive des données du système, collectées sous la forme de traces d'exécution (*post-mortem monitoring*).

Une trace (information enregistrée lors de l'exécution) permet de connaître *a posteriori* les caractéristiques locales du système lors de son enregistrement. Certains outils, développés pour contrôler les performances d'applications parallèles, sont fondées sur l'analyse de telles traces. *ParaGraph* [Heath and Etheridge, 1991], *Paradyn* [Millerand et al., 1995], *Pablo* [Reed et al., 1993] et *Pajé* [Stein, 1999] sont des exemples d'environnements de visualisation utilisant cette méthode. Ils implémentent un *monitoring* de l'exécution des processeurs au sein de *clusters* de grande taille.

L'adaptation de ces techniques de *post-mortem monitoring* pour l'analyse des SMA est soumis aux problèmes suivants [Joumaa et al., 2009] :

- La possibilité de représenter l'exécution d'un système de grande taille sur de longues périodes de temps.
- La possibilité d'ajouter de nouvelles fonctionnalités ou d'étendre les préexistantes sans avoir à modifier le reste des outils.
- Le problème de l'affichage des entités qui peuvent être créées ou détruites dynamiquement lors de l'exécution du système (ouverture des SMA).

MAS-Pajé [Joumaa et al., 2009] tente de résoudre ces problèmes. Il s'agit d'un environnement de visualisation développé dans le cadre de la collaboration entre l'équipe MESCAL et l'équipe MAGMA du Laboratoire d'Informatique de Grenoble. Cet outil utilise les traces déposées par les agents lors de l'exécution du système pour représenter leur activité. Il adapte l'interface interactive de visualisation offerte par *Pajé* pour présenter cette activité sous la forme de diagrammes espace-temps. *Pajé* est particulièrement intéressant ici puisqu'il permet le passage à l'échelle de la visualisation en s'adaptant à la quantité des données lorsque le nombre de nœuds augmente [Stein, 1999].

Discussion

L'observation des SMA, dans le cadre des approches telles que proposées par *MAS-Pajé*, est implémentée par l'étape de traçage. Elle rencontre cependant de lourdes difficultés techniques. Deux des difficultés génériques rencontrées lors du traçage des applications parallèles sont (difficulté 1) l'absence d'horloge globale dans les systèmes distribués, ce qui rend difficile l'ordonnancement des événements tracés sur des nœuds différents, et (difficulté 2) l'intrusion (ou « effet de sonde ») qui modifie le comportement des applications tracées [Chassin de Kergommeaux et al., 2001]. Ces difficultés augmentent avec la taille des applications et surtout avec la quantité des données nécessaires à l'analyse *post-mortem*. Plusieurs techniques sont proposées pour réduire la quantité des données tracées : *Pablo* ajuste dynamiquement la

fréquence d'enregistrement des traces, *Paradyn* adapte le traçage des données en fonction de problèmes de performance détectés *on-line*, d'autres techniques statistiques de *clustering* dynamique limitent le traçage aux processeurs représentatifs de chaque *cluster* [Nickolayev et al., 1997].

La décentralisation des systèmes et leur asynchronisme (*cf.* section 1.1.2) se traduisent par une difficulté à reconstruire l'état global du système à partir des données collectées (difficulté 1). Cela résulte du fait qu'il est impossible de tracer « globalement » le système, puisque les données ne sont pas centralisées, et que les observations locales des agents (par le biais des traces qu'ils déposent) ne sont pas synchronisées.

L'enregistrement des données modifie le comportement des systèmes tracés (difficulté 2). Plus les données collectées sont nombreuses, plus il y a de chances pour que le processus de collection perturbe l'exécution du système. La réduction de l'intrusion impose donc une limite sur la quantité des données à enregistrer. C'est pourquoi cette approche semble difficile à utiliser pour l'analyse des SMA de très grande taille (*cf.* section 1.1.5).

De plus, les lieux et dates d'apparition des phénomènes émergents sont imprévisibles *a priori*. Il est souvent difficile de les détecter dynamiquement afin d'adapter le traçage pour l'analyse *post-mortem* de ces phénomènes (méthodes proposées par *Pablo* et *Paradyn*).

2.1.3 Simulation

Présentation

Il existe de très nombreuses plates-formes destinées à la simulation des SMA. Les plus connues sont *Swarm* [Minar et al., 1996] et *RePast* [North et al., 2007], mais beaucoup d'autres sont utilisées en fonction de leur différentes domaines de spécialisation.

L'objectif de telles plates-formes est la centralisation et la synchronisation de SMA *réels* sous la forme de SMA *modélisés* et *simulés*. La plate-forme permet de contrôler entièrement l'exécution du système simulé. Les contraintes de décentralisation et d'asynchronisme (*cf.* section 1.1.2) sont relâchées grâce à l'utilisation d'un noyau d'exécution central qui ordonnance celles des agents. Toutes les ressources du systèmes sont organisées par ce noyau. Tous les événements peuvent être datés par son horloge.

Le contrôle du temps de l'exécution présente un grand avantage. Celle-ci peut être stoppée à tout moment, ce qui permet de réaliser autant d'analyses que nécessaire à la compréhension de l'état dans lequel le système est figé. De plus, cet état est accessible globalement. Il n'a pas besoin d'être construit à partir d'informations partielles décentralisées dans la mesure où l'ensemble est synchronisé par le noyau central (résolution de la difficulté 1).

Par ailleurs, l'observation des agents (si elle est correctement implémentée) ne modifie par l'exécution du système puisque qu'elle est figée. Les problèmes liés à l'intrusion des sondes d'observation sont ainsi écartés (résolution de la difficulté 2).

Discussion

La simulation permet donc de résoudre les problèmes de décentralisation et d'asynchronisme rencontrés par les méthodes d'analyse de traces. Elle autorise une analyse illimitée de l'état du SMA simulé à un temps donné.

Cependant, deux problèmes majeurs sont à opposer à l'utilisation de telles plates-formes dans le cadre de ce projet.

1. L'architecture des agents, *modélisée* et *simulée*, peut ne pas être conforme à leur *description interne*, c'est-à-dire leur architecture *réelle* (*cf.* section 1.1.3). Le modèle exécutable ainsi

utilisé pour la simulation, n'est pas nécessairement conforme au modèle conceptuel du SMA réel. [Bonneaud et al., 2009] parlent de *biais computationnels*. Il est donc difficile de simuler efficacement un SMA lorsqu'on ne connaît pas l'architecture de ses agents.

L'analyse de l'émergence est alors faussée par l'étape de modélisation. En effet, les phénomènes émergents d'un SMA simulé de manière approximative peuvent diverger fondamentalement de ceux qui émergent dans les SMA réels.

2. Les ressources nécessaires pour la simulation d'un SMA sont obligatoirement supérieures à celles nécessaires pour l'exécution du SMA lui-même. Dans le cas de SMA de très grande taille, il est parfois impossible de restituer entièrement l'activité du système sur un environnement d'exécution limité. Il est possible de simplifier le modèle utilisé pour la simulation, mais ceci accroît alors l'importance du problème précédent.

2.1.4 Analyse des données

Cette section présente rapidement quelques méthodes utilisées pour l'analyse des données collectées par le traçage ou par l'utilisation d'une plate-forme de simulation.

Analyse statistique

De nombreux opérateurs statistiques permettent de synthétiser les données collectées. Ils donnent une information globale sur l'état du système en intégrant des informations locales. *MAS-Pajé*, *RePast* et *Swarm* proposent ainsi de nombreuses opérations pour effectuer une telle analyse statistique. Celle-ci est couplée avec des outils de visualisation (histogrammes, diagrammes, courbes, *etc.*) afin d'en présenter les résultats.

Analyse formelle

Le π -calcul est un langage de programmation théorique développé dans [Milner et al., 1992]. Il est une extension du *Calculus of Communicating Systems* (CCS) [Milner, 1982]. Il définit, à l'aide d'une grammaire formelle, une algèbre sur les processus communicants des systèmes distribués afin d'exprimer la dynamique des structures de ces processus. Cet outil d'abstraction peut ainsi être utilisé pour faire l'analyse de la structure des SMA. Par exemple, le langage *Picol* [Ferber and Gutknecht, 2000] est une adaptation du π -calcul au domaine des SMA, pour la formalisation du méta-modèle d'organisation AGR (Agent, Groupe et Rôle) présenté dans [Ferber and Gutknecht, 1998].

Cependant, en tant que langage de programmation théorique, le π -calcul vise principalement l'analyse des concepts généraux de programmation distribuée. Il n'a pas été conçu pour l'analyse particulière de leur exécution. C'est pourquoi aucune des plates-formes présentées précédemment ne les utilise.

Le π -calcul a également inspiré des langages de programmation concrets (*e.g.* *JoCaml* et *Acute*). Ceux-ci sont destinés à la *conception* de programme distribué exécutables et non à leur *analyse*.

2.2 Positionnement

2.2.1 Les trois étapes de l'analyse

Si on écarte les techniques d'analyse *a priori*, les approches exposées dans la section précédente décomposent l'analyse des systèmes en trois étapes [Joumaa et al., 2009] :

1. collection des données (étape d'observation)

2. analyse des données collectées (étape d'analyse des données)
3. présentation des informations révélées par l'analyse des données (étape de visualisation)

Ces trois étapes sont synthétisées dans la diagramme présenté par la figure 2.1. Le rôle de chacune d'entre-elles est explicité dans ce qui suit.

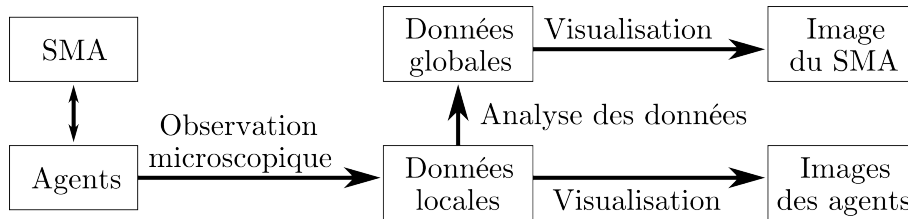


FIGURE 2.1 – Analyse des SMA et observation microscopique

Étape d'observation

Dans le cadre de l'analyse empirique des SMA (analyse *a posteriori*), l'étape d'observation préliminaire est incontournable.

Le sondage est l'« exploration locale et méthodique d'un milieu à l'aide d'une sonde ou de procédés techniques particuliers. » [Robert et al., 2002] Il s'agit d'une méthode d'observation. Les outils précédents utilisent des sondes (sondes à base de traces ou sondes simulées) pour connaître les états et comportement locaux du système (état et comportement des agents). En ce sens, ils effectuent une série d'observations microscopiques.

Les données recueillies par les observations microscopiques doivent être réorganisées pour être cohérentes. C'est le problème de l'état global cohérent lié à l'asynchronisme des SMA observés (difficulté 1). A ce stade, on est capable de fournir des images locales et coordonnées de l'exécution des agents du système.

Étape d'analyse des données

Cette étape a pour but de faire apparaître les caractéristiques globales du système à partir des données collectées par l'observation microscopique. Elle révèle les phénomènes émergents, par intégration des données, agrégation des structures, synthèses formelles ou statistiques.

Étape de visualisation

La dernière étape consiste à présenter les résultats de l'analyse des données. Elle fournit une image globale de l'exécution du SMA, sur la base du travail effectué en amont. Elle n'est pas entièrement dissociée de l'étape d'analyse des données, dans la mesure où les méthodes de visualisation constituent souvent des outils pour une analyse *manuelle*.

Les outils existant pour implémenter les étapes 2 et 3 n'ont pas été discutés en profondeur dans la mesure où des problèmes majeurs résident dans l'implémentation de la première étape. Ce projet a pour objectif principal de résoudre les problèmes de l'observation avant de se confronter aux étapes suivantes.

2.2.2 Les deux paradigmes de l'observation

Il existe deux approches antagonistes pour l'analyse des systèmes complexes : l'*émergentisme* et le *réductionnisme*. Cette section présente deux exemples pour illustrer ces approches. Le premier exemple est celui du recensement, pour l'étude de groupes sociaux, et le second celui du thermomètre, pour la mesure des systèmes gazeux. Il est possible, par analogie, de préciser deux approches antagonistes en ce qui concerne l'analyse des SMA.

Les deux paradigmes présentés ci-dessous ont été développés au cours du XX^e siècle par les empiristes logiques. Ils sont notamment liés aux notions de *théories* et d'*observables* dont on trouvera des références dans [Bogen, 2009] et dans [Carnap and Gardner, 1966].

Émergentisme

Le recensement est une méthode d'investigation utilisée pour révéler et mesurer les caractéristiques globales de systèmes sociaux. Elle est fondée sur l'*observation microscopique* des systèmes, c'est-à-dire l'observation des caractéristiques de leurs individus. Dans un second temps, les opérations de mesure statistique permettent de faire émerger des caractéristiques globales du système, que l'on représente souvent sous la forme de diagrammes (pyramide des âges, proportion des secteurs d'activité, *etc.*). Le sondage est une variante statistique du recensement où les données ne sont pas collectées sur la totalité du système, mais par un processus d'échantillonnage.

Il s'agit de l'approche classique des techniques d'analyse présentées dans la section 2.1. Elle peut être qualifiée d'*approche émergentiste* (ou *approche bottom-up*) dans la mesure où les caractéristiques émergentes du système sont produites à partir de l'observation microscopique de ses individus. Le recensement est alors la composition de *sondes microscopiques*.

Réductionnisme

Le thermomètre est un appareil permettant de mesurer la température des systèmes gazeux. La température est une caractéristique du système au voisinage du thermomètre ou une caractéristique globale du système (si il y a équilibre thermodynamique). A ce titre, le thermomètre réalise une *observation macroscopique*.

La thermodynamique affirme que la température est une grandeur physique qui émerge de l'activité des individus du système, à savoir des particules qui composent le gaz. Il est ainsi possible de réduire la propriété macroscopique sur la base d'un modèle physique adéquat, c'est-à-dire de trouver une explication locale à la mesure globale. Il s'agit donc d'une *approche réductionniste* (ou *approche top-down*). Le thermomètre est une *sonde macroscopique* qui intègre l'énergie cinétique de plusieurs milliers de particules en une unique observation.

Remarques

Il existe des méthodes d'observation macroscopique des systèmes sociaux. Observer par exemple le comportement d'un gouvernement permet d'avoir une idée globale sur la nation dont il est la tête. Cependant, les observations fournies par ce genre de sondes sont fortement soumises à interprétation. Dans le cas de l'observation d'un gouvernement, il est possible que la relation entre la base microscopique et l'activité gouvernementale soit biaisée, et biaise ainsi la sonde (par exemple dans le cas d'un système totalitaire).

Il existe également des méthodes d'observation microscopique des systèmes gazeux. Mais une sonde capable d'observer au niveau de la particule coûte extrêmement cher. Il est donc difficile de multiplier ces sondes pour reconstruire une information globale sur le système, comme la température.

2.2.3 Un changement de paradigme

Les approches classiques de l'analyse des SMA (analyse de traces et simulation) sont des *approches émergentistes*. Comme dans le cas du recensement, elles implémentent une étape d'analyse des données afin de révéler les phénomènes émergents.

Orientation du projet

Ce projet propose de changer de paradigme et de poser les bases d'une *approche réductionniste*. Dans ce but, il encourage la conception et l'implémentation de sondes macroscopiques pour l'observation des SMA. De telles sondes permettent de collecter des données informant globalement sur l'état du système. L'étape d'analyse des données n'est alors plus nécessaire pour révéler les phénomènes émergents, comme dans le cas du thermomètre où la température (grandeur émergente) est *directement* observée par l'appareil.

La figure 2.2 résume les étapes de l'approche réductionniste. L'étape d'analyse des données est évitée par l'observation macroscopique. Par contre, les images locales des agents ne sont plus accessibles. Dans une perspective plus vaste, ouverte par ce projet, une analyse des données globales pourrait aboutir, par réduction, à une connaissance approximative des comportements locaux en présence dans le système (*cf.* section 5.2.1).

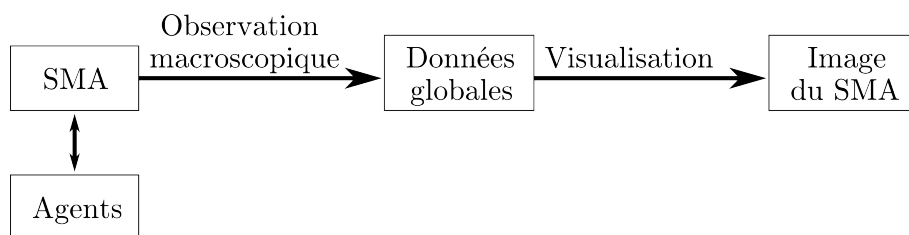


FIGURE 2.2 – Analyse des SMA et observation macroscopique

La systémique

De par sa démarche, ce projet s'inscrit dans les théories systémiques. Une bonne introduction à cette *Théorie générale des systèmes* [von Bertalanffy, 1968] est donnée par de Rosnay dans *Le Macroscopie* [de Rosnay, 1975] : « L'approche systémique, c'est la nouvelle approche que symbolise le macroscopie. Elle s'appuie sur une démarche globale des problèmes ou des systèmes que l'on étudie et se concentre sur le jeu des interactions entre leurs éléments. » La systémique est un complément de l'approche analytique (au sens cartésien de *résolution analytique*). Elle tente d'appréhender les systèmes complexes sans en isoler les parties pour les soumettre à l'analyse (approche émergentiste).

Le Macroscopie est, quant à lui, un appareil symbolique imaginé par de Rosnay pour répondre aux exigences de la systémique. « Le microscope permet d'observer l'infiniment petit, le télescope l'infiniment loin, le macroscopie l'infiniment complexe. » [de Rosnay, 1975] Il permet à l'homme d'observer les organisations complexes dont il est un élément (organisations sociales, économiques, écologiques, *etc.*).

Ce projet propose une piste pour la conception de *macroscopes* permettant l'observation des SMA.

Observatoire de l'Internet

D'un point de vue plus technique, il existe des travaux orientés vers l'étude topologique de l'Internet qui utilisent des techniques d'observation macroscopique. C'est le cas de plusieurs travaux du projet Européen DELIS (*Dynamically Evolving, Large-scale Information Systems*) qui font de l'analyse statistique de données globales, recueillies notamment à l'aide de sondes de type *traceroute* (cf. [Dall'Asta et al., 2005] et [Viger et al., 2005]). Une connaissance théorique de la topologie du réseau est ensuite engendrée par réduction des données globales. Ces travaux, parce qu'ils refusent de recourir à une observation microscopique de l'Internet, peuvent apporter un point de vue intéressant sur le sujet, dans un domaine différent de celui des SMA.

2.2.4 Justifications

Distribution

Les sondes macroscopiques collectent des données de manière *naturellement* distribuée. En effet, elles intègrent les interactions coordonnées de plusieurs agents dans une unique observation. Les problèmes liés à la synchronisation et à la décentralisation de l'information (difficulté 1) sont ainsi évités.

Cette idée est soutenue par les théories systémiques exposées dans la section précédente. Le fait d'aborder les systèmes sans en isoler les parties, en se concentrant sur leurs interactions (par opposition à l'observation microscopique), permet de prendre naturellement en compte les phénomènes de coordinations et de synchronisations qui sont à l'œuvre localement.

Émergence

Ce projet soutient le fait que l'émergence n'est pas une propriété ontologique du système (*i.e.* qui appartient à sa nature propre), mais une propriété épistémologique (*i.e.* qui appartient à l'image du système, sous le regard d'un observateur). Si il se focalise sur les détails de l'activité microscopique du système, il ne voit que la complexité des comportements locaux. « C'est le changement de regard de l'observateur qui lui permet de percevoir [l']émergence. » [M.R. Jean et al., 1997].

Ainsi, l'observation macroscopique est un moyen de détecter immédiatement les phénomènes émergents, sans passer par l'étape d'analyse des données. Cette définition de l'émergence, comme interne à un appareil de détection organisé selon des niveaux d'observation (« changement de regard »), est également défendue par [Bonabeau and Dessalles, 1997].

La notion d'*observateur* sous-entend une « idée » des phénomènes émergents que l'on veut observer. L'observation macroscopique n'est pas réalisée « au hasard », elle est liée au *design* de l'observateur et au modèle théorique qu'il a de l'objet observé.

Coût

L'observation microscopique nécessite l'utilisation simultanée de nombreuses sondes et il est nécessaire d'effectuer des ajustements pour leur synchronisation. Elle représente un coût d'implémentation considérable, proportionnel à la taille du système observé. De plus, l'étape d'analyse des données nécessite un grand temps de calcul (en fonction du phénomène émergent que l'on souhaite détecter). Les approches émergentistes sont donc très coûteuses en terme d'implémentation et de calcul. Ce coût augmente avec la taille des systèmes analysés. Il explose dans le cas de SMA de très grande taille.

L'observation macroscopique traite naturellement la distribution des systèmes et ne comporte pas d'étape d'analyse des données. De plus, elle nécessite beaucoup moins de sondes pour observer le système. Elle est ainsi beaucoup moins coûteuse et, de ce fait, elle supporte mieux le passage à l'échelle.

Chapitre 3

Modélisation

Ce chapitre présente un modèle d’observation macroscopique des systèmes multi-agents (SMA). Il est nécessaire, pour formaliser la notion de SMA *observé*, d’avoir un modèle de SMA *réel*. Ce dernier est construit à partir de la généralisation d’un modèle de système distribué.

La section 3.1 présente le modèle de système distribué à l’origine de ce travail de modélisation et justifie ce choix. La section 3.2 présente le modèle de SMA établi par généralisation du modèle précédent. La section 3.3 présente un modèle d’observation macroscopique des SMA. La section 3.4 donne des exemples théoriques de sondes macroscopiques implémentant ce modèle d’observation.

Remarque. Les modèles présentés dans ce chapitre décrivent en réalité l’exécution de systèmes distribués (section 3.1) ou l’exécution de SMA (section 3.2). Nous devrions alors parler de *modèles d’exécution de systèmes*. Par abus de langage, nous parlons plus simplement de *modèles de système*. Lorsque cette distinction est importante (notamment lors de la définition de l’observation temporelle, section 3.3.3) nous faisons attention d’utiliser les bons termes.

3.1 Modèle de système distribué

3.1.1 Origines du modèle

Le modèle de SMA introduit dans ce chapitre prend ses sources dans les travaux effectués à la fin des années 80 concernant l’analyse des systèmes distribués.

Chandy et Lamport mettent en exergue pour la première fois les problématiques liées à la notion d’état global des systèmes en 1985 [Chandy and Lamport, 1985]. L’état global est défini à un instant donné par l’ensemble des états internes des processus du système à cet instant, plus l’ensemble des messages en attente dans les canaux de communications entre les processus.

L’observation de l’état global est facile si l’on peut synchroniser l’initialisation d’un algorithme distribué pour chacun des processus. Cependant, cela est impossible en pratique, même pour des systèmes distribués de petite taille. Chandy et Lamport proposent donc un algorithme distribué permettant de construire une image virtuelle de l’état global du système : le *snapshot algorithm*. Cet algorithme fournit un état global potentiel, *i.e.* un état dont on ne peut être sûr qu’il fut effectivement celui du système à un temps donné. Cependant, Chandy et Lamport montrent que la connaissance de l’état global possible suffit à l’étude des propriétés stables du système, *i.e.* les propriétés qui, une fois vérifiées, restent vraies.

Plus tard, Mattern développe la notion de *temps virtuel* [Mattern, 1989]. Celui-ci est une approximation du *temps global réel* construite à partir des temporalités locales de chaque pro-

cessus. Il est nécessaire de recourir à cette notion dans le cas de systèmes distribués asynchrones puisqu'aucune horloge commune n'est disponible pour dater les actions des processus de manière absolue.

3.1.2 Pourquoi utiliser ce modèle ?

Les travaux cités précédemment répondent particulièrement aux attentes de ce projet pour diverses raisons.

Sémantique « minimale »

Les modèles théoriques de systèmes distribués ont une sémantique « minimale », c'est-à-dire qu'ils décrivent les système de manière formelle sur la base de deux notions simples : les *processus* et les *messages*. Ces deux notions forment un vocabulaire minimal pour traiter de la distribution des systèmes étudiés, de la décentralisation et de l'asynchronisme. Le cadre formel ainsi développé est assez simple pour pouvoir être généralisé au domaines des SMA et pour participer à l'analyse de leurs structures.

La généralisation repose sur le fait que : *un processus peut être interprété comme un agent et un message peut être interprété comme une interaction.*

Causalité

Ces modèles sont particulièrement efficaces pour décrire la structure causale de l'exécution des systèmes distribués. Cette causalité a une source locale (les événements successifs de l'exécution des processus) et une extension globale (donnée par les messages échangés). Elle définit des relations dans l'espace (causalité spatiale entre les processus) et dans le temps (causalité temporelle entre deux moments de l'exécution). Ces relations, fondées sur le vocabulaire minimal du modèle, sont aisément généralisables aux SMA et permettent de modéliser les causalités dimensionnelles qui structure leur exécution.

Émergence

Ces modèles enfin sont efficaces pour définir la notion de comportement global des système. La notion d'émergence sous-jacente à ces comportements globaux est liée à la notion de calcul émergent (*emergent computation*) développée dans [Forrest, 1990]. Elle y est définie comme :

- un ensemble d'entités en interaction : les processus
- un épiphénomène produit par ce processus : un état stable, un invariant ou une trace d'exécution
- une interprétation de cet épiphénomène comme un calcul ou le résultat d'un calcul

Selon [M.R. Jean et al., 1997], cette définition « est particulièrement intéressante car assez proche conceptuellement de ce que l'on souhaite formaliser en SMA ». Les processus sont des cas particuliers d'agents qui produisent également des épiphénomènes (invariants ou traces). Ces épiphénomènes, dans le cadre des SMA, sont généralement interprétés comme un processus de décision autonome qui mène à une action de l'agent sur l'environnement (comportement de l'agent). L'interprétation du comportement comme un calcul est fondamentalement liée à la notion d'exécution de l'agent (la base formelle de ce comportement).

Dans le cadre de ce projet, la notion de calcul est suffisante pour analyser les structures des SMA (*i.e.* leurs formes). Le comportement émergent d'un SMA est donc interprété comme un *calcul émergent*, de manière entièrement formelle. Celui-ci est détecté par l'observation macroscopique. Pour saisir et analyser les organisations propres aux SMA, il est nécessaire d'ajouter à cette interprétation formelle une dimension sémantique (*cf.* section 5.2.1).

3.1.3 Système distribué

Le modèle présenté dans cette section est en grande partie issu de [Mattern, 1989]. Un *système distribué* est constitué d'un ensemble de processus et d'un ensemble de messages échangés entre ces processus. Les messages définissent des relations entre les processus.

Définition 1. Un *système distribué* S est un couple (P, M) où :

- P est un ensemble fini de processus
- M est un ensemble fini de messages échangés entre les processus de P

3.1.4 Processus et temps local

Processus et états internes

Un *processus*, au sens informatique du terme, est l'instance d'un programme en cours d'exécution. Il comprend entre autre une liste d'instructions à exécuter, une mémoire qui lui est propre et une horloge interne.

L'*état interne* d'un processus, défini à un temps t , est l'ensemble des états de ses composants (instruction courante, état de la mémoire au temps t , temps de l'horloge, *etc.*).

Il est possible d'observer, à tout moment, certaines propriétés de l'état interne d'un processus. Cette observation fournit une représentation partielle de l'état interne du processus. Pour des raisons pratiques, il est souvent impossible d'effectuer une observation totale de l'état interne. Les difficultés sont liées à la complexité de cet état et à l'intrusion trop importante d'une observation totale. L'acquisition d'informations concernant les processus est fondamentalement limitée en ce sens. Dans la suite, l'*état interne* d'un processus désigne en fait son *état interne observé*.

Événements et phases

Un processus peut être vu comme une *séquence d'événements*, où un événement est une *transition de son état local* [Mattern, 1989]. De manière conceptuelle, une telle transition est une *action atomique* du processus. Elle est sans durée et elle sépare deux *phases*.

Une *phase* d'un processus est donc définie sur la période séparant deux événements consécutifs de ce processus, période durant laquelle l'état local apparaît comme inchangée pour l'observateur.

L'*histoire d'un processus* p est la séquence de ses événements. Nous notons $hist(p)$ cet ensemble. De la même manière, l'*histoire d'un système distribué* S est l'union des histoires de ses processus, notée $Hist(S)$.

Espace logique

L'ensemble des processus définit un *espace logique* pour l'histoire du système. Un événement $e \in Hist(S)$ est toujours situé sur un unique processus p . C'est la position de e dans l'espace logique que l'on note $pos(e) = p \in P$.

Les notions de localité et de globalité sont associées à cet espace logique. Par exemple, une *observation locale* (ou *microscopique*) fournit de l'information à propos d'une unique position de l'espace logique (*e.g.* l'observation de l'état interne d'un processus). On parle donc d'*information locale*. Une *observation globale* (ou *macroscopique*), au contraire, fournit une information étendue sur une portion de l'espace logique. Elle regroupe plusieurs processus et on parle d'*information globale*.

Horloge

Une *horloge* est un dispositif permettant de mesurer le temps. Elle définit un *temps physique* (relatif à l'horloge) et un *temps logique* (absolu).

L'annexe A donne une définition précise de ces notions. Elle expose également le modèle standard de temps donné par [van Benthem, 1991]. Pour résumer, la *composante logique* du temps est l'ordre partiel induit par l'horloge sur l'ensemble des événements qu'elle date. Cet ordre représente la notion de *causalité temporelle* - propriété fondamentale du temps - qui se résume ainsi : *le passé peut influencer le futur et le futur ne peut pas influencer le passé*.

Temps logique local

Dans ce modèle de système distribué, un processus ne perçoit que sa propre histoire, c'est-à-dire qu'il ne peut pas utiliser son horloge interne pour dater les événements des autres processus. Par contre, il peut l'utiliser pour dater ses propres événements. L'horloge interne d'un processus p définit ainsi un *temps physique* pour son histoire locale. Il est *relatif* au processus, puisqu'il est donné par son horloge interne. Il s'agit donc d'un *temps physique local*.

Dans le projet d'obtenir une *information globale* concernant le système, il est nécessaire de disposer d'un *temps global* du système, *i.e.* un temps *absolu* permettant de dater tous les événements de l'histoire du système, indépendamment de leur position. Les différents temps physiques locaux, puisqu'ils sont *relatifs* sont en théorie inefficaces. C'est pourquoi il ne faut retenir de ces temps locaux que la *composante logique* qui, elle, a valeur d'absolu (*cf.* annexe A).

Remarque 1. Dans certains cas, il est possible d'avoir une approximation de ce *temps global*, en garantissant des contraintes de synchronisation sur les *temps locaux*. La composante physique des horloges est partiellement utilisée et les incertitudes sont rectifiées par l'analyse [Chassin de Kergommeaux et al., 2001]. Ce projet se place dans le pire des cas, celui où aucune garantie n'est faite *a priori* sur la synchronisation. La composante physique est donc entièrement éliminée du modèle.

Remarque 2. Si elles ont la même source, les horloges internes de plusieurs processus peuvent être parfaitement synchronisées. Par exemple, plusieurs processus peuvent disposer de fonctions `getTime()` reliées au même dispositif. Ces fonctions, si elles sont simultanément déclenchées, renvoient la même date. Dans ce cas, le temps physique de ce dispositif peut être utilisé directement comme temps global du système. Néanmoins, dans la plupart des systèmes complexes auxquels ce projet s'intéresse, les horloges internes des processus ne sont pas *toutes* synchronisées.

Un *temps logique local* est défini pour chaque processus du système. Il ordonne son histoire locale à partir de son horloge interne. Nous notons $<_p$ cet ordre temporel local. Dans la mesure où un événement est une transition de l'état interne du processus, deux événements sur le même processus ne peuvent être simultanés. $<_p$ est donc un ordre total.

L'événement e d'un processus p partitionne son histoire en trois sous-parties disjointes : le *passé*, le *présent* et le *futur* (ces notions étant définies localement).

- Le *passé local* de e est l'ensemble des événements e' de $hist(p)$ tels que $e' <_p e$. On note $passé_e(p)$ cet ensemble et on dit que e' est *avant* e .
- Le *présent local* de e est l'ensemble constitué de l'unique événement e . On note $présent_e(p)$ cet ensemble.
- Le *futur local* de e est l'ensemble des événements tels que $e <_p e'$. On note $futur_e(p)$ cet ensemble et on dit que e' est *après* e .

Il est nécessaire de préciser que la *causalité temporelle* introduite par le temps logique induit des liens de *causalité potentielle* entre les événements. En ce sens, un événement est une *cause potentielle* d'un autre événement si et seulement si *il est possible que le premier affecte le second*. La notion de causalité temporelle (un événement précède un autre) n'implique pas nécessairement une causalité effective (un événement est cause d'un autre) mais une causalité seulement

potentielle.

Représentation graphique

La figure 3.1 donne la représentation graphique de l'exécution d'un processus p . Il s'agit d'un segment gradué par des événements. Les intervalles représentent les phases du processus et une flèche en abscisse donne la direction du temps logique local du processus.

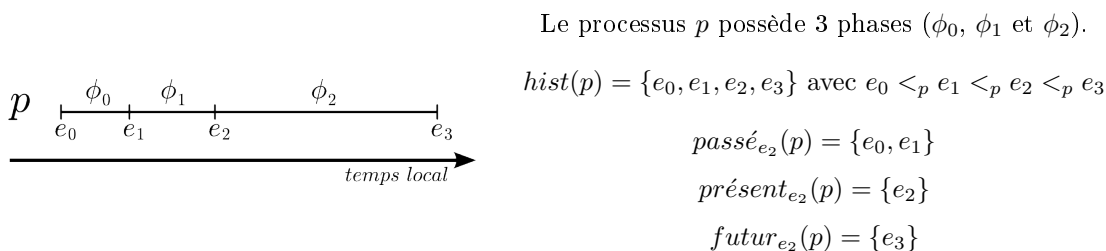


FIGURE 3.1 – Représentation graphique de l'exécution d'un processus p

3.1.5 Messages et temps global

Messages

Un message m est associé à un couple d'événement (e_0, e_1) . e_0 est l'*émission* du message m par un processus émetteur et e_1 est sa *réception* par un processus récepteur, ces deux processus étant nécessairement distincts.

Temps logique global

Il y a un lien de *causalité temporelle* entre les événements e_0 et e_1 en ce sens que l'émission du message m a nécessairement lieu avant sa réception. Celle-ci influence alors potentiellement celle-là et, par transitivité, le passé local de e_0 du processus émetteur peut affecter le futur local de e_1 du processus récepteur. Les messages d'un système distribué induisent un ordre sur les événements d'émission et de réception. On note $e_0 <_m e_1$.

Les messages, puisqu'ils sont ainsi étendus dans l'espace, déterminent des liens de *causalité temporelle* entre les différents temps logiques locaux des processus. Ils permettent ainsi de construire un *temps logique global*.

Un ordre partiel sur l'histoire du système est défini par fermeture transitive des relations d'ordres locales et de celles induites par les messages. Cet ordre définit à son tour la notion de *causalité temporelle globale*.

Définition 2. Soient S un système distribué, P l'ensemble de ses processus et M l'ensemble de ses messages. Nous notons $<_S$ l'ordre partiel sur $Hist(S)$ et nous le définissons de la manière suivante : $<_S$ est la *plus petite relation transitive* telle que, pour tout e_0 et e_1 dans $Hist(S)$, nous avons $e_0 <_S e_1$ si l'une au moins des deux conditions suivantes est réalisée :

- Condition d'ordre local : $\exists p \in P, (e_0, e_1) \in hist(p) \text{ et } e_0 <_p e_1$
- Condition d'ordre liée aux messages : $\exists m \in M, e_0 <_m e_1$

Puisque $<_S$ est un ordre partiel, il est possible que nous n'ayons ni $e_0 <_S e_1$, ni $e_1 <_S e_0$. Dans ce cas, nous disons que les événements e_0 et e_1 sont *causalement indépendants* et nous notons $e_0 \parallel_S e_1$.

L'événement e d'un système distribué S partitionne son histoire en trois sous-parties disjointes : le *passé*, le *présent* et le *futur* (ces notions étant définies globalement).

- Le *passé global* de e est l'ensemble des événements e' de $Hist(S)$ tels que $e' <_S e$. On note $Passé_e(S)$ cet ensemble et on dit que e' est *avant* e .
- Le *présent global* de e est l'ensemble des événements tels que $e' \parallel_S e$, plus l'événement e lui-même. On note $Présent_e(S)$ cet ensemble. La notion de présent global est en fait liée à une indétermination des relations de causalités temporelles. Ainsi, un événement est globalement présent vis-à-vis d'un autre événement si et seulement si il n'est ni avant, ni après. On dit que la causalité de e et e' est *indéterminée*.
- Le *futur global* de e est l'ensemble des événements tels que $e <_S e'$. On note $Futur_e(S)$ cet ensemble et on dit que e' est *après* e .

Remarque 1. Le passé global d'un événement e est une extension de son passé local, de même pour son présent ou futur global. Nous avons ainsi, $\forall e \in Hist(S), pos(e) = p$:

- $passé_e(p) \subset Passé_e(S)$
- $présent_e(p) \subset Présent_e(S)$
- $futur_e(p) \subset Futur_e(S)$

Le temps global défini ici est *virtuel*, en ce sens qu'il est *construit* à partir de différentes observations de causalités temporelles locales. Un temps global *réel* peut être défini à partir d'une horloge commune à tous les processus du système. Une telle horloge induit alors un ordre total sur l'histoire du système distribué, puisque n'importe quel événement peut être daté. Cependant, la notion de *temps réel* n'est pas indispensable ici, dans la mesure où les notions de *causalités potentielles* et d'*indépendances causales* sont entièrement définies grâce au *temps virtuel*. Un *temps réel* n'apporte aucune information supplémentaire quand à ces relations de causalité. Or, ce sont elles seulement qui sont intéressantes pour l'analyse de l'exécution des systèmes.

Représentation graphique

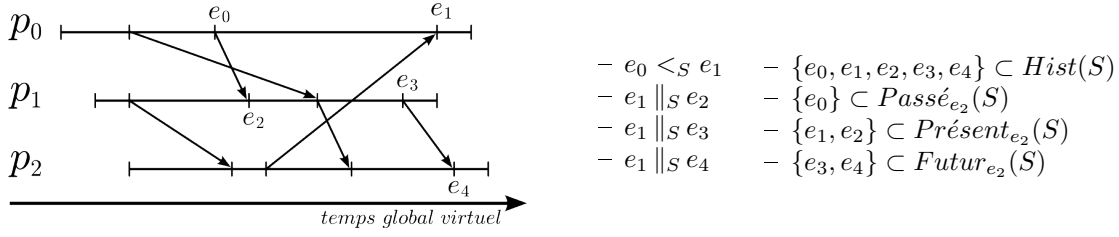
La figure 3.2 donne la représentation graphique de l'exécution d'un système distribué S . Il s'agit d'un ensemble de segments gradués (les processus et leurs événements) et d'un ensemble de flèches (les messages) reliant entre-elles les graduations de ces segments. Ce type de diagrammes, représentant l'espace et le temps de l'exécution, est utilisé dans [Mattern, 1989] sous le nom de *time diagrams*. Nous parlerons plutôt dans ce rapport de *diagrammes d'exécution*.

Le système distribué représenté dans la figure 3.2 est constitué de 3 processus (p_0 , p_1 et p_2) et on a les relations suivantes :

- $e_2 <_S e_3$, car ils sont situés sur le même processus p_1 et $e_2 <_{p_1} e_3$.
- $e_0 <_S e_2$ et $e_3 <_S e_4$, car e_0 et e_3 sont les émissions de messages dont e_2 et e_4 sont les réceptions respectives.
- $e_0 <_S e_4$, par transitivité de la causalité temporelle globale.

Coupes temporelles

La notion de coupe temporelle permet de segmenter l'exécution des systèmes en tranches de temps cohérentes. Elle est utilisée dans la section 3.3.3 pour formaliser l'observation temporelle des SMA. Le reste de cette section formalise la notion de coupe telle qu'elle est introduite par

FIGURE 3.2 – Représentation graphique de l'exécution d'un système distribué S

[Mattern, 1989] dans le cadre des systèmes distribués.

Une *coupe temporelle* c d'un système distribué S est un ensemble de N nouveaux événements, nommés *événements de coupe*, où N est le nombre de processus du système ($N = \text{card}(P)$). Ces événements sont ajoutés à l'histoire du système, à raison d'un événement de coupe sur chaque processus. Celui-ci étant ordonné par le *temps logique local* du processus auquel on l'ajoute.

Une coupe partitionne l'histoire du système en trois sous-parties disjointes :

- Le *passé* d'une coupe est l'union des passés locaux des événements de coupe. Formellement :

$$\text{Passé}_c(S) = \bigcup_{e_c \in c} \text{passé}_{e_c}(\text{pos}(e_c))$$

- Le *présent* d'une coupe est défini de la même manière.

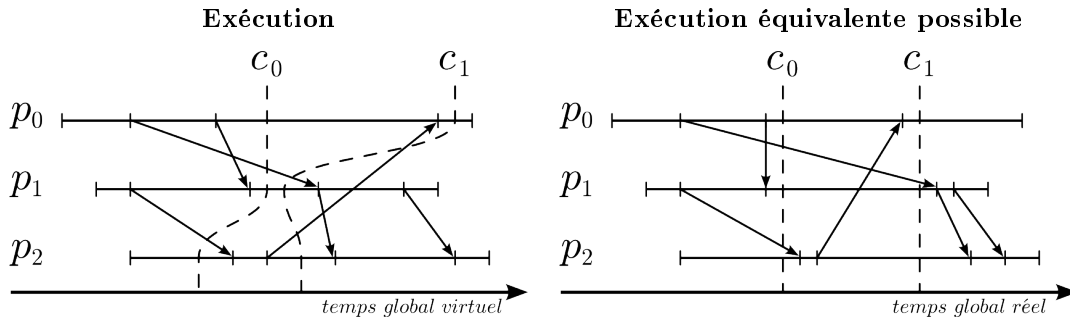
Or, $\forall e \in \text{Hist}(S), \text{présent}_e(\text{pos}(e)) = e$. On a donc :

$$\text{Présent}_c(S) = c$$

- Le *futur* d'une coupe est l'union des futurs locaux des événements de coupe. Formellement :

$$\text{Futur}_c(S) = \bigcup_{e_c \in c} \text{futur}_{e_c}(\text{pos}(e_c))$$

La figure 3.3 présente deux coupes c_0 et c_1 de l'exécution d'un système distribué S . Ces coupes sont représentées par des lignes en pointillés qui séparent le diagramme en deux. Les événements de coupe sont placés sur les intersections entre la ligne en pointillés et les segments représentant les processus. Leurs passés respectifs sont les ensembles des événements à gauche des lignes en pointillés, leurs futurs les ensembles des événements à droite.

FIGURE 3.3 – Deux coupes cohérentes de l'exécution d'un système distribué S

Coupes cohérentes

Une coupe est *cohérente* lorsque son futur n'influence pas son passé. Formellement :

$$C \text{ est cohérente} \iff \forall e_0 \in \text{Passé}_c(S), \nexists e_1 \in \text{Futur}_c(S), e_1 <_S e_0$$

Les coupes c_0 et c_1 de la figure 3.3 sont cohérentes.

Dans le cas contraire, la coupe est dite *incohérente*. La figure 3.4 présente une telle coupe. En effet, on a :

$$e_{c0} <_{p_0} e_0 \implies e_0 \in \text{Futur}_c(S)$$

$$e_1 <_{p_1} e_{c1} \implies e_1 \in \text{Passé}_c(S)$$

$$\text{Et } e_0 <_S e_1$$

Un événement du futur influence donc potentiellement un événement du passé : C est incohérente.

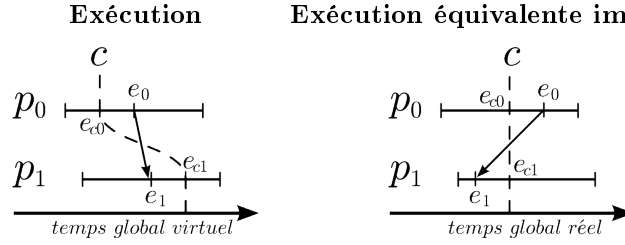


FIGURE 3.4 – Une coupe incohérente

Une propriété intéressante des coupes cohérentes est qu'elles sont toujours *possibles* vis-à-vis d'un *temps global réel*. C'est-à-dire qu'il existe une exécution du système telle que les événements de coupes sont *simultanés* vis-à-vis de ce temps global. Au contraire, une coupe incohérente n'est jamais possible.

Le diagramme de droite de la figure 3.3 montre comment on peut avoir un temps global réel sur le système distribué tel que l'ordre partiel est conservé et tel que les événements de la coupe c_0 soient *simultanés* (ainsi que ceux de la coupe c_1). La figure 3.4 montre que ce n'est pas possible pour une coupe incohérente car, vis-à-vis de n'importe quel temps global réel, une simultanéité des événements de coupe va à l'encontre de la propriété fondamentale du temps. En effet, l'émission du message est postérieure à sa réception : le futur influence le passé !

Une coupe cohérente *précède* une autre coupe cohérente lorsque le passé de la première est inclus dans celui de la seconde. Dans ce cas, le futur de la seconde est alors inclus dans le futur de la première. Par exemple, la coupe c_0 de la figure 3.3 précède la coupe c_1 . On a :

- $\text{Passé}_{c_0}(S) \subset \text{Passé}_{c_1}(S)$
- $\text{Futur}_{c_1}(S) \subset \text{Futur}_{c_0}(S)$

Nous reviendrons sur cette notion de *coupe cohérente* dans la section 3.3.3 pour formaliser la notion d'*observation temporelle*.

3.2 Méta-modèle de système multi-agents

Cette section généralise la notion de *système distribué* (telle que définie précédemment) à la notion de *système multi-agents*. Comme il a été dit précédemment, cette généralisation repose

sur le fait que : *un processus peut être interprété comme un agent et un système distribué peut être interprété comme un SMA.*

Attention : un SMA est une sous-catégorie de système distribué au sens large de système *qui est distribué*. Ici, c'est la notion plus stricte de système *composé de processus et de messages* (telle que définie dans la section précédente) qui est généralisée.

Le modèle AEIO [Demazeau, 1995] expose une méthode de programmation orientée agent fondée sur la décomposition des SMA selon quatre dimensions : les *Agents*, les *Environnements*, les *Interactions* et les *Organisations* (cf. également la méthode VOYELLE exposée dans [Demazeau, 2001]). Cette section définit chacune de ces dimensions à partir des éléments formalisés précédemment. Ces définitions permettent de construire un méta-modèle de SMA sur lequel les notions introduites précédemment peuvent être appliquées.

Un *modèle* de SMA est la description abstraite de l'architecture de ses agents et leurs comportements. Cette section présente un *méta-modèle* de SMA puisqu'il permet de modéliser de manière générique les modèles connus de SMA.

3.2.1 A comme Agents

[Ferber, 1995] donne une définition extensive des caractéristiques fondamentales qui peuvent être attribuées à un agent. La liste suivante retient de ces caractéristiques celles qui peuvent être modélisées à partir du modèle de système distribué.

Définition 3. « On appelle *agent* une entité physique ou virtuelle :

- a. qui est capable d'agir dans un environnement,
- b. qui peut communiquer directement avec d'autres agents,
- e. qui est capable de percevoir (mais de manière limitée) son environnement. »

Les caractéristiques a. et e. sont traitées dans la partie *E comme Environnements*. La caractéristique b. est traitée dans la partie *I comme Interactions*.

Les caractéristiques suivantes sont celles regroupées sous les notions d'*architecture interne* (statique) et d'*état interne* (dynamique) de l'agent. Elles ne sont pas modélisées dans la mesure où ce projet traite uniquement de la *description externe* des agents (cf. section 1.1.3), en terme de *phases* et de *transitions*. Sa structure interne est alors considérée en toute généralité (excepté pour le modèle de l'horloge interne). Les caractéristiques qui suivent sont donc prévues par le modèle de SMA, mais elles n'y sont pas modélisées.

« On appelle *agent* une entité physique ou virtuelle :

- c. qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
- d. qui possède des ressources propres,
- f. qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
- i. dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit. »

L'horloge est la seule composante interne des processus qui a été modélisée. Un agent ne dispose pas nécessairement d'une telle horloge, mais la notion de *temps logique propre* peut être généralisée, au moins du point de vue de l'observateur. L'observation locale d'un agent (par l'intermédiaire de traces, d'interfaces, etc.) fournit donc un ordre total sur la séquence des évé-

nements observés. Cet ordre permet de définir le *temps logique local* de l'agent.

Enfin, les deux caractéristiques suivantes ne sont pas prises en compte dans ce projet.

« On appelle *agent* une entité physique ou virtuelle :

- g. qui possède des compétences et offre des services,
- h. qui peut éventuellement se reproduire. »

3.2.2 E comme Environnements

Plusieurs travaux proposent de modéliser l'environnement des SMA comme un ensemble d'agents qui sont *extérieurs* au système. On parle ainsi d'*agents environnementaux* [Demazeau and Rocha Costa, 1996].

Dans [Omicini et al., 2008], le méta-modèle Agents & Artefacts (A&A) « est caractérisé par trois abstractions fondamentales :

- Les *agents*, pour représenter les composantes pro-actives du système, encapsulant l'exécution autonome d'un certain type d'activité au sein d'un certain type d'environnement ;
- Les *artefacts*, pour représenter les composantes réactives du système telles que les ressources et les médias qui sont construits, partagés, manipulés et utilisés intentionnellement par les agents pour soutenir leurs activités, qu'elles soient coopératives ou compétitives ;
- Les *espaces de travail*, conteneurs conceptuels des agents et des artefacts, utiles pour définir la topologie de l'environnement et pour donner un moyen de définir une notion de localité. »

L'environnement peut donc être modélisé par un ensemble d'entités réactives (par opposition aux agents pro-actifs du système) : les *artefacts* [Omicini et al., 2008]. Dans les termes du modèle de système distribué, un *agent réactif* est un agent qui ne change pas spontanément d'état interne. Ainsi, les seuls événements de l'histoire des artefacts sont des événements d'interactions (comme défini dans la section *I comme Interactions*).

Ces interactions permettent de définir la perception de l'environnement par les agents (envoi spontané d'information concernant l'état de l'artefact). Il y a une symétrie interactionnelle entre *action des agents sur les artefacts* et *perception des artefacts par les agents*. Dans la suite, aucune distinction n'est faite entre agents et artefacts, celle-ci pouvant être introduite très facilement pour modéliser l'environnement.

Les *espaces de travail* forment de manière abstraite la structure logique du SMA. Il s'agit de l'*espace logique* défini par l'ensemble des agents et des artefacts.

3.2.3 I comme Interactions

La notion de *message* peut être généralisée par la notion d'*interaction*. Une interaction est, de la même manière qu'un message, une relation étendue dans le temps et dans l'espace (c'est-à-dire entre plusieurs agents).

Cependant, une interaction n'est pas nécessairement asymétrique, contrairement à un message qui lie un *émetteur* à un *récepteur*. En effet, elle peut définir une relation de causalité symétrique entre deux *événements d'interaction*. C'est le cas par exemple lorsqu'on formalise une interaction « physique » entre deux agents (par exemple la collision de deux agents transporteurs du modèle de SMA présenté dans [Joumaa et al., 2009]). Une interaction symétrique est alors une double interaction synchronisée (via l'environnement, dans le cas d'une collision).

Cette différence avec les messages échangés dans les systèmes distribués introduit la notion de *co-dépendance* entre des événements de deux agents distincts. Ainsi, deux événements sont *co-dépendants* si et seulement si ils sont une cause potentielle l'un de l'autre. Deux événements

co-dépendants sont alors nécessairement simultanés. La figure 3.5 donne un exemple d'interaction symétrique entre deux agents a_1 et a_2 . Les événements e_0 et e_1 y sont co-dépendants.

Ce type de relation est en fait formalisé par Fidge dans le cadre de systèmes distribués sous le terme de *message synchrone* [Fidge, 1991]. Les *messages asynchrones* étant alors les messages tels que définis précédemment (messages asymétriques).

L'ensemble des interactions permet enfin de définir le *temps logique global virtuel* du SMA, par extension des *temps logiques locaux*.

3.2.4 O comme Organisation

Les *organisations* d'un SMA sont l'ensemble des rôles attribués aux agents afin qu'ils puissent atteindre, par coopération ou par compétition, un objectif global. Elle est l'architecture abstraite du système, que l'on retrouve décentralisée dans les architectures internes de chaque agent. Ainsi, la notion d'organisation est un outil de conception des SMA, permettant de modéliser un cadre sur lequel le système atteindra les objectifs que le concepteur s'est fixé.

Lors de l'analyse d'un système, l'organisation n'est pas nécessairement explicite, elle n'est pas connue *a priori*. L'observateur modélisé dans la section suivante est dans cette situation. Il ne connaît ni l'architecture interne des agents, ni l'architecture globale du système (*i.e.* ses organisations). Le but de la méthode d'analyse proposée dans ce projet est justement de décrire *a posteriori*, par l'observation des comportements visibles des agents, les organisations des SMA observés.

Représentation graphique

L'exécution d'un SMA est représentée de la même manière que celle des systèmes distribués : par un *diagramme d'exécution*. La figure 3.5 présente un tel diagramme pour un système S composé de 3 agents.

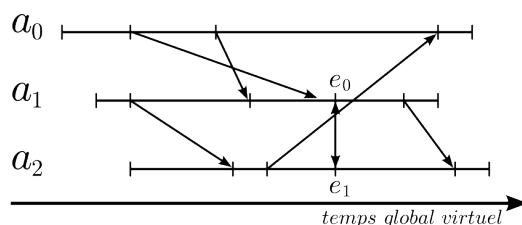


FIGURE 3.5 – Représentation graphique d'un SMA S

3.3 Modèle d'observation macroscopique

Cette section formalise la notion d'*observation macroscopique* d'un SMA. Il s'agit de définir une observation globale (relative à des parties du système) et non plus locale (relative à un agent). Cette section introduit d'abord la notion d'observation en toute généralité (*cf.* section 3.3.1), elle en donne les propriétés fondamentales. Elle explicite ensuite deux catégories d'observation : la première liée à l'*espace* (*cf.* section 3.3.2) et la seconde liée au *temps* (*cf.* section 3.3.3).

3.3.1 Modèle générique

L'*observation macroscopique* d'un SMA est un processus d'investigation qui tente de déterminer les caractéristiques du système sans faire l'analyse complète de ses parties. L'observateur (celui qui procède à l'observation) aperçoit alors le système de manière grossière, supprimant la multitude des détails qui fondent la complexité de celui-ci et se focalisant sur ce qu'il veut voir et comprendre. Cela présuppose que l'observateur a déjà une idée de ce qu'il veut observer, qu'il dispose d'un modèle du système et qu'il sait *comment* l'observer.

Plus formellement, une *observation macroscopique* est une opération de transformation d'un SMA source, nommé *SMA réel* et noté S_r , en un SMA cible, nommé *SMA observé* et noté S_o . Les observations présentées dans cette section sont des *transformations syntaxiques* de SMA, en ce sens qu'elles en modifient formellement la structure (spatiale et temporelle). Elles redéfinissent ainsi leur espace logique et leurs temps logiques (locaux et global). De telles transformations peuvent également être exprimées à l'aide de grammaires formelles, à la manière des travaux présentés dans [Milner et al., 1992] et [Jorrand, 1993].

La nature des *transformations sémantiques* des SMA par l'observation macroscopique est un travail présenté en perspective de ce projet (*cf.* section 5.2.1).

Morphisme d'histoires

Une observation *conserve la relation globale de causalité potentielle* du SMA qu'elle transforme. Ainsi, pour toute observation qui transforme un SMA S_r en un SMA S_o , il existe un *morphisme surjectif* f de $Hist(S_r)$ vers $Hist(S_o)$ tel que $\forall (e_1, e_2) \in Hist(S_r)^2$:

$$e_1 <_{S_r} e_2 \quad \longrightarrow \quad f(e_1) <_{S_o} f(e_2)$$

Attention : deux événements indépendants peuvent devenir dépendants sous l'effet du morphisme.

$$e_1 \parallel_{S_r} e_2 \quad \not\longrightarrow \quad f(e_1) \parallel_{S_o} f(e_2)$$

Ce morphisme est *surjectif*, car il n'existe aucun événement du système observé qui ne peut être relié à au moins un événement du système réel. En d'autres termes, une observation ne fait pas apparaître d'événements *ex nihilo*. Il y a toujours un lien avec ceux du système réel. Par contre, deux événements indépendants peuvent se superposer lors de l'observation et devenir ainsi indiscernables.

Enfin, nous nommons une telle fonction surjective de $Hist(S_r)$ vers $Hist(S_o)$, conservant les ordres définis sur ces ensembles, un *morphisme d'histoire*.

3.3.2 Observation spatiale

Agréger des agents

Une *observation spatiale* transforme un SMA en agrégeant une partie de ses agents. L'opération d'agrégation consiste à remplacer l'ensemble des *agents agrégés*, nommé *agrégat d'agents*, par un unique agent, nommé *agent observé*. L'*espace logique* du SMA est ainsi modifié.

L'observation spatiale pose deux problèmes : comment définir l'*histoire* et les *interactions* de l'agent observé ?

Ce modèle présente l'observation d'une unique agrégation, par souci de lisibilité. Il peut aisément être étendu à de multiples agrégations en partitionnant l'espace du SMA (chaque partie définissant un agrégat). En effet, les transformations par agrégation sont associatives et commutatives.

L'observation spatiale peut également être effectuée à partir d'un ensemble de parties éventuellement superposées. L'étude de la notion de superposition, de l'associativité et de la commutativité de telles observations, est envisagée en perspective de ce projet dans la section 5.2.2.

Exemple par analogie

Lors d'une observation spatiale, le regard de l'observateur ne porte plus distinctement sur les agents de l'ensemble en tant que détails du système. Il les considère de manière grossière comme un unique agent dont l'architecture est composé d'éléments indiscernables.

Si on conçoit un chat comme un système dont les cellules sont les agents, un observateur pour qui le niveau de détail est celui de la cellule ne peut apercevoir le chat. Celui-ci est caché derrière la complexité des agencements de ses cellules. L'agrégation spatiale de celles-ci donne alors l'image d'un agent plus simple, au comportement mieux défini : le chat lui-même. Du même coup, les cellules du chat et l'ensemble de leurs interactions ne sont plus visibles. L'agrégation spatiale, en supprimant les détails, apporte du sens au système. Celui-ci peut être identifié de manière globale, on peut dire « c'est un chat ».

Définir une histoire

Soient A un ensemble d'agents (*i.e.* un *agrégat*) et a l'*agent observé* à partir de cet ensemble. L'histoire de l'agent observé est construite à partir de l'histoire de l'agrégat (définie comme l'union des histoires des agents agrégés et notée $Hist(A)$) à l'aide d'un *morphisme d'histoires* f .

- La restriction de ce morphisme au sous-ensemble $Hist(S_r) \setminus Hist(A)$ est l'identité : il ne modifie pas les événements qui n'appartiennent pas à l'agrégat.
- Sa restriction au sous-ensemble $Hist(A)$ est à valeur dans $hist(a)$: il associe à chaque événement de l'agrégat un événement de l'agent observé.

Il peut exister plusieurs morphismes d'histoires conservant la causalité d'un SMA lors d'une agrégation d'agents. Ceci est dû au fait que l'histoire de l'agrégat est partiellement ordonnée (*temps logique global*) et que celle de l'agent observé est totalement ordonnée (*temps logique local*).

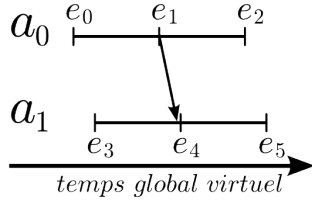
Soient e_0 et e_1 deux événements de $Hist(A)$ tels que e_0 et e_1 sont indépendants ($e_0 \parallel_{S_r} e_1$). Les images selon f de ces événements, si elles ne sont pas identiques, sont nécessairement soumises à la relation de *causalité potentielle locale* de l'agent a . Trois cas sont donc possibles :

- $f(e_0) = f(e_1)$ (pas de relation de causalité)
- $f(e_0) <_a f(e_1)$
- $f(e_1) <_a f(e_0)$

Il existe donc dans ce cas au moins trois morphismes d'histoires différents. Ils définissent chacun un agent observé différent, il y a donc *trois observations possibles*.

Un morphisme d'histoire peut être caractérisé par un *temps global réel*. Dans ce cas, l'horloge globale du système impose un ordre total sur les événements et l'ensemble des contraintes définit un unique morphisme d'histoires.

L'exemple suivant présente différents agents observés selon différents morphismes d'histoires lors d'une agrégation de deux agents. Chaque morphisme est présenté en fonction du temps global réel qui le caractérise.

Exemple

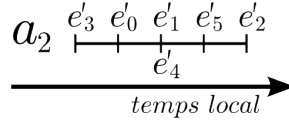
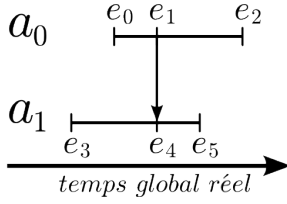
Cette figure représente deux agents que l'on veut agréger. On note $A = \{a_0, a_1\}$ cet agrégat et a_2 son agent observé. Le *temps global virtuel* donne les contraintes suivantes sur $Hist(A) = \{e_0, e_1, e_2, e_3, e_4, e_5\}$:

$$\begin{aligned} e_0 <_A e_1 <_A e_2 \\ e_3 <_A e_4 <_A e_5 \\ e_1 <_A e_4 \end{aligned}$$

Tout morphisme d'histoire f de $Hist(A)$ sur $hist(a_2)$ doit donc vérifier les contraintes suivantes (on note $f(e_i) = e'_i$ pour une meilleure lisibilité) :

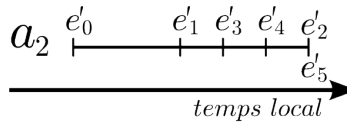
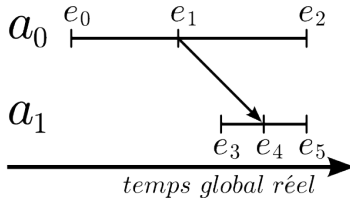
- $e'_0 <_{a_2} e'_1 <_{a_2} e'_2$
- $e'_3 <_{a_2} e'_4 <_{a_2} e'_5$
- $e'_1 <_{a_2} e'_4$

Il existe plusieurs morphismes vérifiant ces contraintes. Les figures 3.6 et 3.7 en présentent deux, sous la forme de diagrammes d'exécution avec le temps global réel qui les caractérise.

Morphisme d'histoires**Agent observé**

$$\begin{aligned} e'_3 <_{a_2} e'_0 <_{a_2} (e'_1 = e'_4) \\ <_{a_2} e'_5 <_{a_2} e'_2 \end{aligned}$$

FIGURE 3.6 – Morphisme d'histoires et agent observé

Morphisme d'histoires**Agent observé**

$$\begin{aligned} e'_0 <_{a_2} e'_1 <_{a_2} e'_3 <_{a_2} e'_4 \\ <_{a_2} (e'_2 = e'_5) \end{aligned}$$

FIGURE 3.7 – Morphisme d'histoires et agent observé

Redéfinir les interactions

L'ensemble des interactions du SMA réel est également modifié par l'observation spatiale. En effet, ces relations doivent être redéfinies pour rester cohérentes avec le nouvel espace logique du système. Pour chaque interaction du SMA réel, trois cas sont possibles :

- L'interaction est *interne* à l'agrégat : elle met en relation deux agents agrégés.
Dans ce cas, l'interaction n'apparaît plus dans le système observé comme une relation entre

deux agents distincts, mais comme une relation de l'agent observé avec lui-même. En ce sens, elle n'est plus une interaction. Elle est donc supprimée de l'ensemble des interactions du système.

- L'interaction est *externe* à l'agrégat : elle met en relation deux agents dont aucun n'est agrégé. Dans ce cas, l'observation spatiale ne modifie pas cette interaction. Celle-ci reste alors inchangée.
- L'interaction est *mixte* : elle relie un agent agrégé à un agent non-agrégé. Dans ce cas, le morphisme d'histoires utilisé pour l'agrégation permet de redéfinir l'événement d'interaction appartenant à l'agent agrégé. Cet événement est remplacé par son image, celle-ci appartenant à l'agent observé.

Exemple

La figure 3.8 représente à gauche un SMA réel S_r dont on veut agréger les agents a_1 et a_2 . On note $A = \{a_1, a_2\}$ cet agrégat et a_4 l'agent observé. Le diagramme à droite représente le SMA S_o , observé à partir de S_r selon un morphisme d'histoires (non précisé).

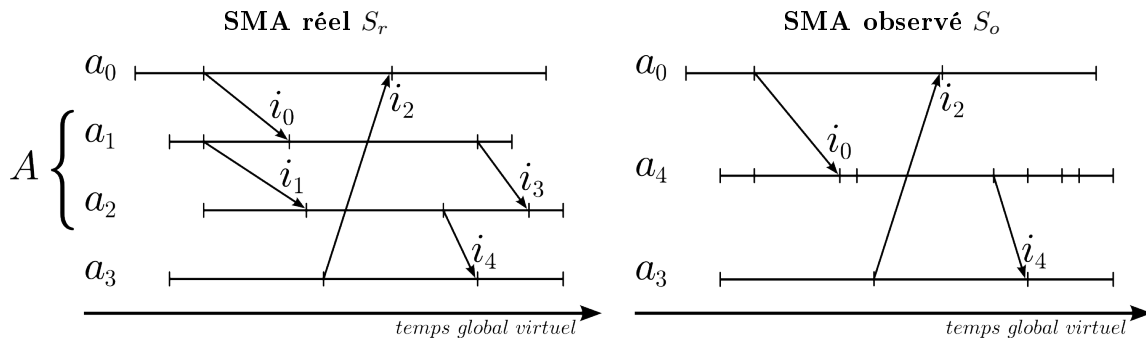


FIGURE 3.8 – Agrégation d'agents

On remarque que :

- Les interactions i_1 et i_3 , internes à A dans S_r , n'apparaissent pas dans S_o .
- L'interaction i_2 , externe à A dans S_r , est restée inchangée lors de l'observation.
- Les interactions i_0 et i_4 , mixtes dans S_r , relient l'agent a_4 (observé dans S_o) au reste du système.

Conservation de la temporalité

En plus de conserver la *causalité temporelle* et *potentielle* du système, l'observation spatiale garantit les deux inclusions suivante $\forall e \in Hist(S_r)$:

- $f(Passé_e(S_r)) \subset Passé_{f(e)}(S_o)$
- $f(Futur_e(S_r)) \subset Futur_{f(e)}(S_o)$

En effet, pour la première inclusion :
 soit $e' \in f(Passé_e(S_r))$
 $\exists e'' \in HistS_r$ tel que $e'' <_{S_r} e$ et $f(e'') = e'$
 $e'' <_{S_r} e \implies f(e'') < f(e) \implies e' < f(e)$
 donc $e' \in Passé_{f(e)}(S_o)$

La preuve est similaire pour la seconde inclusion.

L'observation spatiale augmente ainsi le passé et le futur des événements du système. Ceci est dû au fait que des événements causalement indépendants peuvent devenir dépendants sous le regard de l'observateur. La notion de causalité *potentielle* prend alors tout son sens, puisque deux événements qui semblent dépendre l'un de l'autre selon un certain point de vue, peuvent se révéler indépendants selon une observation plus précise. Lorsqu'on a $e'_0 <_{S_o} e'_1$, il est seulement *possible* que e_0 soit une cause de e_1 . Un examen du comportement interne des agents permet de savoir, en fonction de la structure causale interne, si *effectivement* e_0 est une cause de e_1 .

Bilan

Pour conclure, l'observation spatiale consiste à :

- remplacer les processus agrégés par un processus observé,
- définir l'histoire du processus observé à partir de celle de l'agrégat et d'un morphisme surjectif,
- supprimer les interactions internes à l'agrégat,
- redéfinir les interactions entre le processus observé et les autres agents du système.

3.3.3 Observation temporelle

L'observation temporelle de l'exécution d'un SMA est soumise aux difficultés liées à l'asynchronisme des systèmes (*cf.* section 2.1.2, difficulté 1). Les techniques d'analyse classiques utilisent parfois un *algorithme à tranches de temps* pour découper l'exécution des agents [Joumaa et al., 2009]. Cependant, l'ensemble des tranches ainsi récupérées ne sont pas toujours cohérentes dans la mesure où leurs dates de début et de fin ne coïncident pas nécessairement.

Cette section a pour but de définir une approche similaire en conservant la cohérence temporelle de l'exécution lors de l'observation macroscopique. Pour cela, elle fait appelle à la notion de *coupe temporelle cohérente* définie dans la section 3.1.5.

Agréger des coupes cohérentes

Une *observation temporelle cohérente* transforme l'exécution d'un SMA en agrégeant cette exécution entre deux coupes cohérentes c_0 et c_1 telles que c_0 précède c_1 . Ces coupes sont nommées *coupes cohérentes agrégées* et leur ensemble est nommé *agrégat de coupes*. Elles sont remplacées lors de l'observation par une unique coupe, nommée *coupe observée*, et les événements entre les coupes agrégées sont supprimés. Le *temps logique global* du SMA est ainsi modifié.

L'observation temporelle pose deux problèmes : comment définir l'*histoire* de la coupe observée et les *phases* qu'elle divise ?

Comme précédemment, ce modèle présente l'observation d'une unique agrégation de coupes cohérentes. Il peut être étendu à de multiples agrégations par partition de l'exécution ou même par superposition des agrégats (*cf.* perspectives présentées dans la section 5.2.2). De plus, les observations spatiales et observations temporelles sont commutatives et associatives. L'ordre de ces transformations importe peu lorsqu'on observe un SMA.

Exemple par analogie

Lors d'une observation temporelle, le regard de l'observateur ne porte plus distinctement sur les événements situés entre deux coupes cohérentes. Il ne voit plus les détails de l'histoire du système sur cet intervalle. A la place, il observe un ensemble d'événements simultanés.

Lorsqu'un historien analyse le déroulement d'une bataille, par exemple la bataille de Waterloo, il décompose celle-ci en une séquence d'événements importants ou décisifs (attaque du premier corps, charge de la cavalerie, prise d'un point stratégique, *etc.*). Le travail d'analyse consiste alors à expliciter les liens de causalité entre ces différents événements. L'historien dira : les premiers tirs de canons ont permis aux quatre divisions d'infanterie de progresser, l'attaque du premier corps a incité l'ennemi à charger, la riposte de la cavalerie alliée a permis de conserver la position des canons, *etc.*

Mais lorsque l'historien tente d'analyser les causes de la fin de la France napoléonienne, ces événements précis n'ont plus la même pertinence. La bataille de Waterloo apparaît alors comme un événement unique, ponctuel, dont les détails chronologiques importent peu. Ce qui compte alors, c'est par exemple l'issue de la bataille. Ainsi, suivant ce que l'historien cherche à analyser, il considérera certains événements comme un tout cohérent dont il n'est pas nécessaire de préciser la structure interne.

Définir une histoire

Soit un agrégat C composé des deux coupes cohérentes agrégées c_0 et c_1 dans un SMA S_r . Soit c_2 la coupe observée à partir de cet agrégat.

- L'*intervalle* des coupes c_0 et c_1 est l'intersection du futur de c_0 et du passé de c_1 . On note $inter_{(c_0, c_1)}(S) = Futur_{c_1}(S) \cap Passé_{c_0}(S)$.
- L'*histoire de l'agrégat* est $Hist(C) = Présent_{c_0}(S) \cup inter_{(c_0, c_1)}(S) \cup Présent_{c_1}(S)$.

L'observation temporelle consiste à remplacer l'histoire de l'agrégat par les événements de la coupe observée, sans modifier la causalité potentielle entre les événements du système réel. Elle inclut donc un morphisme d'histoires f de $Hist(S_r)$ vers $Hist(S_o)$ défini tel que :

- les événements de l'histoire du système réel qui ne font pas partie de l'histoire de l'agrégat sont inchangés par le morphisme.

$$\forall e \in Hist(S_r), e \notin Hist(A) \longrightarrow f(e) = e$$

- les événements de l'histoire de l'agrégat sont associés à l'événement de la coupe c_2 correspondant à leur position dans l'espace des agents.

$$\forall e \in Hist(A) \text{ tel que } pos(e) = a, \exists ! e_c \in c_2 \text{ tel que } pos(e_c) = a \text{ et } f(e) = e_c$$

Redéfinir les phases

L'ensemble des phases des agents du SMA est également modifié par l'observation temporelle. En effet, elles doivent être redéfinies pour rester cohérentes avec le nouveau temps logique global du système. Pour chaque phase du SMA réel, quatre cas sont possibles :

- La phase est *interne* à l'agrégat : elle commence après c_0 et termine avant c_1 .
Dans ce cas, la phase n'apparaît plus dans le système observé car son événement de début et son événement de fin ont la même image selon le morphisme d'histoire. Elle est donc supprimée de l'ensemble des phases du processus à laquelle elle appartient.

- La phase est *externe* à l'agrégat : soit elle commence et termine avant c_0 , soit elle commence et termine après c_1 .
Dans ce cas, l'observation temporelle ne modifie pas cette phase. Celle-ci reste alors inchangée.
- La phase *traverse* l'agrégat : elle commence avant c_0 et termine après c_1 .
Dans ce cas également, l'observation temporelle ne modifie pas la phase qui reste inchangée.
- L'interaction est *mixte* : soit elle commence après c_0 et termine entre c_0 et c_1 , soit elle commence entre c_0 et c_1 et termine après c_1 .
Dans ce cas, le morphisme d'histoires utilisé pour l'agrégation permet de redéfinir l'événement de phase situé entre c_0 et c_1 . Cet événement est remplacé par son image, c'est-à-dire l'événement de la coupe observée situé sur le même agent.

Exemple

La figure 3.9 représente à gauche un SMA réel S_r et deux coupes cohérentes c_0 et c_1 telles que c_0 précède c_1 . On veut agréger ces deux coupes. On note $C = \{c_0, c_1\}$ cet agrégat et c_2 la coupe observée. Le diagramme à droite représente le SMA S_o , observé à partir de S_r .

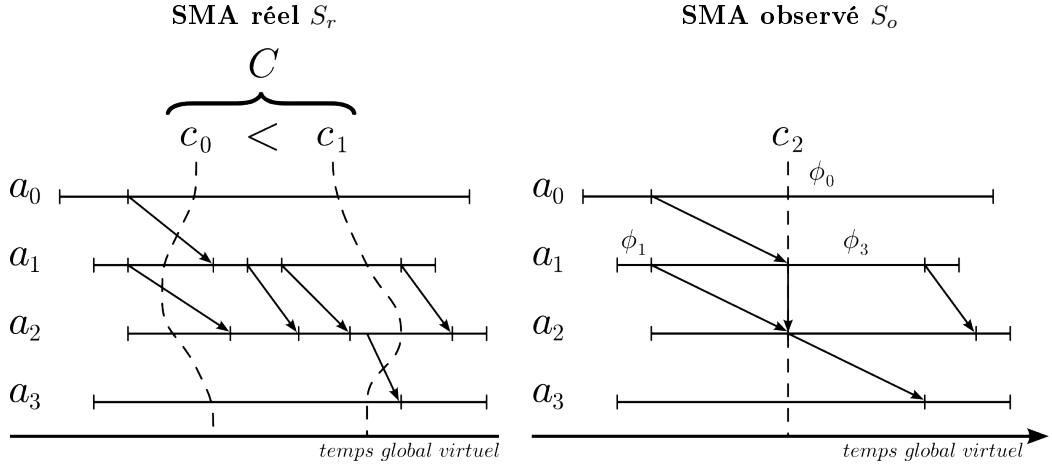


FIGURE 3.9 – Agrégation de coupes cohérentes

On remarque que :

- La phase ϕ_2 , interne à A dans S_r , n'apparaît pas dans S_o .
- La phase ϕ_1 , externe à A dans S_r , est restée inchangée lors de l'observation.
- La phase ϕ_0 , traversant A dans S_r , est restée inchangée lors de l'observation.
- La phase ϕ_3 , mixte dans S_r , commence sur la coupe c_2 (observée dans S_o).

Conservation de la temporalité

En plus de conserver la *causalité temporelle* et *potentielle* du système, l'observation temporelle garantit les deux inclusions suivantes $\forall e \in \text{Hist}(S_r)$:

- $\text{Passé}_{f(e)}(S_o) \subset f(\text{Passé}_e(S_r))$
- $\text{Futur}_{f(e)}(S_o) \subset f(\text{Futur}_e(S_r))$

Les preuves sont similaires à celles faites pour la conservation de la causalité par l'observation spatiale (*cf.* section précédente).

L'observation temporelle diminue ainsi le passé et le futur des événements du système. Ceci est dû au fait que les événements dans l'intervalle de l'agrégat sont supprimés.

Bilan

Pour conclure, l'observation temporelle consiste à :

- remplacer les deux coupes cohérentes agrégées par une coupe observée,
- supprimer les phases internes à l'agrégat,
- redéfinir les phases commençant ou terminant sur la coupe observée.

3.4 Modèle de sonde macroscopique

Dans un SMA, une sonde est un dispositif qui interagit localement avec un agent afin de collecter une information. Elle effectue une observation des caractéristiques de cet agent.

Une *sonde macroscopique* est un dispositif qui interagit globalement avec un système et qui effectue une *observation macroscopique* de ce système (telle que définie dans la section précédente) sans procéder à l'observation complète de l'exécution des agents en présence. Une telle sonde n'a pas besoin, pour fournir une information globale, d'avoir accès à la totalité des événements survenant dans un espace donné durant un temps donné. Elle donne *naturellement* la structure globale de cet espace et la structure globale de son exécution.

Cette section présente deux exemples théoriques de sondes macroscopiques. La première permet d'effectuer une observation spatiale du système et la seconde une observation temporelle.

3.4.1 Sonde spatiale

L'implémentation de l'observation spatiale d'un système peut être réalisée par le biais d'un appareil agrégeant naturellement le comportement externe d'un ensemble d'agents. Dans le cas des SMA, l'environnement peut aider à la conception d'interfaces de communication globales. Ces interfaces centralisent les interactions entre un ensemble d'agents et le reste du système. Ne sont alors retenues que les interactions *mixtes* à l'agrégat (*cf.* section 3.3.2).

De telles interfaces peuvent être modélisées sous la forme d'agrégation d'artefacts. Un *artefact* est une entité réactive utilisée pour modéliser l'environnement des SMA (*cf.* section 3.2.2 et [Omicini et al., 2008]). Elle modélise par exemple un *vecteur d'interaction local* entre deux agents : un canal de communication, une boîte aux lettres, des phéromones attractives ou répulsives (comme utilisées dans la section 4.3.2), *etc.* L'agrégation pertinente de ces artefacts permet d'observer l'activité du système à travers un *vecteur d'interaction global*.

Cette méthode peut être utilisée pour modéliser les sondes macroscopiques telles qu'elles ont été simulées dans le cadre de l'expérience présentée dans la section 4.3.1. Il s'agit d'analyser l'activité d'une grille informatique par le biais de sondes spatiales observant *naturellement* le

comportement d'ensembles de processus (machines) ou d'ensembles de machines (*clusters*). Le réseau de communication entre les processus, les machines et les *clusters* est alors construit de manière à centraliser les messages échangés dans des canaux de communication globaux (inter-machines ou inter-*clusters*). Ces canaux sont ainsi des vecteurs d'interaction globaux sur lesquels il est possible d'implanter des sondes macroscopiques.

Cette approche pour la conception de sondes macroscopiques spatiales n'est valable que si le nombre d'artefacts utilisés par le SMA est bien moins important que le nombre de ses agents. Dans le cas contraire, cette solution ne supporte pas le passage à l'échelle.

3.4.2 Sonde temporelle

L'observation temporelle de l'exécution d'un SMA peut être réalisée en adaptant les algorithmes distribués développés dans [Chandy and Lamport, 1985] et [Mattern, 1989]. Ces travaux ont été introduits dans la section 3.1.1.

Le *snapshot algorithm* [Chandy and Lamport, 1985] permet de connaître l'état global d'un système distribué asynchrone. Il utilise un protocole de collection des états locaux dont l'initialisation est distribuée dans le système. Cela revient à récupérer l'état global du système selon une coupe cohérente (au sens de [Mattern, 1989]). Cet algorithme peut être adapté pour initialiser un algorithme distribué quelconque. Un second *snapshot* détermine la fin de l'algorithme.

La notion de *temps vectoriel* [Mattern, 1989] permet de synchroniser les temps locaux des processus par le biais des messages échangés. Un processus a une connaissance de la temporalité des autres processus en fonction des messages échangés dans le passé. Par transitivité, le processus a une connaissance de son *passé global*.

L'utilisation de cette notion de temps vectoriel permet de retrouver les liens de causalité entre les processus ou entre les agents. Ainsi, un agent a été potentiellement influencé par un autre agent si il a une connaissance de son temps local.

L'observation temporelle peut être réalisée entre deux coupes cohérentes effectuées par deux *snapshot algorithms* espacés dans le temps. En comparant les temps vectoriels de chaque agent avant et après ces coupes, il est possible de déterminer quels agents ont été potentiellement influencés par d'autres entre ces deux coupes. Les interactions responsables de ces relations de causalités ne sont pas connues et le détail de l'exécution entre les deux coupes est inaccessible. Elle est agrégée par cette méthode d'observation macroscopique temporelle.

3.4.3 Composition de sondes

Il est possible de composer les modèles de sondes macroscopiques. Par exemple, la réalisation d'un *snapshot* sur des interfaces globales plutôt que sur des agents permet de réduire le coût de l'observation temporelle décrite précédemment. On réalise ainsi « l'observation temporelle d'une observation spatiale ».

L'étude de la commutativité et de l'associativité des observations issues de sondes macroscopiques est envisagée en perspective de ce projet dans la section 5.2.2.

Chapitre 4

Expérimentations et évaluation

4.1 Présentation

4.1.1 Protocole d'évaluation

Ce chapitre est consacré à l'évaluation des modèles d'observation macroscopique qui ont été formalisés dans le chapitre précédent. Elle consiste à simuler le résultat de sondes macroscopiques pour observer l'exécution de systèmes multi-agents (SMA) eux-mêmes simulés. Le protocole d'évaluation est le suivant.

- Définition de critères d'évaluation.
- Choix d'une plate-forme pour la simulation des SMA.
- Implémentation générique des modèles d'observation macroscopique sur la plate-forme retenue. Ces modèles permettent de simuler les informations collectées par des sondes macroscopiques.
- Choix de modèles de SMA et implémentation.
- Choix de règles d'observation en fonction des modèles de SMA retenus et implémentation concrète d'observations macroscopiques à partir des modèles génériques.
- Visualisation des données recueillies par les sondes macroscopiques ainsi simulées.
- Interprétation des résultats et évaluation selon les critères prédéfinis.

4.1.2 Critères d'évaluation

L'utilisation de sondes macroscopique pour l'analyse des SMA a été défendue dans la section 2.2.4 en trois points : *traitement de la distribution*, *détection de l'émergence* et *coût de l'observation*. Les critères pour l'évaluation de cette approche reprennent deux d'entre-eux.

1. **Détection de l'émergence.** La visualisation des résultats obtenus par la simulation de sondes macroscopiques doit permettre la détection de phénomènes émergents, sans requérir d'étape supplémentaire d'analyse des données collectées.
2. **Coût de l'observation.** Le coût de l'observation macroscopique doit être inférieur au coût de l'observation microscopique permettant de détecter le même phénomène émergent. Dans les expériences qui suivent, le coût de l'observation est égale au *nombre d'interactions collectées par les sondes*. Cette valeur augmente avec la taille du SMA et avec la durée de l'exécution. Elle est donc adéquate pour mesurer la complexité des observations.

En ce qui concerne le caractère *naturellement* distribué de l’observation macroscopique (**traitement de la distribution**, cf. section 2.2.4), ce critère d’évaluation ne peut être pris en compte ici. En effet, par souci de simplicité, les expériences présentées dans ce chapitre sont effectuées sur une plate-forme de simulation des SMA. Celle-ci centralise et synchronise les données du système pour en contrôler l’exécution (cf. le rôle de la simulation dans la section 2.1.3). L’observation microscopique, tout comme l’observation macroscopique, peuvent être effectuées sans se préoccuper de ces difficultés et la seconde ne présente pas d’avantage sur la première dans ce cadre-là.

Il est nécessaire, pour compléter l’évaluation de l’approche défendue par ce projet, de mener des expériences similaires en *conditions réelles* (i.e. sur des SMA non-simulés). L’intérêt des sondes macroscopiques pour traiter la décentralisation des données et l’asynchronisme du système devrait alors être mis en évidence. Cette prochaine étape est exposée en perspective dans la section 5.2.2.

4.2 Plate-forme d’expérimentation

4.2.1 Contribution

L’objectif à long terme de ce projet *ne consiste pas* à développer des outils pour une plate-forme de simulation SMA. En effet, l’utilisation de plates-formes de simulation pour l’analyse de systèmes de grande taille a été critiquée dans la section 2.1.3. Cependant, l’implémentation des modèles d’observation macroscopique sur de telles plates-formes constitue une contribution *à court terme* de ce projet. Cette section explique en quoi les outils que nous avons développés peuvent aider les utilisateurs de ces plates-formes.

Selon [Railsback et al., 2006], les plates-formes de simulation orientées SMA présentent de manière générale certaines lacunes. Les problèmes auxquels ce projet contribue à apporter des éléments de réponse sont les suivants.

- Il existe des lacunes dans les outils d’analyse proposés par les plates-formes. « Tools for scientific analysis are needed. » [Railsback et al., 2006] Les outils disponibles pour la modélisation des SMA sont souvent présents et efficaces, mais ceux qui servent à l’analyse de leurs simulations font défaut. Dans la mesure où ce projet propose des modèles d’observation destinés à l’analyse de l’émergence, il répond directement à ce besoin (cf. également les recommandations de [Gilbert and Bankes, 2002]).
- Le second problème concerne plus particulièrement l’analyse de la causalité dans les SMA. « Understanding causality is an unfulfilled need. » [Railsback et al., 2006] Il s’agit de comprendre les causes de résultats inattendus, pour savoir si ceux-ci sont utiles à l’analyse des SMA simulés ou alors liés à un biais de modélisation [Bonneaud et al., 2009]. Les sondes et *debuggers* disponibles actuellement sur les plates-formes sont utiles, mais uniquement pour l’observation de petites parties du système [Railsback et al., 2006]. Ce projet tente de pallier directement ce problème en effectuant le passage à l’échelle de l’analyse des structures causales (cf. également [Ropella et al., 2002] pour ce problème).
- La dernière recommandation formulée dans [Railsback et al., 2006] est la suivante. « Make sure the framework accomodates complex, multi-level models. » L’observation macroscopique permet justement de prendre en considération la notion de niveaux dans la structure des SMA (niveaux microscopiques, mésoscopiques et macroscopiques). Elle traite de la complexité de ceux-ci par la simplification des structures en fonction du niveau de recul.

4.2.2 MASON

MASON est une application Java composée d'un moteur de simulation et de bibliothèques de visualisation développées par des enseignants-chercheurs de l'université George Mason. Elle est conçue pour faire de la simulation multi-agents et a été choisie pour les raisons suivantes.

- Elle est orientée vers la simulation de systèmes de type *swarm* (essaims). Elle est donc conçue pour gérer jusqu'au million d'entités (*cf.* [Luke et al., 2005]) et simuler des SMA de très grande taille.
- Le méta-modèle de SMA qu'elle utilise est minimal. Il est donc très facile pour un programmeur expérimenté d'y implémenter un méta-modèle plus spécifique.
- Elle est composée de deux couches entièrement indépendantes : la couche *modèle* et la couche *visualisation*. Cette caractéristique est intéressante pour l'ajout d'outils correspondant aux différentes étapes de l'analyse (*cf.* section 2.2.1). La couche *modèle* sert à l'implémentation des modèles d'observation macroscopique, la seconde couche à l'ajout de méthodes de visualisation (*cf.* section 5.2.3).

4.3 Applications

Plusieurs modèles de SMA ont été envisagés pour l'évaluation des modèles d'observation macroscopique. Un *modèle de SMA* est la description abstraite de l'architecture de ses agents et leurs comportements. L'instanciation d'un tel modèle produit une application multi-agent qui peut être exécutée, observée et analysée. Cette section présente les deux modèles retenus.

4.3.1 Grille informatique

Puisque le méta-modèle de SMA qui a été développé dans le cadre de ce projet est fondé sur la généralisation d'un modèle de système distribué, le premier système qui a été simulé est une *grille informatique* constituée de processus échangeant des messages.

Plusieurs travaux dont ce projet s'inspire ont rapport à l'analyse de telles grilles. C'est par exemple le cas de *Pajé* [Stein, 1999] et de *Triva* [Schnorr et al., 2010], environnements de visualisation décrits respectivement dans les sections 2.1.2 et 5.2.3. Cependant, le modèle de grille présenté ici est relativement simple. Les résultats de l'analyse ne peuvent donc pas être *directement* comparés à ceux des travaux cités.

Modèle de grille

Une grille informatique est une infrastructure virtuelle constituée d'un ensemble de composants matériels ou logiciels distribués et autonomes. Il peut exister un certain nombre de relations logiques entre ces composants. Dans le cas présent, les composants sont des processus (les agents du modèle) et les relations logiques des messages échangés (les interactions du modèle).

Organisation L'ensemble des processus est partitionné en machines et l'ensemble des machines en *clusters* (ou grappes de serveurs).

Environnement L'environnement de la grille est la structure physique qui soutient les processus, leurs moyens de communication et l'organisation de la grille. Il s'agit de l'environnement d'exécution de ses composants et de l'ensemble des canaux de communication entre les processus, les machines et les *clusters*. Ces canaux peuvent par exemple être modélisés sous la forme d'artefacts réactifs [Omicini et al., 2008].

Comportement A chaque pas de la simulation, les processus émettent un message avec une probabilité de 10 %. Le destinataire du message est choisi aléatoirement dans la même machine que le processus émetteur (70 % des cas), dans le même *cluster* (20 % des cas) ou dans toute la grille (10 % des cas). Les messages sont ensuite transmis par les canaux de communication.

A chaque pas de la simulation, si au moins un message est en cours de transmission vers un processus, celui-ci récupère un message avec une probabilité de 30 %. Le message récupéré est choisi aléatoirement parmi tous les messages actuellement en transit vers le processus. Enfin, il y a une interaction entre deux processus dès lors que le message émis par le processus émetteur a été reçu par le processus récepteur.

Émergence Deux machines sont choisies aléatoirement dans la grille. A chaque pas de la simulation, chaque processus de la première machine émet un message vers un processus de la seconde machine avec une probabilité de 30 %.

Ce comportement local est la cause d'une saturation globale des canaux de communication entre les deux machines. Le but de l'analyse est de détecter cette saturation.

Remarque. Les chiffres utilisés dans ce modèle pour décrire le comportement des processus ont été choisis à titre d'exemple. Nous ne souhaitons pas ici modéliser de manière précise une grille informatique, mais illustrer efficacement notre travail. Ces chiffres ont donc été ajustés de manière à définir une application simple et claire pour le lecteur, telle que les comportements locaux et les phénomènes émergents soient le plus intelligible possible.

Sondes utilisées

Nous observons la grille à l'aide de plusieurs ensembles de sondes simulées.

Observation microscopique Une sonde est utilisée pour chaque processus. Il est donc possible d'observer entièrement l'activité du système dans le moindre détail.

Observation spatiale au niveau des machines Des sondes macroscopiques sont mises en place au niveau des machines. Elles ne détectent que les interactions entre un processus de la machine et un processus extérieur. Il est donc impossible d'observer l'activité au sein des machines.

S'il fallait implémenter ces sondes et non en simuler le résultat, elles pourraient être construites comme des appareils observant l'environnement autour des machines. La structure physique des grilles comporte en effet des interfaces matérielles ou logicielles entre les différents niveaux d'organisation. Elles centralisent les communications et leur observation permet d'observer l'activité globale d'une machine ou d'un *cluster*. De telles interfaces appartiennent à l'environnement du système. Elles peuvent être modélisées par des artefacts dont l'activité serait surveillée par des sondes macroscopiques [Omicini et al., 2008].

Observation spatiale au niveau des *clusters* De la même manière que précédemment, des sondes macroscopiques sont mises en place au niveau des *clusters*. Leur activité interne ne peut être observée.

Visualisation de l'activité

La grille peut être représentée par un graphe dynamique visualisé en deux dimensions par des points et des lignes. Dans le cas d'une observation microscopique, les points (ou nœuds) du graphe sont les processus de la grille et les lignes (ou arêtes) sont les messages en attente entre les processus à un instant donné. Ce graphe évolue au cours de la simulation, à mesure que des messages sont émis ou reçus.

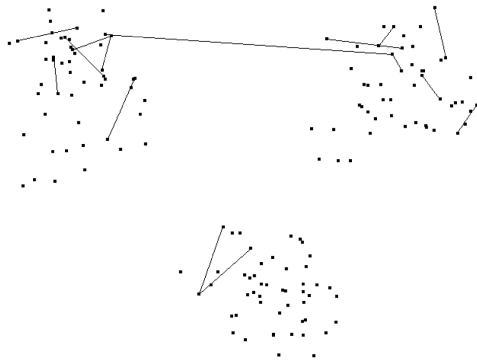


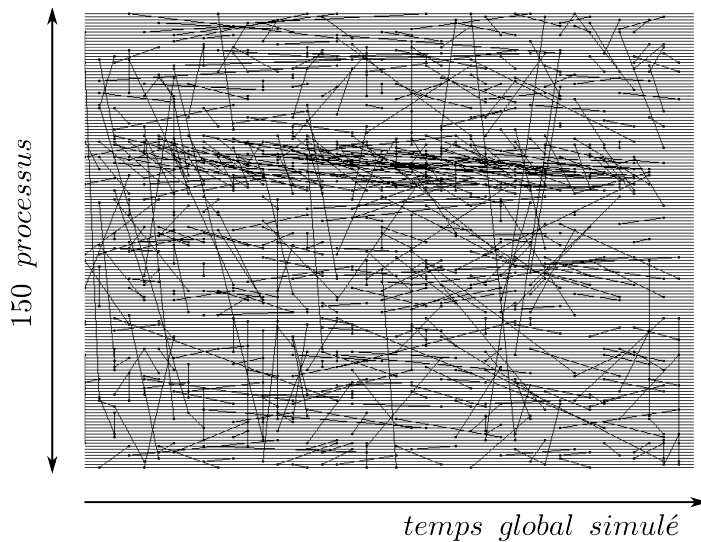
FIGURE 4.1 – Graphe représentant une grille à un instant donné

Cette figure donne la représentation d'une grille de 3 *clusters* (nuages de points) composés chacun de 5 machines, composées à leur tour de 10 processus chacune. 15 messages sont actuellement en attente.

Ces chiffres relativement petits (il y a en tout 150 processus) ont été choisis pour simplifier l'exemple. Les mêmes phénomènes sont observables avec une grille beaucoup plus grande.

L'activité de la grille peut également être représentée par un diagramme d'exécution (*cf.* section 3.2.4). Suivant la méthode d'observation utilisée, le diagramme produit ne sera pas le même. Les digrammes présentés à la page 43 ont été obtenus pour une simulation de 40 pas.

- Dans le cas d'une observation microscopique, toute l'activité du système est représentée (*cf.* figure 4.2).
- Dans le cas d'une observation au niveau des machines, seule leur activité externe est représentée (*cf.* figure 4.3).
- Il en est de même dans le cas d'une observation au niveau des *clusters* (*cf.* figure 4.4).



Dans le cas d'une observation microscopique, les processus responsables de la saturation des canaux de transmission peuvent être identifiés (zone saturée). Les machines globalement responsables peuvent également être retrouvées par analyse du diagramme. Le coût de l'observation est de 537 interactions observées.

FIGURE 4.2 – Diagramme d'exécution résultant d'une observation microscopique

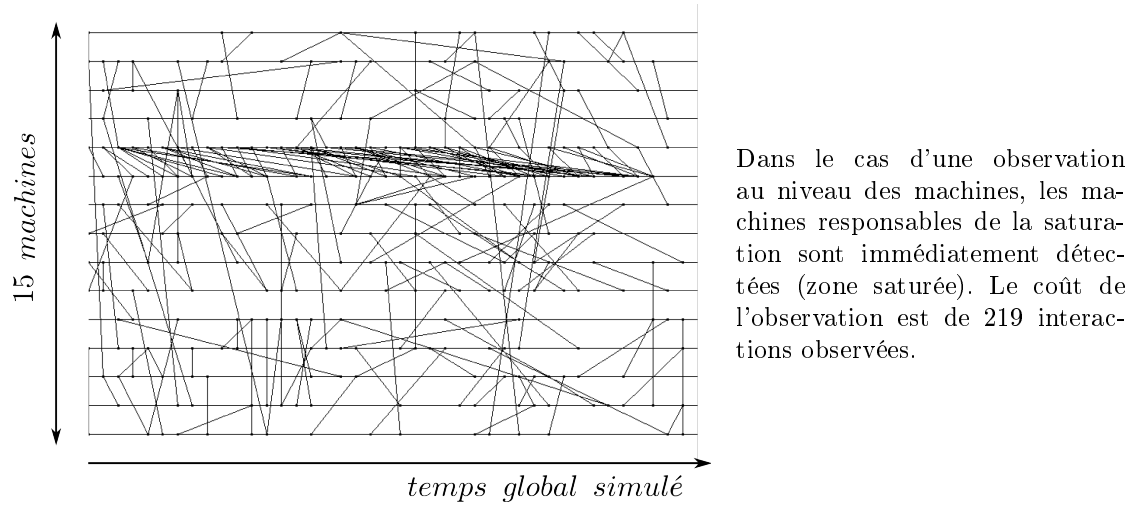
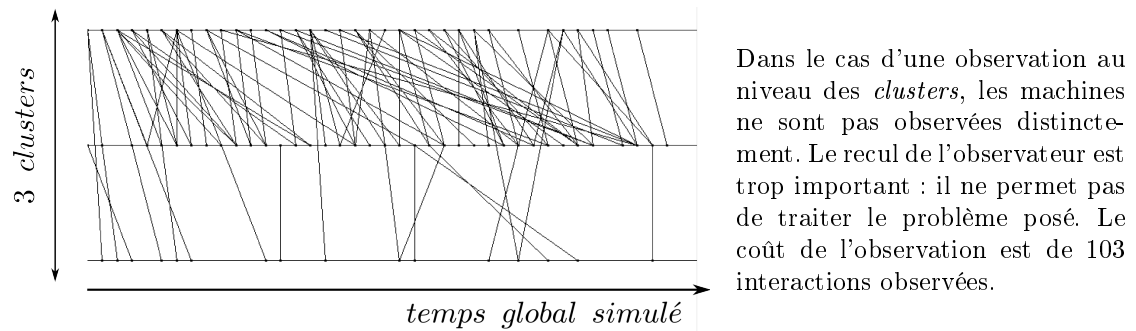


FIGURE 4.3 – Diagramme d'exécution résultant d'une observation au niveau des machines

FIGURE 4.4 – Diagramme d'exécution résultant d'une observation au niveau des *clusters*

Bilan de l'expérience

L'observation microscopique et celle au niveau des machines permettent toutes les deux de détecter le phénomène émergent et sa position (canaux saturés et machines responsables). Cependant, le coût de l'observation au niveau des machines représente 40,8 % du coût de l'observation microscopique. **Il est donc préférable de placer le regard de l'observateur au niveau des machines pour traiter à moindre coût ce problème particulier de saturation globale des canaux.**

L'observation au niveau des *clusters* ne permet pas de détecter la position exacte du phénomène de saturation. Cette expérience confirme que l'utilisation de sondes macroscopiques doit être réfléchi en fonction de ce que l'on souhaite observer et le niveau d'observation choisi de manière adéquat. Cela est dû à l'existence de niveaux dans les phénomènes émergents [Deguet et al., 2006].

La conception de l'observation macroscopique, que nous avons appelée « *design* de l'observateur » dans la section 2.2.4, doit prendre en compte cette notion de niveau. Par exemple, l'émergence est définie dans [Bonabeau and Dessalles, 1997] comme interne à des appareils de détection *hiérarchisés selon des niveaux d'observation*. Une hiérarchie des sondes est également introduite dans ce projet (*cf.* section 3.4.3) par la notion de *composition*. Cette idée est mise en valeur dans la section 5.2.2, en perspective de ce projet, pour accéder aux différents niveaux d'émergence à moindre coût.

4.3.2 Colonie de fourmis

Choix

La seconde application retenue pour l'évaluation est le modèle de la *colonie de fourmis*. Ce modèle et les différents phénomènes émergents qu'il voit apparaître sont présentés dans [Drogoul et al., 1995]. *MASON* propose un modèle simple de colonie sous le nom *AntsForage*. Nous avons exécuté ce modèle et observé le comportement du SMA sous différents points de vues.

Modèle de colonie

Une colonie de fourmis est constituée d'un foyer et de plusieurs individus (les fourmis) qui évoluent dans un environnement pour trouver et ramener de la nourriture. Les fourmis communiquent par le biais de phéromones déposées dans l'environnement.

Environnement L'environnement de la colonie comporte donc : un foyer, des obstacles infranchissables et de la nourriture. Il est également rempli de phéromones au fur et à mesure que les fourmis en parcourent l'espace.

Comportement Tant qu'elle n'a pas trouvé de nourriture, une fourmi se déplace aléatoirement dans l'environnement en zigzaguant. Quand elle rencontre un obstacle, elle change de direction.

Lorsqu'elle trouve de la nourriture, une fourmi en ramène une partie au foyer. Sur le chemin du retour, elle dépose des phéromones indiquant l'existence de nourriture. Les autres fourmis sont naturellement attirées par ces phéromones. Leur puissance attractive est d'autant plus forte qu'elles sont proche de la nourriture, si bien que la direction à suivre est marquée par une augmentation de l'attraction.

Le mécanisme utilisé par les fourmis pour retrouver le foyer nécessite un second type de phéromones. Néanmoins, la compréhension de ce mécanisme n'est pas nécessaire à la compréhension de l'expérience que nous avons réalisée.

Émergence Le dépôt local de phéromones provoque l'apparition de *pistes* menant du foyer aux différents endroits où les fourmis peuvent trouver de la nourriture. Une interaction est caractérisée par le dépôt de phéromones par une fourmi (événement d'émission) et l'attraction d'une autre fourmi par ces mêmes phéromones (événement de réception). Les interactions à la base de l'apparition des pistes (phénomène qualifié d'émergent) sont donc véhiculées par l'environnement et peuvent être très espacées de le temps.

Sondes utilisées

Nous observons le comportement de la colonie à l'aide de plusieurs ensembles de sondes.

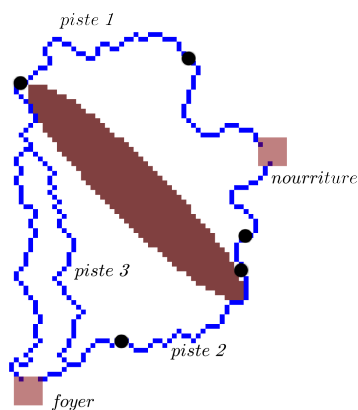
Observation microscopique Toutes les interactions/attractions sont observées.

Observation temporelle à l'échelle de la création des pistes Le comportement de la colonie est divisée en plusieurs périodes. Chaque début de période correspond à l'élaboration d'une nouvelle piste par une fourmi. Celle-ci est détectée par le dépôt des premières phéromones. Ces périodes sont agrégées par les sondes macroscopiques pour présenter le comportement de la colonie en fonction de ces étapes charnières.

Observation temporelle à l'échelle de la collecte de nourriture Le comportement de la colonie est divisée en plusieurs périodes. Chaque début de période correspond à la collecte de nourriture par une fourmi. Celle-ci peut être détectée par une sonde au niveau des réserves de nourriture, observant les fourmis qui y passent. Ces périodes sont agrégées par les sondes macroscopiques pour présenter le comportement de la colonie en fonction du va-et-vient des fourmis.

Visualisation de l'activité

La colonie et son environnement peuvent être représentés à un instant donné par un dessin en deux dimensions.



La figure à gauche présente une colonie de 5 fourmis (ronds noirs). Le grand ovale sombre au milieu est un obstacle. Le carré en haut à droite représente une réserve de *nourriture* et celui en bas à gauche représente le *foyer*. Les chemins sombres entre les deux sont des pistes de phéromones. Il y en a trois.

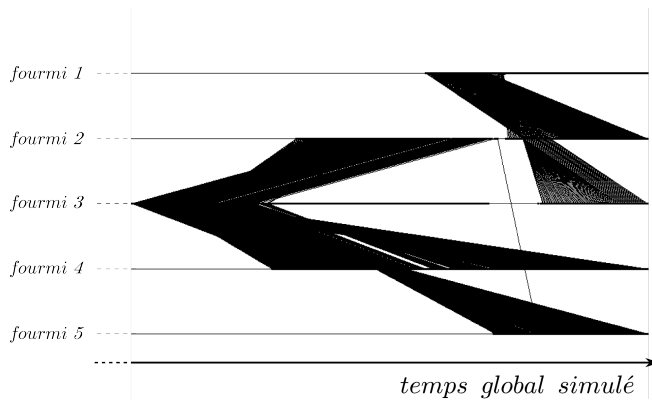
Comme précédemment, nous avons limité le nombre de fourmis à 5 pour conserver la lisibilité de l'exemple. Ce nombre suffit pour faire émerger des pistes de phéromones. Cependant, il est possible qu'une colonie de plus grande taille voit émerger des phénomènes plus complexes.

FIGURE 4.5 – Dessin représentant la colonie à un instant donné

L'activité de la colonie peut également être représentée par un diagramme d'exécution (*cf.* section 3.2.4). Suivant la méthode d'observation utilisées, le diagramme produit ne sera pas le même. Les digrammes présentés dans les pages suivantes ont été obtenus pour une simulation de 1780 pas.

- Dans le cas d'une observation microscopique, toute l'activité de la colonie est représentée (*cf.* figure 4.6).

- Dans le cas d’une observation à l’échelle de la création des pistes, le temps est découpé en périodes correspondant à la création de nouvelles pistes (cf. figure 4.7).
- Dans le cas d’une observation à l’échelle de la collecte de nourriture, le temps est découpé en périodes correspondant aux arrivées des fourmis à la réserve de nourriture (cf. figure 4.8).



Dans le cas d’une observation microscopique, le nombre des interactions est très élevé (il y en a ici 1309). En effet, les fourmis font des allers-retours incessants sur les pistes de phéromones, interagissant de très nombreuses fois avec la fourmi ayant créée la piste à l’origine. Le diagramme est ainsi saturé et assez peu intelligible.

On distingue cependant quelques influences importantes, notamment celle de la *fourmi 3* sur les *fourmis 2, 4 et 5* et celle de la *fourmi 1* sur les *fourmis 2 et 3*.

FIGURE 4.6 – Diagramme d’exécution résultant d’une observation microscopique

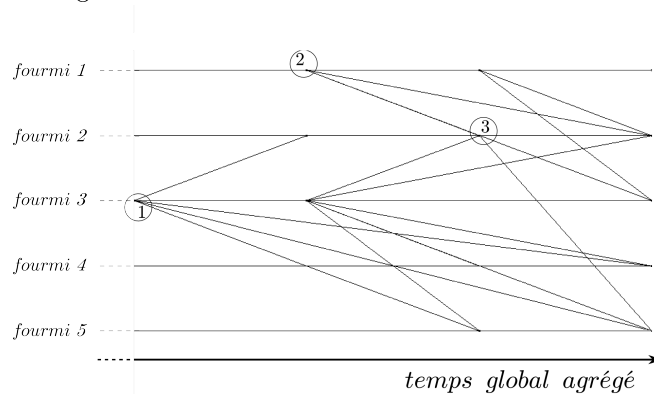


FIGURE 4.7 – Diagramme d’exécution résultant d’une observation à l’échelle de la création des pistes

L’observation à l’échelle de la création des pistes permet de comprendre rapidement comment de nouvelles pistes ont été créées. Le diagramme présente 32 interactions. Les événements doivent être interprétés de la manière suivante :

Événement 1 La *piste 1* a été créée en premier, par la *fourmi 3*. Elle a par la suite influencé le déplacement des *fourmis 2, 3, 4 et 5* (cf. interactions émises en 1).

Événement 2 La *piste 2* a ensuite été créée par la *fourmi 1*, indépendamment de la *piste 2* (aucune interaction reçue en 2). Elle a influencé le déplacement des *fourmis 1, 2 et 3*.

Événement 3 La *piste 3* a enfin été créée par la *fourmi 2*, à partir des *pistes 1 et 2* (cf. interactions reçues en 3).

La figure 4.5, présentant un dessin de la colonies et de ses pistes, confirme que la *piste 3* raccorde les *pistes 1 et 2*. Pour la créer, la *fourmi 2* a nécessairement interagi avec les fourmis responsables de ces pistes.

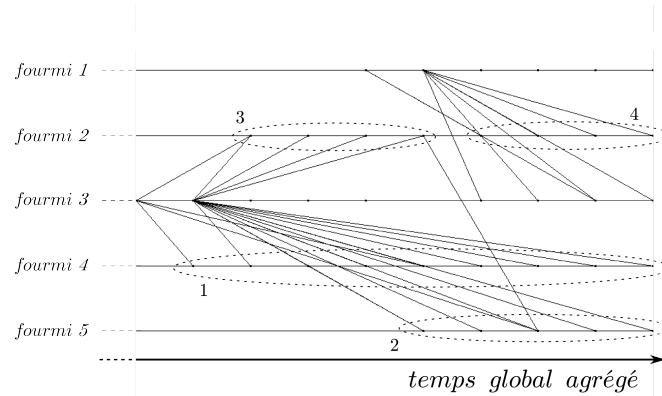


FIGURE 4.8 – Diagramme d'exécution résultant d'une observation à l'échelle de la collecte de nourriture

L'observation à l'échelle de la collecte de nourriture permet d'analyser le va-et-vient des fourmis. Le diagramme présente 54 interactions. Les ensembles d'événements doivent être interprétés de la manière suivante :

Ensembles 1 et 2 Les *fourmis 4 et 5* ont effectués tous leurs va-et-vient sur la piste créée par la *fourmi 3* (*piste 1*).

Ensemble 3 La *fourmi 2* a d'abord emprunté la *piste 1*.

Ensemble 4 Puis, elle a emprunté la piste créée par la *fourmi 1* (*piste 2*). Nous ne pouvons pas savoir si elle a également emprunté la piste qu'elle avait elle-même créée (*piste 3*) pendant cette seconde période. En effet, les interactions d'un fourmi avec elle-même, par le biais de phéromones, ne sont pas prises en compte par le modèle.

Nous aurions pu modéliser les phéromones par des artefacts, entités réactives indépendantes des fourmis qui les déposent, pour pallier ce problème (*cf.* section 3.2.2). Des phéromones bien choisies (une par piste) auraient donné un diagramme d'exécution plus complexe et plus complet.

Bilan

L'observation de l'activité de la colonie selon des échelles de temps macroscopiques permet de détecter et d'analyser l'apparition de comportements globaux (création de pistes et va-et-vient des fourmis). Le coût de ces observations est largement inférieur au coût de l'observation microscopique (qui par ailleurs ne présente pas *immédiatement* des données intelligibles pour identifier les phénomènes et en comprendre les causes). Le coût de l'observation temporelle à l'échelle de la création des pistes (*resp.* de la collecte de nourriture) représente 2,4 % (*resp.* 4,1 %) du coût de l'observation microscopique.

Il est donc préférable, lors de l'analyse de ces phénomènes, de recourir à des sondes macroscopiques bien disposées, notamment lorsque la taille de la colonie et le temps de simulation augmentent (à l'origine *MASON* propose de simuler une colonie de 1000 fourmis, sans limite sur le nombre de pas effectués par la simulation). De telles sondes peuvent être élaborées par l'observation de phéromones pertinentes, représentant la création d'une piste ou l'activité qui y est associée.

Chapitre 5

Bilan et perspectives

5.1 Bilan

Ce projet s'intéresse à l'analyse des phénomènes émergents apparaissant lors de l'exécution, pendant de longues périodes de temps, de systèmes multi-agents (SMA) de grande taille. Cette analyse repose sur l'étude de la structure causale de l'exécution, définissant les notions d'*espace* et de *temps* du système.

Le chapitre 2 fait l'état de l'art des méthodes d'analyses existantes dans le domaine des SMA et expose l'approche défendue par ce projet. Les approches classiques procèdent en trois étapes.

- L'étape de collecte de données consiste à observer localement le comportement des agents. Elle implémente ainsi une *observation microscopique* de l'exécution. Dans un second temps, il est nécessaire de synchroniser ces observations afin de maintenir leur cohérence globale.
- L'étape d'analyse des données permet de mettre en évidence les structures émergentes dissimulées dans les résultats de l'observation microscopique.
- La dernière étape consiste à visualiser le résultat de l'étape d'analyse des données pour en présenter les phénomènes émergents.

L'étape de collecte des données se heurte à deux difficultés : l'asynchronisme des systèmes et l'intrusion du processus d'observation. Ces difficultés augmentent avec la taille des systèmes et le coût de l'observation microscopique (en termes d'implantation et de temps de calcul) est trop élevé pour supporter le passage à l'échelle.

L'approche défendue par ce projet est fondée sur un changement de paradigme. Elle consiste à réaliser une *observation macroscopique* de l'exécution pour dépasser les limites rencontrées par les méthodes classiques. Cette observation donne une information globale sur le système et permet ainsi de détecter *directement* l'émergence. Elle traite l'asynchronisme des SMA de manière *naturelle* et elle s'avère moins coûteuse que l'observation microscopique. Pour toutes ces raisons, elle supporte mieux le passage à l'échelle.

Le chapitre 3 formalise la notion d'*observation macroscopique* dans le cadre des SMA.

Un modèle « minimal » de système distribué est à l'origine de ce travail. Celui-ci modélise, à partir des notions de *processus* et de *messages*, la structure causale de l'exécution. Il formalise ainsi les notions d'*espace logique* et de *temps logique* du système. Un méta-modèle de SMA est généralisé à partir de ce modèle de système distribué.

L'observation macroscopique est définie comme une transformation des SMA, modifiant la structure causale de leur exécution. Elle agrège l'espace et le temps des SMA observés. Elle

conserve cependant les relations de causalités globales pour présenter une image cohérente de l'exécution. Des exemples théoriques de sondes macroscopiques sont données à la fin du chapitre.

Le chapitre 4 présente des expériences que nous avons réalisées afin d'évaluer la pertinence de l'observation macroscopique pour l'analyse des SMA. Elles ont été menées sur une plate-forme de simulation (*MASON*). Les deux applications choisies sont des modèles de SMA relativement simples, mais qui suffisent à faire émerger des comportements globaux et à valider le modèle d'observation macroscopique. Il s'agit de modèles de *grille informatique* et de *colonie de fourmis*.

L'activité de ces SMA a été observée en simulant la collecte de données par des sondes microscopiques et par des sondes macroscopiques (spatiales et temporelles). Les résultats de ces expériences confirment l'efficacité de l'*observation macroscopique* pour détecter les phénomènes émergents et montrent, comme nous l'espérions, qu'elle est beaucoup moins coûteuse que l'*observation microscopique*. Elle est donc bien mieux adaptée à l'analyse des SMA à grande échelle.

Les autres avantages de l'observation macroscopique restent à mettre en évidence par la réalisation d'une expérience en *conditions réelles*. Une telle expérience, en prolongement de ce projet, permettrait de compléter l'évaluation du travail effectué. Plusieurs autres perspectives sont envisagées pour continuer les recherches concernant l'*observation macroscopique*.

5.2 Perspectives

5.2.1 A propos du modèle

Transformations sémantiques

Le modèle exposé dans la section 3.3 définit une observation macroscopique comme la *transformation formelle* d'un SMA. Elle modifie la structure causale de son exécution. Les différentes agrégations présentées peuvent également être formalisées à l'aide d'*algèbres d'agents* et d'*algèbres de coupes cohérentes*, à la manière des langages formels de type π -calcul [Milner et al., 1992].

Il est intéressant de compléter le modèle d'observation macroscopique par la notion de *transformation sémantique* pour prendre en compte les interprétations particulières que le domaine des SMA donne aux notions d'agent, d'interaction et d'émergence (interprétations qui fondent la spécificité du domaine). L'agrégation des agents et des coupes entraîne donc une agrégation de leur signification. On parle alors d'*émergence sémantique*, et non plus seulement d'*émergence structurelle*, détectée par un changement dans le langage de l'observateur pour désigner le phénomène émergent (cf. [M.R. Jean et al., 1997], [Kubík, 2003] et [Deguet et al., 2006]).

Piste. La conception d'un méta-modèle d'état interne sert à définir la notion d'*agrégation d'états* lors de la superposition des phases des agents agrégés (cf. section 3.3.2). Il peut être dérivé d'un méta-modèle d'agent préexistant, tel que l'architecture BDI (pour *Belief, Desire and Intention*) [Rao and Georgeff, 1991] ou le modèle d'architecture des agents *Tileword* [Pollack and Ringuette, 1990] (conçu pour l'évaluation de leur architecture interne). Une algèbre sur les termes de ce méta-modèle, spécifique au domaine des SMA, définit alors les *transformation sémantique* liée à l'agrégation de structures.

Il en est de même avec la conception d'un méta-modèle d'interaction, pour en définir la notion d'agrégation.

Réduction des systèmes

La réduction des SMA a été évoquée dans la section 2.2.3. Il s'agit de retrouver, par analyse des données collectées lors de l'observation macroscopique, le mode comportemental des agents du système. Il est impossible de connaître l'activité précise de ces agents sur la base d'information globale. Par contre, il est envisageable de réduire le SMA à des catégories de comportements locaux.

Piste. La classification des modes génériques d'interaction locale (entre agents), présentée dans [Lacomme et al., 2009], donne un bon point de départ pour définir les termes d'une réduction comportementale. Le résultat de la réduction du comportement observé globalement est une combinaison de ces termes. La mise en place d'une telle technique d'investigation est une perspective à long terme ouverte par ce projet. Elle est résumée par le diagramme de la figure 5.1.

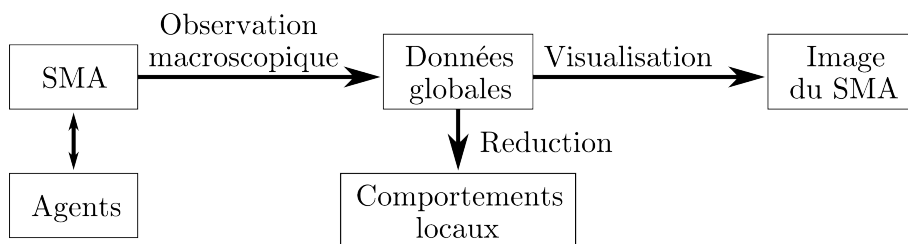


FIGURE 5.1 – Réduction des SMA

5.2.2 A propos des sondes

Les expériences effectuées dans le cadre de ce projet (*cf.* chapitre 4) ont consisté à simuler le résultat d'observations macroscopiques de SMA eux-mêmes simulés. Ce travail d'évaluation nécessite, pour être complétée, une expérience similaire en *conditions réelles*. L'étape suivante de ce projet est donc la conception et l'implémentation de sondes macroscopiques réelles.

Intrusion

Le problème de l'intrusion ou « effet de sonde » est à traiter avec attention. Il a été introduit précédemment comme une des difficultés majeures rencontrées par les techniques d'analyse de traces (*cf.* section 2.1.2, difficulté 2).

L'intrusion des sondes macroscopiques doit être qualifiée et, dans l'idéal, elle doit pouvoir être quantifiée. L'observateur doit être capable de connaître et de mesurer les effets de l'intrusion des sondes qu'il utilise. A terme, il doit pouvoir contrôler ces effets et les réduire.

Piste. Ce problème peut être abordé à partir des études élaborées dans des domaines proche des SMA. Par exemple, une technique de contrôle et de correction de l'intrusion est décrite dans [Chassin de Kergommeaux et al., 2001] en ce qui concerne le *monitoring* des systèmes distribués.

Système de sondes

L'utilisation simultanée de plusieurs sondes pose un certain nombre de problèmes qu'il faut résoudre afin de construire la notion cohérente de *système de sondes*.

- La *superposition de sondes* est brièvement évoquée dans la section 3.3. Elle correspond à l'utilisation simultanée de sondes macroscopiques dont l'espace observé (les agents agrégés) ou l'intervalle du temps de l'observation (les coupes cohérentes agrégées) sont d'intersection non-vide. Il est important de comprendre la sémantique de telles observations superposées. D'autres propriétés comme l'effet des interactions inter-sondes, leur rôle quant à l'intrusion, la commutativité des observations superposées, sont autant de points à étudier pour construire des systèmes de sondes supportant la superposition.
- La *composition de sondes* est introduite dans la section 3.4.3. Elle concerne la construction hiérarchique de sondes macroscopiques à partir de sondes plus « petites ». Un système de sondes peut alors être interprété comme une sonde de niveau supérieur. La construction de sondes par composition est un moyen de prendre du recul sur le système et de détecter de nouveaux niveaux d'émergence (cf. [Deguet et al., 2006]). Cependant, le coût de la sonde composée est au moins égal à la somme des coûts des sondes qu'elle intègre. La conception de *sondes non-composées* reproduisant le comportement de *sondes composées* permet de construire à moindre coût des sondes de niveau supérieur.
- La *synchronisation* (spatiale et temporelle) des sondes utilisées dans un système de sondes (qu'elles soient superposées, composées ou entièrement distinctes) est également un point important à traiter pour produire une observation cohérente.

Piste. Ces questions peuvent être formalisées à l'aide d'algèbres de sondes. Un vocabulaire définit l'ensemble des sondes que l'on peut construire et des règles syntaxiques définissent leur superposition, composition et synchronisation. Cette grammaire formelle, définissant les différents modes d'observation d'un système, permettrait d'étudier de manière abstraite les concepts importants en ce qui concerne l'observation macroscopique des SMA.

5.2.3 A propos de la visualisation

Les outils de visualisation développés pour les expériences de ce projet sont sommaires (cf. section 4.3). La conception d'un véritable environnement de visualisation, adapté aux possibilités de l'observation macroscopique, est la dernière perspective ouverte par ce projet.

Passage à l'échelle de la visualisation

La visualisation de l'exécution des SMA est traitée par *MAS-Pajé* (cf. section 2.1.2) sous la forme de diagrammes espace-temps (similaires aux *diagrammes d'exécution* utilisés dans les chapitres 3 et 4). Le passage à l'échelle de la visualisation est réalisé en adaptant la quantité des données à afficher lorsque la taille du système augmente [Stein, 1999]. Les outils développés dans le cadre de ce projet peuvent contribuer à améliorer de telles techniques de passage à l'échelle.

L'annexe B présente le sujet de stage ingénieur que nous avons proposé, dans le cadre de la collaboration entre les équipes MESCAL et MAGMA, à propos de ces perspectives. Celui-ci aura lieu pendant l'été 2010.

Piste. La technique *Treemap*, utilisée à l'origine pour la visualisation d'espaces mémoires sous forme hiérarchique [Shneiderman, 1992], a été adaptée à la visualisation de grilles informatiques par [Schnorr, 2009]. Cette technique fonctionne par agrégation des données collectées pour

adapter la finesse de la visualisation, se concentrer sur certaines zones ou observer l'ensemble de la grille de manière globale. La dynamique des processus observés est traitée par un *algorithme à tranches de temps*, partitionnant le temps de l'exécution avec plus ou moins de précision. Cette méthode a donné naissance à un environnement de visualisation pour l'analyse de performance des applications réparties : *Triva* [Schnorr et al., 2010].

La technique *Treemap* peut être adaptée à la représentation de diagrammes d'exécution. Les deux types d'agrégaions effectués par l'observation macroscopique (agrégaion d'agents et agrégaion de coupes) sont alors représentés par la composition de deux techniques *Treemap* (une pour l'espace, l'autre pour le temps) utilisant les deux dimensions de l'écran pour la visualisation.

MAS-Pajé peut donc être perfectionné par l'utilisation combinée des méthodes d'observation macroscopique et de la technique *Treemap*.

Interactivité

L'implémentation et le déploiement de systèmes de sondes (*cf.* section 5.2.2) permettent de définir plusieurs observations du même phénomène et plusieurs niveaux de recul. La visualisation des SMA peut avantageusement prendre en compte ces points de vue multiples sur l'exécution du système pour offrir à l'observateur un environnement interactif.

Piste. Pour chaque surface de l'exécution (définie par une portion de l'espace et un intervalle de temps) l'observateur peut accéder à une visualisation différente, en fonction des sondes macroscopiques couvrant cette surface (dans le cas de sondes superposées). L'observateur peut ainsi observer le même espace ou le même intervalle de temps selon différents points de vue.

La résolution des surfaces à explorer (définie par le nombre d'agents et le nombre de coupes) peut être ajustée en fonction des niveaux d'observation offertes par les sondes macroscopiques (dans le cas de sondes composées). L'observateur peut ainsi se rapprocher du système ou au contraire prendre du recul par rapport à lui.

Perspectives méthodologiques

L'implémentation des SMA est effectuée au niveau des agents et de leur comportement local. Le concepteur, qui vise l'émergence d'une solution globale engendrée par le système, doit être capable d'ajuster le modèle de ces agents et les différents paramètres en fonction du comportement globale effectivement arboré par le système. Cette technique de conception/développement en boucle a été annoncée comme un objectif à long terme de ce projet (*cf.* section 1.1.1). Le développement d'un environnement de visualisation pour l'analyse des comportements émergents des SMA contribue à la réalisation de cet objectif.

Bibliographie

- [Bogen, 2009] Bogen, J. (2009). Theory and Observation in Science. *Stanford Encyclopedia of Philosophy*.
- [Bonabeau and Dessalles, 1997] Bonabeau, E. and Dessalles, J.-L. (1997). Detection and Emergence. *Intellectica*, 25(2) :85–94.
- [Bonneaud et al., 2009] Bonneaud, S., Redou, P., Desmeulles, G., and Chevaillier, P. (2009). Biais computationnels dans les modèles de peuplements d’agents. In *17ème Journées Franco-phones sur les Systèmes Multi-Agents*. Cépaduès Editions.
- [Carnap and Gardner, 1966] Carnap, R. and Gardner, M. (1966). *Philosophical Foundations of Physics*. Basic Books.
- [Chaib-Draa et al., 2001] Chaib-Draa, B., Jarras, I., and Moulin, B. (2001). Systèmes multi-agents : principes généraux et applications. In *Principes et architecture des systèmes multi-agents*, pages 27–70. Hermes.
- [Chandy and Lamport, 1985] Chandy, K. M. and Lamport, L. (1985). Distributed Snapshots : Determining Global States of Distributed Systems. *ACM Transactions on Computer Systems*, 3 :63–75.
- [Chassin de Kergommeaux et al., 2001] Chassin de Kergommeaux, J., Maillet, E., and Vincent, J.-M. (2001). Monitoring Parallel Programs for Performance Tuning in Cluster Environments. *Parallel Program Development for Cluster Computing : Methodology, Tools and Integrated*, pages 131–150.
- [Dall’Asta et al., 2005] Dall’Asta, L., Alvarez-Hamelin, I., Barrat, A., Vázquez, A., and Vespignani, A. (2005). Traceroute-Like Exploration of Unknown Networks : A Statistical Analysis. *Lecture Notes in Computer Science*, 3405 :140–153.
- [de Rosnay, 1975] de Rosnay, J. (1975). *Le Macroscopie*. Éditions du Seuil, Paris.
- [Deguet et al., 2006] Deguet, J., Demazeau, Y., and Magnin, L. (2006). Elements about the Emergence Issue : A Survey of Emergence Definitions. In *European Conference on Complex Systems*, volume 3, pages 24–31.
- [Demazeau, 1995] Demazeau, Y. (1995). From Interactions to Collective Behavior in Agent-Based Systems. In *First European Conference on Cognitive Science*, pages 117–132.
- [Demazeau, 2001] Demazeau, Y. (2001). *VOYELLES*.
- [Demazeau and Müller, 1991] Demazeau, Y. and Müller, J.-P. (1991). From Reactive to Intentional Agents. *Decentralized Artificial Intelligence*, 2.
- [Demazeau and Rocha Costa, 1996] Demazeau, Y. and Rocha Costa, A. (1996). Populations and Organizations in Open Multi-Agent Systems. In *First National Symposium on Parallel and Distributed AI*.

- [Drogoul et al., 1995] Drogoul, A., Corbara, B., and Lal, S. (1995). Manta : New experimental results on the emergence of (artificial) ant societies. *Artificial Societies : The Computer Simulation of Social Life*, pages 190–211.
- [Drogoul et al., 2004] Drogoul, A., Ferrand, N., and Müller, J.-P. (2004). Émergence : l’articulation du local au global. In *Systèmes multi-agents*, pages 105–136. OFTA, Paris.
- [Ferber, 1995] Ferber, J. (1995). *Les Systèmes multi-agents : vers une intelligence collective*. InterEditions, Paris.
- [Ferber and Gutknecht, 1998] Ferber, J. and Gutknecht, O. (1998). A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. In *3rd International Conference on Multi-Agent Systems*, pages 128–135. IEEE Computer Society.
- [Ferber and Gutknecht, 2000] Ferber, J. and Gutknecht, O. (2000). Operational Semantics of Multi-Agent Organizations. *Lecture Notes in Computer Science*, 1757 :205–217.
- [Fidge, 1991] Fidge, C. (1991). Logical Time in Distributed Computing Systems. *Computer*, 4 :28–33.
- [Forrest, 1990] Forrest, S. (1990). Emergent Computation : Self-organizing Collective and Co-operative Phenomena in Natural and Artificial Computing Networks. *Physica D : Nonlinear Phenomena*, 42(1–3) :1–11.
- [Gilbert and Banks, 2002] Gilbert, N. and Banks, S. (2002). Platforms and Methods for Agent-based Modeling. *National Academy of Sciences*, 99(3) :7197–7198.
- [Heath and Etheridge, 1991] Heath, M. T. and Etheridge, J. A. (1991). Visualizing the Performance of Parallel Programs. *IEEE Software*, 8(5) :29–39.
- [Jorrand, 1993] Jorrand, P. (1993). Communication as Unification in Process Algebras : Operational Semantics. *Formal Methods in Programming and Their Applications*, 735 :190–216.
- [Joumaa et al., 2009] Joumaa, H., Demazeau, Y., and Vincent, J.-M. (2009). Performance Visualization of a Transport Multi-agent Application. In *7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009)*, pages 188–196. Springer, Berlin.
- [Kubík, 2003] Kubík, A. (2003). Toward a Formalization of Emergence. *Artificial Life*, 9(1) :41–65.
- [Lacomme et al., 2009] Lacomme, L., Demazeau, Y., and Camps, V. (2009). Classification des mécanismes organisationnels dans les réseaux d’agents. In *17ème Journées Francophones sur les Systèmes Multi-Agents*, pages 79–88. Cépaduès Editions.
- [Luke et al., 2005] Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., and Balan, G. (2005). MASON : A Multi-agent Simulation Environment. *Simulation*, 81(7) :517–527.
- [Mattern, 1989] Mattern, F. (1989). Virtual Time and Global States of Distributed Systems. *Parallel and Distributed Algorithms*, pages 215–226.
- [Millerand et al., 1995] Millerand, B. P., Callaghan, M. D., Cargille, J. M., Hollingsworth, J. K., Irvin, R. B., Karavanic, K. L., Kunchithapadam, K., and Newhall, T. (1995). The Paradyn Parallel Performance Measurement Tool. *IEEE Computer*, 28 :37–46.
- [Milner, 1982] Milner, R. (1982). *A Calculus of Communicating Systems*. Springer-Verlag.
- [Milner et al., 1992] Milner, R., Parrow, J., and Walker, D. (1992). A Calculus of Mobile Processes (part I & II). *Information and Computation*, 100 :1–77.
- [Minar et al., 1996] Minar, N., Burkhart, R., Langton, C., and Askenazi, M. (1996). The Swarm Simulation System : A Toolkit for Building Multi-Agent Simulations. Working Paper 96-06-042, Santa Fe Institute.

- [M.R. Jean et al., 1997] M.R. Jean, Groupe de travail « Collectif » IAD/SMA, and AF-CET/AFIA (1997). Émergence et SMA. In *5ème Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents*, volume 3405, pages 323–342. Hermes, Paris.
- [Nickolayev et al., 1997] Nickolayev, O. Y., Roth, P. C., and Reed, D. A. (1997). Real-Time Statistical Clustering For Event Trace Reduction. *International Journal of Supercomputer Applications and High Performance Computing*, 11(2) :144–159.
- [North et al., 2007] North, M. J., Tatara, E., Collier, N. T., and Ozik, J. (2007). Visual Agent-based Model Development with Repast Symphony. In *Conference on Complex Interaction and Social Emergence*.
- [O'Connor and Wong, 2006] O'Connor, T. and Wong, H. Y. (2006). Emergent Properties. *Stanford Encyclopedia of Philosophy*.
- [Omicini et al., 2008] Omicini, A., Ricci, A., and Viroli, M. (2008). Artifacts in the A&A Meta-Model for Multi-Agent Systems. *Autonomous Agents and Multi-Agent Systems*, 17 :432–456.
- [Pollack and Ringuette, 1990] Pollack, M. E. and Ringuette, M. (1990). Introducing the Tileword : Experimentally Evaluating Agent Architectures. In *Eighth National Conference on Artificial Intelligence*, pages 183–189.
- [Railsback et al., 2006] Railsback, S. F., Lytinen, S. L., and Jackson, S. (2006). Agent-based Simulation Platforms : Review and Development Recommendations. *Simulation*, 82(9) :609–623.
- [Rao and Georgeff, 1991] Rao, A. S. and Georgeff, M. P. (1991). Modeling Rational Agents within a BDI-Architecture. In *2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 473–484.
- [Reed et al., 1993] Reed, D. A., Roth, P. C., Aydt, R. A., Shields, K. A., Tavera, L. F., Noe, R. J., and Schwartz, B. W. (1993). Scalable Performance Analysis : The Pablo Performance Analysis Environment. In *Scalable Parallel Libraries Conference*, pages 104–113. IEEE Computer Society.
- [Robert et al., 2002] Robert, P., Rey, A., and Rey-Debove, J. (2002). *Le Petit Robert : dictionnaire alphabétique et analogique de la langue française*. Dictionnaires Le Robert, Paris.
- [Ropella et al., 2002] Ropella, G. E., Railsback, S. F., and Jackson, S. K. (2002). Software Engineering Considerations for Individual-based Models. *Natural Resource Modeling*, 15(1) :5–22.
- [Schnorr, 2009] Schnorr, L. M. (2009). *Some Visualization Models Applied to the Analysis of Parallel Applications*. PhD thesis, Institut Polytechnique de Grenoble.
- [Schnorr et al., 2010] Schnorr, L. M., Huard, G., Olivier, P., and Navaux, A. (2010). Triva : Interactive 3D Visualization for Performance Analysis of Parallel Applications. *Future Generation Computer Systems Journal*, 26(3) :348–358.
- [Shneiderman, 1992] Shneiderman, B. (1992). Tree Visualization with Tree-maps : 2-d Space-filling Approach. *ACM Transactions on Graphics*, 11(1) :92–99.
- [Stein, 1999] Stein, B. (1999). *Visualisation interactive et extensible de programmes parallèles à base de processus légers*. PhD thesis, Université Joseph-Fourier, Grenoble.
- [van Benthem, 1991] van Benthem, J. F. A. K. (1991). *The Logic of Time : A Model-Theoretic Investigation into the Varieties of Temporal Ontology and Temporal Discourse*. Springer, 2nd édition.

- [Viger et al., 2005] Viger, F., Barrat, A., Dall'Asta, L., Zhang, C.-H., and Kolaczyk, E. D. (2005). Network Inference from TraceRoute Measurements : Internet Topology ‘Species’. *Computing Research Repository*, abs/cs/0510007.
- [Viroli and Omicini, 2002] Viroli, M. and Omicini, A. (2002). Modelling Agents as Observable Sources. *Journal of Universal Computer Science*, 8 :423–452.
- [von Bertalanffy, 1968] von Bertalanffy, L. (1968). *General System Theory : Foundations, Development, Applications*. George Braziller, New York.
- [Weiss, 1999] Weiss, G. (1999). *Multiagent Systems : A Modern Approach to Distributed Artificial Intelligence*. MIT Press.
- [Wooldridge, 2009] Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. John Wiley & Sons, 2nd edition.

Annexe A

Modèle standard de temps

Le modèle standard de temps donné dans [van Benthem, 1991] est un ensemble d'« instants » ordonné par une relation de précédence temporelle notée $<$ et vérifiant les conditions suivantes :

- Transitivité
- Irréflexivité (notons que la transitivité et l'irréflexivité implique l'asymétrie)
- Linéarité
- Éternité ($\forall x, \exists y, y < x$ et $\forall x, \exists y, x < y$)
- Densité ($\forall x, \forall y, x < y \rightarrow \exists z, x < z < y$)

Dans la plupart des cas, quand on utilise des horloges, toutes ces propriétés ne sont pas indispensables. Notamment, avec des horloges digitales, la condition de densité du temps mesuré n'est pas vérifiée (l'ensemble des entiers relatifs est alors un modèle de temps très utilisé). Souvent, la condition d'éternité n'est pas non plus nécessaire.

Une horloge h permet à un processus d'associer à un événement perçu $e \in Hist(S)$ une date, notée $h(e)$. Elle permet aussi d'établir la notion de durée entre deux événements e_0 et e_1 de $Hist(S)$ par la différence de leurs dates. La date est la *composante physique* du temps mesuré par l'horloge, dans la mesure où elle répond aux trois premiers critères de la définition du temps donnée par Van Benthem.

Un *temps physique* est *relatif* à l'horloge qui le définit. En effet, étant donnée deux horloges h_0 et h_1 datant le même événement e , rien ne garantit qu'on ait l'égalité $h_0(e) = h_1(e)$. De même, la durée mesurée entre deux événements e_0 et e_1 par h_0 n'est pas nécessairement la même que celle mesurée par h_1 . Ceci est dû au fait que le rythme de pulsation des horloges et leur compteur de pulsations ne sont pas synchronisés. De plus, elles ont des erreurs qui leurs sont propres et imprévisibles. C'est pourquoi *les temps physiques sont relatifs aux horloges*.

La *composante logique* du temps est l'ordre partiel induit par l'horloge sur l'ensemble des événements qu'elle date. Cet ordre représente la notion de *causalité temporelle* - propriété fondamentale du temps - qui se résume ainsi : *le passé peut influencer le futur et le futur ne peut pas influencer le passé*.

- Un événement $e_0 \in Hist(S)$ *précède* un autre événement $e_1 \in Hist(S)$ selon l'horloge h si et seulement si $h(e_0) < h(e_1)$. On note alors $e_0 <_h e_1$ la relation d'ordre entre les deux événements.
- Deux événements sont *simultanés* si ils ont la même date selon l'horloge h . On a alors aucune des deux relations $e_0 <_h e_1$ et $e_1 <_h e_0$ et on note $e_0 \parallel_h e_1$.

Un *temps logique* est absolu. En effet, étant donnée deux horloges h_0 et h_1 ordonnant les mêmes événements e_0 et e_1 , nous avons les équivalences suivantes :

$$\begin{aligned} - e_0 <_{h_0} e_1 &\iff e_0 <_{h_1} e_1 \\ - e_1 <_{h_0} e_0 &\iff e_1 <_{h_1} e_0 \\ - e_0 \parallel_{h_0} e_1 &\iff e_0 \parallel_{h_1} e_1 \end{aligned}$$

C'est pourquoi *les temps logiques sont absolus*.

Annexe B

Sujet de stage 2A ingénieur

Visualisation et analyse d'applications multi-agents de très grande taille

Les applications multi-agents permettent de résoudre des problèmes naturellement distribués. Cependant, la complexité de leur exécution rend leur analyse extrêmement difficile, notamment lorsqu'elles font interagir un très grand nombre d'agents sur une longue période de temps. Des méthodes d'analyse sont cependant nécessaires pour le design de telles applications, afin de tester et d'évaluer les formes globales qui émergent de leur exécution. Ce stage participe à la perspective de développer des outils pour l'analyse de telles structures émergentes (dans le temps et dans l'espace) au sein d'applications réparties à grande échelle.

Ce sujet s'inscrit dans le cadre d'une collaboration entre deux équipes du LIG : l'équipe MAGMA, groupe de recherche en systèmes multi-agents, et l'équipe MESCAL, qui s'occupe de l'exploitation logicielle d'architectures distribuées à très grande échelle. Cette collaboration a donné naissance à deux approches au problème exposé précédemment. La première est fondée sur l'adaptation au contexte des systèmes multi-agents de techniques de traçage d'applications parallèles. Un environnement d'analyse et de visualisation de traces, *MAS-Pajé*, a été implémenté dans cette optique. La seconde approche consiste à implémenter un système de sondes multi-niveaux permettant d'observer l'application de manière à la fois locale (observation des agents) et globale (observation des structures émergentes). Ces sondes sont hiérarchisées en niveaux d'observation. Chacun d'eux fournit une image de l'exécution de l'application selon un certain recul (dans l'espace et dans le temps). Il est nécessaire, pour expliquer les phénomènes émergents (formes détectées par les sondes globales), de synchroniser les niveaux d'observation et d'explicitier les liens entre les différentes images qu'ils produisent.

Ce stage consiste à développer un environnement graphique permettant de visualiser l'exécution d'une application multi-agents à partir d'un système donné de sondes multi-niveaux. Un tel environnement doit être interactif afin de permettre à l'utilisateur de naviguer entre les différents niveaux d'observation et de pouvoir concentrer son attention sur certaines zones (spatiales et temporelles) de l'exécution. Ce travail doit aider à la visualisation et à la compréhension de la hiérarchie des observations. Enfin, il doit faciliter l'analyse des relations entre observations locales et observations globales afin d'expliquer l'émergence.

Observation macroscopique pour l'analyse de systèmes multi-agents à grande échelle

Résumé

Ce projet s'intéresse à l'analyse, sur de longs temps d'exécution, de systèmes multi-agents (SMA) de très grande taille. Cette analyse s'appuie sur la structure causale de l'exécution pour étudier les comportements globaux des systèmes. Les relations de causalités à l'œuvre localement définissent l'espace et le temps de ces systèmes.

Les approches classiques présentées dans ce rapport fondent leurs techniques d'analyse sur une observation microscopique des agents. Le coût de cette observation, les difficultés liées à l'intrusion des sondes, à la décentralisation et à l'asynchronisme des SMA rendent difficile le passage à l'échelle de l'analyse. L'approche défendue par ce projet est fondée sur la notion d'observation macroscopique pour pallier ces problèmes.

Le méta-modèle de SMA élaboré dans ce projet repose sur un modèle minimal de système distribué asynchrone. L'observation macroscopique est alors formalisée comme une transformation de la structure causale de l'exécution des SMA. Elle est le résultat immédiat d'informations collectées par des sondes macroscopiques.

Deux catégories d'observation macroscopique sont exposées dans ce rapport (observation spatiale et observation temporelle). Des expériences sont présentées pour évaluer leur intérêt quant à l'analyse de l'émergence dans les SMA de grande taille.

Mots clés : *systèmes multi-agents, analyse de l'émergence, passage à l'échelle*

Macroscopic Observation for the Analysis of Large-Scale Multi-Agent Systems

Abstract

This project focuses on the analysis of large-scale multi-agent systems (MAS) for long-time executions. This analysis relies on the causal structure of execution in order to study global behaviours of the system. The causal relations locally define space and time.

The classical approaches introduced in this report ground their analysis techniques on a microscopic observation of the agents. The cost of this observation, the difficulties related to the intrusion of probes, to the decentralisation and to the asynchronism of MAS make the scale-up of the analysis difficult. The approach elaborated by this project is grounded on the notion of macroscopic observation to overcome these limitations. The meta-model of MAS developed in this project relies on a minimal model of asynchronous distributed system. Macroscopic observation is then formalised as a transformation of the causal structure of the execution. It is the result of the information collected by macroscopic probes.

Two categories of macroscopic observation are exposed in this report (spatial observation and temporal observation). Some experiments have been made to evaluate their relevance for the analysis of emergence in large-scale MAS.

Keywords : *multi-agent systems, analysis of emergence, scalable systems*