



**HAL**  
open science

## Sur l'assignation de buts comportementaux à des coalitions d'agents

Christophe Chareton, Julien Brunel, David Chemouil

► **To cite this version:**

Christophe Chareton, Julien Brunel, David Chemouil. Sur l'assignation de buts comportementaux à des coalitions d'agents. *Approches Formelles dans l'Assistance au Développement de Logiciels*, Jun 2017, Montpellier, France. hal-01863437

**HAL Id: hal-01863437**

**<https://hal.science/hal-01863437v1>**

Submitted on 28 Aug 2018

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

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

# Sur l'assignation de buts comportementaux à des coalitions d'agents

Christophe Chareton  
LORIA CNRS  
54506 Vandœuvre-lès-Nancy

Julien Brunel  
ONERA/DTIS  
31055 Toulouse cedex 2

David Chemouil  
ONERA/DTIS  
31055 Toulouse cedex 2

## Résumé

Dans cet article, nous présentons un cadre de modélisation formelle pour l'ingénierie du besoin qui prenne simultanément en compte les buts comportementaux et les agents. Pour ce faire, nous introduisons un langage noyau, appelé KHI, ainsi que sa sémantique dans une logique de stratégies appelée USL. Dans KHI, les agents sont décrits par leurs capacités et les buts sont définis par des formules de logique temporelle linéaire. Une « assignation » associe alors chacun des buts à un ensemble (une coalition) d'agents, qui sont responsables de sa satisfaction. Nous présentons et discutons ensuite différents critères de correction pour cette relation d'assignation. Ceux-ci permettent d'évaluer la « pertinence » d'une assignation de buts à des coalitions. Ils diffèrent selon les interactions qu'ils permettent entre les coalitions d'agents. Nous proposons alors une procédure décidable de vérification pour la satisfaction des critères de correction pour l'assignation. Elle consiste à réduire la satisfaction des critères à des instances du problème de *model-checking* pour des formules d'USL dans une structure dérivée des capacités des agents.

## 1 Contexte

Si, en toute rigueur, la discipline de la modélisation du besoin ne se restreint pas à elles seules [17, 14], les approches dites par buts [18] ou par agents [2, 9] ont le vent en poupe dans la communauté idoïne (cf. les citations précédentes mais aussi [12, 15]).

En KAOS [18], la question première est de déterminer les besoins dont il faut tenir compte pour rendre compte d'un *système* au sein d'un *environnement*, le tout formant un *système global* à mettre au point. Celui-ci doit répondre à des *buts* et est constitué d'*agents* (entités actives).

Un *but* est défini comme un *énoncé prescriptif* sous la responsabilité d'agents du système global. Les buts peuvent être de toutes sortes (on retrouve les traditionnelles taxonomies autour des buts *non-fonctionnels* [11]). Mais on distingue en particulier les buts *comportementaux* qui caractérisent des traces et peuvent donc faire l'objet d'une formalisation dans une logique temporelle telle que LTL.

Bien que partageant superficiellement de nombreuses notions avec KAOS, TROPOS se concentre avant tout sur la notion d'*acteur*, défini comme un agent *intentionnel*. Un tel agent est muni de buts qu'il souhaite voir remplis mais dont la satisfaction, partielle comme complète, n'est pas nécessairement de sa responsabilité. Celle-ci peut être déléguée à d'autres acteurs. TROPOS [2] pousse ainsi à l'explicitation des liens de dépendance et de collaboration entre acteurs. Ceci s'explique en particulier par le fait que les systèmes visés par la méthode sont susceptibles de comprendre des acteurs « humains » ou institutionnels.

TROPOS a aussi fait l'objet d'une proposition formelle visant à étudier dans quelle mesure des acteurs peuvent contribuer à satisfaire des buts pour d'autres acteurs. L'approche en question [9, 10] introduit à cette fin les notions, dites « sociales », de *rôle*, d'*engagement* (*commitment*) et de *protocole*. Le rôle représente le comportement *attendu* des acteurs. Une *assignation* de rôles à des acteurs est alors évaluée au moyen d'un critère de correction. Celui-ci revient essentiellement à vérifier que les capacités d'un acteur entraînent les conséquents des engagements où le rôle assigné apparaît comme débiteur.

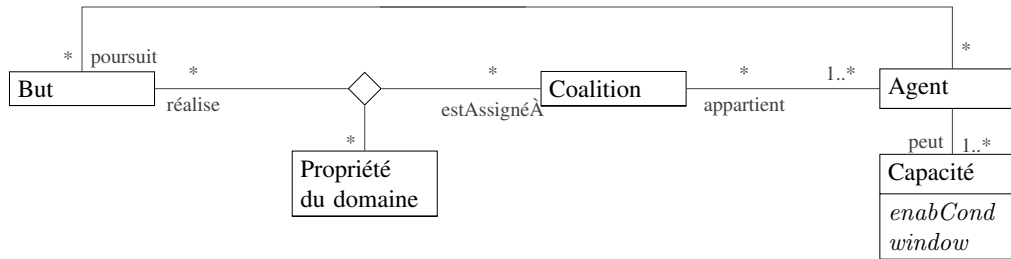


FIGURE 1 – Métamodèle du langage KHI.

Si la dimension « sociale » de cette proposition est plus forte qu'en KAOS, un certain nombre de faiblesses demeurent. Premièrement, on ne peut pas assigner de rôle à un ensemble d'acteurs qui collaboreraient pour l'assurer. Comme en KAOS, la question du partage de variables d'état entre plusieurs acteurs n'est pas traitée ; et les capacités des acteurs demeurent indépendantes du contexte. Surtout, la restriction du formalisme à la logique propositionnelle constitue une lacune importante : l'emploi d'un formalisme temporel, à l'image de ce qui est fait en KAOS, permettrait de véritablement caractériser des comportements attendus des acteurs.

Ces considérations nous ont mené à l'étude formelle ce que nous avons appelé le *problème de l'assignation*. Informellement, étant donnée une assignation de buts à des coalitions (ensembles) d'agents, la question est de déterminer dans quelle mesure ces dernières sont en mesure de satisfaire les premiers, y compris en tenant compte des interactions avec les autres coalitions. Cette question nous a amenés à définir un langage de modélisation appelé KHI et à élaborer une logique temporelle multi-agent, USL, fournissant un domaine sémantique permettant de formaliser et d'évaluer le problème de l'assignation selon différents critères.

## 2 KHI

Le cadre de modélisation KHI (figure 1) comprend un ensemble restreint de concepts destinés à permettre la mise en œuvre d'une modélisation du besoin à *la fois* par buts et par agents. Il peut en un sens être vu comme une union enrichie (des aspects principaux) de KAOS et des variations autour de TROPOS. Une première version de KHI a été introduite dans [4, 5] ; nous présentons ici une version simplifiée d'une seconde version, plus aboutie et décrite en détail dans [3].

Comme en TROPOS, les buts sont poursuivis par des agents, possiblement mais pas nécessairement aptes à contribuer à leur satisfaction. Les agents disposent de capacités plus fines que dans les propositions précédentes : une capacité ne peut s'exercer que dans certaines conditions (*enabCond*) et consiste à pouvoir agir sur une variable du système mais seulement dans une certaine fenêtre (*window*), c'est-à-dire dans un ensemble fini de valeurs possible. Originalité supplémentaire : les agents sont regroupés en coalitions (ensembles) qui se voient assigner les buts à remplir (autrement dit, l'assignation n'est pas restreinte à des agents isolés).

Par ailleurs, la formalisation des buts suit le même principe qu'en KAOS : les buts sont décrits dans la logique LTL. Ces formules de LTL apparaîtront comme sous-formules de formules de la logique USL sur laquelle s'appuie la formalisation complète de KHI. Or cette logique repose sur des modèles admettant des exécutions finies. La sémantique utilisée pour LTL pour le critère de correction du raffinement des buts par des sous-buts et de l'opérationnalisation s'appuie donc sur des traces possiblement finies (et non-vides), de manière par ailleurs standard (techniquement comme la logique  $LTL^n$  d'[13]).

L'existence d'exécutions finies résulte du choix de modélisation de l'assignation entre buts et coalitions d'agents : celle-ci est définie comme une application de l'ensemble des buts dans les coalitions non vides d'agents (donc de type  $\text{But} \rightarrow \mathcal{P}_{>0}(\text{Agent})$ ). On remarque qu'il est tout à fait possible pour un agent de se retrouver dans plusieurs coalitions. Dès lors, la participation de cet agent à ces différentes coalitions peut induire des spécifications contradictoires pour son comportement. Cet agent pourra alors

être engagé envers la réalisation d'actions *incompatibles* entre elles, depuis un même état. La présence d'exécutions finies dans notre formalisme provient donc du fait que nous avons souhaité favoriser la description fine des capacités des agents à composer leurs comportements pour la satisfaction de différents buts. Par ailleurs, il faut noter que notre formalisme permet ensuite de spécifier des modèles où les exécutions sont forcément infinies.

### 3 Problème de l'assignation

L'évaluation d'une assignation ne se fait pas de manière binaire. Un ensemble de critères permet de la caractériser plus finement ; en particulier en déterminant si des coalitions peuvent interagir de sorte à voir assurée la satisfaction de leurs buts :

**Correction locale** Ce critère consiste à s'assurer que chaque but est assigné à une coalition capable d'assurer sa satisfaction, quoi que fassent les autres agents.

**Correction globale** Le critère de correction locale est insuffisant dès lors qu'un agent appartient à plusieurs coalitions, car rien ne dit qu'il est apte à participer à toutes ces coalitions (en vue de réaliser leurs buts respectifs) *à la fois* (les comportements demandés pourraient être contradictoires). Le critère de *correction globale* demande donc s'il y a *un* comportement (par agent) qui permette à chaque coalition de satisfaire les buts qui lui sont assignés. Cet unique comportement permet de s'assurer de la cohérence des choix des agents.

**Collaboration** Le critère précédent peut être trop fort au sens où il stipule que chaque coalition doit pouvoir satisfaire ses buts assignés, quoi que fassent les autres agents. Or on peut souhaiter que certaines coalitions *collaborent* avec d'autres pour assurer la satisfaction de leurs buts.

**Contribution** Les trois critères précédents reposent sur l'hypothèse qu'il est possible de contrôler (et spécifier) tous les agents dans le système. Or il se pourrait que ce ne soit pas le cas. Le critère de *contribution* pose alors la question de savoir si une coalition indépendante, en satisfaisant ses propres buts assignés, est en mesure de produire des « effets de bord » qui contribuent à la satisfaction des buts assignés à d'autres coalitions.

Ces critères sont tous traduits en problèmes de *model-checking* dans la logique USL. On aboutit donc à des problèmes de la forme  $\mathfrak{R} \models c$ , où  $c$  est une formule d'USL traduisant un critère et où  $\mathfrak{R}$  est une CGS (*Concurrent Game structure*), soit un système de transitions particulier.

Intuitivement, les états de la CGS  $\mathfrak{R}$  sont déterminés par les valuations possibles des variables décrivant le système (et compatibles avec les propriétés du domaine). En ce qui concerne la fonction de transition, les états suivants d'un état donné dépendent de celui-ci ainsi que de l'intersection des choix des agents, choix eux-mêmes déterminés par leurs capacités (pour cette raison, la fenêtre d'action d'un agent doit être exprimée comme un ensemble fini de valeurs possibles pour une variable donnée).

Pour la formule  $c$ , il s'agit à chaque fois d'exprimer que des coalitions d'agents sont en mesure d'assurer la satisfaction de propriétés temporelles, en tenant compte du fait que les autres coalitions agissent elles-aussi et qu'un agent peut appartenir à plusieurs coalitions. Les coalitions pouvant poursuivre des objectifs variés, il faut pouvoir spécifier qu'un agent peut enrichir de *différentes* manières son comportement selon les objectifs auxquels il concourt.

### 4 Logique des stratégies actualisables

Ces motifs mènent naturellement vers les *logiques temporelles multi-agents*. Toutefois, aucune proposition ne présentant les caractéristiques nécessaires<sup>1</sup> à l'élaboration des critères, nous avons défini notre propre logique : USL. Dans les logiques telles qu'ATL [1] ou SL [8, 16], chaque agent est muni d'une *stratégie* qui, en fonction du déroulement du jeu jusqu'alors, indique quelle action il choisit, ce

---

1. En particulier, donc, la possibilité pour un agent de suivre diverses « stratégies » et de les raffiner de diverses façons.

qui concourt à établir une décision et donc à déterminer l'état successeur. Eu égard à KHI, cette construction contribue à matérialiser la notion selon laquelle un agent a le moyen (lire : une stratégie) d'assurer une propriété. Toutefois, ici, en conformité avec les critères d'assignation, il est nécessaire de pouvoir composer des stratégies de plusieurs manières différentes. Dans USL, un agent auquel est assigné une stratégie  $\varsigma$  peut se voir à nouveau assigner une autre stratégie  $\varsigma'$ . Il composera alors son comportement de manière à satisfaire à la fois  $\varsigma$  et  $\varsigma'$ . On dit qu'il *refine* sa stratégie. Ceci nous a menés, pour USL, à l'utilisation de stratégies *non-déterministes* :

**Définition 4.1** (Multistratégie). *Une multistratégie est une application qui, à tout déroulement<sup>2</sup> d'un jeu, associe un ensemble non-vide d'actions.*

Nous sommes maintenant en mesure de définir une syntaxe pour USL, avec les contraintes suivantes : (a) pouvoir raisonner sur les multistratégies, ce qui mène à l'introduction de modalités *ad hoc* ; (b) pouvoir se référer à des multistratégies données, d'où une notion de *variable* associée (et donc de *quantificateur*) ; (c) pouvoir composer, pour un même agent, plusieurs multistratégies différentes dans des sous-formules distinctes, et ce sans *révoquer* les multistratégies déjà associées, ce qui nécessite des opérateurs liant ou révoquant explicitement une multistratégie à un agent.

**Définition 4.2** (Formules d'USL). *Soient  $Ag$  un ensemble d'agents,  $At$  un ensemble de propositions, et  $X$  un ensemble de variables de multistratégies. Alors, l'ensemble des pseudoformules d'USL sur  $(Ag, At, X)$  est engendré par la grammaire suivante :*

- Pseudoformules d'états :  $\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle x \rangle\rangle\varphi \mid (A \triangleright x)\psi \mid (A \not\triangleright x)\psi$
- Pseudoformules de chemins :  $\psi ::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid \psi \cup \psi$

où  $p \in At$ ,  $x \in X$  et  $A \subseteq Ag$ .

*Une formule (bien formée) d'USL est une pseudoformule dans laquelle chaque variable de multistratégie introduite par un quantificateur est fraîche par rapport à la portée dans laquelle elle apparaît.*

Naturellement, lors de la détermination de la possible satisfaction d'une formule dans une CGS, la présence de quantificateurs et d'opérateurs de liaison et révocation doit être prise en compte dans un *contexte* (d'évaluation). Un tel contexte  $\kappa$  est un couple comprenant : (a) une *assignation*  $\alpha$ , qui a pour objet de mémoriser la multistratégie effective associée à chaque variable liée rencontrée jusqu'alors ; (b) et un *engagement*  $\gamma$ , c'est-à-dire l'inventaire, pour chaque agent, des (variables de) multistratégies selon lesquelles il joue. Ce couple doit par ailleurs être cohérent : toute variable de multistratégie listée dans l'engagement doit disposer d'une instanciation, donnée par l'assignation.

Nous renvoyons le lecteur à [6] pour une exposition plus fine, qui fait appel à un certain nombre de détails techniques, mais nous donnons ici une intuition des points saillants de la sémantique :

- L'opérateur  $\langle\langle x \rangle\rangle$  est un quantificateur existentiel sur les multistratégies : une formule  $\langle\langle x \rangle\rangle\varphi$  est vraie dans un état  $s$ , dans un contexte  $\kappa$ , ssi il existe une multistratégie  $\varsigma$  t. q. la formule  $\varphi$  est vraie dans le contexte  $\kappa$  enrichi du fait que  $x$  est instanciée par  $\varsigma$ . On peut aussi définir un quantificateur universel de la sorte :  $\llbracket x \rrbracket\varphi := \neg\langle\langle x \rangle\rangle\neg\varphi$ .
- Une formule  $(A \triangleright x)\psi$  est vraie dans un état  $s$ , dans le contexte  $\kappa$ , ssi la formule  $\psi$  est vraie dans toute exécution *issue* de  $s$  et  $\kappa[A \oplus x]$ , où  $\kappa[A \oplus x]$  est le contexte  $\kappa$  enrichi du fait que les agents dans  $A$  sont maintenant liés à la multistratégie instanciant  $x$  dans  $\kappa$  (en plus des multistratégies auxquelles ils étaient éventuellement déjà liés).

Les *issues*  $out(\kappa, s)$  d'un état dans un certain contexte constituent les exécutions possibles à partir de cet état dans la CGS si chaque agent ne joue que des actions autorisées par toutes les multistratégies auxquelles il est lié dans ce contexte.

Il faut remarquer que parmi ces issues, certaines exécutions peuvent être finies, ce en raison de la possibilité pour un agent de jouer en même temps selon des multistratégies contradictoires (c'est-à-dire renvoyant des ensembles disjoints d'actions). Ceci implique l'usage d'une sémantique idoine pour les opérateurs temporels dans les formules de chemin.

2. C'est-à-dire un préfixe non-vide d'exécution dans la CGS.

—  $(A \not\triangleright x)$  délie les agents dans  $A$  de la multistratégie instanciant  $x$  dans le contexte courant.

**Theorème 4.1** ([6]). *Pour finir, on remarque que : (a) USL est strictement plus expressive que SL [8, 16]; (b) comme pour cette dernière, la satisfaisabilité est indécidable; (c) en revanche, le model-checking sur une CGS finie<sup>3</sup> est décidable, quoique en temps non-élémentaire, ce qui est aussi le cas de SL.*

## 5 Évaluation d'une assignation

Les caractéristiques uniques d'USL (en particulier le raffinement de multistratégies) permettent de formaliser les critères de correction présentés en § 3. On ne présente pas ici la procédure (complexe) de construction d'une CGS *finie* à partir d'un modèle KHI, cf. [3, 7]. Cette finitude dépend en particulier du type de formules atomiques et de la forme des fenêtres que KHI permet de spécifier. Elle rend décidable la vérification des critères ci-dessous.

Soit un modèle KHI et soit  $G$  un ensemble de buts. On note  $\mathcal{A}$  l'application qui à tout but associe la coalition qui lui est assignée. Étant donnée une coalition  $A = \{a_1, \dots, a_n\}$  et  $\vec{x}$  un vecteur de variables de multistratégies; on note  $(A \triangleright \vec{x})$  pour  $(a_1 \triangleright x^{a_1}) \dots (a_n \triangleright x^{a_n})$ .

**Correction locale** L'assignation est localement correcte ssi pour tout but  $g \in G$ , il existe un vecteur de multistratégies t. q., en les jouant, les agents concernés peuvent assurer la satisfaction de  $g$  :

$$\text{LC}_{\mathcal{A}}[G] := \bigwedge_{g \in G} [\langle\langle \vec{x}_g \rangle\rangle (\mathcal{A}_g \triangleright \vec{x}_g) g]$$

**Correction globale** Ici, on considère un unique vecteur de multistratégies, ce qui impose aux agents de jouer en cohérence ( $\vec{x}_g$  représente la partie de  $\vec{x}$  qui concerne les agents dans  $\mathcal{A}_g$ ) :

$$\text{GC}_{\mathcal{A}}[G] := \langle\langle \vec{x} \rangle\rangle \left[ \bigwedge_{g \in G} (\mathcal{A}_g \triangleright \vec{x}_g) g \right]$$

**Collaboration** Pour qu'une coalition associée à un but  $h$  collabore à la satisfaction de l'ensemble de buts  $G$ , il faut un vecteur de multistratégies  $\vec{x}_h$  t. q. si les agents dans  $\mathcal{A}_h$  les jouent, alors ces multistratégies assurent *à la fois* que le but  $h$  est satisfait et que l'évolution du modèle est contrainte de sorte à ce que l'assignation devienne globalement correcte pour  $G$ . Soit donc  $h$  un but t. q.  $h \notin G$ . :

$$\text{Coll}_{\mathcal{A}}[h, G] := \langle\langle \vec{x}_h \rangle\rangle (\mathcal{A}_h \triangleright \vec{x}_h) (h \wedge \text{GC}_{\mathcal{A}_G}[G])$$

**Contribution** La contribution est définie comme une variante universellement quantifiée de la collaboration; les agents concernés doivent pouvoir assurer  $h$  et tout vecteur de multistratégies qui permet aux dits agents d'assurer  $h$  doit aussi contraindre l'évolution du système de sorte à ce que l'assignation devienne globalement correcte pour  $G$  :

$$\text{Contr}_{\mathcal{A}}[h, G] := [\langle\langle y_h \rangle\rangle (\mathcal{A}_h \triangleright \vec{y}_h) h] \wedge \left[ \llbracket \vec{x}_h \rrbracket \left\{ (\mathcal{A}_h \triangleright \vec{x}_h) h \rightarrow ((\mathcal{A}_h \triangleright \vec{x}_h) \text{GC}_{\mathcal{A}_G}[G]) \right\} \right]$$

## 6 Perspectives

KHI constitue à notre connaissance la première proposition de cadre de modélisation *formelle* pour l'ingénierie du besoin qui prenne simultanément en compte les buts comportementaux et les agents.

Ceci étant, le cadre doit clairement être amélioré. Tout d'abord, la sémantique de KHI ne fait pas appel à toute l'expressivité d'USL. Une question importante en termes d'applicabilité de nos propositions serait de déterminer s'il est possible de caractériser un fragment d'USL suffisant pour la traduction de KHI et pour lequel le *model-checking* disposerait d'une complexité « raisonnable » en pratique.

3. C'est-à-dire t. q. les ensembles d'états et d'actions sont finis.

De son côté, USL constitue aussi une proposition originale. Elle bénéficie par ailleurs de propriétés métathéoriques comparables aux logiques similaires et dispose d'un fort pouvoir expressif dont il s'agirait de creuser les conséquences en profondeur.

Hors de l'ingénierie du besoin proprement dite, USL pourrait bien avoir des applications intéressantes. Ainsi de l'analyse de « systèmes de systèmes », ensembles dans lesquels, en particulier, les sous-systèmes apparaissent comme des entités autonomes dont les capacités sont connues et qui disposent de leurs objectifs propres en sus de ceux du système global. Une autre application possible réside dans la sécurité : en effet, USL permet en principe de raisonner sur des agents dont les objectifs sont malveillants.

Pour finir, USL permet de raisonner sur le comportement (les multistratégies) possible des agents. Une question supplémentaire serait celle du comportement *effectif* du système. Cette question appelle certainement plusieurs directions de recherche autour de la *synthèse* de contrôleurs, de normes, de stratégies...

## Références

- [1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5) :672–713, 2002.
- [2] J. Castro, M. Kolp, and J. Mylopoulos. A requirements-driven development methodology. In *Proceedings of the 13th International Conference on Advanced Information Systems Engineering, CAiSE '01*, pages 108–123, London, UK, UK, 2001. Springer-Verlag.
- [3] C. Chareton. *Modélisation formelle d'exigences et logiques temporelles multi-agents*. PhD thesis, Institut supérieur de l'aéronautique et de l'espace (ISAE), Université de Toulouse, 2014.
- [4] C. Chareton, J. Brunel, and D. Chemouil. A Formal Treatment of Agents, Goals and Operations Using Alternating-Time Temporal Logic. In *Formal Methods, Foundations and Applications (SBMF)*, 2011.
- [5] C. Chareton, J. Brunel, and D. Chemouil. Vers une sémantique des jeux pour un langage d'ingénierie des exigences par buts et agents. In *Approches Formelles dans l'Assistance au Développement de Logiciels (AFADL)*, 2012.
- [6] C. Chareton, J. Brunel, and D. Chemouil. A Logic with Revocable and Refinable Strategies. *Information and Computation*, 242 :157–182, 2015.
- [7] C. Chareton, J. Brunel, and D. Chemouil. Evaluating the assignment of behavioral goals to coalitions of agents. In *Proc. 18th Brazilian Symposium, SBMF 2015, Belo Horizonte, Brazil, 2015*. Springer, 2015.
- [8] K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. *Inf. Comput.*, 208(6) :677–693, 2010.
- [9] A. K. Chopra, F. Dalpiaz, P. Giorgini, and J. Mylopoulos. Modeling and reasoning about service-oriented applications via goals and commitments. In *Proceedings of the 22nd International Conference on Advanced Information Systems Engineering (CAiSE)*, volume 6051 of *LNCS*, pages 113–128. Springer, 2010.
- [10] A. K. Chopra and M. P. Singh. Multiagent commitment alignment. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems*, pages 937–944. IFAAMAS, 2009.
- [11] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos. *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers, 2000.
- [12] E. Dubois, P. Du Bois, and M. Petit. ALBERT : An agent-oriented language for building and eliciting requirements for real-time systems. In *HICSS (4)*, pages 713–722, 1994.
- [13] C. Eisner, D. Fisman, J. Havlicek, Y. Lustig, A. McIsaac, and D. V. Campenhout. Reasoning with temporal logic on truncated paths. In *Proceedings of the 15th International Conference on Computer Aided Verification (CAV)*, pages 27–39, 2003.
- [14] M. Jackson. *Problem Frames*. Springer, 2001.

- [15] T. P. Kelly and R. A. Weaver. The goal structuring notation - a safety argument notation. In *Proceedings of the Dependable Systems and Networks 2004 Workshop on Assurance Cases*, 2004.
- [16] F. Mogavero, A. Murano, and M. Y. Vardi. Reasoning about strategies. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010*.
- [17] B. Nuseibeh and S. M. Easterbrook. Requirements engineering : a roadmap. In *22nd International Conference on Software Engineering, Future of Software Engineering Track*, 2000.
- [18] A. van Lamsweerde. *Requirements Engineering — From System Goals to UML Models to Software Specifications*. Wiley, 2009.