

Étude sur les cardinalités des interactions

Plan du chapitre :

Afin d'expliquer au mieux l'apparition d'un phénomène, le modèle d'une simulation doit toujours en fournir une description la plus simple possible. Pour aider cet effort, le modèle ne doit pas reposer sur des interactions et des comportements d'agents plus complexes que le nécessitent ces explications. Puisque les actions impliquant simultanément plusieurs agents ne sont pas modélisées sous la forme d'interactions dans les approches actuelles, ces dernières éludent un problème pourtant réel, apparaissant explicitement dans notre approche centrée sur les interaction : « Des interactions simples suffisent-elles à exprimer des comportements complexes ? »

Nous étudions et analysons ce problème à l'aide de deux cas d'études se focalisant sur deux problèmes de simulation différents :

- la coordination de plusieurs agents dans une interaction ;*
- les interactions ayant un effet simultané sur plusieurs agents.*

Nous identifions à cette occasion :

- deux patrons de conception, permettant de modéliser la plupart des interactions complexes impliquant plus de deux agents à l'aide d'interactions individuelles et dégénérées (section 6.1) ;*
 - un nouveau type d'interactions appelé « interactions multicast » permettant de modéliser simplement certaines interactions complexes ne pouvant être décomposées simplement en interactions individuelles et dégénérées (section 6.2).*
-

6.1 Des interactions simples suffisent-elles à modéliser des comportements complexes ?

Un phénomène réel peut être modélisé de manière arbitrairement complexe, en particulier si l'on cherche à en représenter les moindres détails. Bien que de tels modèles soient réalistes, ils n'offrent pas pour autant une « bonne » explication au phénomène puisqu'il y est difficile d'identifier les éléments ayant une réelle influence sur l'apparition du phénomène. Ainsi, un modèle formel offrant des concepts arbitrairement complexes ne fournit pas systématiquement de meilleures explications à un phénomène que des modèles simples. La représentation des connaissances doit donc maximiser le rapport entre simplicité du modèle et ensemble des simulations pouvant être spécifiées.

Puisque les actions impliquant simultanément plusieurs agents ne sont pas modélisées sous la forme d'interactions dans les approches actuelles, ces dernières éludent un problème pourtant réel, apparaissant explicitement dans notre approche centrée sur les interaction : « Des interactions simples suffisent-elles à exprimer des comportements complexes ? ». Dans la description de IODA fournie dans la partie précédente, la cardinalité des interactions est restreinte à un unique agent source, et à aucun ou un seul agent

cible. Bien que ces interactions soient particulièrement adéquates pour représenter des interactions impliquant au plus deux agents, qu'en est-il des interactions en impliquant plus ? On peut en effet aisément imaginer des cas où plus de deux agents participent à une interaction :

- une réaction chimique catalysée par une enzyme, dans laquelle plus de deux substrats sont impliqués ;
- le transport d'un objet volumineux (par exemple une armoire), qui ne peut être transporté que par plusieurs personnes ;
- une bombe qui explose et blesse tous êtres vivants situés dans un certain périmètre ;
- un malade qui propage une maladie au contact d'autres personnes ;
- une parcelle de l'environnement qui transfère une part des phéromones qu'elle contient aux parcelles lui étant contiguës.

Dans cette section, nous discutons ce point, et montrons en quoi il est possible de représenter la plupart des interactions de cardinalité $(n, m) \in \mathbb{N}^2$ uniquement à l'aide d'interactions de cardinalité $(1,1)$ et $(1,0)$. Nous nous appuyons pour cela sur les exemples mentionnés ci-avant, et nous en déduisons deux modèles génériques permettant de décomposer une interaction en un ensemble d'interactions plus simples.

6.1.1 Problème de coordination

Dans cette première section, nous étudions comment modéliser deux phénomènes impliquant des interactions de cardinalité $(n, m) \in \mathbb{N}^2$, dans deux domaines d'application différents. Dans un premier temps, nous nous intéressons à la modélisation du transport d'un objet volumineux par plusieurs personnes, et donc à la modélisation d'interactions de cardinalité $(n, 1)$, avec $n \in \mathbb{N}$. Dans un second temps, nous étudions comment modéliser des réactions enzymatiques en biochimie, ce qui nous amène à considérer comment décomposer des interactions de cardinalité $(1, n)$, avec $n \in \mathbb{N}$. Ces deux études nous amènent à identifier un premier modèle générique permettant de décomposer toute interaction en un ensemble d'interactions de cardinalité $(1,1)$ et $(1,0)$, que nous présentons dans la section qui suit.

Modélisation du transport d'un objet volumineux

Nous illustrons la modélisation d'interactions impliquant plusieurs sources et une cible dans le contexte de simulations où un objet volumineux, par exemple une armoire, ne peut être déplacé que par les efforts conjoints de $n \in \mathbb{N}$ personnes. Ce phénomène peut être représenté dans IODA en modélisant les personnes par une famille d'agents **Personne**, en modélisant l'armoire par une famille d'agents **Armoire**, et en modélisant le déplacement de l'armoire par une interaction **TRANSPORT**, de cardinalité $(n,1)$.

La restriction des cardinalités effectuées dans la description du modèle IODA ne permet pas de modéliser une interaction de cardinalité $(n,1)$. Il est toutefois possible de modéliser ce phénomène à l'aide d'interactions de cardinalité $(1,1)$ et $(1,0)$, et s'affranchir ainsi de la limitation des cardinalités. Pour s'en convaincre, nous nous intéressons plus en détails au phénomène lié à cette interaction.

Le déplacement d'une armoire par n personnes est un problème de coordination. Une fois les personnes nécessaires rassemblées, le déplacement de l'armoire peut être effectué.

Modéliser ce processus avec IODA consiste par exemple à modéliser une personne par une famille d'agents **Personne**, une armoire par une famille d'agents **Armoire**, et un rassemblement de personnes par une famille d'agents **Rassemblement**. Le transport d'une armoire est décomposé en quatre phases, résumées sur la matrice d'interaction brute de la figure 6.1.

Source \ Cible	\emptyset	Armoire	Rassemblement
Personne		(CRÉER RASSEMBLEMENT, $d = 1$)	(REJOINDRE, $d = 1$)
Rassemblement	(DISSOUDRE)	(DÉPLACER, $d = 1$)	

FIGURE 6.1 – Matrice d'interaction brute exprimant comment décomposer le transport d'une armoire à plusieurs personnes en interactions de cardinalités $(1,1)$ et $(1,0)$.

Lors de la première phase, une personne crée un rassemblement de personnes ne contenant qu'elle

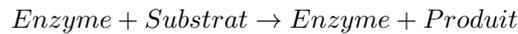
même, dont l'objectif est de transporter une armoire située à proximité (interaction CRÉER RASSEMBLEMENT de la figure 6.1). Lors de la seconde phase, chaque personne souhaitant aider au transport de l'armoire rejoint le rassemblement de personnes, en initiant l'interaction REJOINDRE. Une fois que le rassemblement est suffisamment conséquent pour transporter l'armoire, le rassemblement de personnes déplace l'armoire à l'endroit souhaité en initiant l'interaction DÉPLACER. La dernière phase consiste à dissoudre le rassemblement de personnes en initiant l'interaction DISSOUDRE.

Ce modèle montre qu'il est possible de décomposer une interaction de cardinalité $(n, 1)$ en trois interactions de cardinalité $(1, 1)$, et une interaction de cardinalité $(1, 0)$. La restriction des cardinalités énoncée dans le chapitre 3 n'empêche donc pas la modélisation de phénomènes impliquant ce type d'interactions.

Modélisation de réactions chimiques

Nous illustrons la modélisation d'interactions entre une source et plusieurs cibles dans le contexte de la biochimie. Plus précisément, nous nous intéressons à la modélisation de réactions chimiques impliquant une enzyme.

Le rôle d'une enzyme dans une réaction chimique est d'accélérer la formation d'un ensemble de molécules, appelé produit, à partir d'un ensemble de molécules appelé substrat. Ces réactions chimiques prennent donc la forme générale :



L'interaction entre l'enzyme et le substrat a pour effet de transformer le substrat en produit. Ce phénomène peut donc être représenté dans IODA en modélisant l'enzyme par un agent, chaque molécule du substrat par un agent, et en représentant la réaction chimique sous la forme d'une interaction dont la source est l'enzyme, dont les cibles sont les molécules du substrat, et dont les actions consistent à retirer de l'environnement les molécules du substrat, et d'y ajouter les molécules du produit.

Si le substrat est formé de deux molécules, que l'on note A et B , et que le produit est un complexe, que l'on note C , alors la réaction chimique prend la forme :

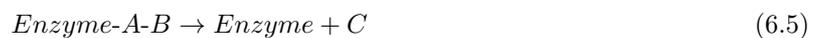
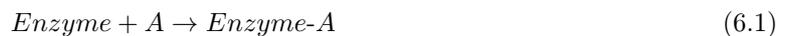


Les agents y sont les espèces chimiques Enzyme, A , B et C , et la réaction chimique y est une interaction ASSOCIER, dont la cardinalité est $(1, 2)$, qui consiste à retirer de l'environnement ses deux cibles, et à créer une troisième entité correspondant à l'association de ses deux cibles.

La restriction des cardinalités effectuées dans la description du modèle IODA ne permettent pas de modéliser une interaction de cardinalité $(1, 2)$. Il est toutefois possible de modéliser ce phénomène avec des interactions de cardinalité $(1, 1)$ et $(1, 0)$, et donc de s'affranchir de la limitation des cardinalités. Pour s'en convaincre, nous proposons d'étudier plus en détails le phénomène lié à cette réaction chimique. En nous basant sur cette décomposition, nous proposons un modèle permettant de décrire l'association de respectivement deux et trois molécules de substrat, en n'utilisant que des interaction permises par la restriction des cardinalités émise dans le chapitre 3.

Décomposition d'une réaction chimique contenant deux molécules de substrat Une réaction chimique a lieu entre les molécules d'un substrat lorsque ces dernières restent suffisamment longtemps dans une configuration spatiale particulière. Une enzyme est une protéine qui catalyse les réactions chimiques en favorisant le contact entre les molécules du substrat. Elle dispose pour cela d'un site actif, sur lequel les molécules du substrat peuvent venir se fixer (voir figure 6.2).

D'après ces informations, cette réaction chimique peut être décomposée en cinq réactions chimiques plus simples. En posant Enzyme-A (resp. Enzyme-B et Enzyme-A-B) une protéine contenant une enzyme sur le site actif de laquelle est fixée une molécule A (resp. B et C), ces réactions sont :



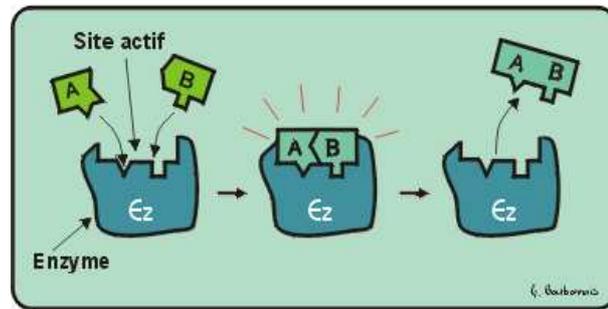


FIGURE 6.2 – Description détaillée d’une réaction chimique catalysée par une enzyme, dans laquelle le substrat est composé de deux molécules A et B , et où le produit est un complexe C formé d’une molécule A et d’une molécule B .

Les réactions 6.1 et 6.2 décrivent le fait qu’une molécule du substrat peut venir se fixer sur le site actif d’une enzyme, ce qui aura pour résultat de créer une protéine complexe contenant à la fois l’enzyme et une molécule de substrat. Les réactions 6.3 et 6.4 décrivent le fait que la seconde molécule du substrat peut venir se fixer sur l’espace restant du site actif d’une enzyme, ce qui aura pour résultat de créer une protéine complexe contenant l’enzyme et le substrat. Enfin, la réaction 6.5 décrit la décomposition de la protéine complexe en une enzyme et le produit de la réaction chimique.

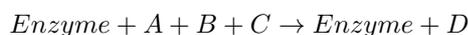
Ces nouvelles réactions font apparaître six espèces chimiques, et cinq réactions. Un modèle IODA possible contient alors six familles d’agents (qui représentent chacune une espèce chimique), et de cinq interactions (qui représentent chacune une réaction). Les réactions 6.1, 6.2, 6.3 et 6.4 ne font réagir que deux espèces chimiques. Par conséquent, elles sont représentées dans IODA à l’aide d’interactions de cardinalité (1,1). De plus, la réaction 6.5 ne fait réagir qu’une seule espèce chimique. Elle est donc représentée à l’aide d’une interaction de cardinalité (1,0). La matrice d’interaction brute de la figure 6.3 résume ce que nous énonçons ici, sous l’hypothèse qu’une molécule A (resp. B) peut se fixer sur le site actif de l’enzyme que si la distance entre cette molécule et l’enzyme est inférieure ou égale à δ_A (resp. δ_B).

Source \ Cible	\emptyset	Enzyme	Enzyme-A	Enzyme-B
A		(RÉACTION6.1, $d = \delta_A$)		(RÉACTION6.4, $d = \delta_A$)
B		(RÉACTION6.2, $d = \delta_B$)	(RÉACTION6.3, $d = \delta_B$)	
Enzyme-A-B	(RÉACTION6.5)			

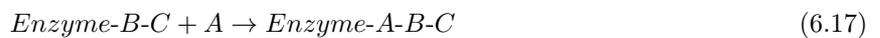
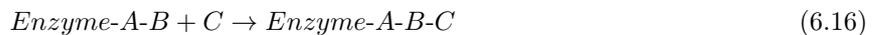
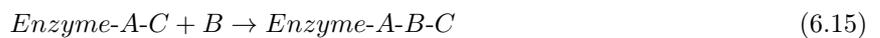
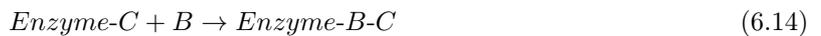
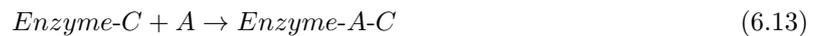
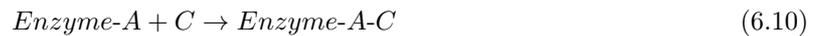
FIGURE 6.3 – Matrice d’interaction brute décrivant la décomposition de la réaction chimique $Enzyme + A + B \rightarrow Enzyme + C$ en cinq réactions chimiques plus simples.

Il est donc possible de décomposer l’interaction ASSOCIER de cardinalité (1,2) en un ensemble d’interactions de cardinalité (1,1) et (1,0). Cette décomposition peut de plus se généraliser à des réactions chimiques impliquant plus de deux molécules de substrat. Nous en donnons l’exemple avec une réaction chimique dont le substrat contient trois molécules.

Décomposition d’une réaction chimique contenant trois molécules de substrat Considérons la réaction chimique suivante, ayant pour substrat trois molécules A , B et C , et pour produit un complexe D :



Cette réaction chimique peut être décomposée en un ensemble de réactions ne faisant réagir à chaque fois au plus deux espèces chimiques :



En construisant le modèle IODA de cette réaction chimique selon les principes énoncés précédemment, l'interaction de cardinalité (1,3) impliquant les trois espèces chimiques A , B et C est décomposée en 11 familles d'agents, 12 interactions de cardinalité (1,1), et une interaction de cardinalité (1,0). Ce modèle est résumé sur la matrice d'interaction brute de la figure 6.4.

Source \ Cible	\emptyset	Enzyme	Enzyme-A	Enzyme-B
A		(RÉACTION6.6, $d = \delta_A$)		(RÉACTION6.11, $d = \delta_A$)
B		(RÉACTION6.7, $d = \delta_B$)	(RÉACTION6.9, $d = \delta_B$)	
C		(RÉACTION6.8, $d = \delta_C$)	(RÉACTION6.10, $d = \delta_C$)	(RÉACTION6.12, $d = \delta_C$)
Enzyme-A-B-C	(RÉACTION6.18)			

Source \ Cible	Enzyme-C	Enzyme-A-B	Enzyme-A-C	Enzyme-B-C
A	(RÉACTION6.13, $d = \delta_A$)			(RÉACTION6.17, $d = \delta_A$)
B	(RÉACTION6.14, $d = \delta_B$)		(RÉACTION6.16, $d = \delta_B$)	
C		(RÉACTION6.15, $d = \delta_C$)		
Enzyme-A-B-C				

FIGURE 6.4 – Matrice d'interaction brute décrivant la décomposition de la réaction chimique $\text{Enzyme} + A + B + C \rightarrow \text{Enzyme} + D$ en 13 réactions chimiques plus simples. Pour des raisons d'affichage, cette matrice est coupée en deux parties.

Cette décomposition est forcément très lourde. Il ne s'agit pourtant pas d'une complexification artificielle du phénomène. En effet, cette description correspond exactement à la décomposition d'une équation-bilan selon la cinétique chimique. Chaque interaction simple correspond alors à une réaction élémentaire et chaque nouvelle famille d'agents à un intermédiaire réactionnel.

Praticabilité de la décomposition d'une interaction Jusqu'à maintenant, nous avons montré qu'il était possible de décomposer une interaction de cardinalité (1,n), avec $n \geq 2$, en un ensemble d'interactions de cardinalité (1,1) ou (1,0), dans le cadre de la description de réactions chimiques. Toutefois, on peut remarquer que le nombre de familles d'agent et le nombre d'interactions obtenus par la décomposition augmente exponentiellement par rapport à n . Considérons la décomposition d'une réaction chimique catalysée par une enzyme dont le substrat contient $n \in \mathbb{N}$ molécules différentes, et dont le produit contient $m \in \mathbb{N}$ molécules. La modélisation de cette décomposition telle que présentée jusqu'à présent implique l'utilisation de $2^n + n + m - 1$ familles d'agents, de $\sum_{i \in [0, n]} (n - i) C_i^n$ interactions de cardinalité (1,1) et d'une interaction de cardinalité (1,0). Cette décomposition est donc très coûteuse en termes de complexité

du modèle, puisqu'elle y introduit un nombre non négligeable de familles d'agents et d'interactions. Le modèle en devient moins intelligible, et moins facile à modifier. Cette décomposition ne peut donc être pratiquée telle quelle.

La description générique des interactions, et son indépendance aux spécificités des agents permet de s'affranchir de ce problème. En effet, avec une modélisation appropriée, le nombre de familles d'agents peut être ramené à $m + n + 1$ (soit le même nombre de familles que la modélisation de ce phénomène avec une interaction de cardinalité $(1, n)$), et le nombre d'interactions à 2 (et donc une interaction de plus que la modélisation de ce phénomène avec une interaction de cardinalité $(1, n)$). Nous illustrons cette affirmation en appliquant la méthodologie IODA pour décrire la décomposition de la réaction chimique $Enzyme + A + B + C \rightarrow Enzyme + D$.

Soit un phénomène mettant en jeu les réactions chimiques 6.6 à 6.6. Dans ce modèle, les espèces chimiques Enzyme-A, Enzyme-B, Enzyme-C, Enzyme-A-B, Enzyme-A-C, Enzyme-B-C et Enzyme-A-B-C représentent une enzyme, sur laquelle des substrats se sont fixés. Nous pouvons les représenter par une unique famille d'agents **Enzyme**, qui contiendra un nombre variable de substrats. La première phase de la méthodologie IODA nous amène donc à identifier quatre familles d'agents, dont les identificateurs sont **Enzyme**, **A**, **B** et **C**.

Source \ Cible	\emptyset	A	B	C	Enzyme
A					(SE FIXER SUR, $d = 0$)
B					(SE FIXER SUR, $d = 0$)
C					(SE FIXER SUR, $d = 0$)
Enzyme	(PRODUIRE D)				

FIGURE 6.5 – Matrice d'interaction brute modélisant une décomposition sous une forme synthétique de la réaction chimique d'équation bilan $Enzyme + A + B + C \rightarrow Enzyme + D$.

Les réactions chimiques 6.6 à 6.18 expriment le fait qu'une espèce chimique A , B ou C a la capacité de se fixer sur une enzyme. Cela mène à l'identification d'une interaction SE FIXER SUR dans la matrice d'interaction brute (voir figure 6.5). Cette interaction, dont la spécification est fournie sur la figure 6.6, consiste à ajouter la source sur le site actif de la cible (primitive *ajouterAuSiteActif*), et à retirer la source de l'environnement (puisque'elle est maintenant associée à l'cible), seulement si le site actif de la cible peut contenir la source (primitive *peutAjouterAuSiteActif*).

« interaction individuelle » SE FIXER SUR(<i>Source</i> , <i>Cible</i>)	
Signatures	nom
	identifiant de la signature
	Source Cible Environnement
Préconditions	Cible.peutAjouterAuSiteActif(Source)
Déclencheur	vrai
Actions	Cible.ajouterAuSiteActif(Source)
	Environnement.retirer(Source)

FIGURE 6.6 – Spécification de l'interaction SE FIXER SUR, permettant aux molécules du substrat de se fixer sur le site actif d'une enzyme

La dernière réaction chimique 6.18 représente le fait qu'une enzyme sur laquelle sont fixés les trois substrats A , B et C peut produire une molécule D . Cela amène à l'identification d'une interaction PRODUIRE D dans la matrice d'interaction brute (voir figure 6.5). Cette interaction, dont la spécification est fournie sur la figure 6.7, consiste à retirer une molécule de l'espèce chimique A , une molécule de l'espèce chimique B et une molécule de l'espèce chimique C du site actif de la source (primitive *retirerDuSiteActif*), à les combiner pour produire une molécule de l'espèce chimique D (primitive *créerMoléculeD*), et à ajouter

cette nouvelle molécule dans l'environnement, à la position de l'enzyme. Cette interaction ne peut être effectuée que si le site actif de la source contient au moins une molécule des espèces chimiques A , B et C (primitive *siteActifContient*).

« interaction dégénérée » PRODUIRE D(<i>Source</i>)		
Signatures	nom	identifiant de la signature
		Source Environnement
Préconditions	Source.siteActifContient(1, "A") et Source.siteActifContient(1, "B") et Source.siteActifContient(1, "C")	
Déclencheur	vrai	
Actions	Agent substratA = Source.retirerDuSiteActif(1, "A") Agent substratB = Source.retirerDuSiteActif(1, "B") Agent substratC = Source.retirerDuSiteActif(1, "C") Agent d = Source.créerMoléculeD(substratA, substratB, substratC) Environnement.ajouter(d, Source.abscisse(), Source.ordonnee())	

FIGURE 6.7 – Spécification de l'interaction PRODUIRE D, permettant à l'enzyme de retirer de son site actif une molécule A , une molécule B , et une molécule C , de les combiner afin de créer une molécule D , et enfin de placer cette molécule dans l'environnement.

Cette façon de modéliser permet de décomposer une interaction de cardinalité $(1,3)$ en un ensemble réduit d'interactions de cardinalités $(1,1)$ et $(1,0)$. Les principes mis en place peuvent aisément être généralisés aux réactions chimiques faisant intervenir un nombre quelconque de molécules dans son substrat. Ainsi, les interactions de cardinalité $(1,n)$, $n \geq 2$, peuvent être décrites dans IODA à l'aide d'interactions individuelles et dégénérées, n'accroissant que légèrement la complexité du modèle. Il est donc possible de modéliser des réactions chimiques avec les restrictions de cardinalités imposées dans IODA.

6.1.2 Première décomposition générique : modélisation par agrégation d'agents

Les modèles décrits dans la section précédente montrent qu'il est possible de représenter des interactions de cardinalité $(1,n)$ et $(n,1)$, avec $n \in \mathbb{N}$, en utilisant uniquement des interactions de cardinalité $(1,1)$ et $(1,0)$. Elles reposent toutes deux sur le même principe général, qui décompose l'exécution de telles interactions en deux phases. Lors d'une première phase, les $n + 1$ participants à l'interaction sont agrégés en un unique agent, dont la fonction est d'initier l'interaction souhaitée. La seconde phase consiste à initier l'interaction avec l'agent agrégat en tant que source. Ce principe de modélisation peut être généralisé à tout domaine d'application, et à toute interaction de cardinalité $(n, m) \in \mathbb{N}^* \times \mathbb{N}$. Dans cette section, nous proposons une description possible d'un tel modèle. Nous fournissons uniquement une description schématique des primitives impliquées dans ces interactions, car ces dernières dépendent grandement de la sémantique de l'interaction modélisée.

Principes généraux du modèle Le principe général permettant la décomposition d'une interaction \mathcal{I} de cardinalité $(n, m) \in \mathbb{N}^* \times \mathbb{N}$ en un modèle ne contenant que des interactions de cardinalités $(1,1)$ et $(1,0)$ consiste dans un premier temps à créer un agent A , auquel on agrègera les sources (s_i) et les cibles (c_j) de l'interaction \mathcal{I} . Pour cela, l'une des sources s_i initie une interaction dégénérée DÉBUT, dont la spécification est schématisée sur la figure 6.8. Cette interaction consiste à créer l'agent A , et à agrèger s_i à A . Elle n'est effectuée que si s_i a l'intention d'initier l'interaction \mathcal{I} , et si s_i vérifie des pré-requis logiques associés à \mathcal{I} .

Les autres sources s'agrègent A en initiant une interaction individuelle ENTRER ayant pour cible l'agent A , dont la spécification est schématisée sur la figure 6.9. Elle n'est initiée que si sa source souhaite

« interaction dégénérée » DÉBUT(<i>Source</i>)		
Signatures	nom	identifiant de la signature
	Source	SourceDeDébut
Environnement	EnvironnementDansDébut	
Déclencheur	% <i>Source souhaite initier l'interaction I</i>	
Préconditions	% <i>Source vérifie des pré-requis logiques associés à I</i>	
Actions	% <i>Un agent agrégat A est créé</i>	
	% <i>A est ajouté à l'environnement</i>	
	% <i>Source est retiré de l'environnement</i>	
	% <i>Source est agrégé à A</i>	

FIGURE 6.8 – Spécification schématique de l'interaction DÉBUT, permettant à un agent source d'une interaction \mathcal{I} de cardinalité $(n, m) \in \mathbb{N}^* \times \mathbb{N}$ de débiter l'agrégation des agents participant à \mathcal{I} .

participer en tant que source à l'interaction \mathcal{I} , si cette source vérifie des pré-requis logiques associés à \mathcal{I} , et si le nombre de sources contenues dans A est strictement inférieur à n . Cette interaction assure que la participation d'un agent source s_i à l'interaction \mathcal{I} est le fruit de son processus décisionnel.

« interaction individuelle » ENTRER(<i>Source, Cible</i>)		
Signatures	nom	identifiant de la signature
	Source	SourceDeEntrer
	Cible	CibleDeEntrer
Environnement	EnvironnementDansEntrer	
Déclencheur	% <i>Source souhaite initier l'interaction I</i>	
Préconditions	% <i>Le nombre de sources contenues dans Cible est strictement inférieur à n et</i> % <i>Source vérifie des pré-requis logiques associés à I</i>	
Actions	% <i>Source est retiré de l'environnement</i>	
	% <i>Source est agrégé à Cible</i>	

FIGURE 6.9 – Spécification schématique de l'interaction ENTRER, permettant à un agent source d'une interaction \mathcal{I} de cardinalité $(n, m) \in \mathbb{N}^* \times \mathbb{N}$ de s'ajouter à l'agent agrégeant toutes les sources et les cibles de l'interaction \mathcal{I} .

Chaque cible c_j est ajoutée à A via l'interaction individuelle CAPTER, qui est initiée par l'agent A et prend pour cible c_j , dont la spécification est schématisée sur la figure 6.10. Elle n'est initiée que si sa source A souhaite que sa cible soit l'une des cibles de l'interaction \mathcal{I} , si cette cible vérifie des pré-requis logiques associés à \mathcal{I} , et si le nombre de cibles contenues dans A est strictement inférieur à m . Cette interaction assure que la participation d'un agent cible c_j à l'interaction \mathcal{I} est le fruit du processus décisionnel de l'ensemble des sources (s_i).

La seconde phase de ce processus commence une fois l'agrégation terminée. Elle consiste dans un premier temps à effectuer les actions de l'interaction \mathcal{I} , en initiant l'interaction dégénérée INITIER avec A pour source dont la spécification est schématisée sur la figure 6.11. Cette interaction n'est initiée que si les agents sources et cibles agrégés à A sont en nombre suffisant, et vérifient le déclencheur, et les préconditions de \mathcal{I} .

Une fois ces actions effectuées, l'agent A désagrège tous les agents qu'il contient, et les replace dans l'environnement en initiant l'interaction dégénérée FIN. Cette interaction a aussi pour effet de retirer A de l'environnement, et n'est effectuée que si l'interaction \mathcal{I} a bien eu lieu. La spécification de cette interaction est schématisée sur la figure 6.12.

Si \mathcal{F} est la famille de l'agent A , S_i est la famille de l'agent source s_i , et C_j est la famille de l'agent source c_j , alors ce principe se traduit par la matrice d'interaction de la figure 6.13.

« interaction individuelle » CAPTER(<i>Source</i> , <i>Cible</i>)		
Signatures	nom	identifiant de la signature
	Source	SourceDeCapter
	Cible	CibleDeCapter
	Environnement	EnvironnementDansCapter
Déclencheur	% <i>Source</i> souhaite que <i>Cible</i> soit l'une des cibles de l'interaction \mathcal{I}	
Préconditions	% Le nombre de cibles contenues dans <i>Source</i> est strictement inférieur à m et % <i>Cible</i> vérifie des pré-requis logiques associés à \mathcal{I}	
Actions	% <i>Cible</i> est retiré de l'environnement % <i>Cible</i> est agrégé à <i>Source</i>	

FIGURE 6.10 – Spécification schématique de l'interaction CAPTER, permettant aux agents sources d'une interaction \mathcal{I} de cardinalité $(n, m) \in \mathbb{N}^* \times \mathbb{N}$ d'ajouter un agent cible à l'agent agrégeant toutes les sources et les cibles de l'interaction \mathcal{I} .

« interaction dégénérée » INITIER(<i>Source</i>)		
Signatures	nom	identifiant de la signature
	Source	SourceDeInitier
	Environnement	EnvironnementDansInitier
Déclencheur	% Le déclencheur de \mathcal{I} est vrai pour les agents sources et cibles contenus dans <i>Source</i>	
Préconditions	% <i>Source</i> agrège n agents sources et m agents cibles et % Les préconditions de \mathcal{I} sont vraies pour les agents sources et cibles contenus dans <i>Source</i>	
Actions	% Les actions de \mathcal{I} sont effectuées pour les agents sources et cibles contenus dans <i>Source</i> % On spécifie dans <i>Source</i> que l'interaction \mathcal{I} a été effectuée	

FIGURE 6.11 – Spécification schématique de l'interaction INITIER, permettant aux agents sources et cibles d'une interaction \mathcal{I} de cardinalité $(n, m) \in \mathbb{N}^* \times \mathbb{N}$ d'initier l'interaction \mathcal{I} .

« interaction dégénérée » FIN(<i>Source</i>)		
Signatures	nom	identifiant de la signature
	Source	SourceDeFin
	Environnement	EnvironnementDansFin
Déclencheur	vrai	
Préconditions	% Il est spécifié dans <i>Source</i> que l'interaction \mathcal{I} a été effectuée	
Actions	% Ajouter dans l'environnement tous les agents agrégés à <i>Source</i> % Retirer <i>Source</i> de l'environnement	

FIGURE 6.12 – Spécification schématique de l'interaction FIN, permettant de replacer dans l'environnement les agents sources et cibles ayant participé à une interaction \mathcal{I} de cardinalité $(n, m) \in \mathbb{N}^* \times \mathbb{N}$.

Source \ Cible	\emptyset	\mathcal{F}	C_1	...	C_n
\mathcal{F}	(INITIER, p=1) (FIN, p=0)		(CAPTER, $d = d_2$, p=1)	...	(CAPTER, $d = d_2$, p=1)
S_1	(DEBUT, p=0)	(ENTRER, $d = d_1$, p=1)		...	
\vdots			\vdots		
S_n	(DEBUT, p=0)	(ENTRER, $d = d_1$, p=1)		...	

FIGURE 6.13 – Matrice d'interaction raffinée décrivant comment modéliser une interaction de cardinalité $(n, m) \in \mathbb{N}^* \times \mathbb{N}$ en utilisant uniquement des interactions de cardinalité (1,1) et (1,0).

Simplifications ponctuelles du modèle générique Le modèle générique présenté ici est valable pour toute interaction \mathcal{I} de cardinalité $(n, m) \in \mathbb{N}^* \times \mathbb{N}$, et permet de modéliser ces interactions à l'aide d'au maximum trois interactions de cardinalité $(1, 1)$, de deux interactions de cardinalité $(1, 0)$, et d'une famille d'agents supplémentaire. Comme tout modèle générique, il peut être simplifié dans certains cas particuliers.

Premièrement, l'une des sources de l'interaction \mathcal{I} peut être utilisée en tant qu'agent agrégat. C'est par exemple le cas de la famille d'agents **Enzyme** qui agrège le substrat d'une réaction chimique. Cela évite de créer une nouvelle famille d'agents pour agréger les sources et les cibles de \mathcal{I} .

Secondement, la distinction entre les interactions INITIER et FIN est faite pour marquer la différence entre la sémantique de l'interaction \mathcal{I} , qui est décrite dans l'interaction INITIER, et la désagrégation des agents ayant participé à l'interaction \mathcal{I} , qui est décrite dans l'interaction FIN. Dans certaines interactions, la désagrégation fait partie de la sémantique de l'interaction. C'est par exemple le cas de l'interaction PRODUIRE A-B-C de la décomposition d'une réaction chimique, puisque les agents libérés dans l'environnement à la fin de l'interaction diffèrent des agents présents au début de l'interaction. Dans de tels cas, la distinction entre INITIER et FIN n'est pas souhaitable. Une seule interaction est donc utilisée pour modéliser ces dernières.

Enfin, si jamais l'interaction \mathcal{I} n'implique qu'une seule cible, alors il est possible de se passer de l'interaction CAPTER en considérant que l'interaction INITIER est de cardinalité $(1, 1)$ et prend pour cible la cible de \mathcal{I} . C'est par exemple le cas de l'interaction DÉPLACER un objet lourd à plusieurs.

6.1.3 Problème d'actions simultanées

Dans cette troisième section, nous étudions sur deux exemples une autre façon de modéliser des interactions de cardinalité $(1, n)$, où $n \in \mathbb{N}$, avec les restrictions de cardinalité de IODA. Dans un premier temps, nous nous intéressons à la modélisation de la propagation d'une maladie contagieuse. Dans un second temps, nous étudions comment modéliser une explosion ayant un effet sur tous les agents situés dans le périmètre de l'explosion. Ces deux études nous amènent à identifier un second modèle générique permettant de décomposer certaines interactions en un ensemble d'interactions de cardinalité $(1, 1)$ et $(1, 0)$, sans utiliser d'agent agrégateur.

Modélisation de la propagation d'une maladie contagieuse

Considérons une simulation visant à reproduire la propagation d'une épidémie dans une population. Dans cette simulation, une personne peut être saine ou malade. Une personne malade contamine par contact toute personne saine affaiblie (par exemple parce qu'elle souffre d'un syndrome immuno-déficience, parce qu'elle vient d'être opérée ou encore parce qu'elle a des plaies ouvertes).

La modélisation d'un tel phénomène se fait dans IODA à l'aide d'une famille d'agents **Personne**. Une **Personne** malade contamine toutes les personnes affaiblies situées à proximité. Par conséquent, l'interaction PROPAGER CONTAGION prend pour cibles un nombre variable d'agents, qui dépend de la position dans l'environnement du malade et des personnes saines. Dans le modèle IODA présenté dans le chapitre 3, la notion de cardinalité permet de caractériser uniquement un nombre fixe de cibles. Elle n'est donc pas appropriée pour modéliser l'interaction CONTAMINER telle quelle. Il est toutefois possible de représenter un tel phénomène à l'aide d'interactions de cardinalité $(1, 1)$.

Dans le phénomène décrit ici, contaminer n cibles revient à effectuer successivement n fois la contamination d'une cible. On peut donc représenter la contamination par une interaction de cardinalité $(1, 1)$ que nous appelons CONTAMINER, qui sera répétée autant de fois qu'il y a de cibles possibles (voir figure 6.14).

Modélisation d'une explosion

Considérons une simulation visant à reproduire le comportement de soldats devant traverser un champ de mines. Dans cette simulation, une mine est un explosif se déclenchant lorsque qu'un soldat marche dessus, c'est à dire lorsque la distance entre le soldat et la mine est de 0. Lorsque l'explosion est déclenchée, la déflagration qu'elle engendre blesse tous les soldats situés dans un certain périmètre (par exemple un disque de rayon 3 mètres), et a pour conséquence la destruction de la mine.

Source \ Cible		\emptyset	Personne
		Personne	(CONTAMINER, d = 0)

« interaction individuelle » CONTAMINER(Source,Cible)		
Signatures	nom	identifiant de la signature
	Source	SourceDeContaminer
	Cible	CibleDeContaminer
	Environnement	EnvironnementDansContaminer
déclencheur	vrai	
préconditions	Source.estMalade() et Cible.estAffaiblié()	
actions	Cible.devenirMalade()	

FIGURE 6.14 – Matrice d’interaction brute (en haut) permettant de modéliser la propagation d’une infection à l’aide de l’interaction individuelle CONTAMINER (en bas), qui est effectuée par un agent source malade afin de contaminer un agent cible affaibli.

La modélisation d’un tel phénomène se fait dans IODA à l’aide de deux familles d’agents *Mine* et *Soldat*. Un *Soldat* se déplace dans l’environnement, et peut déclencher l’explosion d’une *Mine* en marchant dessus. Une interaction DÉCLENCHER de cardinalité (1,1) est donc ajoutée au modèle, afin de modéliser l’évènement déclencheur d’une explosion. La déflagration d’une explosion affecte tous les soldats situés à proximité de la mine. **Par conséquent, l’interaction EXPLOSER prend pour cible un nombre variable d’agents**, qui dépend de la position de la *Mine*, ainsi que de la position des *soldats* dans l’environnement. Dans le modèle IODA présenté dans le chapitre 3, la notion de cardinalité permet uniquement de caractériser un nombre fixe de cibles. Elle n’est donc pas appropriée pour modéliser l’interaction EXPLOSER telle quelle. Il est toutefois possible de représenter un tel phénomène à l’aide d’interactions de cardinalité (1,1) et (1,0). **Néanmoins, contrairement à l’exemple précédent, une explosion affectant n cibles n’est pas équivalent à n explosions successives affectant à chaque fois une cible.** En effet, une explosion a un effet de bord sur la *Mine*, qui consiste à retirer la mine de l’environnement.

Dans le phénomène décrit ici, l’explosion d’une mine se décompose en trois phases : le déclenchement de l’explosion par un soldat marchant sur la mine, la déflagration de la mine sur les soldats situés à moins de trois mètres, et la disparition de la mine dont la déflagration sur les soldats a eu lieu. Il est donc possible de modéliser ce phénomène à l’aide de trois interactions résumées sous la forme de la matrice d’interaction raffinée de la figure 6.15.

Source \ Cible		\emptyset	Mine	Soldat
		Mine	(DISPARAÎTRE, p=0)	(EXPLOSER SUR, d = 3, p=1)
	Soldat	(SE DÉPLACER, p=0)	(DÉCLENCHER, d = 0, p=1)	

FIGURE 6.15 – Matrice d’interaction brute décrivant l’explosion d’une mine et son effet sur les soldats situés à proximité, à l’aide d’interactions de cardinalité (1,1) et (1,0).

La première interaction, que nous nommons DÉCLENCHER, a pour cardinalité (1,1), et pour fonction de notifier à la mine que son explosion doit avoir lieu. Cette interaction a pour source un *Soldat*, a pour cible une *Mine*, et se déclenche seulement si la distance entre le *Soldat* et la *Mine* est de 0. Ses actions consistent à spécifier que la *Mine* a été activée, et qu’elle doit donc exploser (voir figure 6.16).

La seconde interaction, nommée EXPLOSER SUR, a pour cardinalité (1,1), et pour fonction de blesser un à un chaque *Soldat* situé à moins de trois mètres de la mine. Il s’agit d’une interaction ayant pour source une *Mine*, pour cible un *Soldat*, et qui se déclenche seulement si la mine a été activée, et si la déflagration de cette mine n’a pas déjà touché le soldat. Ses actions consistent à diminuer la santé du

« interaction individuelle » DÉCLENCHER(Source, Cible)		
Signatures	nom	identifiant de la signature
	Source	SourceDeDéclencher
	Cible	CibleDeDéclencher
	Environnement	EnvironnementDansDéclencher
déclencheur	vrai	
préconditions	vrai	
actions	Cible.activer()	

FIGURE 6.16 – Spécification de l'interaction DÉCLENCHER.

Soldat (voir figure 6.17).

« interaction individuelle » EXPLOSER SUR(Source, Cible)		
Signatures	nom	identifiant de la signature
	Source	SourceDeExploserSur
	Cible	CibleDeExploserSur
	Environnement	EnvironnementDansExploserSur
déclencheur	not Source.aDejaBlessé(Cible)	
préconditions	Source.estActivée()	
actions	Cible.diminuerSanté(Source.puissance())	

FIGURE 6.17 – Spécification de l'interaction EXPLOSER SUR.

Enfin, la dernière interaction, nommée DISPARAÎTRE, a pour cardinalité (1,0), et pour fonction de détruire la Mine ayant explosé. Il s'agit d'une interaction ayant pour source une Mine, et qui se déclenche seulement si la Mine a été activée, et si tous les Soldats qui devaient être les victimes de la Mine ont subi l'interaction BLESSER. Ses actions consistent à retirer la Mine de l'environnement (voir figure 6.18). La condition relative au fait que tous les soldats doivent avoir été blessés est acquise à l'aide d'une priorité plus forte attribuée à l'interaction EXPLOSER SUR.

« interaction dégénérée » DISPARAÎTRE(Source)		
Signatures	nom	identifiant de la signature
	Source	SourceDeDisparaître
	Environnement	EnvironnementDansDisparaître
déclencheur	vrai	
préconditions	Source.estActivée()	
actions	Environnement.retirer(Source)	

FIGURE 6.18 – Spécification de l'interaction DISPARAÎTRE.

Ainsi, bien que le formalisme de IODA décrit dans le chapitre 3 ne permette pas de modéliser une explosion par une interaction ayant plusieurs cibles, il est possible de modéliser ce phénomène en initiant une interaction de cardinalité (1,1) autant de fois qu'il n'y a de cibles. Il faut toutefois noter que, contrairement à la propagation d'une contagion, l'explosion a un effet de bord sur la source de l'interaction (la disparition de l'agent source de l'environnement). Pour modéliser cet effet de bord, il se révèle nécessaire d'ajouter une interaction supplémentaire au modèle. Cette décomposition reste toutefois une solution naïve, sujette à des problèmes temporels exposés dans la section 6.1.5.

6.1.4 Seconde décomposition générique : modélisation par répétition d'interactions

Les modèles décrits dans la section précédente montrent qu'il est possible de représenter certaines interactions impliquant une source et $n \in \mathbb{N}$ cibles en utilisant une unique interaction de cardinalité (1,1), qui est répétée n fois. Toutefois, ce principe de modélisation ne peut être appliqué que si l'effet de l'interaction sur chaque cible est identique, si cette interaction n'a pas d'effet de bord sur la source de l'interaction, et si ses conditions ou ses actions ne dépendent pas du nombre de ses cibles. Si l'effet de l'interaction diffère selon ses cibles, ou si ses conditions ou ses actions dépendent du nombre de ses cibles, il est alors impossible d'appliquer le principe de modélisation que nous exposons dans cette section. Il reste toutefois possible de représenter cette interaction avec le principe décrit dans la section 6.1.2.

Principes généraux du modèle Le principe général permettant la décomposition de certaines interaction \mathcal{I} de cardinalité $(1, n) \in \mathbb{N}$ en interactions de cardinalités (1,1) et (1,0) consiste à diviser l'interaction \mathcal{I} en une interaction AFFECTER CIBLE de cardinalité (1,1), et une interaction AFFECTER SOURCE, de cardinalité (1,0). L'interaction AFFECTER CIBLE est une interaction dont les actions appliquent l'effet de \mathcal{I} sur une de ses n cibles. Cette interaction n'est effectuée que si sa cible est l'une des n cibles de l'interaction \mathcal{I} . La spécification de cette première interaction est telle que schématisée sur la figure 6.19.

« interaction individuelle » AFFECTER CIBLE(Source, Cible)		
Signatures	nom	identifiant de la signature
	Source	Cible
	Environnement	EnvironnementDansAffecterCible
déclencheur	% Vérifier que Cible est l'une des n cibles de \mathcal{I}	
préconditions	% Vérifier que l'effet de \mathcal{I} n'a pas déjà été appliqué sur Cible	
actions	% Appliquer les effet de \mathcal{I} uniquement sur Cible	

FIGURE 6.19 – Description schématique de l'interaction AFFECTER CIBLE, qui permet d'appliquer les effet d'une interaction \mathcal{I} de cardinalité (1,n), avec $n \in \mathbb{N}$, uniquement sur l'une de ses cibles.

L'interaction AFFECTER SOURCE joue une fonction similaire à AFFECTER CIBLE, en appliquant toutefois les effets de l'interaction \mathcal{I} à sa source, si ces effets ne lui ont pas déjà été appliqués. La spécification de cette seconde interaction est telle que schématisée sur la figure 6.20.

« interaction dégénérée » AFFECTER SOURCE(Source)		
Signatures	nom	identifiant de la signature
	Source	Environnement
déclencheur	vrai	
préconditions	% Vérifier que l'effet de \mathcal{I} n'a pas déjà été appliqué sur Source	
actions	% Appliquer les effet de \mathcal{I} uniquement sur Source	

FIGURE 6.20 – Description schématique de l'interaction AFFECTER SOURCE, qui permet d'appliquer les effet de \mathcal{I} uniquement sur sa source, et avant d'avoir appliqué un quelconque effet sur ses cibles.

Afin de ne pas biaiser la modification des n cibles de l'interaction \mathcal{I} , la modification de sa source ne peut avoir lieu qu'avant ou après l'exécution des interactions AFFECTER CIBLE. Cela aboutit donc à deux modèles possibles, dont la description diffère dans l'ordre dans lequel exécuter ces interactions. Dans un cas, l'interaction AFFECTER SOURCE n'est initiée que si l'interaction AFFECTER CIBLE ne peut plus être initiée. Dans l'autre, les interactions AFFECTER CIBLE ne sont initiées que si l'interaction AFFECTER SOURCE a été préalablement initiée.

6.1.5 Limites des interactions de cardinalités (1,1) et (1,0)

Dans cette section, nous avons décrit deux modèles génériques permettant de décomposer des interactions de cardinalité $(n, m) \in \mathbb{N}^* \times \mathbb{N}$ en un ensemble d'interactions de cardinalité (1,1) et (1,0). Le premier repose sur l'agrégation des $n + m$ agents participant à l'interaction en un seul agent, et permet de décomposer toute interaction. Le second repose sur le fait que certaines interactions de cardinalité (1,n) peuvent être dans certains cas (qui dépendent de la sémantique de l'interaction) n répétitions d'une interaction de cardinalité (1,1). Puisque le premier modèle permet de décomposer toute interaction, la restriction aux interactions individuelles et dégénérées effectuée dans le chapitre 3 ne limite pas l'ensemble des simulations pouvant être spécifiées avec IODA. Cette décomposition n'implique l'ajout que d'un nombre restreint de familles d'agents et d'interactions, et ne complexifie donc que légèrement la spécification des interactions, des familles d'agents et des matrices d'interaction.

Dans certains cas, cette décomposition pose toutefois des problèmes du point de vue de la représentation du temps, et de l'implémentation de l'algorithme d'ordonnement de l'activité des agents.

Nature du problème rencontré

Là où une interaction de cardinalité (n,m) nécessite l'initiation d'une seule interaction, la décomposition de cette interaction nécessite l'initiation de plusieurs interactions. Si l'on se place dans le cadre des simulations en temps discret dont le modèle est décrit dans le chapitre 4, cela signifie que l'interaction n'est plus effectuée en un seul pas de temps, mais en plusieurs pas de temps, dont le nombre dépend de la cardinalité de l'interaction, et de la façon utilisée pour décomposer l'interaction.

La première décomposition effectue une interaction de cardinalité (n, m) en au mieux $3 + m$ pas de temps : un pas de temps lors duquel toutes les interactions DÉBUT et ENTRER sont initiées, m pas de temps lors desquels l'agent agrégat initie l'interaction CAPTER avec chacune des m cibles, un pas de temps lors duquel l'agent agrégat initie l'interaction INITIER, et enfin un pas de temps lors duquel l'agent agrégat initie l'interaction FIN. La seconde décomposition effectue une interaction de cardinalité (1, n) en $n + 1$ pas de temps : n pas de temps lors desquels l'agent source effectue l'interaction AFFECTER CIBLE avec chacune de ses n cibles, et un pas de temps lors duquel l'agent source effectue l'interaction AFFECTER SOURCE.

Cet étalement dans le temps peut poser problème si les agents sont capables d'initier ou subir d'autres interactions pendant cette période. En effet, dans certains cas, des agents qui auraient dû être la source ou la cible de l'interaction peuvent ne pas l'être, et inversement, des agents qui n'auraient pas dû être la source ou la cible de l'interaction peuvent le devenir. Ce fait peut aboutir à des résultats de simulation faussés.

Par exemple, dans le cas du modèle décrit dans la section 6.1.3, des soldats peuvent se déplacer et déclencher une mine, et une mine qui a été déclenchée peut exploser et tuer les soldats victimes de sa déflagration. Dans cet exemple, pendant que la mine initie l'interaction EXPLOSER SUR sur un Soldat, les autres Soldats peuvent se déplacer, et échapper à la déflagration (ceci est illustré sur la figure 6.21). S'ensuit alors un biais : un Soldat ayant déclenché la Mine peut s'enfuir pendant que la Mine est occupée à EXPLOSER SUR les autres Soldats à proximité.

Toutefois, l'utilisation des décompositions proposées ici n'aboutissent pas systématiquement à des biais. Par exemple, dans le cas de la biochimie, cette représentation du temps est tout à fait appropriée pour modéliser la décomposition d'une réaction chimique, car les différentes réaction faisant partie de cette décomposition ne sont pas supposées survenir en parallèle, contrairement aux interactions EXPLOSER SUR. Le problème mentionné ici concerne le cas restreint des interactions ayant nécessairement un effet simultané sur les agents qu'elle implique.

Solutions possibles

Plusieurs approches permettent de résoudre ce problème, mais nécessitent pour la plupart de prendre en compte une représentation plus complexe du temps. Une première approche consiste à permettre aux agents d'initier plusieurs interactions par pas de temps. Bien qu'elle permette de résoudre le problème mentionné ici, cette solution nécessite de fournir un modèle spécifiant de manière générique quelles interactions peuvent être initiées simultanément par un agent, intégrer ces spécifications au modèle de sélection

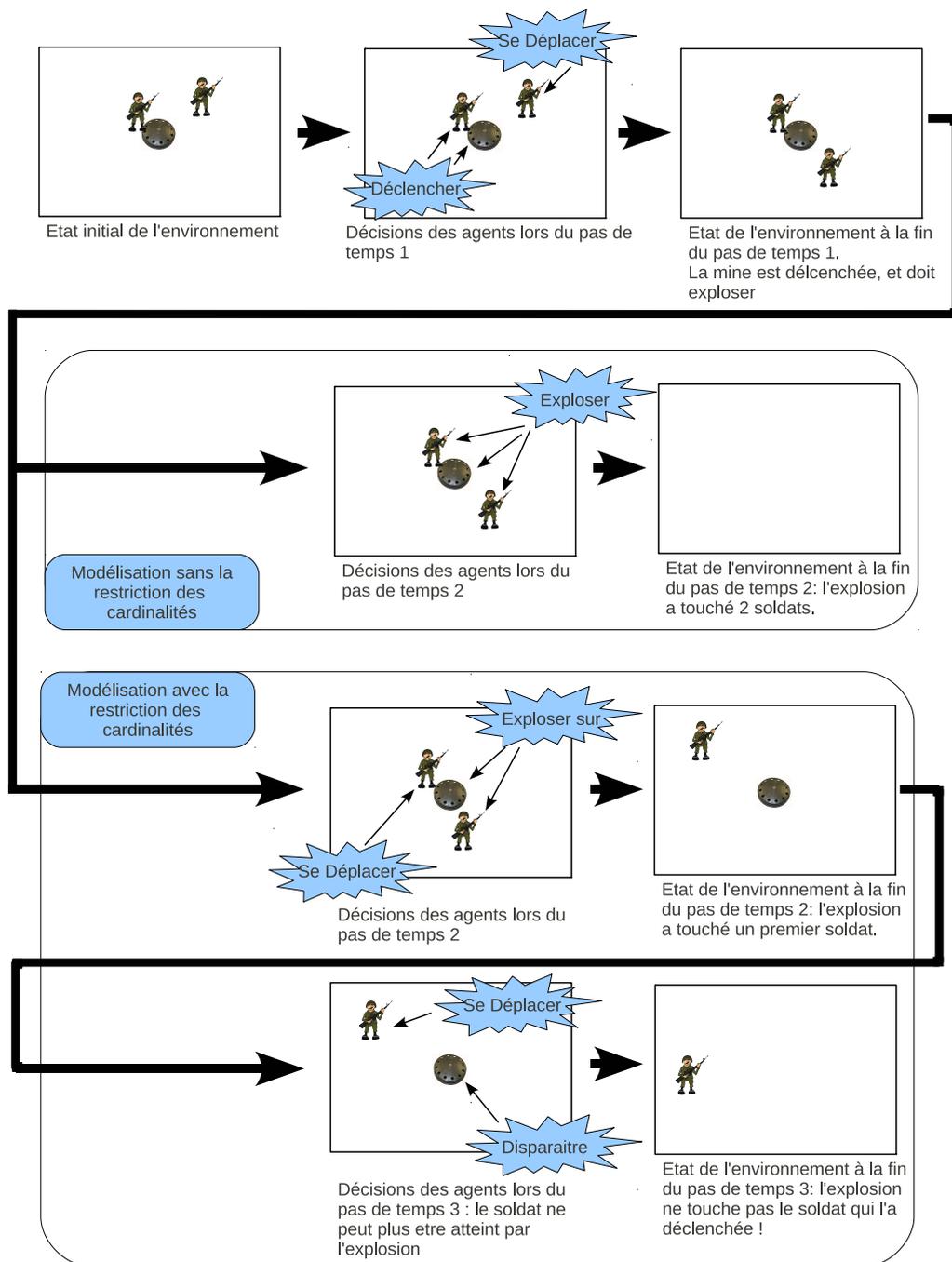


FIGURE 6.21 – Illustration du problème de représentation du temps lié à la décomposition d'une interaction en un ensemble d'interactions de cardinalité (1,1) et (1,0). Cette illustration s'appuie sur le modèle décrit dans la section 6.1.3, dans lequel des soldats peuvent déclencher l'explosion d'une mine. Pour simplifier la compréhension de ce schéma, nous supposons que les actions des interactions EXPLOSER et EXPLOSER SUR ont pour effet de retirer les soldats blessés de l'environnement.

d'interaction, décrire les algorithmes lui étant associés, et permettre de telles spécifications dans la méthodologie de conception. Ainsi, pour résoudre un problème propre à un nombre restreint d'interactions (*i.e.* les interactions qui doivent être exécutées en parallèle), l'ensemble du modèle et de la méthodologie doit être complexifié. Cette solution n'est donc pas souhaitable.

Une seconde solution consiste à utiliser une représentation du temps très fine, où une épaisseur temporelle (*i.e.* une durée) serait attribuée à chaque interaction. Attribuer une épaisseur temporelle presque nulle à une interaction permettrait alors d'obtenir une sorte de parallélisme. Toutefois, tout comme la solution mentionnée précédemment, cette solution nécessite de complexifier la structure du modèle existant, et est donc à éviter.

Une troisième approche consiste à construire le simulateur de sorte qu'il puisse détecter l'initiation d'une interaction située en amont de la décomposition (par exemple l'interaction DÉBUT) lors d'un pas de temps. Lorsque cette situation est détectée, le simulateur passe dans un état transitoire avant l'exécution du prochain pas de temps. Dans cette situation transitoire, les agents ne peuvent initier que les interactions faisant partie de la décomposition (dans l'exemple donné ici, il s'agit des interactions CAPTER, ENTRER, INITIER et FIN). Cette situation transitoire prend alors fin lorsque la dernière interaction de la décomposition est effectuée (ici, l'interaction FIN). Cette approche requiert l'annotation des interactions, afin de déterminer quelles sont celles faisant partie d'une décomposition, et fait perdre leur généralité aux algorithmes utilisés. Elle est donc aussi à éviter.

Une autre solution pourrait être de s'assurer que le modèle ne spécifie que des interactions survenant à la même échelle temporelle, comme c'est par exemple le cas dans la décomposition d'une réaction chimique. Cette solution a l'avantage de ne pas nécessiter d'intervention au niveau de la structure du modèle, et porte seulement sur son sens. Toutefois, cette solution ne permet pas de modéliser les interactions devant survenir en parallèle (par exemple EXPLOSER SUR).

Enfin, une dernière solution serait de prendre en compte les cardinalités plus complexes dans IODA.

Synthèse et discussion

En résumé, les patrons de conception proposés dans cette section peuvent décomposer la plupart des interactions de cardinalité $(n, m) \in \mathbb{N}^* \times \mathbb{N}$ en n'utilisant que des interactions individuelles et dégénérées. Toutefois, certaines interactions de cardinalité (n, m) ont nécessairement un effet simultané sur toutes leurs cibles et ne peuvent donc pas être décomposées. Différentes solutions permettent de remédier à ce problème, mais nécessitent pour la plupart la complexification de l'ensemble du modèle. S'ensuit alors un paradoxe : l'ensemble du modèle est complexifié pour modéliser une minorité d'interactions.

Afin de concilier la possibilité de modéliser de telles interactions sans pour autant complexifier la spécification des autres interactions, nous définissons une extension de IODA où :

- le modèle formel permet de modéliser un troisième type d'interactions appelées interactions multicast ;
- le modèle des interactions individuelles et des interactions dégénérées ne sont pas modifiés ;
- la seule modification de IODA consiste à identifier faire le lien entre tuple réalisable et interaction multicast.

L'ajout des interactions multicast n'induit que peu de modifications dans IODA. De plus, comme il s'agit d'une extension méthodologique, les modèles ne contenant que des interactions pouvant être décomposées peuvent toujours être modélisés simplement à l'aide du cœur de l'approche IODA.

6.2 Extension de IODA aux interactions de type multicast

Certaines interactions initiées par une seule source peuvent avoir un nombre de cible dynamique dépendant du voisinage de l'agent source. Par exemple :

- l'interaction EXPLOSER qui prend pour source un engin explosif et pour cible tous les agents se situant à distance de déflagration de cette source ;
- l'interaction INFECTER qui prend pour source une personne malade et pour cible tous les agents sains se trouvant à son contact ;

- l’interaction ALERTER qui prend pour source un animal ayant perçu un danger ou une alarme anti-incendie et pour cible d’agents alertés par la source.

Ces interactions ont un effet nécessairement simultané sur toutes les cibles. Ces interactions ne peuvent être décomposées en utilisant nos patrons de conception si la représentation du temps ne permet d’initier qu’une seule interaction par pas de temps. En effet, pendant que cette interaction individuelle est initiée sur une cible particulière, les autres cibles de l’interaction ont l’opportunité d’initier ou subir d’autres interactions pouvant modifier leur état. Cela aboutit à des situations où certaines cibles échappent à l’interaction. La figure 6.21 illustre ce point dans le cas d’un Soldat pouvant échapper à la déflagration d’une mine qu’il a pourtant DÉCLENCHÉE.

La spécification de telles interactions induit alors un choix :

- soit complexifier la représentation du temps et permettre à un agent d’initier plus d’une interaction à la fois ;
- soit complexifier les cardinalités supportées par IODA, et ajouter un type d’interaction pour couvrir ce cas problématique en plus des interactions individuelles et dégénérées.

Il n’est pas envisageable de complexifier la représentation du temps, puisque cela reviendrait à complexifier l’ensemble du modèle pour prendre en compte des situations très minoritaires. Nous proposons donc d’étendre le modèle ainsi que la méthodologie et les algorithmes présentés dans le chapitre 4 aux **interactions ayant un nombre dynamique de cibles, que nous appelons interactions de multicast.**

6.2.1 Extension du modèle des interactions

La différence principale entre les interactions individuelles et dégénérées d’une part, et les interactions de multicast d’autre part, tient dans la façon d’exprimer le nombre de cibles de l’interaction. Alors que dans le premier cas, le nombre de cibles est fixe, et égal à 0 ou 1, le nombre de cibles d’une interaction de multicast dépend agents présents dans le voisinage de sa source. Par conséquent, la définition de telles interactions est complétée par un critère permettant de déterminer quelles sont les cibles de l’interaction. Nous appelons *Critère d’acceptation d’une cible* un tel critère. Il prend pour paramètres l’agent source et un agent du voisinage de l’agent source, et retourne une valeur booléenne, vraie si l’agent est une cible de l’interaction et fausse sinon.

Définition 33. Interaction de multicast

Une **interaction de multicast** est une interaction ayant une seule source et un nombre dynamique de cibles, qui dépend du voisinage de l’agent source. Ses cibles sont identifiées parmi le voisinage de l’agent source à l’aide d’un **critère d’acceptation d’une cible**, qui retourne vrai si l’agent est une cible de l’interaction.

Le critère d’acceptation de ces interactions prend la forme :

$$\begin{aligned} \text{accepterCible} : \\ \mathbb{A} \times \mathbb{A} &\rightarrow \{ \text{vrai}, \text{faux} \} \\ (source, agent) &\rightarrow \begin{cases} \text{faux} & \text{si source n'accepte pas agent comme cible de l'interaction} \\ \text{vrai} & \text{sinon} \end{cases} \end{aligned}$$

On note $\mathbb{I}_{(1,*)}$ l’ensemble des interactions de multicast définies pour une simulation.

Dans la suite de cette section, nous décrivons comment compléter le modèle de IODA pour prendre en compte de telles interactions.

Caractérisation du modèle formel

Les principes généraux énoncés jusqu’à ne suffisent pas à la spécification précise d’une interaction de multicast. En effet, dans le modèle formel de IODA, nous avons caractérisé une interaction I comme un 9-uplet contenant un identifiant (id), un ensemble de noms attribués aux agents impliqués dans l’interaction ($noms$), une fonction déterminant si l’agent ayant un nom particulier est une source ou une cible de l’interaction ($nature$), une cardinalité $card = (card_S(I), card_T(I)) \in \mathbb{N}^* \times \mathbb{N}$, une fonction associant une signature à chaque nom d’agent dans l’interaction ($signatures$), une signature de l’environnement

dans l'interaction (*sign_{env}*), et enfin des préconditions, déclencheur et actions. Nous proposons ici de caractériser la valeur de ces divers éléments pour les interactions de multicast. Cette phase est nécessaire à la construction d'une représentation graphique permettant de spécifier ces interactions.

Caractérisation de "noms" : Pour spécifier de telles interaction, le premier problème qui se pose est de pouvoir trouver le nom de chaque agent participant à l'interaction de multicast, afin d'en décrire les conditions et les effets. Pour cela, nous utilisons deux conventions de notation. Premièrement, la source d'une telle interaction doit avoir pour nom "*source*". Ensuite, puisque le nombre de cibles d'une interaction de multicast peut varier de 0 à une valeur n quelconque, le nom des cibles est la concaténation de la chaîne "*cibles[i]*" où i est un nombre allant de 1 à n , $n \in \mathbb{N}$ étant le nombre de cibles participant à l'interaction.

Caractérisation de "nature" : La spécification de la fonction *nature* est triviale, et se base sur l'analyse du nom des entités. La source est l'agent dont le nom dans l'interaction est "*source*". Les agents ayant pour nom "*cibles[i]*" sont les cibles de l'interaction.

Caractérisation de "actions" : Comme nous l'avons mentionné dans le second modèle générique que nous avons présenté, les actions d'une telle interaction se décomposent :

- d'une part en un effet appliqué successivement à chaque cible ;
- d'autre part en un effet appliqué à la source soit avant, soit après la modification des cibles.

Afin de garder un maximum de liberté dans l'expression de ces interactions, nous permettons les deux cas. Pour cela, la description des actions est libre, et nous introduisons un mot clé particulier nommé *nombreCibles* propre à ces interactions, permettant de connaître le nombre de cibles de l'interaction. Ces informations sont suffisantes pour décrire les actions de telles interactions.

Exemple. Description des actions de l'interaction CONTAMINER

Dans le contexte d'une simulation visant à modéliser la propagation d'une infection, l'interaction de multicast CONTAMINER peut avoir la spécification qui suit :

« interaction de multicast » CONTAMINER(<i>source,cibles,nombreCibles</i>)	
...	
Actions	Pour tout i allant de 1 à <i>nombreCibles</i> , faire : <i>Environnement.retirer(cibles[i])</i> Fin pour

Exemple. Description des actions de l'interaction EXPLOSER

Dans le contexte d'une simulation visant à modéliser le déplacement de soldats dans un champ de mines, l'interaction de multicast EXPLOSER peut avoir la spécification qui suit :

« interaction de multicast » EXPLOSER(<i>source,cibles,nombreCibles</i>)	
...	
Actions	Pour tout i allant de 1 à <i>nombreCibles</i> , faire : <i>Environnement.blessier(cibles[i])</i> Fin pour <i>Environnement.retirer(source)</i>

Caractérisation de "signatures" et de "identifiant" : Dans les interactions de multicast, les cibles ont toutes un rôle réflexif, puisqu'elles subissent toutes le même effet en parallèle. Par conséquent, seules deux signatures d'entités sont définies : une pour leur source, et l'autre pour leurs cibles. Nous utilisons les mêmes notations que les interactions individuelles pour les désigner : la signature de la source de l'interaction est notée *sign_{source}*, et celle de ces cibles est notée *sign_{cibles}*. La signature de l'environnement dans de telles interactions, ainsi que l'identifiant de ces interactions, s'exprime exactement comme dans les interactions individuelles et dégérées.

Caractérisation de "card" : Le nombre de cibles de l'interaction n'est pas fixe, et ne peut donc être une valeur dans \mathbb{N} . Nous prenons pour convention de noter la cardinalité de telles interactions $(1, *)$, où $*$ représente un nombre dynamique d'agents. Afin de déterminer quels sont les agents participant à l'interaction lors de l'exécution de la simulation, nous ajoutons à la spécification de telles interactions un critère d'acceptation *accepterCible*. Ce critère prend en paramètre un agent situé dans le voisinage de la source, et retourne une valeur booléenne. Il permet de déterminer quels agents parmi le voisinage de la source sont des cibles de cette interaction de multicast. Ce critère manipule des primitives abstraites de la source, ainsi que des cibles.

Caractérisation de "préconditions" et de "déclencheur" : Le critère d'acceptation de cibles permet de définir si un agent situé dans le voisinage de la source doit être une cible de l'interaction. Ce critère ne définit que les cibles pouvant participer à l'interaction. Pour que cette interaction puisse être initiée par un agent source, les préconditions et le déclencheur doivent être vérifiés. Ils définissent des conditions portant sur la source ainsi que l'ensemble des cibles de l'interaction.

Intégration à la matrice d'interaction

Le modèle que nous venons de décrire permet de caractériser la sémantique d'une interaction de multicast. Pour les utiliser dans une simulation, ces dernières doivent pouvoir figurer dans la matrice d'interaction brute. Puisque les cibles de cette interaction ont un rôle réflexif, nous considérons qu'elles sont placées dans la matrice d'interaction de la même manière que les interactions individuelles, de cardinalité $(1,1)$. Une interaction de multicast I figurant à l'intersection de la ligne associée à une famille d'agents F_1 et de la colonne associée à une famille d'agents F_2 se lit alors « I est une interaction initiée par une instance de la famille F_1 sur un nombre variable déterminé dynamiquement d'instances de la famille F_2 ». Ce nombre d'instances est déterminé à l'aide du critère d'acceptation de l'interaction, ainsi que la garde de distance de l'interaction dans la matrice. Un agent ne peut être la cible de cette interaction que si elle vérifie à la fois le critère d'acceptation, et si la distance la séparant de la source est inférieure ou égale à la garde de distance. L'intégration des interactions de multicast ne modifie donc que légèrement le modèle d'une matrice d'interaction brute.

Dans le cas de la simulation où des soldats se déplacent dans un environnement contenant des mines, où ces derniers peuvent déclencher une mine en marchant dessus, et où une mine déclenchée explose et blesse tous les soldats avoisinants, la matrice d'interaction brute prend l'apparence décrite sur la figure 6.22.

Source \ Cible	\emptyset	Mine	Soldat
Mine			(EXPLOSER, $d = 3$)
Soldat	(SE DÉPLACER)	(DÉCLENCHER, $d = 0$)	

FIGURE 6.22 – Exemple de matrice d'interaction brute intégrant la notion d'interaction de multicast. Cet exemple décrit un modèle dans lequel des soldats peuvent se déplacer dans l'environnement ou déclencher une mine, et où une mine peut exploser et blesser les soldats se trouvant à une distance inférieure ou égale à 3.

Dans le modèle formel, cela se traduit dans un premier temps par l'ajout de la notion d'*élément d'assignation de multicast*, qui représente l'association d'une interaction de multicast, d'une famille d'agents source, d'une famille d'agents cible et d'une garde de distance. Un élément d'assignation de multicast représente donc une interaction de multicast qu'un agent source peut initier avec un nombre déterminé dynamiquement d'agents cibles, tous instances de la même famille d'agents cible.

Définition 34. Élément d'assignation de multicast

Un **élément d'assignation de multicast** est la représentation d'une interaction de multicast $\mathcal{I} \in \mathbb{I}_{(1,*)}$ qu'un agent instance d'une famille d'agents source $\mathcal{F}_S \in \mathbb{F}$ est capable d'initier avec un ensemble déterminé dynamiquement d'agents instances d'une famille d'agents cible $\mathcal{F}_T \in \mathbb{F}$, qui se situe au maximum à une distance $dist \in \mathbb{R}^+$ de l'agent source. Il est représenté sous la forme du n -uplet $(\mathcal{I}, dist, \mathcal{F}_S, \mathcal{F}_T) \in \mathbb{I}_{(1,*)} \times \mathbb{R}^+ \times \mathbb{F}^2$.

L'ensemble des interactions de multicast qu'un agent a la capacité d'initier avec un ensemble d'agents instances d'une famille d'agents cible est alors représenté sous la forme d'une *assignation de multicast*.

Définition 35. Assignation de multicast

Une **assignation de multicast** $a_{S/T}^*$ est un ensemble d'éléments d'assignation de multicast. Elle représente l'ensemble des interactions de multicast qu'une instance de la famille d'agents S peut initier en tant que source conjointement avec un ensemble d'instances de la famille d'agents T en tant que cible.

Plus formellement, on a :

$$\forall S \in \mathbb{F}, T \in \mathbb{F}, a_{S/T}^* \subset (\mathbb{I}_{(1,*)} \times \mathbb{R}^+ \times \mathbb{F}^2)$$

La matrice d'interaction brute représente l'ensemble des interactions que des instances d'une famille d'agents source est capable d'initier avec l'environnement ou une instance d'une famille d'agents cible. Elle correspond à l'ensemble des assignations pour une simulation, et se représente sous la forme d'une fonction associant une assignation dégénérée à chaque famille d'agent source, et une assignation individuelle à chaque couple de familles d'agents source et cible. L'ajout de la notion d'assignation de multicast enrichit cette fonction, puisqu'elle associe aussi une assignation de multicast à un couple de familles d'agents source et cible. Plus formellement, la matrice d'interaction brute se note alors :

$$\mathcal{M} : \begin{array}{l} \mathbb{F} \times (\mathbb{F} \cup \{\emptyset\}) \\ (\mathcal{S}, \mathcal{T}) \end{array} \begin{array}{l} \rightarrow \mathcal{P}(\mathbb{I}_{(1,0)} \times \mathbb{F}) \cup \mathcal{P}(\mathbb{I}_{(1,1)} \times \mathbb{R}^+ \times \mathbb{F}^2) \cup \mathcal{P}(\mathbb{I}_{(1,*)} \times \mathbb{R}^+ \times \mathbb{F}^2) \\ \rightarrow \begin{cases} a_{S/\emptyset} & \text{si } \mathcal{T} = \emptyset \\ a_{S/\mathcal{T}} \cup a_{S/\mathcal{T}}^* & \text{sinon} \end{cases} \end{array}$$

6.2.2 Intégration à la méthodologie

Pour intégrer les interactions de multicast à la méthodologie, et donc permettre leur spécification, nous devons reconsidérer les étapes ayant une influence sur la spécification du modèle d'une interaction, ainsi que les étapes où les spécifications dépendent du contenu du modèle d'une interaction. Ces étapes sont au nombre de trois : l'étape de description de la sémantique d'une interaction, l'étape de description de la matrice d'interaction brute, et l'étape de spécification des primitives abstraites des agents. Nous reconsidérons ces trois étapes afin de répondre à trois problèmes se posant suite à la définition du modèle d'une interaction de multicast :

- Quelle représentation graphique utiliser pour spécifier la sémantique d'une interaction de multicast ?
- Comment déterminer si une interaction est individuelle ou de multicast d'après la matrice d'interaction ?
- Quelles sont les primitives abstraites devant être implémentées par les agents pouvant être la source ou la cible d'une interaction de multicast ?

Spécification graphique d'une interaction de multicast

Une interaction de multicast reste avant tout une interaction. Sa spécification graphique est donc très similaire aux interactions individuelles et dégénérées. Toutefois, puisque ces interactions définissent un critère d'acceptation de cibles, cette représentation graphique doit être étendue pour décrire ce critère. La représentation graphique d'une interaction de multicast prend la forme générale décrite dans la figure 6.23.

Par exemple, dans le cadre de la modélisation de la propagation d'une contagion, la représentation graphique de l'interaction PROPAGER CONTAGION est décrite sur la figure 6.24. Puisque cette interaction ne cible que les agents sains, le critère d'acceptation y est « une cible doit être saine », exprimé sous la forme d'une primitive "estSain" que doivent implémenter toutes les cibles. Puisque l'infection n'a de sens que si la source est malade, et s'il y a au moins une cible à infecter, les préconditions de cette interaction portent sur une primitive "porteUneMaladie" que la source doit implémenter, et sur le nombre de cibles devant être différent de 0. Les actions de cette interaction consistent à rendre malade chaque cible, via leur primitive "devenirMalade".

« interaction de multicast » IDENTIFIANT DE L'INTERACTION(source,cibles,nombreCibles)		
Signatures	nom	identifiant de la signature
	Source	<i>identifiant de la signature d'agent pour "Source"</i>
	Cible	<i>identifiant de la signature d'agent pour "Cible"</i>
	Environnement	<i>identifiant de la signature de l'environnement</i>
Déclencheur	<i>% Déclencheur de l'interaction, décrit sous la forme d'un algorithme. Il manipule des primitives de l'agent nommé "source" dans cette interaction, ainsi que des agents cibles nommés "cible[1]", "cible[2]" ... "cible[nombreCibles]". "nombreCibles" est le nombre d'agents vérifiant le critère d'acceptation de l'interaction, ainsi que la garde de distance dans la matrice d'interaction.</i>	
Préconditions	<i>% Préconditions de l'interaction, décrites sous la forme d'un algorithme. Elles manipulent des primitives de l'agent nommé "source" dans cette interaction, ainsi que des agents cibles nommés "cible[1]", "cible[2]" ... "cible[nombreCibles]". "nombreCibles" est le nombre d'agents vérifiant le critère d'acceptation de l'interaction, ainsi que la garde de distance dans la matrice d'interaction.</i>	
Actions	<i>% Actions de l'interaction, décrites sous la forme d'un algorithme. Elles manipulent des primitives de l'agent nommé "source" dans cette interaction, ainsi que des agents cibles nommés "cible[1]", "cible[2]" ... "cible[nombreCibles]". "nombreCibles" est le nombre d'agents vérifiant le critère d'acceptation de l'interaction, ainsi que la garde de distance dans la matrice d'interaction.</i>	
Critère d'acceptation (source,cible)	<i>% Critère d'acceptation d'un agent cible par un agent source. Il manipule des primitives de l'agent "source" et de l'agent "cible"</i>	

FIGURE 6.23 – Forme générale de la représentation graphique permettant la spécification d'une interaction de multicast.

« interaction de multicast » PROPAGER CONTAGION(source,cibles,nombreCibles)		
Signatures	nom	identifiant de la signature
	Source	SourceDePropagerContagion
	Cible	CibleDePropagerContagion
	Environnement	EnvironnementDansPropagerContagion
Déclencheur	vrai	
Préconditions	source.porteUneMaladie() et nombreCibles \neq 0	
Actions	Pour tout i allant de 1 à nombreCibles, faire : cibles[i].devenirMalade() Fin pour	
Critère d'acceptation (source,cible)	cible.estSain()	

FIGURE 6.24 – Spécification d'une interaction de multicast PROPAGER CONTAGION, qui propage la maladie portée par une source sur un ensemble de cibles saines.

Interaction individuelle ou de multicast ?

Les interactions de multicast sont représentées dans la matrice d'interaction de la même manière que les interactions individuelles. Par conséquent, cette représentation graphique seule ne permet pas de déterminer si une interaction est de multicast ou individuelle. Seule la description sémantique de l'interaction permet de faire la différence. Ainsi, lors de l'étape de construction de la matrice d'interaction brute, placer l'identifiant d'une interaction encore non spécifiée dans une cellule de la matrice d'interaction située à l'intersection d'une ligne et d'une colonne associées à des familles d'agents ne permet de déduire que partiellement sa cardinalité : il s'agit soit d'une interaction individuelle, soit d'une interaction de multicast. Les interactions placées dans la colonne \emptyset échappent à cette incertitude : ce sont des interactions dégénérées.

Inversement, si une interaction est définie comme de multicast lors de la description de sa sémantique, elle ne peut être placée que dans les cellules dont la colonne est associée à une famille d'agents : elle ne peut être placée dans la colonne \emptyset .

Quelles primitives spécifier dans les agents ?

Les agents source et cibles d'une interaction sont caractérisés par une signature dans l'interaction, qui décrit l'ensemble des primitives qu'ils doivent spécifier. Dans le cas des interactions individuelles et de dégénérées, ces signatures fournissent un résumé des diverses primitives abstraites manipulées dans les préconditions, le déclencheur et les actions de l'interaction. Les interactions de multicast ne se limitent pas à ces spécifications, et définissent de plus un critère d'acceptation d'une cible. Ce critère manipule lui aussi des primitives devant être implémentées par les sources ou les cibles de l'interaction, qui doivent par conséquent figurer dans les signatures des entités participant à l'interaction.

6.2.3 Intégration dans les algorithmes de simulation

Intégrer les interaction de multicast dans les algorithmes de simulation revient à déterminer comment recenser les interactions de multicast qu'un agent a l'opportunité d'initier, et à intégrer de telles interactions dans le modèle de sélection d'interaction.

Dans le cadre restreint de IODA, une interaction qu'un agent a potentiellement l'opportunité d'initier est représentée sous la forme d'un tuple composé d'un agent source, d'un élément d'assignation, et d'aucun ou un agent cible. L'ensemble des interactions qu'un agent a l'opportunité d'initier, appelé potentiel d'interaction dans la section 4.3 du chapitre 4, est uniquement composé de tels tuples. Sa construction repose sur les concepts de tuples éligibles et tuples réalisables, que nous étendons ici aux interactions de multicast.

Un tuple est dit éligible si l'agent source du tuple a la capacité d'initier l'interaction du tuple avec les cibles du tuple, ce qui se mesure sur trois critères. Premièrement, la matrice d'interaction doit attester que l'interaction peut avoir lieu entre la famille d'agent source et des instances de la famille d'agents cible. Cela se traduit par le fait que l'élément d'assignation du tuple est présent dans la matrice d'interaction. Secondement, il faut que l'ensemble des cibles du tuple fassent partie des agents perçus par l'agent source (son voisinage). Enfin, la distance séparant la source de chacune de ses cibles doit être inférieure ou égale à la garde de distance de l'élément d'assignation.

Définition 36. Éligibilité pour une interaction de multicast

Soient $s \in \mathbb{E}$ une entité dont la famille est $\mathcal{F}_S \in \mathbb{F}$, $(t_i)_{i \in \mathbb{N}} \in \mathcal{P}(\mathbb{E})$ un ensemble d'entités dont la famille est $\mathcal{F}_T \in \mathbb{F}$, et $a_{multi} = (\mathcal{I}, dist, \mathcal{F}_S, \mathcal{F}_T) \in \mathbb{I}_{(1,*)} \times \mathbb{R}^+ \times \mathbb{F}^2$ un élément d'assignation de multicast.

Un tuple $T = (a_{multi}, s, (t_i)_{i \in \mathbb{N}})$ est dit **éligible** si toutes les conditions qui suivent sont vérifiées :

- s a la capacité d'initier l'interaction de multicast associée à l'élément d'assignation a_{multi} avec les agents $(t_i)_{i \in \mathbb{N}}$ pour cible, *i.e.* $a_{multi} \in \mathcal{M}(\mathcal{F}_S, \mathcal{F}_T)$;
- la distance entre la source s et chaque cible t_i est inférieure ou égale à la garde de distance associée à l'élément d'assignation a_{multi} , *i.e.* si $\forall i, \text{environment.distance}(s, t_i) \leq dist$;
- toutes les cibles t_i appartiennent au voisinage de la source s , *i.e.* si $\forall i, t_i \in \mathcal{V}(e)$.

Ce critère est syntaxique, et se fonde sur la matrice d'interaction brute de la simulation, ainsi que sur le voisinage de l'agent source. La validité sémantique d'un tuple est vérifiée à l'aide du critère de réalisabilité, qui vérifie si les préconditions et le déclencheur sont vérifiés pour la source et l'ensemble de cibles du tuple, si chaque cible vérifie le critère d'acceptation de l'interaction de multicast, et s'il n'existe pas d'autre agent dans le voisinage de la source qui aurait pu être ajouté aux cibles de l'interaction de multicast.

Définition 37. Réalisabilité pour une interaction de multicast

Soient $s \in \mathbb{E}$ une entité dont la famille est $\mathcal{F}_S \in \mathbb{F}$, $(t_i)_{i \in [1, n]} \in \mathbb{E}^n$ un ensemble de n entités dont la famille est $\mathcal{F}_T \in \mathbb{F}$, et $a_{multi} = (\mathcal{I}, dist, \mathcal{F}_S, \mathcal{F}_T) \in \mathbb{I}_{(1, *)} \times \mathbb{R}^+ \times \mathbb{F}^2$ un élément d'assignation de multicast.

Un tuple $T = (a_{multi}, s, (t_i)_{i \in \mathbb{N}})$ est dit **réalisable** si toutes les conditions qui suivent sont vérifiées :

- T doit être éligible ;
- les préconditions de l'interaction \mathcal{I} sont vérifiées pour l'agent source s et les agents cibles $(t_i)_{i \in \mathbb{N}}$, *i.e.* $preconditions(\mathcal{I})(s, (t_i)_{i \in [1, n]}) = vrai$;
- le déclencheur de l'interaction \mathcal{I} est vérifié pour l'agent source s et les agents cibles $(t_i)_{i \in \mathbb{N}}$, *i.e.* $declencheur(\mathcal{I})(s, (t_i)_{i \in [1, n]}) = vrai$;
- le critère d'acceptation de cible est vérifié par chaque cible du tuple, *i.e.* $\forall i \in [1, n], accepterCible(\mathcal{I})(s, (t_i)_{i \in [1, n]})$;
- s'il n'existe pas d'autre agent dans le voisinage de la source qui aurait pu être ajouté aux cibles de l'interaction de multicast, *i.e.* si $\forall c \in \mathcal{V}(e), (a_{multi}, s, (t_i)_{i \in [1, n]} \cup \{c\})$ est réalisable $\Rightarrow c \in (t_i)_{i \in [1, n]}$.

La figure 7 décrit comment calculer le potentiel d'interaction d'un agent, en se reposant directement sur les éléments déclaratifs du modèle IODA.

Le modèle de sélection réactive d'interaction n'est pas modifié par l'ajout des interactions de multicast. Il consiste toujours à attribuer une priorité aux éléments d'assignation de la matrice d'interaction brute, et à sélectionner un tuple réalisable de priorité maximale.

6.3 Synthèse du chapitre

Dans ce chapitre, nous montrons que les interactions individuelles et les interactions dégénérées suffisent à modéliser la majorité des simulations. Pour cela, nous établissons des patrons de conception permettant de décomposer la plupart des interactions de cardinalité (n, m) en un ensemble d'interactions individuelles et dégénérées. Nous pouvons ainsi modéliser des interactions complexes à l'aide d'interactions simples.

Patrons de conception. Le premier patron est appelé modélisation par agrégation d'agents. Il base la décomposition d'une interaction complexe sur un agent auquel toutes les sources et les cibles sont agrégées. Les sources y sont ajoutées en initiant une interaction individuelle prenant pour cible l'agent agrégat. Les cibles y sont ajoutées en subissant une interaction individuelle prenant pour source l'agent agrégat. L'interaction complexe est alors initiée par l'agent agrégat, à l'aide d'une interaction dégénérée. Les principes de ce patron permettent en théorie de modéliser tout type d'interaction complexe.

Le second patron de conception est appelé modélisation par répétition d'une interaction. Il permet de modéliser les interactions de cardinalité $(1, n)$ ayant un effet identique sur toutes les cibles (par exemple une explosion prenant pour cible des soldats). La décomposition y est basée sur :

- une interaction individuelle initiée par l'agent source successivement sur chaque agent cible, afin d'appliquer l'effet de l'interaction complexe sur chacune de ses cibles ;
- sur une interaction dégénérée initiée par la source avant ou après les interactions individuelles, afin d'appliquer une seule fois l'effet de l'interaction complexe sur sa source.

Ces patrons de conception ne permettent pas en pratique de modéliser tout type d'interaction si les agents ne peuvent initier qu'une interaction par pas de temps. En effet, seule une cible peut être modifiée ou agrégée par pas de temps, laissant l'opportunité aux autres cibles d'échapper à l'interaction.

Algorithme 7 : Algorithme permettant de calculer le potentiel d'interaction d'une entité x . Il s'agit d'une extension de l'algorithme 2 de la page 121 aux interactions de multicast.

```

potentiel d'interaction( $x$ )
début
   $\mathcal{R} \leftarrow \emptyset$ ;
  % Recensement des tuples contenant des interactions individuelles ou de multicast
  pour tous les  $\mathcal{F} \in \mathbb{F}$  faire
    pour tous les  $a \in \mathcal{M}(\text{famille}(x), \mathcal{F})$  faire
      % Si l'interaction de l'élément d'assignation est une interaction individuelle
      si  $\text{card}(\mathcal{I}(a)) = (1, 1)$  alors
        pour tous les  $y \in \mathcal{V}(x)$  faire
          si  $(a, x, y)$  est réalisable alors
             $\mathcal{R} \leftarrow \mathcal{R} \cup \{(a, x, y)\}$ ;
          sinon
            % Si l'interaction de l'élément d'assignation est une interaction de multicast
            si  $\text{card}(\mathcal{I}(a)) = (1, *)$  alors
               $\mathcal{T} \leftarrow \emptyset$ ;
              pour tous les  $y \in \mathcal{V}(x)$  faire
                si  $\text{accepterCible}(\mathcal{I}(a))(x, y)$  et  $\text{environnement.distance}(x, y) \leq \text{dist}(a)$ 
                alors
                   $\mathcal{T} \leftarrow \mathcal{T} \cup \{y\}$ ;
                si  $(a, x, \mathcal{T})$  est réalisable alors
                   $\mathcal{R} \leftarrow \mathcal{R} \cup \{(a, x, \mathcal{T})\}$ ;
            sinon
              % Recensement des tuples contenant des interactions dégénérées
              pour tous les  $a \in \mathcal{M}(\text{famille}(x), \emptyset)$  faire
                si  $(a, x)$  est réalisable alors
                   $\mathcal{R} \leftarrow \mathcal{R} \cup \{a, x\}$ ;
      retourner  $\mathcal{R}$ ;
fin

```

Extension aux interactions multicast. Deux principes différents permettent de modéliser des interactions ayant un effet simultané sur leurs cibles :

- utiliser une représentation du temps complexe, et conserver la représentation actuelle des interactions ;
- utiliser une représentation des interactions complexe, et conserver la représentation du temps.

L'utilisation d'une représentation du temps complexe n'est pas envisageable, car cela revient à complexifier la spécification de l'ensemble des simulations pour traiter des cas minoritaires. Nous montrons que la représentation des interactions complexes ne nécessite dans IODA qu'un nombre restreint de modifications et est donc praticable.

Afin de modéliser ces interactions, notre extension de IODA repose sur un type d'interaction plus complexe appelé interactions multicast. Les interactions multicast impliquent simultanément un nombre de cibles déterminé dynamiquement par l'agent source lors de sa prise de parole. Le nombre de cibles de telles interactions dépend du voisinage de l'agent source et d'un critère défini dans l'interaction multicast déterminant si un voisin est pris pour cible par l'interaction. Cet ajout dans notre taxinomie ne modifie pas de manière significative le modèle formel de IODA, et ne nécessite pour sa grande partie que des ajustements dans les algorithmes déjà existants.

Cette extension permet donc de couvrir un plus grand nombre de simulations avec IODA pour un coût restreint en terme de complexification de la structure du modèle et de la méthodologie de conception.