Définitions et études de cas sur l'émergence

Henri Merciol

Définitions et études de cas sur l'émergence

Henri Merciol
GREYC - Université de Caen
Henri.Merciol@unicaen.fr

Résumé

Cet article présente un travail de synthèse portant sur les phénomènes émergents dans les systèmes multi-agents.
Il a fallut tout d'abord déterminer précisément ce que l'on entendait par phénomène émergent. En effet, bien que le terme soit très en vogue aujourd'hui, il n'est pas précisément défini. Deuxièmement, quelques études de cas simples ont été élaborées afin de pouvoir étudier des exemples simples de phénomènes émergents.
Nous commencerons donc cet article par quelques précisions sur la notion d'émergence. Nous poursuivrons par une introduction à la modélisation GDT4MAS que nous utilisons pour modéliser les systèmes multi-agents. Nous terminerons enfin en présentant quelques cas d'étude simples sur l'émergence.

Mots Clef

Systèmes multi-agents, émergence, preuve formelle.

Abstract

This paper present a synthetic work on emergent phenomenoms in multi-agents systems.
First of all, we had to define precisely what we meant by emergent phenomenom. Indeed, even if this word is used a lot nowadays, it is not clearly defined. Then, we built some small study cases in order to be able to study simple exemples of emergent phenomenoms.
We will begin this paper with some clarifications about emergence. Then, we will introduce the GDT4MAS modelisation, which we use to modelise the multi-agents systems. Finaly, we will end this paper by presenting some simple study cases about emergence.

Keywords

Multi-agents system, emergence, formal proof.

1 Introduction

Les systèmes multi-agents sont de plus en plus étudiés et utilisés, et en particulier les propriétés émergentes (ou phénomènes émergents) survenant dans la plupart des systèmes complexes. Ces propriétés, observables dans de nombreux systèmes naturels, peuvent se révéler très utiles, mais sont encore largement inexpliquées.
Nous présentons dans cet article le travail préliminaire d'une thèse visant à élaborer une méthode de preuve automatique et formelle des phénomènes émergents. Nous tentons donc d'y éclaircir la notion d'émergence, dont la définition ne fait pas encore consensus et a beaucoup évoluée [Gol99]. Nous y présentons plusieurs propriétés ainsi que la définition que nous avons retenue d'une propriété émergente. Nous présentons également plusieurs cas simples de propriétés émergentes permettant une étude approfondie. De plus, nous proposons une modélisation simple associée à un système de preuve formelle automatisable, dans le formalisme de laquelle nous présentons les cas simples mentionnés.
Nous commencerons par définir le cadre précis de notre approche. En particulier nous présenterons plusieurs définitions de ces propriétés trouvées dans la littérature et en déduirons différents types de propriétés émergentes. Ensuite nous présenterons la modélisation GDT4MAS, que nous développons actuellement pour représenter les systèmes multi-agents. Puis nous exposerons les cas d'étude simples élaborés avec leur propriété émergente associée, pour finalement conclure.

2 Émergence

Plusieurs définitions de l'émergence se sont succédées ou ont cohabité depuis ses premières mentions. Tous ces débats et revirements démontrent l'absence de consensus.
Nous devrons donc d'abord préciser quelle est la définition que nous adopterons pour la suite de l'article. Nous nous inspirerons pour cela des différentes définitions déjà existantes et en choisirons une, la plus générale possible. Nous identifierons ensuite plusieurs propriétés de l'émergence et types d'émergence.

2.1 Définition générale

De nombreuses définitions de l'émergence ont été formulées. Par exemple, G. H. Lewes (1875) explique que, contrairement à une résultante, les phénomènes émergents ne sont pas simplement la somme de leurs composants, mais quelque chose de plus [Lew75]. Jeffrey Goldstein (1999) les décrit comme "l'apparition de structures, schémas et propriétés nouveaux et cohérents lors du processus d'auto-organisation d'un système complexe" [Gol99]. Peter A. Corning (2002) dit que "les règles ou les lois n'ont pas d'efficience causale ; de fait, elles ne génèrent rien. Elles ne servent qu'à décrire les régularités et les relations consistantes dans la nature" [Cor02]. Il illustre son propos avec l'exemple du jeu d'échec en rappelant qu'il est impossible de prédire à coup sûr le prochain mouvement d'un

joueur en se basant simplement sur les règles du jeu : il y a quelque chose de plus que ces règles.
Dans le domaine plus spécifique des systèmes multi-agents, James Odell (1998) définit l'émergence comme "l'existence d'un schéma cohérent qui apparaît des interactions entre des objets plus élémentaires" [Ode98]. Il propose donc une explication au phénomène d'émergence en suggérant que tout phénomène émergent naîtrait de l'interaction entre les différents composants du système.
Nous considérons, et proposons, la définition suivante pour les phénomènes émergents dans le cadre spécifique d'un système multi-agents.

Définition 1 (Propriété émergente)**.** *Nous appelons* propriété émergente *dans un système multi-agents toute propriété du système apparaissant au cours de son exécution, qu'elle ait été recherchée ou non par le concepteur du système, qui n'est le but explicite d'aucun agent du système et qui n'est pas le fait d'un agent isolé.*

Autrement dit, nous considérons que les phénomènes émergents peuvent être un effet secondaire de l'interaction entre le code des différents agents.

2.2 Émergences forte et faible

Une répartition possible et commune des phénomènes émergents est de les séparer en deux catégories : les émergences dites fortes et celles dites faibles.
Par définition, l'émergence faible regroupe tous les phénomènes émergents reproductibles par simulation informatique, tandis que l'émergence forte regroupe ceux qui ne le sont pas. L'existence de ce premier type d'émergence est évidente : dés lors que l'on peut simuler un système, on doit pouvoir reproduire toutes ses propriétés. Quant à l'émergence forte, son existence ne fait pas consensus. P. W. Anderson la justifie en disant que "la capacité a tout réduire à des lois fondamentales simples n'implique pas la capacité de reconstruire l'univers à partir de ces lois" [And72]. À l'inverse, Mark A. Bedau la remet en question en observant que "ça ressemble à de la magie" [Bed97] : selon lui, qu'on ne puisse pas reconstruire un phénomène à partir de ses lois fondamentales signifie que l'on produit quelque chose à partir de rien, et de tels phénomènes ne sont donc pas descriptibles par la science.
Départager ces deux points de vue n'est pas l'objectif de cet article. Quoi qu'il en soit, nous ne nous intéressons ici qu'à des cas simulables, et donc à l'émergence faible.

2.3 Synergie

On parle de synergie lorsque les différents éléments d'un système combinent leurs actions/effets pour obtenir un autre effet que tout ce que les éléments auraient pu accomplir isolément.
Peter A. Corning [Cor02], pour ne citer que lui, définit bien l'émergence comme une forme de synergie et donne même quelques exemples. Par exemple, le chlorure de sodium, ou sel de table, est composé de deux éléments très toxiques pour l'homme, le chlore et le sodium, mais les deux ensembles sont au contraire très bénéfiques.
Bon nombre d'autres exemples sont fondés sur le partage des tâches. Notamment, toutes les symbioses fonctionnent ainsi : en fournissant à son partenaire un service qu'il est incapable de produire lui-même, et réciproquement, ils constituent ensemble une entité améliorée, bien supérieure aux organismes isolés. Les exemples de symbiose sont légion, mais nous ne nous étendrons pas plus sur le sujet.
Un autre exemple que donne Corning est l'automobile : l'ensemble forme un moyen de transport, ce qui n'est le cas d'aucune de ses pièces. Mais si on retire certaines pièces particulières, comme le volant, le moteur ou une roue, le tout ne fonctionne plus. Ce qui n'est pas le cas si on retire des pièces moins importantes comme les jantes.
Certains agents d'un système sont donc parfois indispensables ou plus utiles que d'autres, mais ils participent tous à la constitution du tout.

2.4 Types d'émergence

Les propriétés suivantes ne sont pas présente dans tous les phénomènes émergents, mais intéressent beaucoup chercheurs et industriels.

Conventions sociales. Les conventions sociales ont été beaucoup étudiées récemment pour leur capacité à résoudre des problèmes de coordination.
En effet, la définition d'une convention dans un système multi-agents est une contrainte sur le comportement (ou l'état) des agents du système. Certains ajoutent à cette définition la notion d'implicite [Del02], indiquant que cette convention ne doit pas être décidée au départ mais doit émerger du système et être implicitement trouvée par ses agents. D'autres considèrent qu'il s'agit d'un cas particulier, en distinguant alors les conventions hors-lignes inscrites durablement dans le comportement des agents et les conventions émergeant des interactions entre les agents du système (correspondant à la définition précédente) [WW95]. Dans cette dernière définition, seul le second type de convention constitue une propriété émergente, et donc seul ce cas nous intéresse.
Par définition, ce type d'émergence naît donc de l'interaction entre les agents qui doivent se mettre d'accord sur une politique à mener. Elle correspond bien à notre définition car quelle que soit la politique choisie, aucun agent ne cherchait à imposer ladite politique aux autres agents.

Stabilité et auto-réparation. Certains phénomènes émergents représentent un état d'équilibre stable du système et vont donc s'auto-entretenir, voire s'auto-réparer en cas de perturbations. Deux exemples très connus sont le vol en nuée d'oiseaux et la colonie de fourmis.
Commençons par le vol en nuée d'oiseaux. Dans une telle nuée, aucun oiseau n'a de rôle prépondérant. Si un chasseur abat l'oiseau de tête, il est immédiatement remplacé et la nuée peut continuer son vol. De même, si la nuée se sépare en deux groupes pour quelque raison (une tour ou un avion

à contourner par exemple), chaque groupe peut continuer seul ou les deux fusionnent et reforment la nuée juste après. L'exemple de la colonie de fourmis est plus évident : qui n'a pas déjà essayé d'obstruer ou d'effacer la piste d'une file de fourmis ? Nous avons alors tous observé que la piste se reforme ou l'obstacle est contourné rapidement et que la file de fourmis retrouve alors le plus court chemin.
Cette capacité de certains phénomènes à s'auto-réparer est bien sûr très intéressante. Elle n'est d'ailleurs pas l'apanage des phénomènes émergents : les codes correcteurs en sont un autre exemple.

3 Modélisation GDT4MAS

Nous nous baserons, pour présenter les cas d'études, sur une modélisation que développent Bruno Mermet et Gaële Simon appelée GDT4MAS [MS09, MS10, MS12].

3.1 Environnement et variables

Nous représentons simplement l'environnement par une liste de variables (représentant généralement la vision que les agents en ont via leurs détecteurs). Tous les agents peuvent *a priori* modifier ces variables (c'est au modèle de définir quels agents modifient ou non quelles variables en tenant compte des éventuelles contraintes physiques du cas modélisé).
Les agents disposent aussi de variables représentant leur état : des variables internes accessibles uniquement par eux (par exemple, le niveau de leur batterie), et des variables dites de surface visibles par les autres agents mais modifiables uniquement par eux (par exemple, leur position).
Les phénomènes émergents sont des propriétés du système. Nous espérons donc pouvoir les exprimer formellement à partir des variables du système (une intuition qui reste à démontrer).

3.2 Arbre de décomposition de but

Dans cette modélisation, chaque type d'agent est représenté par un GDT ou arbre de décomposition de but (Goal Decomposition Tree) [MSZ].
À la racine de cet arbre, on spécifie le but complet et la condition d'activation de l'agent exprimés formellement. La condition d'activation permet de savoir à quel moment l'agent entre en action, les variables utilisées étant celles visibles par l'agent (soit par ses détecteurs, soit les variables qui lui sont accessibles). Le but est représenté par condition de satisfaction exprimée formellement.
Nous notons, pour une variable x, sa valeur à l'instant considéré x, et son état au pas de temps suivant x'.
Ce but global est ensuite décomposé en plusieurs sous-buts grâce à des opérateurs de décomposition spécifiques (And, SeqAnd, Or, SeqOr, Case et Iter). Ces opérateurs permettent d'indiquer comment ces sous-buts doivent s'articuler pour réaliser le but père (voir l'exemple ci-dessous). On continue de décomposer récursivement les buts en sous-buts plus simples jusqu'à obtenir des buts élémentaires, que nous nommerons buts-feuilles. Du parcours de l'arbre modulé par les opérateurs de décomposition et l'exécution des buts-feuilles atteints résulte alors un état du système vérifiant le but global.

3.3 Exemple

L'exemple le plus simple est le suivant : on veut modifier une variable x en lui associant la valeur $2x+1$. En notant x' la nouvelle valeur, on cherche donc à obtenir $x' = 2x + 1$. Il s'agit du but racine présenté sur la figure 1.

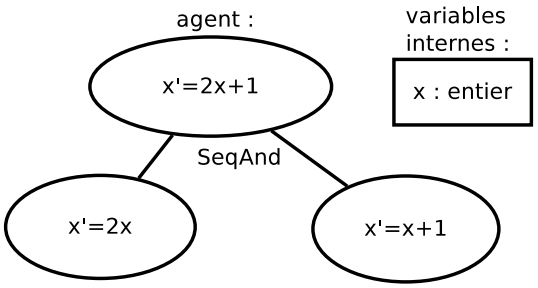

FIGURE 1 – GDT exemple SeqAnd

Pour cela nous allons donc utiliser l'opérateur de décomposition SeqAnd. Cet opérateur nous dit que pour réaliser le but père il faut exécuter tous les buts fils dans l'ordre (de gauche à droite sur la figure). La décomposition présentée sur la figure 1 nous indique donc de d'abord multiplier x par deux puis d'ajouter un au résultat pour obtenir $2x + 1$. En réalité, du point de vue mémoire, on peut distinguer quatre instants : t_0 avant l'exécution de l'agent, t_1 après l'exécution du premier sous-but, t_2 avant l'exécution du second, et t_3 après l'exécution de l'agent. En notant x_{t_i} la valeur de x à l'instant t_i, on obtient : $x_{t_1} = 2 * x_{t_0}$ et $x_{t_3} = x_{t_2} + 1$. La variable x étant interne à l'agent, rien ne peut la modifier à part l'agent lui-même, et sachant qu'il n'effectue aucune action entre les instants t_1 et t_2, alors $x_{t_2} = x_{t_1}$. Au final, on a donc bien $x_{t_3} = 2x_{t_0} + 1$.

3.4 Preuve formelle

Cette modélisation est également associée à un système de preuve formelle développé simultanément [MS12].
Ce système de preuve est conçu pour être automatisable. Il associe à chaque but feuille une obligation de preuve fondée sur son action élémentaire et à chaque opérateur de décomposition une obligation de preuve calculée en fonction des sous-buts. On obtient ainsi une obligation de preuve par nœud dans le GDT des agents.
Il arrive que des arbres ou sous-arbres soient répétés à l'identique ou à un paramètre près par un autre agent ou, pour les sous-arbres, à un autre endroit dans le même arbre. La modélisation étant par essence modulaire, ces cas sont faciles à détecter et à traiter : on peut aisément faire une preuve pour le module et ne pas avoir à la refaire pour chaque répétition.
Les obligations de preuve de chaque agent sont ensuite vérifiées par un prouveur automatique en logique des prédicats, par exemple celui que nous utilisons est PVS [SRI].

Si toutes les obligations sont vérifiées et justes, le système fait bien ce que l'on voulait.

3.5 Exemple, preuve formelle

Reprenons l'exemple précédent.

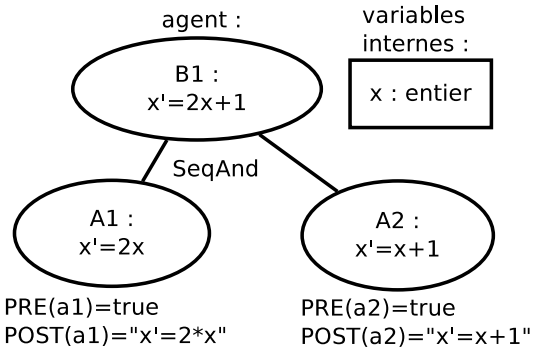

FIGURE 2 – GDT exemple SeqAnd avec actions

Commençons par les feuilles. Chaque but-feuille est associé à une action, qui est en réalité un appel de fonction ou du code effectuant concrètement l'action visée, soit sur un agent virtuel soit sur un robot. Cette action est associée à une précondition PRE_a et une postcondition $POST_a$ spécifiées par le concepteur du programme/robot. Dans notre cas, aucune précondition particulière n'est nécessaire pour aucune de nos actions, nous les instancions donc à $true$. Pour la postcondition, il s'agit de l'effet de l'action. Dans notre cas, nous pouvons recopier le but du nœud correspondant. Pour des cas plus complexes, en particulier si le concepteur du programme/robot et le concepteur de l'agent appelant ne se sont pas concertés, on peut ne pas avoir d'action correspondant exactement au but recherché.
Nous disposons également pour chaque nœud d'un contexte C_N, calculable automatiquement à partir des conditions d'activation de l'agent, et d'une condition de satisfaction SC_N, équivalent au but du nœud. Nous poserons les conditions d'activation de l'agent à $true$, indiquant qu'il peut agir sans contrainte. Nous ne détaillerons pas ici le calcul des contextes des sous-buts, voici les résultats : $C_{A1} = true$ et $C_{A2} = (x_{t_2} = 2 * x_{t_0})$.
Les deux obligations de preuve que nous devons calculer pour chaque but-feuille sont $C_N \rightarrow PRE_a$ et $POST_a \rightarrow SC_N$. Donc :

$$C_{A1} \rightarrow PRE_{a1} \quad (1)$$
$$POST_{a1} \rightarrow SC_{A1} \quad (2)$$
$$C_{A2} \rightarrow PRE_{a2} \quad (3)$$
$$POST_{a2} \rightarrow SC_{A2} \quad (4)$$

Qui devient :

$$true \rightarrow true \quad (5)$$
$$(x_{t_1} = 2 * x_{t_0}) \rightarrow (x_{t_1} = 2 * x_{t_0}) \quad (6)$$
$$(x_{t_2} = 2 * x_{t_0}) \rightarrow true \quad (7)$$
$$(x_{t_3} = x_{t_2} + 1) \rightarrow (x_{t_3} = x_{t_2} + 1) \quad (8)$$

Sur un cas aussi simple, on remarque aisément que toutes ces obligations de preuve sont vérifiées.
Passons maintenant au but racine $B1$. Chaque but non feuille est associé à une obligation de preuve calculée en fonction de son opérateur de décomposition et de ses sous-buts. Dans le cas du SeqAnd, le calcul est le suivant :

$$C_N \wedge SC_{N1} \wedge stab_{N1 \rightarrow N2} \wedge SC_{N2} \rightarrow SC_N \quad (9)$$

Où l'on renomme convenablement les instants de façon à refléter la sémantique opérationnelle et où $stab_{N1 \rightarrow N2}$ représente la stabilité des variables internes entre les deux buts-feuilles (à savoir $x_{t_2} = x_{t_1}$). En l'appliquant, nous obtenons :

$$(x_{t_1} = 2 * x_{t_0}) \wedge (x_{t_2} = x_{t_1}) \wedge (x_{t_3} = x_{t_2} + 1) \rightarrow (x_{t_3} = 2 * x_{t_0} + 1) \quad (10)$$

Ce qui est également vérifié.
Les cinq obligations de preuve étant vérifiées, nous avons prouvé que notre agent faisait bien ce que l'on voulait qu'il fasse.
Comme annoncé précédemment, tous ces calculs sont automatisables et la vérification peut se faire simplement grâce à un prouveur automatique.

4 Jeux de cas simples

Tous les cas que nous présentons maintenant sont simples. Ils ont été conçus ainsi afin d'être aisément compréhensibles et étudiables dans tous leurs aspects. Et c'est justement cette simplicité qui les rend intéressants, permettant d'en faire une étude poussée. Chacun présente une propriété émergente que nous expliciterons.

4.1 Le pompier et le pyromane

Voici la situation pour notre cas. Notre univers se réduit à un pompier, un pyromane, une maison et un lac. Le pyromane veut incendier la maison et le pompier, disposant d'un seau, cherche à éteindre l'incendie en puisant de l'eau dans le lac avec son seau et en la versant sur la maison en feu (voir la figure 3).

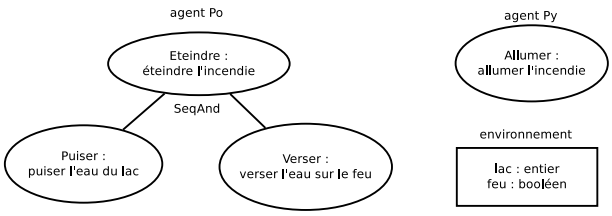

FIGURE 3 – GDT le pompier et le pyromane

Nous n'avons donc que deux agents : le pompier et le pyromane. Le premier agit chaque fois que la maison est en feu, et le second chaque fois qu'elle ne l'est pas. Pour simplifier ce cas au maximum, nous posons que le pyromane réussit toujours à allumer son feu et qu'il suffit d'un seau d'eau pour l'éteindre. Par ailleurs, le lac n'est pas alimenté.

On observe alors que le lac finit systématiquement par se vider. Même si l'explication est évidente, le phénomène correspond bien à la définition d'émergence que nous avons admise en début d'article. En effet, aucun agent ne souhaite vider le lac (le pompier veut puiser de l'eau pour éteindre le feu, il n'essaie pas de le vider). D'ailleurs, aucun des deux agents isolés ne parviendrait à le vider : le pompier seul n'agirait pas (pas de feu à éteindre) et le pyromane ne va jamais puiser d'eau.
Il s'agit donc bien d'un phénomène émergent synergique, où l'association des deux agents produit un nouvel effet. Mais ce système ne comportant que deux agents, il est facile à appréhender.

4.2 Le pompier et le pyromane avec affluent

Notre situation est ici la même que dans le cas précédent à ceci près que nous ajoutons un nouvel élément à notre système : notre lac dispose maintenant d'un affluent, représenté par un agent, qui va remplir le lac à débit constant (voir la figure 4). Sachant cela, en posant que le pompier va puiser de l'eau à un rythme régulier, on peut aisément savoir si le lac va finir par se vider ou déborder.

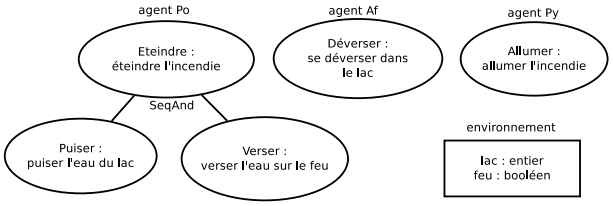

FIGURE 4 – GDT pompier, pyromane et affluent

Si le lac déborde, ce n'est pas un phénomène émergent car seul l'affluent en est responsable. En effet, le jeu du pompier et du pyromane ne va que retarder l'échéance.
À l'inverse, si le lac se vide, se sera grâce à l'association du pompier et du pyromane, l'affluent retardant l'échéance. Nous retrouvons alors le même cas que précédemment.
L'intérêt de ce cas est de présenter le fait que dans un système tous les agents ne participent pas forcément à l'établissement de la propriété émergente, et que parfois même certains y sont opposés. Dans ce cas précis, nous avons les trois types d'agents : le pompier vide le lac, le pyromane n'y touche pas et l'affluent le remplit.

4.3 Le portique

Pour cette situation, nous disposons d'un portique pour déplacer des conteneurs. Nous ne nous intéressons pas à comment le crochet agrippe le conteneur, juste à comment le positionner correctement. Pour cela, le portique dispose de deux moteurs : un pour se déplacer horizontalement le long d'un rail et un autre pour ajuster la hauteur du crochet (voir la figure 5).
Le but du système est d'ajuster précisément le crochet pour prendre ou déplacer les conteneurs. On voit facilement que la propriété "crochet bien ajusté" ne peut être satisfaite que

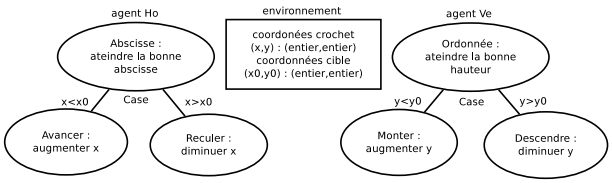

FIGURE 5 – GDT le portique

par le travail combiné des deux moteurs, un moteur seul ne pouvant gérer les deux paramètres (hauteur + alignement) en même temps.
Certaines définitions de l'émergence que nous avons évoquées réclament que le phénomène soit imprévisible, mais de récents articles sur le sujet rejettent cette idée car les phénomènes émergents ont été observés comme étant reproductibles [Gol99, Cor02]. De plus, cet aspect ne figure pas dans la définition que nous considérons dans cet article. D'après notre définition, il s'agit donc bien d'une propriété émergente.

4.4 Les 3 cinéphiles

Trois amis décident d'aller au cinéma pour la soirée. Arrivés sur place, deux films sont à l'affiche et ils doivent alors décider lequel regarder. Préférant rester ensemble, il doit donc y avoir consensus.
Pour cela, chacun va commencer par choisir un film (A ou B aléatoirement). Ils l'annoncent tous et chacun compare le choix des deux autres : si plus de la moitié des autres amis (dans ce cas, les deux) ont fait un choix différent, il change d'avis. Si la moitié ou moins (ici un ou aucun) a fait un choix différent, il ne change pas d'avis.
La figure 6 présente un modèle possible pour ce cas. Dans un soucis de lisibilité, un seul des agents est représenté sur la figure, mais les deux autres sont identiques aux indices des variables près. Ce modèle pourrait notamment être précisé au niveau du but "changer d'avis" : on pourrait le décomposer en deux sous-cas "filmA" et "filmB" à l'aide d'un autre opérateur Case.
Il s'agit bien évidement d'un problème de coordination qui va se résoudre par le phénomène de convention sociale (voir la section 2.4). Le phénomène émergent que nous attendons est donc l'apparition du consensus. Le problème est ici réduit à son minimum : trois protagonistes. Dans une telle situation, le consensus sera atteint au plus tard au second tour. En effet, seules deux situations peuvent se présenter : soit tous trois sont d'accord dès le premier tour, soit un n'est pas d'accord avec les deux autres et il changera d'avis au tour suivant (contrairement au deux autres).
Étant donnée la simplicité extrême de ce cas, un certain nombre de phénomènes pouvant être observés dans des problèmes de convention sociale à grande échelle ne le seront pas ici (notamment tout ce qui touche à la propagation d'un état). Néanmoins, rajouter un quatrième compère et associer au groupe un graphe des relations non complet (chacun ne prend en compte que les avis de certains pour se

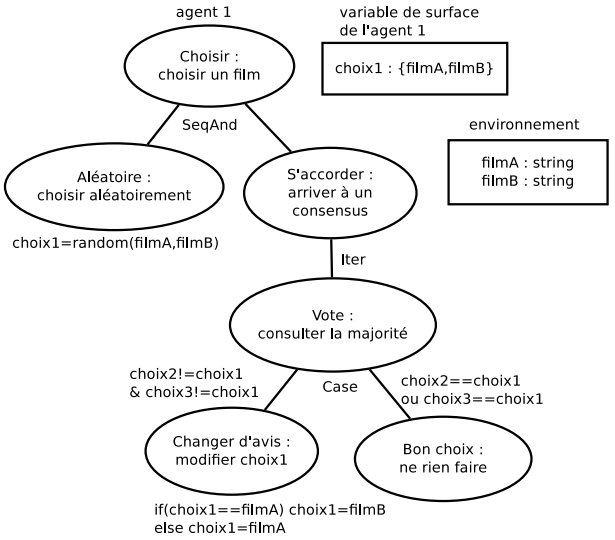

FIGURE 6 – GDT 3 cinéphiles

décider) devrait permettre d'observer certains de ces phénomènes sous leur forme la plus élémentaire. Dans cette situation, l'émergence d'un consensus peut dépendre des conditions initiales.

5 Conclusion

L'émergence est un concept encore très flou et de nombreuses définitions, parfois contradictoires, ont été avancées. Nous en avons vu quelques unes et désigné celle que nous considérons dans nos travaux. Nous avons également remarqué que cette notion assez générale pouvait être précisée par un certains nombres de types d'émergence et nous en avons défini plusieurs. Si nous n'avons pas la prétention d'avoir fait un travail exhaustif et définitif sur le sujet, nous pensons néanmoins avoir contribué à l'éclaircir.

Nous avons ensuite introduit la modélisation GDT4MAS. Cette modélisation permet de représenter visuellement le comportement des agents d'un système en plus d'être associé à un système de preuve formelle automatisable.

Nous avons finalement présenté un certain nombre de cas simples présentant des propriétés émergentes. Les propriétés de ces cas sont suffisamment simples pour être comprises et étudiées dans le détail. Ces cas sont des exemples simples permettant des expérimentations approfondies.

Le but de nos travaux est de pouvoir à terme prouver la présence d'une propriété émergente dans un système multi-agents. Pour cela, nous allons étendre le système de preuve formelle déjà associé à la modélisation et l'adapter pour les propriétés émergentes. La précision de système devrait nous permettre de définir si un phénomène émergent apparaît systématiquement dans un système, et sinon, sous quelles conditions.

Références

[And72] P. W. Anderson. More Is Different. *Science*, 177 :393–396, August 1972.

[Bed97] Mark A. Bedau. Weak emergence. *Tomberlin*, 11 :375–399, 1997.

[Cor02] Peter A. Corning. The re-emergence of "emergence" : A venerable concept in search of a theory. *Complexity*, 7(6) :18–30, July/August 2002.

[Das08] Mehdi Dastani. 2apl : a practical agent programming language. *Autonomous Agents and Multi-Agent Systems*, 16 :214–248, March 2008.

[Del02] Jordi Delgado. Emergence of social conventions in complex networks. *Artificial Intelligence*, 141 :171–185, 2002.

[Dre03] Johann Dreo. *Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues. Application en génie biomédical*. PhD thesis, Université de Paris 12, 2003.

[Gol99] Jeffrey Goldstein. Emergence as a Construct : History and Issues. *Emergence : Complexity and Organization*, 1(1) :49–72, 1999.

[IRC] IRCCyN. Roméo - A tool for Time Petri Nets analysis. *http ://romeo.rts-software.org/.*

[Lew75] *Problems of Life and Mind*. London : Trübner, 1875.

[MS09] Bruno Mermet and Gaële Simon. GDT4MAS : an extension of the GDT model to specify and to verify MultiAgent Systems. *Proc. of AAMAS 2009*, pages 505–512, 2009.

[MS10] Bruno Mermet and Gaële Simon. Specifying and Verifying Holonic Agents with GDT4MAS. *Int. Journal of Agent-Oriented Software Engineering*, 4(3) :281–303, 2010.

[MS12] Bruno Mermet and Gaële Simon. GDT4MAS : a formal model and language to specify and verify agent-based complex systems. *Studia Informatica Universalis*, 10(3) :5–32, 2012.

[MSZ] Bruno Mermet, Gaële Simon, and Bruno Zanuttini. Agent Design with Goal Decomposition Trees.

[Ode98] James Odell. Agents and Emergence. *Distributed Computing*, October 1998.

[Pey03] Sylvain Peyronnet. *Model checking et vérification probabiliste*. PhD thesis, 2003.

[SRI] SRI International. PVS. *http ://pvs.csl.sri.com*.

[Wil02] Uri Wilensky. Modeling nature's emergent patterns with multi-agent languages. In *Proceedings of EuroLogo 2002*, 2002.

[WW95] Adam Walker and Michael Wooldridge. Understanding the emergence of conventions in multi-agent systems. pages 384–389. MIT Press, 1995.

[YS03] Pinar Yolum and Munindar P. Singh. Emergent properties of referral systems, 2003.