
Argumentation et négociation pour le choix d’emballages alimentaires biodégradables

Patrice Buche^{1,2}, Madalina Croitoru², Jérôme Fortin^{1,2},
Patricio Mosse^{1,2}, Nouredine Tamani¹, Rallou Thomopoulos^{1,2}

1. UMR IATE, CIRAD-INRA-Supagro-UM2,
2 place P. Viala, 34060 Montpellier cedex 1
{buche,rollou,tamani}@supagro.inra.fr,{fortin.je,patricio.mosse}@gmail.com
2. INRIA GraphIK, LIRMM, 161 rue Ada, F-34095 Montpellier cedex 5
Madalina.Croitoru@lirmm.fr

RÉSUMÉ. L’objectif de ce papier est d’agrèger les préférences de différents acteurs d’un domaine, en utilisant les techniques d’argumentation puis de négociation, pour l’aide à la prise de décision. Ce travail s’insère dans le cadre du projet européen EcoBioCap (ECOefficient BIOdegradable Composite Advanced Packaging), dont le but est de fournir aux filières agroalimentaires européennes de nouveaux emballages éco-efficaces, personnalisables et biodégradables. Une des tâches du projet EcoBioCap est de fournir un système d’aide à la décision facilitant le choix d’un emballage pour un produit donné. Ce choix d’emballage doit satisfaire au mieux de nombreux critères, pas toujours compatibles entre eux et exprimés par différents acteurs. Pour ce faire, une logique de préférences, composée d’un langage et d’une sémantique, a été proposée afin de permettre aux intervenants d’exprimer leurs préférences dans un cadre formel. Cette logique a été par la suite intégrée dans un système d’argumentation (instance de ASPIC+) afin d’extraire à partir des préférences exprimées les ensembles de conclusions justifiées appelés extensions. Ces dernières présentant des conclusions contradictoires sont enfin soumises à un processus de négociation pour la résolution des conflits nécessaire à l’aide à la prise de décision. Un prototype a été implémenté afin de valider notre approche avec les experts du projet.

ABSTRACT. The aim of the paper is to aggregate preferences expressed by different actors. The aggregation process is based on argumentation techniques enriched with a negotiation phase. The final aim of this process is to support decision making. The work presented here is a part of the European Project EcoBioCap (ECOefficient BIOdegradable Composite Advanced Packaging) that aims at providing new, eco-efficient packaging to european food producers. One of the tasks of the project is to provide a decision support system allowing to help into the choice of a packaging for a given food product. Such packaging must satisfy, at its best, preferences expressed by different actors on several criteria, not always mutually compatible. To reach this goal, we have proposed a preference logic that allows the actors to express formally their preferences. This language has then been used as the basis of an argumentation system, instantiated from

ASPIC+. This allowed, by the means of extensions, to obtain maximally consistent packaging properties. When several extensions contain desirable features, a negotiation process is realized in order to add knowledge to the system which will eventually allow certain inconsistencies to be resolved. An implementation has been performed in order to test our approach with domain experts.

MOTS-CLÉS : argumentation, négociation, prise de décision, consensus, langage de préférences, ASPIC+.

KEYWORDS: argumentation, negotiation, decision making, conflict resolution, preference language, ASPIC+.

1. Introduction

Ce papier traite de la question d'agrèger des préférences non indépendantes en utilisant des méthodes et outils issus de l'argumentation et de la négociation. Ces recherches sont motivées par le projet Européen EcoBioCap (*ECoefficient BIOdegradable Composite Advanced Packaging*).

Dans ce projet, un logiciel d'aide à la conception d'emballage est en cours de développement. Ce logiciel doit inclure un module facilitant la prise de décision de groupe. Il doit permettre d'agrèger les préférences de différents acteurs au sein d'une filière afin de pouvoir recommander une décision finale. L'argumentation est un moyen naturel de formaliser les raisons expliquant pourquoi telle ou telle action est préférable.

Selon (Kiesling, 2002) un langage de représentation de préférences doit avoir une définition et une sémantique précise, et doit à la fois être déclaratif, extensible (c'est-à-dire que l'on peut construire des préférences complexes à partir de préférences simple) et tolérant aux conflits. De nombreux travaux sur l'expression de préférences ont été menés dans un contexte de recherche d'information. Par exemple (Kaci, 2009) définit des préférences permettant de filtrer des réponses d'une requête utilisateur, (Kiesling, Köstler, 2002) permettent de définir des contraintes dures (dont la satisfaction est obligatoire) et souples (qui sont des souhaits non forcément satisfaits), (Bouzeghoub *et al.*, 2007 ; Bouzeghoub, Kostadinov, 2005) proposent de définir des préférences dans des profils utilisateurs et donnent un modèle générique d'expression de préférences.

Dans notre contexte, ces méthodes d'expression et d'agrégation de préférences sont limitées. En effet, elles ne tiennent pas compte des différents points de vue et explications associées aux préférences exprimées par les différents intervenants. Des travaux récents proposent, par ailleurs, des méthodes permettant d'exprimer des préférences sur des arguments pour la prise de décision (Amgoud, Vesic, 2011 ; Kaci, Torre, 2008 ; Modgil, 2009 ; Amgoud, Cayrol, 2002). À l'inverse de ces approches, notre objectif est d'agrèger des préférences soutenues par des arguments (Bourguet *et al.*, 2010), d'où notre méthode basée sur l'utilisation d'un système d'argumentation structuré instancié à l'aide d'un langage logique d'expression de préférences.

Le système décrit dans ce papier est une instanciation du système d'argumentation APSIC+ (Prakken, 2011). L'utilisation de ce modèle est une des bases qui nous permet de garantir la satisfaction d'un certain nombre de critères importants dont les postulats de rationalité exprimés dans (Caminada, Amgoud, 2005). ASPIC+ propose un cadre formel abstrait, mais moins abstrait que le cadre de Dung introduit dans (Dung, 1995), instanciable à l'aide de n'importe quel langage logique. Une grande partie de ce travail repose donc sur la définition d'une telle logique, permettant de représenter des préférences utilisateur concernant les caractéristiques des emballages.

Le papier est organisé de la manière suivante. Dans la section 2, nous présentons le projet applicatif EcoBioCap qui est le support de cette étude, ainsi qu'un scénario réaliste d'agrégation de préférences dans le cadre de ce projet. Dans la section 3, nous introduisons notre système d'argumentation basé sur une logique des préférences. Un langage d'expression des préférences ainsi que sa sémantique associée sont proposés. En section 4, nous détaillons l'étape de prise de décision après le processus d'argumentation à l'aide d'une approche de négociation. La sortie du processus d'argumentation est l'ensemble des extensions préférées, ayant un ensemble de conclusions conflictuelles. La section 5 est consacrée à l'implémentation et aux développements logiciels effectués afin de valider notre approche auprès des experts du projet. Enfin, la section 6 conclut le papier en rappelant les diverses contributions présentées dans le cadre du projet ECOBIOCAP ainsi que les perspectives qui leur sont associées.

2. Un scénario d'agrégation dans le cadre du projet EcoBioCap

Le projet européen EcoBioCap (*ECO*efficient *BIO*degradable *CO*mposite *ADV*anced *PA*ckaging) a pour objectif de proposer de nouveaux emballages à la fois éco-efficaces, biodégradables et personnalisables aux consommateurs européens. La figure 1 illustre le cycle de développement du projet. La conception d'un emballage passe par l'étude des interactions entre l'aliment et son emballage à différents niveaux : moléculaire, nanoscopique et microscopique. Le projet associe des laboratoires de recherche et des partenaires industriels afin de permettre le passage à l'échelle industrielle des nouveaux emballages développés.

Le livrable principal du *Work Package 1* (WP1) du projet est la création d'un logiciel d'aide à la sélection, pour un produit frais respirant, de l'emballage satisfaisant au mieux les préférences exprimées par les différents acteurs de la filière agroalimentaire.

La figure 2 montre les différents composants du système d'aide à la décision actuellement en construction (Destercke *et al.*, 2011). Une base de données permet de connaître les paramètres respiratoires des aliments. Une autre base de données stocke les caractéristiques des différents matériaux d'emballage (perméabilité à l'oxygène, au dioxyde de carbone, biodégradabilité, transparence, etc.). Une troisième base de données permet d'accéder aux préférences des acteurs du projet.

Le système d'aide à la décision procède de la manière suivante :

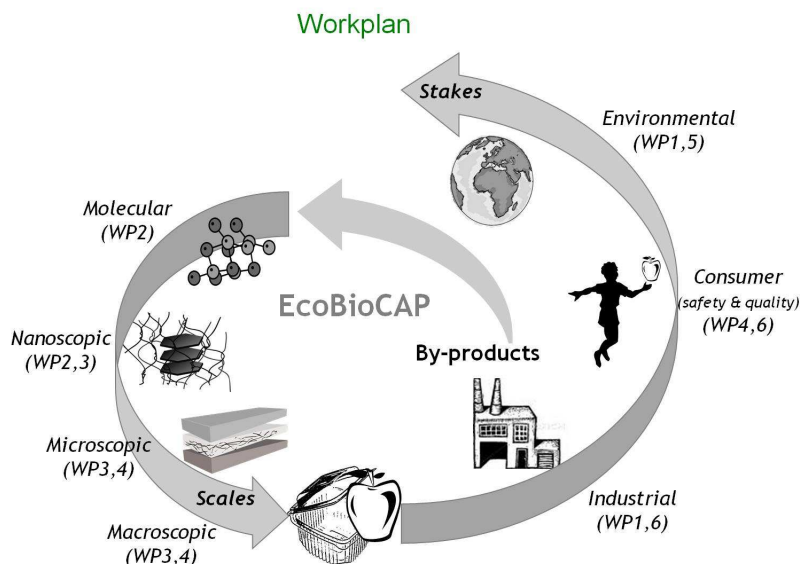


Figure 1. Cycle de développement du projet EcoBioCap

1. À partir des préférences des acteurs (par exemple température de stockage souhaitée, volume de l'emballage, etc.) et des paramètres respiratoires d'un aliment, le système calcule la perméabilité optimale de l'emballage idéal pouvant conserver au mieux l'aliment.

2. Cette perméabilité optimale est ensuite utilisée conjointement avec les autres préférences des acteurs pour créer une requête multicritère adressée à la base de données d'emballages. Cette requête permet d'obtenir une liste d'emballages classés du plus pertinent au moins pertinent au regard des préférences exprimées.

L'agrégation des préférences des acteurs est réalisée avant la phase d'interrogation du système d'aide à la décision. Pour recueillir leurs préférences, différents questionnaires ont été envoyés aux acteurs (chercheurs, fabricants d'emballages, fabricants de produits alimentaires, consommateurs de différents pays). Les questions posées portent sur les caractéristiques souhaitées des emballages (prix, durée de vie et cycle de vie de l'emballage, présence de nanoparticules, biodégradabilité, etc.). Pour chacun de ces critères, les acteurs ont été amenés à quantifier son importance, à indiquer leurs valeurs préférées et la raison qui motive leur choix. L'importance d'un critère doit permettre de privilégier l'optimisation de certains critères vis-à-vis d'autres critères. Par exemple, le respect des critères sanitaires garantissant une bonne conservation peut être considéré plus prioritaire que la transparence de l'emballage.

Nous décrivons dans la suite de cette section un scénario d'agrégation de préférences basé sur les résultats de ces enquêtes.

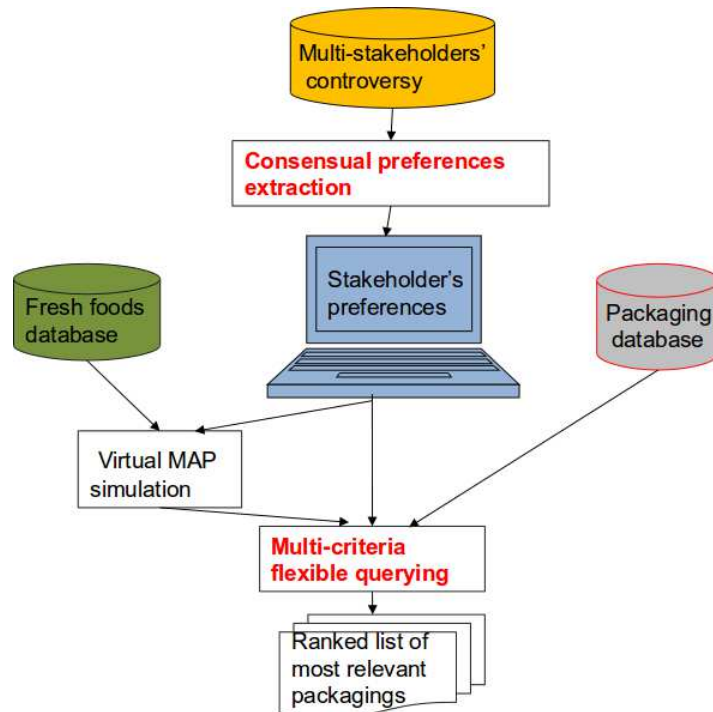


Figure 2. Système d'aide à la décision d'EcoBioCap

Dans ce scénario, nous considérons deux experts. Le premier est un producteur de fromage de chèvre artisanal et le second une multinationale agroalimentaire, produisant entre autre des fromages de chèvre.

Dans la suite, nous présentons le type de préférences que nous avons recueillies dans les enquêtes, et comment nous pouvons les agréger, à l'aide d'arguments expliquant les choix effectués. Cette phase d'agrégation a pour but d'obtenir une liste unique des valeurs préférées pour chaque critère, à partir des différentes listes des valeurs préférées par chaque acteur. Les enquêtes ont été construites pour recenser les techniques actuelles d'emballage, ainsi que les attentes des acteurs en matières d'innovations techniques. Ces enquêtes servent dans le projet de base de réflexion pour les recherches et développement de nouveaux emballages et comme connaissances de base de notre système d'aide à la décision. L'enquête est divisée en plusieurs parties :

- description de matériaux d'emballage,
- conditions de stockage et de transport,
- attentes du consommateur,
- classement des critères,
- autres limitations.

Nous décrivons ici les arguments présentés pour les 4 critères suivants : couleur de l'emballage, durée de vie de l'aliment, type de machine utilisé pour l'emballage, mode d'étiquetage.

– *Couleur* : les valeurs pour ce critère sont $\{vert, mix\ and\ match, transparent\}$;

- L'expert 1 nous dit : “mon premier choix pour la couleur est le *vert* ou *mix and match* (que l'on pourrait traduire par mélange de transparence et de couleur) car c'est de cette façon que les consommateurs nous connaissent. Si ces choix ne sont pas possibles, alors, nous préférons un emballage *transparent* qui permet au consommateur de voir le produit”.

- L'expert 2 nous dit : “Je préfère l'emballage *mix and match* car nos consommateurs nous reconnaissent grâce à lui. Si ce n'est pas possible, je préfère un emballage *transparent* à *vert* car le *vert* est utilisé par des concurrents”.

Pour des raisons de marketing, les valeurs préférées pour l'expert 1 sont *vert* et *mix and match*, *transparent* est une option acceptable pour lui, mais avec une préférence moindre. Comme *mix and match* est la valeur préférée de l'expert 2 et une valeur acceptable pour l'expert 1, c'est donc cette dernière valeur qui devrait être préconisée par l'opérateur d'agrégation.

– *Mode d'étiquetage de l'emballage* : les valeurs possibles sont $\{Autocollant, Impression\ Directe\}$.

- L'expert 1 déclare : “Je préfère une impression directe car les autocollants peuvent être enlevés par les consommateurs”,

- L'expert 2 nous dit : “Je préfère les autocollants plutôt qu'une impression directe car ils sont plus facilement personnalisables en fonction du consommateur visé. De plus cette technique n'est pas chère”.

Les préférences des deux acteurs sont contradictoires.

– *Type de machine utilisé pour l'emballage* : les valeurs possibles sont $\{HFFS, VFFS, HM\}$, où *HFFS* correspond à une machine à scellement de film horizontal, *VFFS* désigne une machine à scellement de film vertical et *HM* correspond à la mise sous emballage manuelle ;

- L'expert 1 déclare : “Il vaut mieux emballer le nouveau produit à la main (*HM*) car c'est la méthode d'emballage que nous avons l'habitude d'utiliser et la machine *HFFS* que nous possédons est déjà utilisée pour emballer un autre fromage. De plus, utiliser une telle machine nous oblige à poser des autocollants sur les emballages (indiquant les informations sur le produit et le fabricant, destinées aux consommateurs). Nous ne pouvons pas utiliser de machine *VFFS* car nous n'en avons pas”.

- L'expert 2 dit : “Pour ce nouveau produit, il faut utiliser une machine *VFFS* ou *HFFS* plutôt qu'un emballage manuel (*HM*) car il n'est pas possible de contrôler l'atmosphère d'un emballage posé manuellement. Je préfère la machine *VFFS* car c'est celle qui est disponible en ce moment.”

– Nous apprenons aussi que l'utilisation d'une machine *VFFS* implique le choix d'un autocollant pour l'étiquetage de l'emballage. Or pour ce critère, l'expert 1 préfère une impression directe, car un autocollant peut être facilement enlevé par le consommateur.

À ce stade, aucune valeur ne peut convenir au premier abord. L'expert 1 privilégie l'organisation, tandis que l'expert 2 est plus concerné par la qualité. Il faut donc examiner les arguments exprimés par les experts lorsqu'ils ont exprimé leurs préférences et procéder par la suite à une phase de négociation pour tenter d'atteindre un consensus. Nous proposons une approche d'argumentation et de négociation permettant de modéliser les préférences des différents experts, ainsi que les arguments associés, puis de détecter les conflits d'intérêts éventuels et d'entamer enfin une phase de négociation afin de dégager un consensus.

3. Expression des préférences et argumentation

En partant des problématiques posées par le projet ECOBIOCAP pour la sélection d'emballage, nous avons opté pour une démarche basée sur l'argumentation afin de formaliser les échanges entre les acteurs du domaine.

Nous rappelons brièvement dans cette section les notions de base de l'argumentation ainsi que le formalisme ASPIC+. Nous détaillons ensuite les spécifications du langage logique (Croitoru *et al.*, 2012) (syntaxe et sémantique) proposé pour la modélisation des préférences utilisateur dans un cadre argumentatif, basé sur ASPIC+.

3.1. Notions générales sur l'argumentation

Un cadre argumentatif abstrait de Dung introduit dans (Dung, 1995), noté AF, est un couple $(\mathcal{A}, \mathcal{C})$, où \mathcal{A} est un ensemble d'arguments et \mathcal{C} est une relation binaire dite d'attaque entre arguments de \mathcal{A} : $\mathcal{C} \subseteq \mathcal{A} \times \mathcal{A}$.

Les définitions 1 et 2 ci-dessous permettent de définir la notion de conflit au sein d'un ensemble d'arguments et l'acceptabilité d'un argument par rapport à un ensemble d'arguments. Ces définitions sont à la base de la notion d'extension.

DÉFINITION 1. — *Arguments non conflictuels.* Un ensemble d'arguments $S \subseteq \mathcal{A}$ est dit sans conflits si et seulement si $\forall X, Y \in \mathcal{A} : (X, Y) \notin \mathcal{C}$.

DÉFINITION 2. — *Argument acceptable.* Soit $(\mathcal{A}, \mathcal{C})$ un AF. Pour tout $X \in \mathcal{A}$, X est dit acceptable par rapport à un ensemble d'arguments $S \subseteq \mathcal{A}$, si et seulement si tout argument attaquant X , est à son tour attaqué par un argument de S .

La notion d'extension est au coeur des théories de l'argumentation. Une extension est un ensemble cohérent (sans conflits) d'arguments qui se confortent. Une extension représente donc un ensemble d'arguments dans lesquels on peut croire. La cohérence peut être envisagée selon les différents critères exprimés dans la définition 3.

DÉFINITION 3. — *Extensions.* Soit $S \subseteq A$ un ensemble d'arguments sans conflits.

- S est une extension admissible si et seulement si pour tout $X \in S$, alors X est acceptable par rapport à S .
- S est une extension complète si et seulement si pour tout X acceptable par rapport à S , alors $X \in S$.
- S est une extension préférée si et seulement si S est une extension complète maximale (au sens d'inclusion d'ensemble).
- S est une extension de base si et seulement si S est une extension complète minimale (au sens d'inclusion d'ensemble).
- S est une extension stable si et seulement si S est une extension préférée et tout argument de A est soit dans S soit attaqué par un argument de S ($\forall Y \notin S, \exists X \in S : (X, Y) \in \mathcal{C}$).

A partir d'une sémantique particulière (admissible, complète, préférée, de base ou stable), un système d'argumentation peut avoir une, plusieurs ou même aucune extension (ou vide).

Les définitions ci-dessus forment la base du cadre abstrait d'argumentation de Dung. Dans ce cadre, les arguments sont seulement des entités abstraites et les relations d'attaques entre arguments doivent être une entrée du système. ASPIC+ propose un cadre pour formaliser les connaissances et les arguments d'une situation pratique, à partir desquels le système calcule automatiquement les extensions en se basant sur des relations d'attaque et de défaite prédéfinies (présentées plus loin) entre arguments. Nous allons voir que pour instancier un système ASPIC+, il nous faut définir un langage logique permettant d'exprimer nos connaissances sur le domaine que l'on considère.

3.2. Système d'argumentation ASPIC+

Une théorie d'argumentation est un couple (AS, \mathcal{K}) où AS est un système argumentatif défini sur une base de connaissances \mathcal{K} . ASPIC est un exemple de système d'argumentation introduit dans (Amgoud *et al.*, 2006). Des correctifs lui ont été aussi apportés pour définir ASPIC+ (Prakken, 2011 ; Prakken, Modgil, 2012), sur lequel nous allons nous baser pour notre solution d'argumentation. Selon (Prakken, Modgil, 2012), un système argumentatif ASPIC+ est un quadruplet $(\mathcal{L}, -, \mathcal{R}, n)$, avec :

- \mathcal{L} est un langage logique,
- ' $-$ ' est une fonction dite de contrariété définie de \mathcal{L} dans $2^{\mathcal{L}}$, qui permet d'associer à toute formule dans \mathcal{L} l'ensemble de ses formules incompatibles, tel que :
 - φ est le contraire de ψ si $\varphi \in \bar{\psi}, \psi \notin \bar{\varphi}$,
 - φ est en contradiction avec ψ , noté $\varphi = -\psi$, si $\varphi \in \bar{\psi}, \psi \in \bar{\varphi}$,
 - toute formule φ de \mathcal{L} doit avoir au moins une formule contradictoire.

– $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ est l'ensemble de règles strictes \mathcal{R}_s , de la forme $\phi_1, \dots, \phi_m \rightarrow \phi$, et de règles non strictes (sujettes à une remise en cause) \mathcal{R}_d , de la forme $\phi_1, \dots, \phi_m \Rightarrow \phi$, avec $\phi_1, \dots, \phi_m, \phi$ sont des metavariables correspondant aux formules bien formées de \mathcal{L} . Il est à noter que $\mathcal{R}_s \cap \mathcal{R}_d = \emptyset$.

– $n : \mathcal{R}_d \rightarrow \mathcal{L}$ est une fonction de nommage pour les règles non strictes.

Une base de connaissances associée à un système d'argumentation $AS = (\mathcal{L}, -, \mathcal{R}, n)$ est un ensemble $\mathcal{K} \subset \mathcal{L}$ tel que $\mathcal{K} = \mathcal{K}_a \cup \mathcal{K}_p$ et $\mathcal{K}_a \cap \mathcal{K}_p = \emptyset$, avec \mathcal{K}_a contient les axiomes (que l'on ne peut remettre en cause) du système et \mathcal{K}_p contient les prémisses ordinaires (que l'on peut remettre en cause) du système.

3.2.1. Arguments dans ASPIC+

DÉFINITION 4. — *Argument ASPIC+*. Soit \mathcal{K} une base de connaissances de \mathcal{L} ($\mathcal{K} \subseteq \mathcal{L}$). Un argument A dans \mathcal{K} a l'une des deux formes suivantes :

1. ϕ si $\phi \in \mathcal{K}$ avec $Prem(A) = \{\phi\}$, $Conc(A) = \phi$, $Sub(A) = \{\phi\}$, $Rules(A) = \emptyset$, $TopRules(A) = \text{non définie}$,

2. $A_1, \dots, A_m \rightarrow / \Rightarrow \psi$ si A_1, \dots, A_m sont des arguments tel qu'il existe une règle stricte/non stricte de la forme $Conc(A_1), \dots, Conc(A_m) \rightarrow / \Rightarrow \psi$ dans $\mathcal{R}_s/\mathcal{R}_d$ et :

- $Prem(A) = Prem(A_1) \cup \dots \cup Prem(A_m)$,
- $Conc(A) = \psi$,
- $Sub(A) = Sub(A_1) \cup \dots \cup Sub(A_m) \cup \{A\}$,
- $Rules(A) = Rules(A_1) \cup \dots \cup Rules(A_m) \cup \{Conc(A_1), \dots, Conc(A_m) \rightarrow / \Rightarrow \psi\}$,
- $TopRule(A) = Conc(A_1), \dots, Conc(A_m) \rightarrow / \Rightarrow \psi$.

avec : $Prem$ est une fonction qui retourne toutes les formules de \mathcal{K} appelées prémisses de l'argument, $Conc$ retourne sa conclusion, Sub retourne tous ses sous-arguments, $Rules$ retourne toutes les règles (strictes et non strictes) appliquées par l'argument, et enfin $TopRule$ retourne la dernière règle appliquée par l'argument.

3.2.2. Relations d'attaque et de défaite

Soit $AS = (\mathcal{L}, cf, \mathcal{R}, n)$ un système d'argumentation ASPIC+. Nous rappelons qu'une base de connaissances associée à AS est un ensemble $\mathcal{K} \subset \mathcal{L}$ tel que $\mathcal{K} = \mathcal{K}_a \cup \mathcal{K}_p$ et $\mathcal{K}_a \cap \mathcal{K}_p = \emptyset$, avec \mathcal{K}_a contient les axiomes du système et \mathcal{K}_p contient les prémisses ordinaires du système.

Les attaques entre arguments définies dans ASPIC+ sont les suivantes :

1. Réfuter (rebut) : l'argument A réfute l'argument B sur l'un de ses sous-arguments B' de la forme $B'_1, \dots, B'_m \Rightarrow \psi$, si $Conc(A) \in cf(\psi)$. Si $Conc(A)$ est le contraire de ψ alors A contrairement réfute B .

2. Amoindrir ou affaiblir (undermine) : l'argument A amoindrit ou affaiblit l'argument B si $Conc(A) \in cf(\psi)$ pour un $\psi \in Prem(B) \setminus \mathcal{K}_a$. Si $Conc(A)$ est le

contraire de ψ ou $\psi \in \mathcal{K}_p$ alors A contrairement amoindrit B .

3. Bloquer (undercut) : l'argument A bloque l'argument B sur l'un de ses sous-arguments B' de la forme $B'_1, \dots, B'_m \Rightarrow \psi$, si A contredit B' .

L'argument A met en échec l'argument B avec succès si A réfute B sur l'un de ses sous-arguments B' et $\text{Conc}(A)$ et le contraire de $\text{Conc}(B)$.

L'argument A amoindrit ou affaiblit la conclusion de l'argument B avec succès si A amoindrit $\text{Conc}(B)$ et $\text{Conc}(A)$ est le contraire de $\text{Conc}(B)$.

Les attaques permettent de définir une relation de défaite entre arguments. A défait B si A bloque, ou réfute avec succès, ou amoindrit (affaiblit) la conclusion de B avec succès. Si A défait B et B ne défait pas A alors A défait strictement B .

Les relations d'attaque et de défaite permettent de définir à partir du système ASPIC+ l'ensemble des extensions (admissibles, préférées, complètes, de base, stables) possibles. Les définitions suivantes permettent de hiérarchiser les différentes conclusions que l'on peut envisager en sortie d'un système d'argumentation comme ASPIC+, selon qu'elles apparaissent dans une, plusieurs, ou toutes les extensions.

DÉFINITION 5. — *Conclusions d'une extension.* Soit E une extension d'un cadre argumentatif AF . Les conclusions de E , notées $\text{Concs}(E)$, est $\text{Concs}(E) = \{\text{Conc}(A), A \in E\}$, avec $\text{Conc}(A)$ est la conclusion de l'argument A .

DÉFINITION 6. — *Conclusion justifiée sceptiquement/crédulement.* Pour toute extension $T \in \{\text{complète, préférée, de base, stable}\}$, X est une conclusion sceptiquement (resp. crédulement) justifiée, vis-à-vis de la sémantique T , si X appartient à toutes (resp. à au moins) les (une) extension(s) de T .

DÉFINITION 7. — *Conclusions sceptiques d'un AF.* Soit E_1, \dots, E_n les extensions de AF selon la sémantique $T \in \{\text{complète, préférée, de base, stable}\}$. Les conclusions sceptiques de AF est $\text{Output}(AF) = \bigcap_{i=1, \dots, n} \text{Concs}(E_i)$.

Nous décrivons dans ce qui suit les éléments de base nécessaires à la définition de notre instanciation ASPIC+ pour l'argumentation basée sur les préférences utilisateur.

3.3. *Éléments de base du système d'argumentation proposé*

Dans cette sous-section, nous introduisons le langage logique de notre système d'argumentation, basé sur la logique propositionnelle, pour la définition d'opérateurs d'expression de préférences (Croitoru *et al.*, 2012). Puis, nous détaillons sa sémantique, les règles d'inférence associées ainsi que sa fonction de contrariété.

3.3.1. *Aspects syntaxiques du langage \mathcal{L}*

Le langage logique \mathcal{L} est défini de manière inductive. Il est généré par l'ensemble des symboles propositionnels SP associés aux connecteurs logiques $\wedge, \neg, \succeq, \not\prec$ comme suit :

- $Prop(SP)$ est l'ensemble des formules de la logique propositionnelle définies à l'aide des connecteurs \wedge, \neg ,
- $A \in Prop(SP)$ alors $A \in \mathcal{L}$,
- Si $A, B \in Prop(SP)$ alors $A \succeq B \in \mathcal{L}$ et $A \not\succeq B \in \mathcal{L}$,
- Si $A, B \in \mathcal{L}$ alors $A \wedge B \in \mathcal{L}$,
- Si $A, B \in Prop(SP)$ alors $\neg(A \succeq B) \in \mathcal{L}$ et $\neg(A \not\succeq B) \in \mathcal{L}$.

Nous utilisons les notations classiques permettant une écriture plus lisible des différentes formules :

- $A \vee B = \neg(\neg A \wedge \neg B)$,
- $A \rightarrow B = \neg A \vee B$,
- $A \leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A)$.

EXEMPLE 8. — Nous reprenons l'exemple d'agrégation des préférences pour la sélection d'un emballage pour un fromage, présenté dans la section 2. Nous exprimons les préférences suivantes :

- les couleurs préférées sont $\{vert, mix\ and\ match, transparent\}$,
- les machines (pour emballer) préférées sont $\{HFFS, VFFS, HM\}$,
- les informations sur le produit peuvent être imprimées directement sur l'emballage ou imprimées sur des autocollants $\{imprimer, coller\}$,
- si la machine utilisée est $VFFS$ alors les informations (label) sur le produit doivent être collées,
- imprimer directement les informations est préféré à l'utilisation d'autocollants,
- la machine $VFFS$ n'est pas préférée à $HFFS$.

Il est possible de modéliser ces préférences par le langage $\mathcal{L} = SP \cup F(SP)$ où SP est un ensemble de symboles propositionnels et $F(SP)$ est l'ensemble des formules logiques définies sur SP tel que :

- $SP = \{vert, mix\ and\ match, transparent, HFFS, VFFS, HM, imprimer, coller\}$,
- $F(SP) = \{VFFS \rightarrow coller, imprimer \succeq coller, VFFS \not\succeq HFFS\}$.

□

3.3.2. Aspects sémantiques du langage \mathcal{L}

L'interprétation des éléments du langage \mathcal{L} est basée sur la notion de modèle $M \in 2^{SP}$. En effet, une interprétation est construite à l'aide d'un ensemble de modèles \mathcal{M} sur lequel une relation d'ordre totale \succeq est définie : $\succeq \subseteq \mathcal{M} \times \mathcal{M}$, avec $M_1 \succeq M_2$ si et seulement si $(M_1, M_2) \in \succeq$ et $M_1 \not\succeq M_2$ si et seulement si $(M_1, M_2) \notin \succeq$.

Un modèle peut alors exprimer une préférence d'un expert et l'ordre entre les modèles exprime aussi un ordre entre les préférences des experts.

EXEMPLE 9. — Nous reprenons le langage défini dans l'exemple 8, nous pouvons lui associer l'interprétation (\mathcal{M}, \succeq) où $\mathcal{M} = \{M_1, M_2, M_3, M_4\}$ et :

- $M_1 = \{vert, VFFS, coller, HM\}$,
- $M_2 = \{mix\ and\ match, transparent, VFFS, imprimer\}$,
- $M_3 = \{mix\ and\ match, HFFS, imprimer, transparent\}$,
- $M_4 = \{vert, HFFS, coller\}$
- $M_2 \succeq M_3 \succeq M_1 \succeq M_4$

□

La relation de satisfiabilité des formules propositionnelles vis-à-vis d'un modèle M , notée \models , est définie comme suit :

- $\forall A \in SP, M \models A$ ssi $A \in M$,
- $\forall A \in SP, M \models \neg A$ ssi $A \notin M$,
- $\forall A, B \in SP, M \models A \wedge B$ ssi $M \models A$ and $M \models B$.

Soit (\mathcal{M}, \succeq) une interprétation, les relations de satisfiabilité \models et de non satisfiabilité $\not\models$ sont définies comme suit :

- $\forall A \in SP, (\mathcal{M}, \succeq) \models A$ ssi $\forall M \in \mathcal{M}, A \in M$, sinon $(\mathcal{M}, \succeq) \not\models A$,
- $\forall A, B \in SP, (\mathcal{M}, \succeq) \models A \succeq B$ ssi $\forall M_1, M_2 \in \mathcal{M}$ satisfaisant :
 - a) $M_1 \models A \wedge \neg B$
 - b) $M_2 \models B \wedge \neg A$
 - c) M_1, M_2 coïncident avec tous les autres éléments de SP , i.e. $M_1 \cup M_2 = SP$,

alors $M_1 \succeq M_2$, sinon $(\mathcal{M}, \succeq) \not\models A \succeq B$

EXEMPLE 10. — En se basant sur l'interprétation de l'exemple 9, on peut déduire que $mix\ and\ match \succeq vert$. En effet, le seul couple de modèles satisfaisant les conditions a), b), c) ci-dessus est (M_3, M_1) . En effet, nous avons bien $mix\ and\ match \in M_3$ et $vert \notin M_3$, donc la condition a) est vérifiée ; de même pour M_1 , nous avons $vert \in M_1$ et $mix\ and\ match \notin M_1$, donc la condition b) est vérifiée ; l'union des symboles $M_3 \cup M_1 = SP$ donc la condition c) est vérifiée ; et comme $M_2 \succeq M_3 \succeq M_1 \succeq M_4$ alors nous avons $M_3 \succeq M_1$. □

- $\forall A, B \in SP, (\mathcal{M}, \succeq) \models A \not\succeq B$ ssi $\exists M_1, M_2 \in \mathcal{M}$ satisfaisant :

- a) $M_1 \models A \wedge \neg B$
- b) $M_2 \models B \wedge \neg A$
- c) M_1, M_2 coïncident avec tous les autres éléments de SP , i.e. $M_1 \cup M_2 = SP$,

alors $M_2 \succeq M_1$, sinon $(\mathcal{M}, \succeq) \not\models A \not\succeq B$

EXEMPLE 11. — En se basant sur l'interprétation de l'exemple 9 et en suivant le même raisonnement de l'exemple 10, on peut déduire que $coller \not\succeq imprimer$, car M_1 et M_3 satisfont bien les conditions a), b), c) de la définition et de plus $M_3 \succeq M_1$. □

– $\forall A, B \in SP, (\mathcal{M}, \succeq) \models \neg(A \succeq B)$ ssi $\forall M_1, M_2 \in \mathcal{M}$ satisfaisant :

a) $M_1 \models A \wedge \neg B$

b) $M_2 \models B \wedge \neg A$

c) M_1, M_2 coïncident avec tous les autres éléments de SP , i.e. $M_1 \cup M_2 = SP$,

alors $M_2 \geq M_1$, sinon $(\mathcal{M}, \succeq) \not\models \neg(A \succeq B)$

EXEMPLE 12. — On peut reprendre l'exemple 10 en inversant l'ordre du couple (M_3, M_1) pour déduire de cette interprétation que $\neg(\text{vert} \succeq \text{mis and match})$. \square

– $\forall A, B \in \mathcal{L}, (\mathcal{M}, \succeq) \models A \wedge B$ ssi $(\mathcal{M}, \succeq) \models A$ and $(\mathcal{M}, \succeq) \models B$,

sinon $(\mathcal{M}, \succeq) \not\models A \wedge B$

– $\forall A \in \mathcal{L}, (\mathcal{M}, \succeq) \models \neg A$ ssi $(\mathcal{M}, \succeq) \not\models A$, sinon $(\mathcal{M}, \succeq) \not\models \neg A$

3.3.3. Les règles d'inférence associées

Nous identifions dans ce qui suit les règles d'inférence sur lesquelles le système d'argumentation se base pour définir la logique de raisonnement associée.

Soit \mathcal{F} un ensemble de formules de \mathcal{L} , $C \in \mathcal{L}$ est dite conséquence de \mathcal{F} (noté $\mathcal{F} \vdash C$) si et seulement si pour toute interprétation (\mathcal{M}, \succeq) et pour toute formule $f \in \mathcal{F}$ tel que $(\mathcal{M}, \succeq) \models f$, nous avons $(\mathcal{M}, \succeq) \models C$.

En plus des règles d'inférence de la logique propositionnelle, le langage \mathcal{L} admet les règles suivantes :

$RS_1 : \neg(\phi \succeq \psi) \vdash (\phi \not\succeq \psi)$

$RS_2 : \neg(\phi \not\succeq \psi) \vdash (\psi \succeq \phi)$

$RS_3 : (\phi \succeq \psi) \wedge (\psi \succeq \sigma) \vdash (\phi \succeq \sigma)$

$RS_4 : (\neg\phi \succeq \phi) \wedge (\psi \rightarrow \phi) \vdash (\neg\psi \succeq \psi)$

$RS_5 : (\phi \succeq \neg\phi) \wedge (\psi \rightarrow \phi) \vdash (\psi \succeq \neg\psi)$

La règle d'inférence RS_1 peut être démontrée en considérant que sa partie gauche ne peut être vraie que si $M_1 < M_2$, ce qui satisfait les exigences de sa partie droite. La preuve de la règle d'inférence RS_2 est analogue à celle de la règle RS_1 . La règle RS_3 découle de la transitivité de la relation \succeq entre modèles. Les deux dernières règles peuvent être démontrées à partir des définitions de l'ensemble de modèles satisfaisant l'opérateur de préférence.

3.3.4. Validité des règles et complétude de l'interprétation

L'interprétation sémantique associée au langage logique présenté ci-dessus est soumise à la condition de complétude de l'ensemble \mathcal{M} de modèles, telle qu'elle est exprimée par l'hypothèse de l'ensemble de modèles maximal centré (Hansson, 1996). La complétude de l'ensemble des interprétations signifie que $\mathcal{M} = 2^{SP}$, où SP est l'ensemble de symboles propositionnels sur lequel le système est défini.

En effet, la propriété de transitivité (RS_3) mais aussi les règles d'inférence RS_4 et RS_5 ne sont plus valides si cette hypothèse n'est pas vérifiée par une interprétation donnée (\mathcal{M}, \succeq) , comme exprimé par les propriétés suivantes.

PROPRIÉTÉ 13. — *Dans une interprétation (\mathcal{M}, \succeq) , si \mathcal{M} n'est pas complet alors la relation de transitivité n'est plus vérifiée, i.e. : $(\mathcal{M}, \succeq) \not\models (a \succeq b) \wedge (b \succeq c) \rightarrow a \succeq c$.*

Preuve : Il est possible de démontrer cette propriété à l'aide d'un contre-exemple. Soit (\mathcal{M}, \succeq) une interprétation définie sur l'ensemble de symboles $SP = \{a, b, c\}$, tel que $\mathcal{M} = \{M_1, M_2\}$ avec $M_1 = \{a, b\}$, $M_2 = \{c, b\}$ et $M_2 \succeq M_1$. Nous pouvons montrer que $(\mathcal{M}, \succeq) \models (a \succeq b)$, $(\mathcal{M}, \succeq) \models (b \succeq c)$ mais $(\mathcal{M}, \succeq) \not\models (a \succeq c)$.

Pour $(\mathcal{M}, \succeq) \models (a \succeq b)$, selon l'interprétation associée, $\forall M_3, M_4 \in \mathcal{M}$, si $M_3 \models a \wedge \neg b$ et $M_4 \models \neg a \wedge b$ et M_3, M_4 coïncident avec tous les autres éléments de SP alors $M_3 \succeq M_4$. Comme $\nexists M \in \mathcal{M} : M \models a \wedge \neg b$ alors la déclaration $(\mathcal{M}, \succeq) \models (a \succeq b)$ est toujours vraie.

Le même raisonnement est suivi pour montrer que $(\mathcal{M}, \succeq) \models (b \succeq c)$.

Pour $(\mathcal{M}, \succeq) \models (a \succeq c)$, il faut avoir $\forall M_3, M_4 \in \mathcal{M}$, si $M_3 \models a \wedge \neg c$ et $M_4 \models \neg a \wedge c$ et M_3, M_4 coïncident avec tous les autres éléments de SP alors $M_3 \succeq M_4$. Nous avons bien $M_1 \models a \wedge \neg c$ et $M_2 \models \neg a \wedge c$ et M_1, M_2 coïncident avec tous les autres éléments de SP , cependant $M_2 \succeq M_1$, donc $(\mathcal{M}, \succeq) \not\models (a \succeq c)$. \square

PROPRIÉTÉ 14. — *Dans une interprétation (\mathcal{M}, \succeq) , si \mathcal{M} n'est pas complet alors les règles d'inférence suivantes (RS_4 et RS_5) ne sont plus vérifiées :*

- $(\mathcal{M}, \succeq) \not\models (\neg a \succeq a) \wedge (b \rightarrow a) \rightarrow \neg b \succeq b$
- $(\mathcal{M}, \succeq) \not\models (a \succeq \neg a) \wedge (b \rightarrow a) \rightarrow b \succeq \neg b$

Preuve : La preuve de la propriété 14 est similaire à celle de la propriété 13 (basée sur la construction d'un contre-exemple). \square

3.3.5. La fonction de contrariété

La fonction de contrariété permet de définir dans notre cas les formules mutuellement exclusives du langage \mathcal{L} . Elle servira par la suite à définir les attaques entre les arguments du système mis en place. Nous définissons la fonction de contrariété $cf : \mathcal{L} \rightarrow 2^{\mathcal{L}}$ de la manière suivante :

Pour toute formule propositionnelle ϕ :

- $cf(\phi) = \{\neg\phi\}$
- $cf(\neg\phi) = \{\phi\}$

Pour toute autre formule dans \mathcal{L} , si l'ensemble de modèles \mathcal{M} est complet alors :

- $cf(\neg(a \succeq b)) = \{(a \succeq b)\}$
- $cf((a \succeq b)) = \{(b \succeq a), \neg(a \succeq b), (a \not\succeq b)\}$
- $cf((a \not\succeq b)) = \emptyset$
- $cf(\neg(a \not\succeq b)) = \{(a \succeq b), \neg(b \succeq a), (b \not\succeq a)\}$

Sinon, si l'ensemble de modèles \mathcal{M} est incomplet alors, pour les mêmes considérations vues pour les propriétés 13 et 14, la fonction de contrariété pour les autres formules (non propositionnelles) de \mathcal{L} est :

$$- cf((a \succeq b)) = \{b \succeq a\}$$

3.4. Système d'argumentation ASPIC+ proposé

À l'aide des éléments présentés ci-dessus, nous définissons notre instanciation d'ASPIC+, ainsi que les notions d'attaque et de défaite et nous discutons du choix de la nature de l'interprétation associée à notre système d'argumentation.

3.4.1. Instanciation du système ASPIC+

Nous définissons notre système d'argumentation ASPIC+, noté $AS = (\mathcal{L}, -, \mathcal{R}, n)$, comme suit :

- \mathcal{L} correspond au langage logique défini dans les sous-sections 3.3.1 et 3.3.2,
- '-' est représentée par la fonction de contrariété cf définie dans la sous-section 3.3.5,
- $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ est l'ensemble de règles strictes (de la forme $\phi_1, \dots, \phi_k \rightarrow \phi$) et non strictes (de la forme $\phi_1, \dots, \phi_k \Rightarrow \phi$) du système, avec :
 - $\mathcal{R}_s = \{RS_1, \dots, RS_5\} \cup RS^{SP}$, si l'on considère que l'ensemble des modèles \mathcal{M} est complet,
 - $\mathcal{R}_s = \{RS_1, RS_2\} \cup RS^{SP}$, si l'on considère que \mathcal{M} est non complet.
- n est une fonction de nommage non définie.

Les arguments qui peuvent être exprimés dans cette instanciation ont les mêmes formats introduits dans la définition 4.

EXEMPLE 15. — Nous reprenons le cas présenté dans l'exemple 8 ainsi que son interprétation présentée dans l'exemple 9. Nous rappelons que le contexte de l'exemple 8 définit le langage $\mathcal{L} = SP \cup F(SP)$ où :

- $SP = \{vert, mix \text{ and } match, transparent, HFFS, VFFS, HM, imprimer, coller\}$,
- $F(SP) = \{VFFS \rightarrow coller, imprimer \succeq coller, VFFS \not\prec HFFS\}$.

Dans ce contexte, on peut exprimer l'argument A suivant composé des sous-arguments A_1 et A_2 :

- $A_1 : VFFS$ (argument de la forme 1),
- $A_2 : A_1 \rightarrow Coller$ (argument de la forme 2).

avec : $Prem(A) = \{VFFS\}$, $Conc(A) = Coller$, $Sub(A) = \{VFFS, VFFS \rightarrow Coller\}$, $Rules(A) = \{VFFS \rightarrow Coller\}$ et $TopRule = \{VFFS \rightarrow Coller\}$.

Cet argument exprime le fait qu'ayant $VFFS$ dans la base de connaissances \mathcal{K} , et la règle d'inférence $VFFS \rightarrow Coller$, alors la valeur $Coller$ est justifiée. \square

La complétude du raisonnement du système d'argumentation ASPIC+ proposé dépend de la nature de l'interprétation (\mathcal{M}, \succeq) choisie (ensemble \mathcal{M} complet/non complet de modèles).

3.4.2. Choix de la nature de l'interprétation

L'ensemble \mathcal{M} est dit complet s'il correspond à l'ensemble de toutes les parties de SP sur lequel le système d'argumentation est défini. Cette condition est difficile à satisfaire en raison de la complexité du calcul. En effet, si $\mathcal{M} = \{M \mid M \in \mathcal{P}(SP)\}$ alors 2^{SP} modèles possibles sont à calculer, et il n'est pas réaliste de demander à l'utilisateur d'exprimer tous les modèles possibles pour l'interprétation de ses préférences, d'autant plus qu'un ordre total (\succeq) doit être aussi spécifié entre ces modèles.

Pour ces considérations, nous privilégions dans la pratique les interprétations basées sur des ensembles de modèles incomplets. Afin de pallier la perte des règles d'inférence, nous combinons à notre système d'argumentation un processus de négociation dans lequel les intervenants seront en mesure d'ajouter des arguments de façon dynamique pour résoudre les situations conflictuelles. Cette possibilité d'ajout d'arguments correspond à l'ajout dans le système d'autres modèles modifiant ainsi les conclusions du processus de raisonnement sous-jacent à l'argumentation. Nous décrivons dans la section suivante le processus de négociation mis en place.

4. Prise de décision/négociation

Nous avons introduit l'idée d'une phase de négociation pour nous permettre de décider par consensus comment choisir une valeur de préférence parmi plusieurs valeurs en conflit.

Lorsque les décideurs expriment une préférence sur un ou plusieurs critères, ils sont invités à exprimer la raison qui soutient leur préférence, raison que l'on peut voir comme un argument justifiant la préférence. Chaque conflit obtenu par notre système d'argumentation représente donc deux arguments qui s'attaquent mutuellement. L'idée derrière la phase de négociation est d'utiliser ces arguments comme point de départ d'une discussion.

Lorsqu'un conflit entre préférences est détecté, nous commençons par associer une désignation, c'est-à-dire donner un nom formel (par exemple A, B, \dots) à chaque argument soutenant une préférence et construisons un second système d'argumentation contenant les arguments mis en jeu dans le conflit, ainsi que ceux rajoutés par les décideurs au travers de nouvelles informations exprimées dans le système, en plus des relations d'attaque qui en découlent.

A chaque itération de la discussion ainsi amorcée, le système calcule et retourne les extensions de Dung. Lorsqu'au cours du dialogue nous obtenons une unique extension préférée incluant un et un seul des arguments du conflit initial, nous proposons de

clure la phase de négociation. La préférence supportée par l'argument faisant partie de l'extension préférée finale reste dans le système, tandis que la préférence rejetée par le système argumentatif est désormais ignorée. Cette phase de négociation permet en pratique de résoudre bon nombre de conflits. En effet, chaque décideur exprime ses préférences en fonction de ces connaissances qui sont par nature incomplètes. Dans notre contexte un décideur peut par exemple vouloir un mode d'emballage *VFFS* ou *HFFS* permettant de contrôler l'atmosphère. Par contre, il peut revoir sa position lorsqu'il apprend dans la phase de négociation que de nouvelles techniques permettent de contrôler l'atmosphère avec un mode d'emballage manuel. Voyons le déroulement d'un tel scénario.

Deux experts sont en conflit sur le type d'emballage à utiliser pour emballer un fromage. Les arguments exprimés sont :

- Expert 1 : “Il faut utiliser une machine *VFFS* plutôt qu'un emballage manuel car l'atmosphère d'un emballage manuel ne peut être contrôlée” (Argument *A*).
- Expert 2 : “Il vaut mieux emballer à la main car c'est la méthode d'emballage que nous avons l'habitude d'utiliser et que la machine *VFFS* que nous possédons est utilisée pour emballer un autre fromage” (Argument *B*).

La figure 3 présente le système d'argumentation nouvellement créé. On y trouve les deux arguments *A* et *B* supportant les deux préférences opposées.

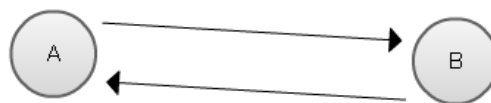


Figure 3. Système d'argumentation permettant de commencer une phase de négociation

Les utilisateurs sont alors amenés à saisir de nouveaux arguments dans le système d'argumentation. Nous pouvons par exemple ajouter successivement les arguments suivants :

A : il faut utiliser une machine *VFFS* plutôt qu'un emballage manuel car l'atmosphère d'un emballage manuel ne peut être contrôlée,

B : il vaut mieux emballer à la main car c'est la méthode d'emballage que nous avons l'habitude d'utiliser et la machine *VFFS* que nous possédons est utilisée pour emballer un autre fromage,

C : les nouvelles technologies permettent de contrôler l'atmosphère dans les emballages manuels,

D : il est très cher de former les employés sur ces nouvelles techniques,

E : la formation des employés est très rapide ainsi que la procédure de mise en place.

Les arguments sont ajoutés successivement au système d'argumentation (figure 4). À chaque itération les extensions préférées au sens de Dung sont calculées. Lors-

qu'une seule extension préférée existe et qu'elle contient exactement un des arguments du conflit initial, les utilisateurs sont invités à clore le conflit, sauf s'ils ont d'autres arguments à opposer. Dans notre cas, l'extension préférée obtenue est $\{B, C, E\}$ et l'argument B sort alors vainqueur du conflit.

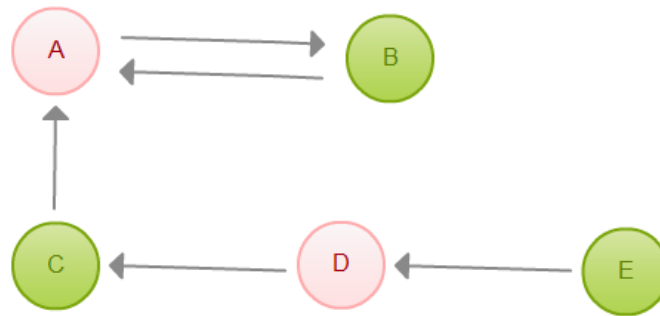


Figure 4. Système d'argumentation en fin de négociation

Malheureusement, en pratique, il n'est pas toujours possible de trouver un consensus entre deux parties poursuivant des buts distincts. Cette situation apparaît dans notre système lorsque nous n'obtenons jamais une seule extension préférée. Dans ce cas, nous considérons que le conflit ne peut être résolu par un simple système d'argumentation et de négociation. Il est alors nécessaire en pratique de pouvoir séparer le scénario d'utilisation de notre système d'aide à la décision en deux scénarios distincts. Dans un des scénarios nous gardons une des préférences dont le conflit n'a pu être résolu par la négociation. Dans l'autre, nous gardons l'autre préférence. Chaque utilisateur du système peut alors continuer à travailler en parallèle sans que le désaccord soit bloquant pour aucun des deux.

Raffiner les scénarios risque de faire apparaître en pratique de nombreux scénarios parallèles. Pour cette raison, nous nous contentons dans notre application de ne poursuivre l'investigation que dans un seul scénario choisi par l'utilisateur parmi tous les scénarios possibles.

5. Développements/exemple illustratif

Afin de valider notre approche à base d'argumentation et de négociation dans le projet EcoBioCap, un premier prototype d'application a été développé en Java.

Lors de l'utilisation de l'application, différentes phases se succèdent :

1. Saisie des données : la saisie des données consiste dans un premier temps à définir les différents critères et leurs valeurs possibles qui seront utilisés par le système. Les principaux critères utilisés dans les exemples de ce papier sont le type de machine, la couleur et le type de label. Ensuite, il faut saisir les différents experts concernés dans le système (fabricants de produits frais, consommateurs, fournisseurs d'emballage, etc.). Puis, il faut saisir les préférences des différents acteurs et ainsi que

leur connaissances implicites. Ces dernières représentent notamment les implications qu'il y a entre différentes valeurs pour différents critères donnés. Par exemple nous avons vu qu'un expert peut savoir que si l'on utilise une machine à scellement de film, alors la seule façon de mettre un label sur l'emballage sera d'y apposer un autocollant. La figure 5 montre les implications entre valeurs telles que nous pouvons les faire afficher dans l'application. Dans cette figure, une flèche montre d'après un certain expert, le choix d'une valeur d'un critère implique le choix d'une valeur d'un autre critère. Par exemple l'expert 1 exprime que le choix de machine *HFFS* implique que l'emballage ne peut être transparent. Cette phase de saisie des données initiale a pour l'instant été faite manuellement une fois pour toutes (les informations sont sauvegardées dans un format adéquat) pour les applications ciblées du projet EcoBioCap. Une automatisation de la saisie des informations est à prévoir pour rendre ce prototype facilement utilisable pour d'autres applications.

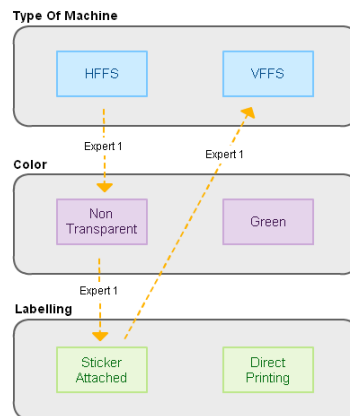


Figure 5. Implication entre différentes valeurs de différents critères

2. Représentation des préférences : une fois toutes les informations saisies lors de l'initialisation, le logiciel permet de visualiser les différentes préférences exprimées par les différents experts. Pour cela, un graphe est construit ayant pour sommet chaque valeur de chaque critère (voir figure 6). Une préférence exprimée par un acteur est modélisée par un arc orienté, ayant pour label l'expert qui considère la préférence. Lorsqu'un conflit entre les préférences exprimées sur les valeurs est détecté par les techniques d'argumentation, un bouton "Solve Conflict" apparaît, permettant directement d'initier une phase de négociation.

3. Négociation : la figure 7 montre une copie d'écran du système lors de la phase de négociation tel qu'il est implémenté actuellement. Dans cette phase, les utilisateurs peuvent ajouter itérativement leurs arguments complémentaires. À chaque itération, si le système d'argumentation en cours d'élaboration ne possède qu'une seule extension préférée, les utilisateurs sont invités à clore la négociation. En effet, si aucun autre argument ne peut être ajouté, cette situation correspond à un consensus. Dans le cas où plus aucun argument ne peut être ajouté et que le système d'argumentation ne propose

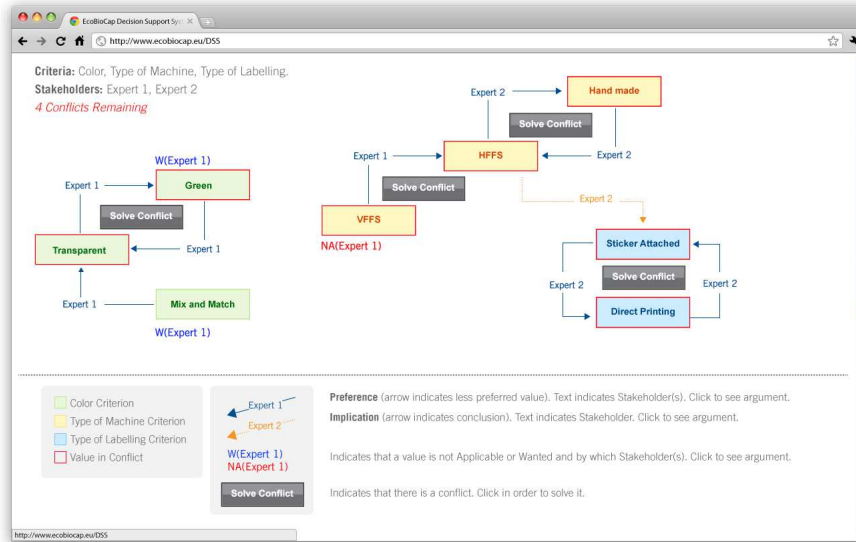


Figure 6. Représentation graphique des préférences

pas une unique extension préférée, cela correspond à une situation de blocage : 2 experts n'arrivent pas à se mettre d'accord. Il faut alors ignorer les arguments d'au moins un des deux experts en conflit, afin de ne pas bloquer le système.

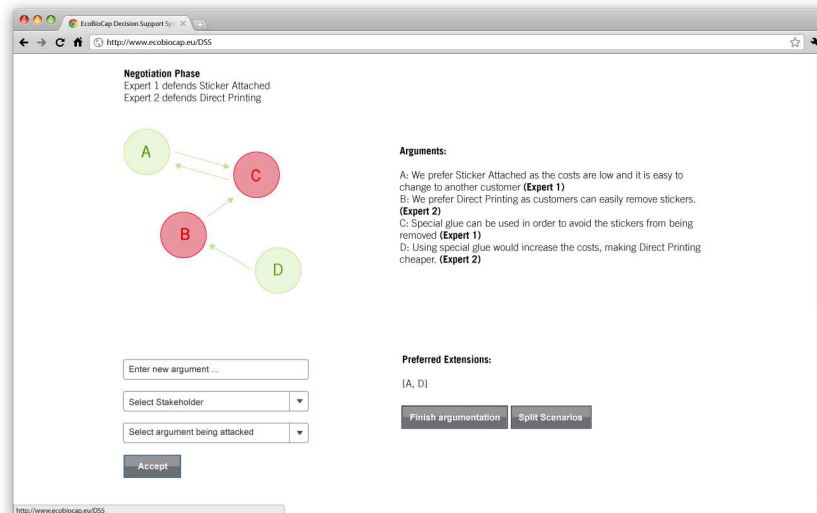


Figure 7. Copie d'écran d'une phase de négociation

4. Décisions : une fois les conflits résolus, le prototype affiche pour chaque critère la valeur retenue et les raisons du choix de cette valeur.

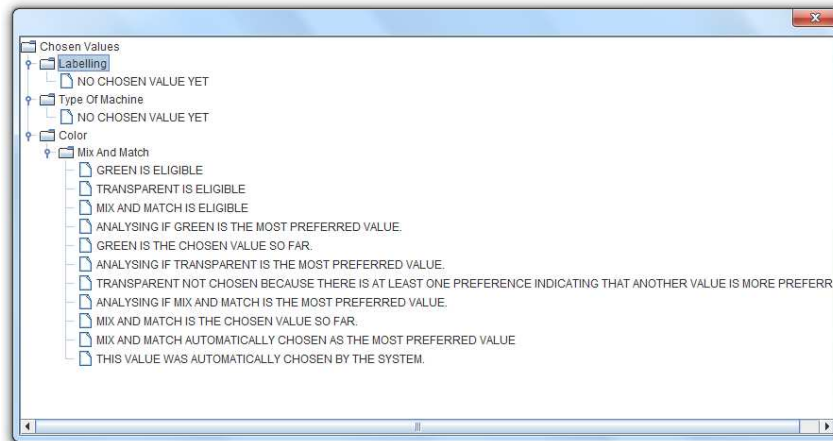


Figure 8. Valeurs choisies et explications dans un scénario d'utilisation

6. Conclusion

Nous avons présenté une approche d'agrégation de préférences, pouvant être conflictuelles, de différents acteurs d'un domaine, en utilisant des techniques d'argumentation et de négociation, pour l'aide à la prise de décision.

Nous avons proposé d'exprimer les préférences utilisateur à l'aide d'un langage d'expression de préférences en logique propositionnelle. Nous avons également proposé une sémantique dont le rôle est d'associer une interprétation aux préférences et de permettre, grâce à des règles d'inférence, la déduction automatique de nouvelles préférences à partir de celles exprimées par les acteurs.

Ce langage logique ainsi que sa sémantique sont la base de notre instanciation du système d'argumentation ASPIC+. Les extensions en sortie du système d'argumentation présentent souvent des conclusions contradictoires nécessitant une étape de négociation pour la résolution des conflits détectés automatiquement par le système. Nous avons montré que cette résolution n'est pas toujours possible si l'on se limite à un ajout d'arguments dans le système et à l'itération du processus d'argumentation. Les situations de blocage peuvent être décomposées en scénarios non conflictuels et indépendants afin de pouvoir aboutir à une prise de décision. Dans notre solution actuelle, et pour des raisons d'efficacité, un seul scénario parmi ceux issus de cette décomposition peut être choisi. Un prototype a été implémenté afin de montrer la faisabilité de notre approche.

Dans un avenir proche, nous étendrons notre modèle afin de pouvoir exprimer non seulement des préférences sur les valeurs de différents critères, mais aussi de pouvoir

privilégier certains critères plus importants au détriment d'autres critères et de pouvoir aussi accorder plus d'importance à certains experts qu'à d'autres. Par exemple, on pourrait considérer que les critères sanitaires sont plus importants que les aspects esthétiques de l'emballage, ou encore qu'il est plus important de privilégier l'avis d'un expert des normes sanitaires, par exemple, pour choisir la température de stockage de l'emballage plutôt que celui d'un gestionnaire d'entrepôt. Pour ce faire, il est possible de considérer dans le langage les attributs sur lesquels les valeurs de préférences sont exprimées, et d'associer aux arguments et à leurs origines une importance ou une relation de précédence qui sera utilisée dans le processus de prise de décision comme suggéré dans (Amgoud, 2005 ; Amgoud, Prade, 2009).

Remerciements

Les travaux de recherche ayant conduit aux résultats présentés dans ce papier ont été financés par le 7e programme cadre de la communauté européenne (FP7/2007-2013) sous la convention de subvention du projet n°FP7-265669-EcoBioCAP.

Bibliographie

- Amgoud L. (2005). A unified setting for inference and decision: An argumentation-based approach. In C. Reed (Ed.), *In 5th workshop on computational models of natural argument, CMNA'05. collocated with IJCAI'05*, p. 40-43.
- Amgoud L., Bodenstaff L., Caminada M., McBurney P., Parsons S., Prakken H. *et al.* (2006). *Final review and report on formal argumentation systems*. Rapport technique. (Deliverable D2.6 ASPIC EC project Argumentation Service Platform with Integrated Components)
- Amgoud L., Cayrol C. (2002, Mars). A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, vol. 34, n° 1-3, p. 197-215. <http://dx.doi.org/10.1023/A:1014490210693>
- Amgoud L., Prade H. (2009). Using arguments for making and explaining decisions. *Artificial Intelligence Journal*, vol. 173, p. 413-436.
- Amgoud L., Vesic S. (2011). A new approach for preference-based argumentation frameworks. *Annals of Mathematics and Artificial Intelligence*, vol. 63, n° 2, p. 149-183.
- Bourguet J.-R., Amgoud L., Thomopoulos R. (2010). Towards a unified model of preference-based argumentation. In *Foiks*, p. 326-344.
- Bouzeghoub M., Calabretto S., Denos N., Harrathi R., Kostadinov D., Nguyen A.-T. *et al.* (2007). *Accès personnalisé aux informations : approche dirigée par la qualité*. In Actes du xxvème congrès inforsid, p. 105-120.
- Bouzeghoub M., Kostadinov D. (2005). Personnalisation de l'information : aperçu de l'état de l'art et définition d'un modèle flexible de profils. In *Actes de la 2e conférence en recherche d'information et applications (coria)*, p. 201-218.
- Caminada M., Amgoud L. (2005). An axiomatic account of formal argumentation. In *Proceedings of the Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference*, p. 608-613. AAAI Press / The MIT Press.

- Croitoru M., Fortin J., Oren N. (2012). Arguing with preferences in ecobiocap. In *Proceedings of the 4th international conference on computational models of argument*, p. 51–58. IOS Press, Frontiers in Artificial Intelligence and Applications series.
- Destercke S., Buche P., Guillard V. (2011). A flexible bipolar querying approach with imprecise data and guaranteed results. *Fuzzy Sets and Systems*, vol. 169, n° 1, p. 51-64.
- Dung P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence Journal*, vol. 77, p. 321–357.
- Hansson S. O. (1996). What is ceteris paribus preference? *Journal of Philosophical Logic*, vol. 25, n° 3, p. 307–332.
- Kaci S. (2009). *Working with preferences: less is more*. Habilitation à diriger des recherches, Centre de Recherche en Informatique de Lens - CNRS UMR 8188.
- Kaci S., Torre L. van der. (2008). Preference-based argumentation: Arguments supporting multiple values. *Int. J. Approx. Reasoning*, vol. 48, n° 3, p. 730–751.
- Kiesling W. (2002). Foundation of preferences in database systems. In *proceedings of the 28th vldb conference, hong kong, china*.
- Kiesling W., Köstler G. (2002). Preference sql - design, implementation, experiences. In *28th vldb conference, hong kong, china*.
- Modgil S. (2009). Reasoning about preferences in argumentation frameworks. *Artif. Intell.*, vol. 173, n° 9-10, p. 901–934.
- Prakken H. (2011). An abstract framework for argumentation with structured arguments. *Argument and Computation*, vol. 1, n° 2, p. 93–124.
- Prakken H., Modgil S. (2012). Clarifying some misconceptions on the aspic+ framework. *Computational Models of Argument. Proceedings of COMMA 2012*.