Calculs à l'aide de mots : vers un emploi de termes linguistiques de bout en bout dans la chaîne du raisonnement

Isis Truck

Université Paris 8

LIASD – EA 4383 **Laboratoire d'Informatique Avancée de Saint-Denis**
Axe ALMD

Calculs à l'aide de mots : vers un emploi de termes linguistiques de bout en bout dans la chaîne du raisonnement

Rapport Scientifique

présenté et soutenu publiquement le 8 septembre 2011

pour l'obtention de l'

Habilitation à diriger des recherches de l'Université Paris 8

(spécialité informatique)

par

Isis Truck

Composition du jury

Président :	Jacques Malenfant	Professeur, Université Pierre et Marie Curie – Paris 6
Rapporteurs :	Didier Dubois	Directeur de recherche au CNRS, IRIT–UMR 5505, Université P. Sabatier, Toulouse
	Luis Martínez-López	Professeur, Université de Jaén, Espagne
	Gabriella Pasi	Professeur, Université de Milan Bicocca, Italie
Garant :	Marc Bui	Professeur, Université Vincennes - Saint-Denis – Paris 8
Examinateur :	Francis Rousseaux	Professeur, Université de Reims Champagne-Ardenne

École Doctorale Sciences Sociales

Remerciements

Tout d'abord, je remercie du fond du cœur Marc Bui, mon garant, qui m'a insufflé le courage et l'envie nécessaires à la réalisation de ce travail.

Ensuite, je voudrais exprimer toute ma gratitude à Gabriella Pasi, Didier Dubois et Luis Martínez, mes rapporteurs, pour avoir accepté cette lourde tâche et pour tout le temps qu'ils y ont consacré. Merci en particulier à Didier Dubois pour nos longues discussions téléphoniques qui m'ont permis de rendre ce document plus cohérent et plus lisible.

Je suis également très reconnaissante à Jacques Malenfant d'avoir accepté de présider le jury et, surtout, de m'avoir tant épaulée, soutenue et encouragée durant toutes ces années. Je n'oublierai jamais son intarissable savoir et son infinie gentillesse.

Un très grand merci à Francis Rousseaux pour ses nombreux conseils et pour avoir bien voulu examiner ce travail.

Je veux aussi remercier tout particulièrement mes collègues et amis Alain Bonardi, Anna Pappa et Nicolas Jouandeau pour tous les échanges que nous eus, pour les nombreuses collaborations passées, en cours et à venir, et pour leurs très grandes compétences, sans lesquels le travail évoqué dans ce mémoire n'aurait pas vu le jour. Je n'oublie pas non plus les doctorants que j'ai co-encadrés ou que je co-encadre encore, auxquels je dois beaucoup : Pierre Châtel, Mohammed-Amine Abchir, Xavier Dutreilh et Olga Melekhova.

Je n'oublierai jamais Elisabeth Bautier qui a tant fait pour moi et m'a tellement soutenue, en particulier dans les moments les plus difficiles. Merci pour tout cela, Elisabeth.

Je voudrais aussi adresser un très grand merci à Claude Carlet et à Jaime Lopez-Krahe pour avoir accepté et pris le temps de rapporter sur ce travail pour le Conseil scientifique.

Merci enfin à mes collègues de Paris 8, en particulier mes amies Viviane, Anne-Marie et Anolga pour leur soutien sans faille,
à Marie-Christine Lamiche pour toute l'énergie qu'elle a mise pour la logistique,
au Service de la Recherche (Annick, Rui, Sophie, Sylvia, Yassamine, etc.) pour son aide et sa bonne humeur.

A JTS, MTK & OA.

"La langue est pour ainsi dire une algèbre qui n'aurait que des termes complexes."

Ferdinand de Saussure, Cours de linguistique générale, Payot, 1916.

Table des matières

Table des figures

Avertissement

J'adresse cet avertissement au lecteur qui pourra peut-être s'étonner de certains accords de participes.

Même si, comme dans toute recherche inscrite dans un laboratoire, celle-ci doit bien évidemment à la dynamique intellectuelle collective (les noms des collègues et étudiants sont cités à la fin de chaque section), les pages qui suivent sont écrites sous ma seule responsabilité, en conséquence, le "nous" sujet adopté dans l'écriture est bien un nous dit de "modestie".

Chapitre 1

Introduction

La numérisation prise au sens large, c'est-à-dire l'attribution d'un nombre à des entités, est un problème non trivial, sauf peut-être en géométrie, physique ou économie, domaines qui ont réussi leur numérisation [Maurin, 2009]. Dans plusieurs autres disciplines, on cherche à y parvenir également, mais cela reste un problème ouvert. En effet, le recours aux mots pour décrire les phénomènes et les entités implique, pour la machine, un traitement linguistique et non plus numérique. A cela s'ajoute la notion d'imperfection qui entache les grandeurs et les observations. C'est pourquoi depuis plusieurs années, un axe de recherche important s'est créé autour du "calcul à l'aide de mots" encore appelé *Computing with Words*, thème abordé par Zadeh dès 1978 (lire, par exemple, [Zadeh, 1978], [Zadeh, 1996] ou encore [Zadeh, 2002]) mais qui n'a suscité un réel engouement qu'assez récemment, environ trente ans après l'invention de la logique floue [Zadeh, 1965].

Ainsi, nous situons le contexte général de nos travaux de recherche dans le domaine de la représentation, modélisation puis combinaison des connaissances en milieu imprécis et vague, en particulier dans la théorie des sous-ensembles flous, d'une part, et dans la théorie des multi-ensembles, d'autre part.

Dans ce contexte, nous nous sommes toujours attachée[1] à concevoir des modèles ou outils linguistiques pour pallier les problèmes d'imprécision. A peu près tous les domaines sont ou peuvent être concernés par ces problèmes d'imprécision ; c'est pourquoi nous avons cherché à appliquer nos modèles ou nos idées dans des contextes divers, où il nous semblait qu'une collaboration interdisciplinaire était pertinente. Par exemple, dans le domaine des architectures logicielles, les informations recueillies — qu'elles proviennent de données observables ou de données prévisibles — qui permettront d'alimenter les différents modules de prise de décision, sont presque toujours entachées d'imprécision et impliquent donc un traitement particulier. Et les choix ou préférences — provenant le plus souvent d'un "expert" — pour aider le système à décider, en fonction des données entrantes, sont plus facilement exprimables à l'aide de mots. Sans doute d'autres disciplines peuvent-elles bénéficier de nos approches, nous allons voir au fil des pages celles avec lesquelles des résultats concluants ont été obtenus.

De façon générale et assez théorique, notre travail s'est d'abord fondé sur la notion de

1. Le lecteur s'étonnant de cet accord de participe peut se reporter à l'avertissement en page 1.

nuance, de variation ou modification, préservant la simplicité d'un espace de variables linguistiques de cardinalité assez faible mais sur lequel un large éventail de nuances (standard) peuvent être appliquées. Puis nous avons utilisé ces modèles et outils dans des contextes où l'expression de nuances est très importante, comme le *perceptual computing* dans le domaine de la classification des couleurs ou encore le jeu théâtral, et, plus récemment, pour la capture des intentions des programmeurs pour établir des politiques de contrôle et d'adaptation dynamique des architectures logicielles selon l'approche du calcul auto-régulé (*autonomic computing*).

Ce document est structuré en trois parties correspondant aux chapitres 2 à 4 : dans la première, nous situons le contexte théorique en tant que tel, mais aussi et surtout, nous rappelons le travail de thèse qui a permis de positionner tout ce qui a été envisagé par la suite. Dans la deuxième, nous expliquons les trois grands axes de travail autour de notre fil conducteur du "calcul à l'aide de mots", à savoir l'axe de la perception visuelle (au sens propre), celui de la perception (au sens plus figuré) dans les arts du spectacle et enfin celui des architectures logicielles. Pour deux de ces trois axes, des co-encadrements de thèse ont eu lieu ou sont encore en cours. La troisième partie, quant à elle, étudie les pistes de réflexions actuelles et encore en cours que ces années de travail ont amenées. Il faut noter qu'un article soumis à la revue IJAMS (*International Journal of Applied Management Science*) est joint en annexe et que, de surcroît, certaines idées non encore publiées sont exposées dans cette troisième partie.

Chapitre 2

Contexte et historique

Sommaire

DANS CE CHAPITRE, on pose d'abord les bases de la réflexion depuis le début de nos recherches, c'est-à-dire depuis maintenant une dizaine d'années.

Sans retranscrire tout le travail de thèse, on évoque le contexte tel qu'il se présentait à l'origine et les hypothèses posées. Puis nous nous attachons à décrire les tout premiers travaux post-thèse qui ont principalement consisté à poursuivre et étendre les résultats obtenus pendant le doctorat, c'est-à-dire, à lier le mieux possible les termes linguistiques et leur représentation interne, en passant par la combinaison de ces termes.

2.1 Contexte

Notre volonté était donc d'abord d'étudier les — certains, en tous cas — outils servant à exprimer et représenter les notions d'imprécision dans les connaissances.

On sait l'intérêt des approches linguistiques pour simuler le raisonnement humain depuis les fameux travaux de Zadeh, *cf.* par exemple [Zadeh, 1965, Zadeh, 1975] et l'on voit le bénéfice que l'étude des modificateurs linguistiques (traduction des *linguistic hedges* de Zadeh) peut apporter pour la modélisation et le raisonnement. En effet, la simulation du mode de pensée humain ne peut véritablement se concevoir qu'en utilisant des connaissances graduelles — et non tranchées — en ayant à l'esprit de les représenter avec la possibilité de les moduler le mieux possible. En théorie du mesurage, mesurer signifie comparer des quantités, c'est-à-dire comparer une dimension inconnue avec une autre, connue. La dimension inconnue est exprimée comme un multiple ou une fraction de l'unité définie. Les modificateurs s'inscrivent précisément dans ce cadre : ils permettent de comparer deux quantités,

dont une seule est connue. Mais le problème peut aussi être pris inversement : connaissant deux quantités distinctes, peut-on trouver un modificateur les rendant égales afin de les comparer ?

2.2 Historique

Pendant le doctorat, nous avions donc l'intention d'étudier les modificateurs linguistiques définis par Zadeh, de tenter une taxonomie de ces modificateurs afin d'être en mesure de comparer n'importe quel sous-ensemble flou avec un autre, et de chercher un parallèle à ces modificateurs dans la logique multivalente de De Glas [De Glas, 1989]. De plus, nous avons conjecturé qu'il existait un lien fort entre agrégation et modification, c'est-à-dire que l'action d'agrégation pouvait être vue comme une succession d'actions de modification [Truck, 2002]. Plus généralement, notre hypothèse était que le raisonnement approximatif peut être réécrit à l'aide de l'agrégation et de la modification.

Nous avons donc mené des travaux et réflexions dans le cadre des logiques non classiques, plus particulièrement de la logique floue (théorie des sous-ensembles flous, Zadeh) et de la logique multivalente (principalement initiée par Łukasiewicz [Łukasiewicz, 1920] et dans laquelle s'inscrit la théorie des multi-ensembles de De Glas).

La différence entre la théorie de De Glas et celle de Zadeh réside principalement dans l'expression des données. Chez Zadeh, on considère un univers de discours (ou ensemble de référence) continu ou discret sur lequel la donnée prend ses valeurs et l'on considère un ensemble continu de valeurs comprises entre 0 et 1 exprimant l'appartenance de la donnée aux valeurs de l'ensemble de référence. Chez De Glas, les multi-ensembles évoquent les *bags* de Yager [Yager, 1986]. Un *bag* est un ensemble fini avec une notion de multiplicité de chaque élément. C'est une généralisation d'un ensemble : alors que chaque élément d'un ensemble a une seule appartenance, un élément d'un multi-ensemble peut avoir plus qu'une appartenance (il peut y avoir plusieurs instances d'un élément dans un multi-ensemble). Par ailleurs, chez De Glas, l'univers de discours est discret, nécessairement. Et fini, bien sûr. L'appartenance peut donc être partielle et s'exprime ainsi : $x \in_\alpha A$ qui signifie que x appartient à A avec un degré α ou bien que x satisfait A avec le degré α. Et cette assertion est booléenne. A est un multi-ensemble, α (encore noté v_α) est un terme linguistique ou une expression adverbiale comme "très", "beaucoup", "suffisamment", "peu", etc. Plus précisément, pour De Glas, "x est $v_\alpha A$" s'écrit "x (est v_α)A" ou encore : "x est A" est τ_α-vraie. A chaque v_α correspond donc un degré de vérité τ_α.

Ainsi, chez De Glas, on s'affranchit de la deuxième dimension dans la représentation des données pour ne garder que l'axe des abscisses sur lequel on trouve les degrés associés aux valeurs pouvant être prises. Ces degrés sont arrangés dans une échelle permettant ainsi d'introduire une relation d'ordre total, notée $\leq$, entre eux.

Nous avons donc mené des travaux qui ont permis la construction d'outils dans ces deux théories : principalement des outils de modification de données linguistiques — tantôt vues comme des symboles, tantôt comme des sous-ensembles flous. Des opérateurs de modification notés GSM pour *Generalized Symbolic Modifiers* ont été proposés, ainsi qu'une taxonomie des modificateurs flous, permettant l'expression de n'importe quel sous-ensemble flou comme

étant le modificateur de n'importe quel autre sous-ensemble flou. Puis nous avons utilisé ces GSM pour exprimer le résultat de données agrégées. Ainsi est né l'opérateur d'agrégation appelé la *médiane symbolique pondérée* ou encore SWM (pour *Symbolic Weighted Median*) qui exploite complètement cette idée puisqu'elle est entièrement définie par des GSM.

Définition 1. *Soit* $\mathcal{L}_b = \{a_{0,b-1}, a_{1,b-1}, \ldots, a_{b-1,b-1}\}$ *une collection de* b *éléments ordonnés. Lorsque l'on pondère les éléments, la collection est notée* $\langle a_{0,b-1}^{w_0}, a_{1,b-1}^{w_1}, \ldots, a_{b-1,b-1}^{w_{b-1}} \rangle \in \mathcal{B}^{\mathcal{L}_b}$ *(ensemble de ces collections) telle que* $\sum w_i = 1$. *La* médiane symbolique pondérée $\mathcal{M}$ *est définie comme suit :*

$$
\begin{array}{rcl}
\mathcal{M} : \ \mathcal{B}^{\mathcal{L}_b} & \rightarrow & \mathcal{L}_{b'} \\
\langle a_{0,b-1}^{w_0}, a_{1,b-1}^{w_1}, \ldots, a_{b-1,b-1}^{w_{b-1}} \rangle & \mapsto & \mathcal{M}(\langle a_{0,b-1}^{w_0}, a_{1,b-1}^{w_1}, \ldots, a_{b-1,b-1}^{w_{b-1}} \rangle) \\
 & & = a'^{w'_i}_{i,b'-1} \text{ tel que : } \left| \sum_{p=0}^{i-1} w'_p - \sum_{p=i+1}^{b'-1} w'_p \right| < \varepsilon \\
 & & = m(a_{j,b-1}^{w_j}) \text{ avec } w_j = 1 \\
 & & = m(a_{j,b-1})
\end{array}
$$

avec $m(a_{j,b-1})$ *un GSM appliqué à un élément de la collection initiale* $\mathcal{L}_b$. $\sum_{p=0}^{i-1} w'_p$ *(respectivement* $\sum_{p=i+1}^{b'-1} w'_p$*) est la somme* $\mathcal{S}_1$ *(respectivement* $\mathcal{S}_2$*) des poids des éléments qui sont avant — car la collection est ordonnée — (respectivement après) l'élément* $a'^{w'_i}_{i,b'-1}$.

Afin d'obtenir une médiane correcte (c'est-à-dire que ε est suffisamment petit ou bien égal à zéro), la méthode choisie est de subdiviser l'élément en un certain nombre de sous-éléments. De cette façon, une nouvelle collection est obtenue et les sommes $\mathcal{S}$ peuvent être calculées avec cette nouvelle collection. Ainsi, la médiane est soit un élément initial (pris directement dans $\mathcal{L}_b$), soit un *sous-élément*. Et un sous-élément est un élément sur lequel on a appliqué un ou plusieurs GSM.

Intuitivement, la SWM est à mi-chemin entre une moyenne pondérée et une médiane : on exprime la valeur agrégée comme étant "à peu près" une des valeurs de l'ensemble initial, ce "à peu près" étant défini grâce aux GSM. La SWM satisfait les huit propriétés suivantes : identité, monotonie, conditions aux bornes, pseudo-continuité (comportement non chaotique), symétrie, idempotence, compensation, contre-balancement.

Le lecteur intéressé pourra retrouver d'autres détails dans les articles suivants : [Akdag et al., 2001, Truck et al., 2001b, Truck et al., 2001a, Truck et al., 2002a, Truck et al., 2002b, Truck et al., 2002c] ou dans notre thèse [Truck, 2002].

2.3 Tout premiers travaux post-thèse

Après le doctorat, nous avons mieux assis ces différents concepts, c'est-à-dire amélioré et enrichi les définitions de nos outils, en construisant, par exemple, un treillis complet des GSM afin d'ordonner chaque famille de modificateurs pour sélectionner le plus approprié selon le problème posé. Nous avons aussi redéfini notre agrégateur, en rendant sa définition plus générique et nous avons étendu l'axiomatique dans la théorie des multi-ensembles de De Glas.

TABLE 2.1 – Résumé des GSM renforçants et affaiblissants.

MODE / **NATURE**	**Affaiblissant**	**Renforçant**
Erosion	$\tau_{i'} = \tau_{\max(0,i-\rho)}$, $\mathcal{L}_{M'} = \mathcal{L}_{\max(1,M-\rho)}$ — **EW**(ρ)	$\tau_{i'} = \tau_i$, $\mathcal{L}_{M'} = \mathcal{L}_{\max(i+1,M-\rho)}$ — **ER**(ρ)
		$\tau_{i'} = \tau_{\min(i+\rho,M-\rho-1)}$, $\mathcal{L}_{M'} = \mathcal{L}_{\max(1,M-\rho)}$ — **ER′**(ρ)
Dilatation	$\tau_{i'} = \tau_i$, $\mathcal{L}_{M'} = \mathcal{L}_{M+\rho}$ — **DW**(ρ)	$\tau_{i'} = \tau_{i+\rho}$, $\mathcal{L}_{M'} = \mathcal{L}_{M+\rho}$ — **DR**(ρ)
	$\tau_{i'} = \tau_{\max(0,i-\rho)}$, $\mathcal{L}_{M'} = \mathcal{L}_{M+\rho}$ — **DW′**(ρ)	
Conservation	$\tau_{i'} = \tau_{\max(0,i-\rho)}$, $\mathcal{L}_{M'} = \mathcal{L}_M$ — **CW**(ρ)	$\tau_{i'} = \tau_{\min(i+\rho,M-1)}$, $\mathcal{L}_{M'} = \mathcal{L}_M$ — **CR**(ρ)

2.3.1 Poursuite des travaux sur les GSM

Les GSM sont définis par le biais d'un ensemble totalement ordonné de M degrés de vérité $\mathcal{L}_M = \{\tau_0, \ldots, \tau_i, \ldots, \tau_{M-1}\}$ [2] ($\tau_i \leq \tau_j \Leftrightarrow i \leq j$) avec τ_0 correspondant à faux et τ_{M-1} à vrai. Quatre opérateurs de base sont définis $\vee$ (max), $\wedge$ (min), $\neg$ (négation ou complémentation symbolique, avec $\neg\tau_j = \tau_{M-j-1}$) et l'implication suivante de Łukasiewicz $\rightarrow_L$:

$$\tau_i \rightarrow_L \tau_j = \min(\tau_{M-1}, \tau_{M-1-(i-j)})$$

Nous rappelons qu'un GSM est une application de $\mathcal{L}_M$ dans $\mathcal{L}_{M'}$, qui associe $\tau_{i'} \in \mathcal{L}_{M'}$ à $\tau_i \in \mathcal{L}_M$. $\tau_{i'}$ est calculé en fonction d'un GSM m ayant un rayon — correspondant à une force — ρ, et noté m_ρ.

Définition 2.

$$\begin{array}{rccc} m_\rho\colon & \mathcal{L}_M & \rightarrow & \mathcal{L}_{M'} \\ & \tau_i & \mapsto & \tau_{i'} \end{array}$$

La position du degré dans l'échelle est notée $p(\tau_i) = i$. Une proportion est associée à chaque degré linguistique : $\text{Prop}(\tau_i) = \dfrac{p(\tau_i)}{M-1}$. Ainsi, m_ρ modifie le couple (τ_i, M) en un nouveau couple $(\tau_{i'}, M')$.

Selon ρ, le nouveau couple associé est plus ou moins voisin du couple initial. Plus le rayon est grand, et moins les couples sont voisins. Trois familles de GSM sont ainsi définies : les affaiblissants, les renforçants et les centraux, selon qu'ils affaiblissent, renforcent ou magnifient (ou "démagnifient", à la manière d'un zoom) la valeur initiale. Les affaiblissants et les renforçants sont rappelés dans le tableau 2.1.

2. M est un entier strictement positif.

Quant aux GSM centraux qui permettent de modifier la granularité de l'échelle, sans pour autant changer la proportion (Prop), voici en guise d'exemple la définition du modificateur DC′ :

$$\mathrm{DC}'(\rho) = \begin{cases} \tau_{i'} = \begin{cases} \tau_{\frac{i*(M\rho-1)}{M-1}} & \text{si } \tau_{\frac{i*(M\rho-1)}{M-1}} \in \mathcal{L}_{M'} \\ \tau_{\left\lfloor \frac{i*(M\rho-1)}{M-1} \right\rfloor} & \text{sinon (pessimiste)} \\ \tau_{\left\lfloor \frac{i*(M\rho-1)}{M-1} \right\rfloor + 1} & \text{sinon (optimiste)} \end{cases} \\ \mathcal{L}_{M'} = \mathcal{L}_{M\rho} \end{cases}$$

Les cas "pessimiste" et "optimiste" correspondent à des arrondis respectivement par défaut et par excès.

L'ordonnancement de chaque famille de modificateurs passe par l'utilisation de Prop, associée à chaque modificateur. En effet, si $\tau_i \overset{m_\rho}{\longmapsto} \tau_{i'}$ alors $\mathrm{Prop}(\tau_i) \overset{m_\rho}{\longmapsto} \mathrm{Prop}(\tau_{i'})$. De plus, si $\tau_{i'} = m_\rho(\tau_i)$ alors $\mathrm{Prop}(\tau_{i'}) = \mathrm{Prop}(m_\rho(\tau_i)) = \mathrm{Prop}(m_\rho)$ (abus d'écriture). Parmi les GSM, certains sont plus puissants que d'autres. On établit donc une relation d'ordre entre eux.

Définition 3. *Soient $m_{\rho,1}$ et $m_{\rho,2}$ deux GSM avec le même rayon ρ. On note $m_{\rho,1}(\tau_i) = \tau_{i'_1}$ avec $\tau_{i'_1} \in \mathcal{L}_{M'_1}$ et $m_{\rho,2}(\tau_i) = \tau_{i'_2}$ avec $\tau_{i'_2} \in \mathcal{L}_{M'_2}$*

$$\textit{Prop}(m_{\rho,1}) = \frac{p(\tau_{i'_1})}{M'_1 - 1} \quad \textit{est comparable à} \quad \textit{Prop}(m_{\rho,2}) = \frac{p(\tau_{i'_2})}{M'_2 - 1} \quad \textit{si et seulement si,}$$

$$\textit{pour tout } M, \quad \begin{cases} \textit{Prop}(m_{\rho,1}) \leq \textit{Prop}(m_{\rho,2}) \quad \forall \tau_i \in \mathcal{L}_M \\ \textit{ou} \\ \textit{Prop}(m_{\rho,2}) \leq \textit{Prop}(m_{\rho,1}) \quad \forall \tau_i \in \mathcal{L}_M \end{cases}$$

Définition 4. *Deux GSM $m_{\rho,1}$ et $m_{\rho,2}$ admettent une relation d'ordre si et seulement si $\textit{Prop}(m_{\rho,1})$ est comparable à $\textit{Prop}(m_{\rho,2})$, pour tout $\tau_i \in \mathcal{L}_M$ donné. Formellement, la relation $\trianglelefteq$ est définie comme suit :*

$$\textit{Pour tout } M, \quad m_{\rho,1} \trianglelefteq m_{\rho,2} \Leftrightarrow \textit{Prop}(m_{\rho,1}) \leq \textit{Prop}(m_{\rho,2}) \quad \forall \tau_i \in \mathcal{L}_M$$

Si deux GSM $m_{\rho,1}$ et $m_{\rho,2}$ sont en relation l'un avec l'autre, la comparaison entre leurs Prop est possible puisque l'on compare des rationnels, mais surtout parce que l'unité est la même pour tout $\tau_i \in \mathcal{L}_M$ (nous rappelons que les degrés sont uniformément distribués sur les échelles). Ainsi, la relation binaire $\trianglelefteq$ sur les GSM est une relation d'ordre partielle (certains GSM ne sont pas comparables avec d'autres) que l'on peut exprimer dans un treillis (*cf.* figure 2.1).

Par ailleurs, nous avons aussi étudié les limites de nos modificateurs, c'est-à-dire que nous avons cherché à savoir pour quels GSM les limites des échelles sont atteintes et pour lesquels elles ne le seront jamais. Nous avons ainsi pu classer les GSM en deux familles : les "finis" et les "infinis".

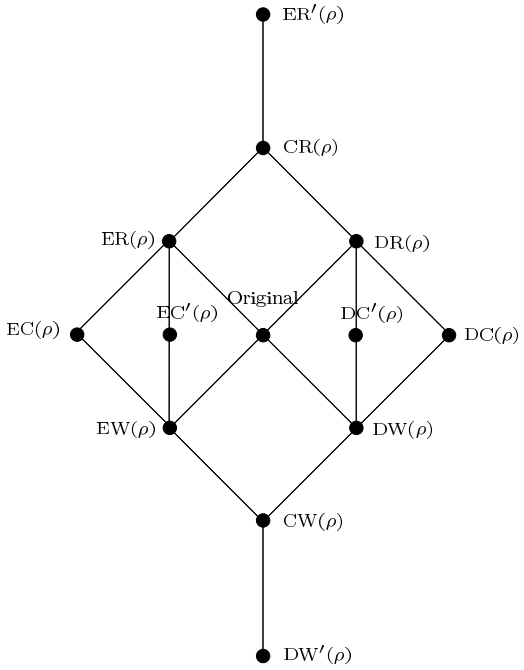

FIGURE 2.1 – Treillis pour la relation $\trianglelefteq$.

Enfin, nous avons composé les GSM entre eux, à la manière de l'opérateur $\circ$. On distingue deux types de compositions : les homogènes et les hétérogènes. Nous appelons compositions homogènes les compositions de modificateurs de la même famille (par exemple, un EW avec un EW), tandis que les hétérogènes impliquent des GSM de familles différentes.

Théorème 1. *Si m_{ρ_1} est un GSM affaiblissant ou renforçant quelconque avec un rayon ρ_1, que m_{ρ_2} est un GSM de même nature que m_{ρ_1} avec un rayon ρ_2, ... et que m_{ρ_n} est un GSM de même nature que m_{ρ_1} avec un rayon ρ_n, alors m_{ρ_s} est un GSM de même nature que m_{ρ_1}, mais avec un rayon d'action ρ_s égal à la somme des rayons d'action, soit $\rho_1 + \rho_2 + \ldots + \rho_n$.*

Théorème 2. *En composant n GSM quelconques, on obtient toujours un couple degré/échelle valide, c'est-à-dire que le degré $\tau_{i'}$ résultant des n modifications est un entier, M' résultant des n modifications est un entier non nul et différent de 1, et $\tau_{i'} < M'$.*

Le lecteur intéressé pourra retrouver tous les détails et des précisions sur notre agrégateur médiane dans les articles suivants : [Truck et al., 2003, Truck et Akdag, 2005, Akdag et Truck, 2009, Truck et Akdag, 2006, Truck et Akdag, 2009].

Nous avons aussi poursuivi sur des questions d'"apprentissage de modifications" et sur la "composition algorithmique" de modificateurs linguistiques : par exemple, que vaut "un peu plus *plus* un tout petit peu moins" ? Et que vaut "beaucoup moins *sachant* un petit peu moins" ? On cherchait à sauvegarder (ou "apprendre") les traitements associés à des modificateurs linguistiques exprimés en langage naturel. Il s'agissait de faire le parallèle entre la composition (au sens mathématique) des GSM et la composition (application successive sous forme additive ou soustractive) de différentes modifications que l'on faisait subir à une donnée. Nous avons donc imaginé deux opérateurs équivalents à un *plus* et à un *sachant* et,

en particulier, l'opérateur $\circ$ est précisé. Ces deux opérateurs permettent d'*apprendre* comme d'*oublier* les significations des modifications à appliquer aux objets considérés.

2.3.2 Poursuite des travaux sur l'apprentissage de modifications

Telles que présentées, les opérations d'apprentissage de modifications sont des opérations d'adjonction que l'on note II. Par exemple, dans le domaine colorimétrique, en considérant une certaine couleur, si l'on la souhaite "beaucoup plus foncée", et que l'on corrige ensuite par "un petit peu moins fade", on réalise l'opération :
"beaucoup plus foncé$_{(2)}$" $\leftarrow$ "beaucoup plus foncé$_{(1)}$" II "un petit peu moins fade"
et on indice entre parenthèses la nouvelle expression pour la distinguer de la première.

Nous utilisons un graphe pour stocker les modifications attachées aux qualificatifs et, indirectement, les modifications des intensités. Chaque qualificatif X est associé à une fonction élémentaire qui effectue le traitement par défaut noté $trtX$, pour "*tr*aitement de X" (*cf.* figure 2.2 où trois qualificatifs *fade*, *bleuté*, *foncé*, parmi un ensemble, sont présentés).

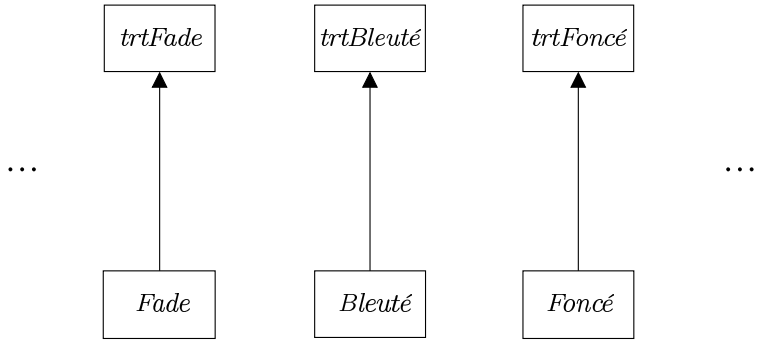

FIGURE 2.2 – Etat du graphe avant tout apprentissage de modifications.

Soit maintenant l'apprentissage de la modification suivante :
"beaucoup plus bleuté$_{(2)}$" $\leftarrow$ "beaucoup plus bleuté$_{(1)}$" II "un peu plus foncé".

Formellement, l'opérateur d'adjonction II associe deux couples formés chacun d'un modificateur m et d'un qualificatif q avec un nouveau couple (modificateur, qualificatif) :

Définition 5. *Soient* $\mathbb{M}$ *l'ensemble des modificateurs et* $\mathcal{Q}$ *l'ensemble des qualificatifs colorimétriques.*

$$\begin{array}{rcl} \text{II} : \quad (\mathbb{M} \times \mathcal{Q})^2 & \rightarrow & \mathbb{M} \times \mathcal{Q} \\ \langle (m_1, q_1), (m_2, q_2) \rangle & \mapsto & (m'_1, q'_1) \end{array}$$

$(m_1, q_1), (m_2, q_2)$ sont deux couples (modificateur, qualificatif) et (m'_1, q'_1) est un nouveau couple qui est identique, linguistiquement parlant, à (m_1, q_1), mais dont le modificateur m'_1 et le qualificatif q'_1 ne correspondent pas aux mêmes traitements sur la couleur que m_1 et q_1. Appliquer le traitement q'_1 est équivalent à appliquer le traitement q_1 puis le traitement q_2. De même, appliquer m'_1 est équivalent à appliquer m_1 puis m_2.

'Il' se décline donc en deux opérateurs d'adjonction : l'un pour composer les qualificatifs entre eux, l'autre pour composer les modificateurs entre eux. L'opérateur de composition des modificateurs a déjà été évoqué plus haut, il s'agit de $\circ$. Ainsi, ce que l'on cherche à calculer est : $m'_1 = m_2 \circ m_1$. Dans notre exemple, m_1 correspond à "beaucoup plus" et m_2 correspond à "un peu plus".

Graphiquement, il s'agit d'indiquer que *bleuté* doit être relié à *trtFoncé*, en plus d'être relié à *trtBleuté*. Ainsi, 'Il' se code en ajoutant un nœud intermédiaire dans le graphe qui permet de relier *bleuté* avec *trtFoncé*, mais sans perdre *trtBleuté* pour autant. Par ailleurs, pour ne pas perdre l'information sur l'*intensité* associée à *foncé* ("un peu plus"), on stocke aussi une intensité dans ce nœud. Ce nouveau nœud intermédiaire possède donc deux fils et une intensité associée au deuxième fils afin de se souvenir qu'appliquer le qualificatif *bleuté* revient en fait à appliquer d'abord *trtBleuté*, puis "un peu plus" *trtFoncé* (*cf.* figure 2.3).

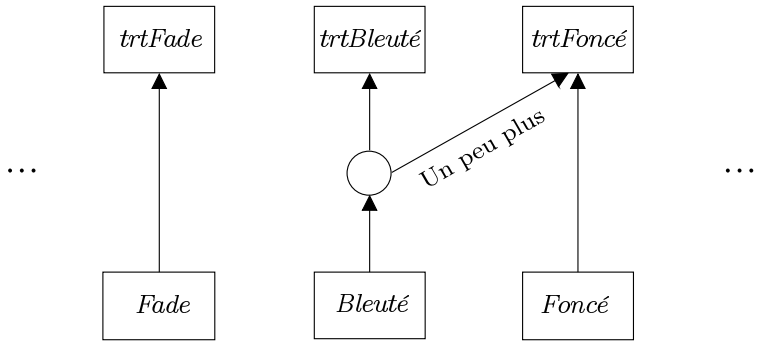

FIGURE 2.3 – Nœud intermédiaire (première étape).

On voit ici tout de suite que les notions de nuances du langage naturel vont prendre toute leur importance : dans l'exemple ci-dessus, l'utilisateur, en souhaitant *bleuter* sa couleur, veut certes la *bleuter* mais aussi la *foncer* ***légèrement*** : tout le problème est dans le mot *légèrement*, comment l'interpréter ? Et *a contrario*, le même utilisateur qui souhaiterait retirer du bleu dans sa couleur (il la demanderait "beaucoup moins bleutée", par exemple) devrait voir appliquer son vœu précédent, mais à l'inverse : la couleur, en plus d'être moins bleutée, devrait être éclaircie ("un peu moins foncée"). Ce qui veut dire que l'on doit être capable d'appliquer le nouveau traitement associé à *bleuté*, quelle que soit la nuance (ou intensité) demandée dans la requête. Il est donc nécessaire d'enregistrer le changement de traitement de *bleuté* ***avec*** la nuance associée, et sans perdre l'information "beaucoup plus". On doit ensuite pouvoir retrouver la modification apprise sur *bleuté* en changeant automatiquement l'intensité du nœud intermédiaire, donc stocker dans le nœud une intensité *tempérée* en fonction de l'intensité de départ. Dans l'exemple, on stocke donc l'expression « "un peu plus" *sachant* "beaucoup plus" » (*cf.* figure 2.4) qui correspond précisément à $m_2 \circ m_1$.

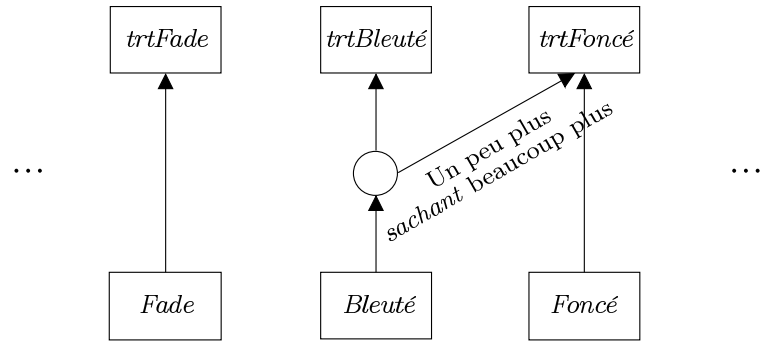

FIGURE 2.4 – Nœud intermédiaire avec une intensité *tempérée* (étape finale).

Pour éviter d'apprendre des modifications irréalistes, on n'autorise que des modifications élémentaires, c'est-à-dire qu'un qualificatif X ne peut être directement associé qu'à deux traitements ($trtX$ et $trtY$). Par exemple, on ne peut pas directement modifier *fade* comme étant *fade*, *bleuté* et *foncé*. Pour faire cela, il faut d'abord modifier *fade* puis *bleuté*, ou bien d'abord *bleuté* puis *fade*.

Ainsi, un nœud intermédiaire aura toujours exactement deux fils et une intensité associée au deuxième fils. Le premier fils pointera toujours sur une modification élémentaire ($trtX$) tandis que le deuxième pointera soit sur un $trtX$ soit sur un autre nœud intermédiaire. On peut ainsi effectuer d'autres apprentissages qui mettent en jeu des qualificatifs déjà modifiés. Par exemple, "plus fade$_{(2)}$" ← "plus fade$_{(1)}$" II "un peu moins bleuté$_{(2)}$" produira le graphe de la figure 2.5.

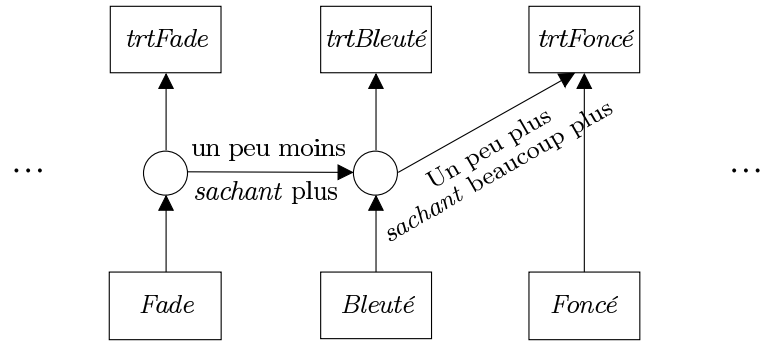

FIGURE 2.5 – Ajout d'un nouveau nœud intermédiaire.

Les nœuds intermédiaires du graphe et un masque associé à chaque qualificatif permettent au graphe de ne jamais contenir de boucle. En effet, le masque permet d'éviter à deux qualificatifs d'avoir leurs nœuds intermédiaires pointant l'un vers l'autre, y compris par transitivité.

Supposons que l'on ait n qualificatifs. Chacun possède un masque de n bits qui lui est associé. Chaque bit d'un masque correspond au traitement d'un qualificatif : s'il est à 1 alors le traitement doit être effectué, sinon il ne doit pas l'être. Par exemple, pour un qualificatif

q_2 parmi 8, si son masque vaut : 00110010, cela signifie qu'appliquer q_2 consistera à effectuer trois traitements consécutivement : le traitement de q_2 car le 2^e bit le moins significatif vaut 1, le traitement de q_5 et le traitement de q_6. Considérons maintenant un exemple complet avec quatre qualificatifs (masques de 4 bits). La figure 2.6 montre l'état initial du graphe.

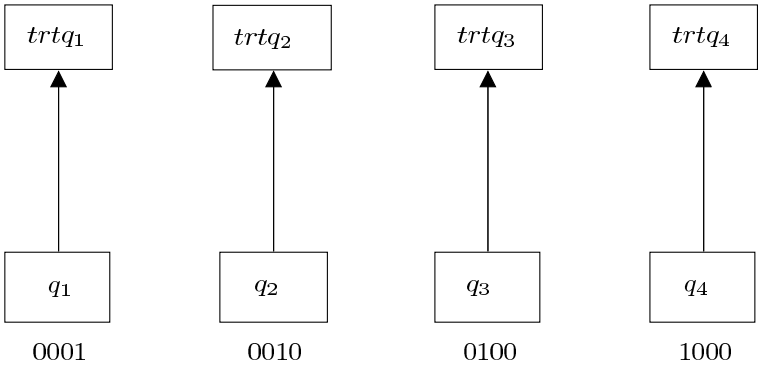

FIGURE 2.6 – Etat initial du graphe.

Dans la première étape, le système apprend que le qualificatif q_4 est aussi associé au qualificatif q_3 : après vérification avec un OU exclusif qui doit donner un résultat nul, les deux masques sont "ajoutés", ce qui correspond à l'opération logique OU (*cf.* figure 2.7). La vérification permet de s'assurer que l'"ajout" entre les deux masques est possible. Ici, $1000 \wedge 0100 = 0000$, on peut donc calculer le nouveau masque de q_4 qui vaut désormais $1000 \vee 0100 = 1100$.

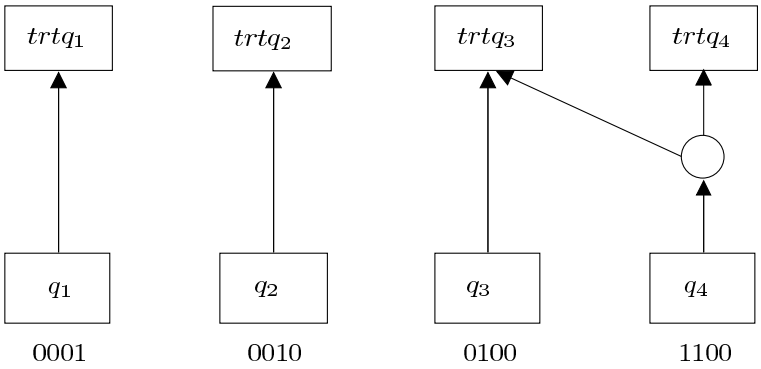

FIGURE 2.7 – Première étape : l'apprentissage peut être effectué.

Dans la deuxième étape, le système apprend que le qualificatif q_3 est aussi associé au qualificatif q_2 : après vérification, les deux masques sont "ajoutés" et, là encore après vérification, il faut mettre à jour le masque de q_4 car q_4 est maintenant associé à q_3 et q_2 (*cf.* figure 2.8). La mise à jour se fait aussi à l'aide d'un OU.

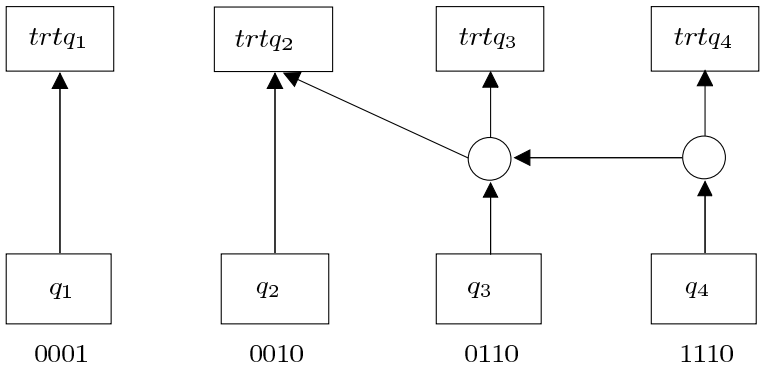

FIGURE 2.8 – Deuxième étape : l'apprentissage peut être effectué.

Dans la troisième étape, le système apprend que le qualificatif q_2 est aussi associé au qualificatif q_4, autrement dit que q_2 doit pointer aussi sur $trtq_4$. On procède à la vérification et on obtient un résultat différent de zéro ($0010 \wedge 1110 = 0010$) donc les deux masques ne peuvent être "ajoutés". Le système conclut ainsi que l'apprentissage de cette modification est impossible (*cf.* figure 2.9).

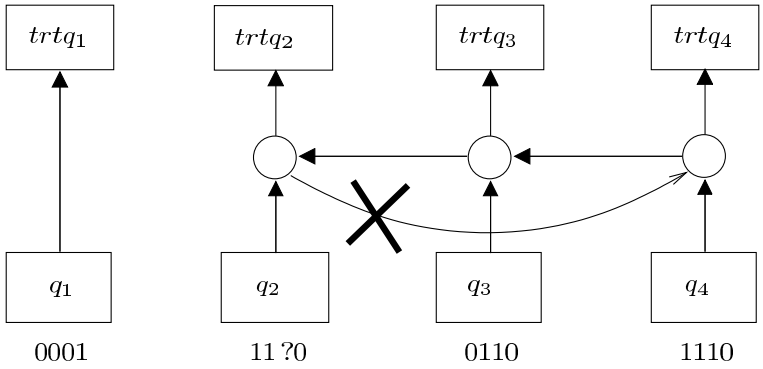

FIGURE 2.9 – Troisième étape : l'apprentissage ne peut être effectué.

Une fois l'apprentissage des modifications effectué, les traitements à exécuter sont donc des **compositions** des traitements élémentaires.

Par ailleurs, si l'on souhaite *oublier* ce que l'on avait appris sur un qualificatif, on redonne au qualificatif sa modification initiale et on supprime les nœuds intermédiaires concernés, sauf si d'autres nœuds pointent encore sur ces nœuds. Si l'on oublie l'apprentissage de *fade*, par exemple, il y a suppression d'un nœud et on obtient le graphe de la figure 2.4. Si, par contre, on n'oublie pas *fade*, mais que l'on oublie l'apprentissage de *bleuté*, aucun nœud n'est supprimé et on obtient le graphe de la figure 2.10 dans laquelle on n'oublie pas l'apprentissage de *bleuté* pour le qualificatif *fade* car on veut conserver de façon intacte la modification de *fade*.

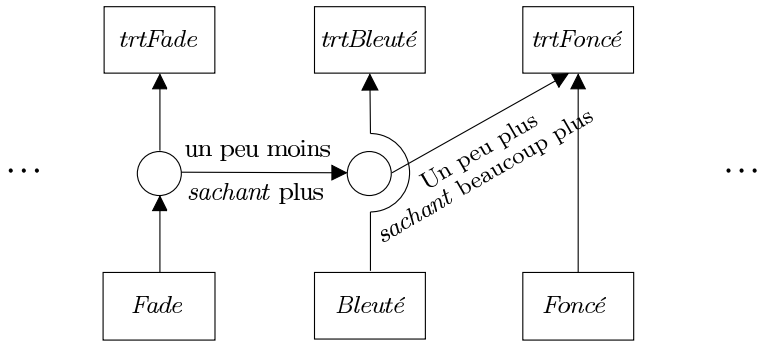

FIGURE 2.10 – Oubli de l'apprentissage de "bleuté".

Concernant la question de l'intensité de la modification, on cherche maintenant à calculer m'_1 (*cf.* Définition 5), c'est-à-dire à définir concrètement l'opérateur de composition $\circ$ puisque $m'_1 = m_2 \circ m_1$. Dans notre premier exemple, on souhaitait ajouter "un peu moins fade" à "beaucoup plus bleuté$_{(1)}$" et obtenir ainsi "beaucoup plus bleuté$_{(2)}$".

Pour ce faire, les contraintes suivantes doivent être satisfaites :

- si l'on applique plus tard "beaucoup plus bleuté$_{(2)}$", on devra obtenir le même résultat que celui que l'on a obtenu avec "beaucoup plus bleuté$_{(1)}$" puis "un peu moins fade",
- si l'on applique "bleuté$_{(2)}$" avec un autre degré d'intensité, le résultat devra être cohérent avec les modifications que l'on vient de faire. Dans notre exemple, appliquer "beaucoup *moins* bleuté$_{(2)}$" devra faire intervenir "un peu *plus* fade".

Pour cela, on procède de la façon suivante :

- à l'apprentissage, on tempère la correction en tenant compte de l'intensité initiale. Par exemple, si l'on ajoute "un peu *moins* fade" alors qu'on est en train de travailler sur "beaucoup *plus* bleuté$_{(1)}$", on doit relativiser le modificateur de "fade" : dans le graphe, on stockera "un tout petit peu *moins* fade". On note :
  "un peu *moins*" *sachant* "beaucoup *plus*" = "un tout petit peu *moins*", c'est-à-dire $m_2 \circ m_1 = m'_1$ et correspond à "un tout petit peu *moins*".
- lorsqu'on cherchera ensuite à appliquer à nouveau "beaucoup *plus* bleuté$_{(2)}$", on composera le modificateur voulu ("beaucoup *plus*") avec le modificateur stocké dans le graphe ("un tout petit peu *moins*"). On note donc :
  "Beaucoup *plus*" + "un tout petit peu *moins*" = "un peu *moins*", c'est-à-dire que l'on introduit un nouvel opérateur $\odot$ correspondant à '+' et l'on note $m_1 \odot m'_1 = m_2$.

Cela implique que, pour m_i et m_j deux modificateurs linguistiques, on a :
$m_j \odot (m_i \circ m_j) = m_i$, condition indispensable pour pouvoir *oublier* ce qu'on a appris. $\odot$ est la fonction inverse de $\circ$.

Construction de la fonction $\odot$

Soient n (n, nombre pair) modificateurs. À chaque modificateur, on associe un entier, qu'on appelle *intensité*, en fonction de son rayon ρ et de son sens (*plus* ou *moins*, c'est-à-

dire famille renforçante ou affaiblissante), noté $I(m)$. Par exemple, si $n = 14$, on aura 14 modificateurs (de "énormément moins" à "énormément plus" en passant par "un tout petit peu moins", etc.) et les intensités suivantes : 7 pour "énormément plus", 1 pour "un tout petit peu plus", ... , -7 pour "énormément moins". Il n'y a pas d'intensité égale à 0 car cela correspondrait à un modificateur sans action.

L'ensemble des modificateurs considéré est tel que, pour tout modificateur m de cet ensemble, on peut trouver un modificateur noté $\overline{m}$ tel que $I(\overline{m}) = -I(m)$. m et $\overline{m}$ ont un sens opposé : si m est renforçant alors $\overline{m}$ est affaiblissant, et *vice-versa*.

On souhaite que la composition d'un modificateur fort avec un modificateur faible donne un modificateur moyen. Pour cela, on considère qu'un des modificateurs est *neutre* et on le note m_0. Son intensité $I(m_0)$ vaut $(n+2)/4$. Dans notre cas, c'est "plutôt plus" (d'intensité 4). C'est-à-dire que composer "plutôt plus" avec un autre modificateur ne change pas celui-ci.

Par ailleurs, la composition de "un petit peu moins" *sachant* "beaucoup plus", par exemple, doit être à peu près égale à "*un tout petit peu moins*", si l'on veut rester cohérent. Ce qui signifie que la composition d'un modificateur linguistique d'intensité négative (en l'occurrence, "un petit peu moins") avec un modificateur linguistique d'intensité positive (en l'occurrence, "beaucoup plus") doit donner un modificateur d'intensité négative (en l'occurrence, "un tout petit peu moins").

Si $I(m_1)$ et $I(m_2)$ sont positifs alors on devra avoir :

1. Si $m_1 = m_0$, $I(m_1 \odot m_2) = I(m_2 \odot m_1) = I(m_2)$ (symétrie et élément neutre)
2. Si $I(m_3) > I(m_1)$, $I(m_3 \odot m_2) > I(m_1 \odot m_2)$ et si $I(m_4) > I(m_2)$, $I(m_1 \odot m_4) > I(m_1 \odot m_2)$ (monotonie)

La solution retenue est : $I(m_1 \odot m_2) = I(m_1) + I(m_2) - I(m_0)$

De plus, si l'on change le signe de $I(m_1)$ ou bien de $I(m_2)$, on doit inverser le signe du résultat. Par exemple, si $I(m_1) > 0$ et $I(m_2) < 0$, alors : $I(m_1 \odot m_2) = -I(m_1 \odot \overline{m_2}) = -(I(m_1) + I(\overline{m_2}) - I(m_0)) = -I(m_1) + I(m_2) + I(m_0)$.

On obtient pour les autres cas :

$$I(m_1 \odot m_2) = \begin{cases} I(m_1) + I(m_2) - I(m_0) & \text{si } (I(m_1) \geq 0 \text{ et } I(m_2) > 0) \text{ ou si } (I(m_1) > 0 \text{ et } I(m_2) \geq 0) \\ -I(m_1) + I(m_2) + I(m_0) & \text{si } I(m_1) > 0 \text{ et } I(m_2) < 0 \\ I(m_1) - I(m_2) + I(m_0) & \text{si } I(m_1) < 0 \text{ et } I(m_2) > 0 \\ -I(m_1) - I(m_2) - I(m_0) & \text{si } (I(m_1) \leq 0 \text{ et } I(m_2) < 0) \text{ ou si } (I(m_1) < 0 \text{ et } I(m_2) \leq 0) \\ 0 & \text{sinon (si } I(m_1) = 0 \text{ et } I(m_2) = 0) \end{cases}$$

Plus généralement, on définit $\odot$ ainsi :

Définition 6. *Soient deux modificateurs m_1 et m_2 avec, pour intensités, respectivement, $I(m_1)$ et $I(m_2)$. La fonction $\odot(m_1, m_2)$ est définie par l'intermédiaire de l'intensité $I(m_1 \odot m_2)$:*

$$\begin{array}{rcl} \odot \ : \ \mathbb{N} \times \mathbb{N} & \rightarrow & \mathbb{N} \\ (I(m_1), I(m_2)) & \mapsto & I(m_1 \odot m_2) \end{array}$$

avec $I(m_1 \odot m_2) = \pm I(m_1) \pm I(m_2) \pm I(m_0)$, selon les valeurs $I(m_1)$ et $I(m_2)$.

TABLE 2.2 – Signes de $I(m_1)$, $I(m_2)$ et $I(m_1 \odot m_2)$ selon les cas.

$I(m_1)$	$I(m_2)$	$I(m_3) = I(m_1 \odot m_2)$
$+$	$+$	$+$
$+$	$-$	$-$
$-$	$+$	$-$
$-$	$-$	$+$

Calcul de $\circ$

$\circ(., m_1)$ est l'inverse de $\odot(m_1, .)$, c'est-à-dire que si $m_3 = m_1 \odot m_2$, on a alors $m_2 = m_3 \circ m_1$.

Les signes de $I(m_1)$, $I(m_2)$ et $I(m_3) = I(m_1 \odot m_2)$ sont liés comme le montre le tableau 2.2.

Les signes de $I(m_3)$ et $I(m_1)$ permettent d'obtenir celui de $I(m_2)$, donc de $I(m_3 \circ m_1)$. Si $I(m_1)$ et $I(m_3)$ sont positifs, on inverse la première forme de $\odot$: $I(m_1 \odot m_2) = I(m_3) = I(m_1) + I(m_2) - I(m_0)$, donc, ayant $m_2 = m_3 \circ m_1$, nous avons : $I(m_3 \circ m_1) = I(m_3) - I(m_1) + I(m_0)$.

On obtient pour les autres cas :

$$I(m_3 \circ m_1) = \begin{cases} I(m_3) - I(m_1) + I(m_0) & \text{si } (I(m_1) \geq 0 \text{ et } I(m_3) > 0) \text{ ou si } (I(m_1) > 0 \text{ et } I(m_3) \geq 0) \\ I(m_3) + I(m_1) - I(m_0) & \text{si } I(m_1) > 0 \text{ et } I(m_3) < 0 \\ -I(m_3) - I(m_1) - I(m_0) & \text{si } I(m_1) < 0 \text{ et } I(m_3) > 0 \\ I(m_3) + I(m_1) - I(m_0) & \text{si } (I(m_1) \leq 0 \text{ et } I(m_3) < 0) \text{ ou si } (I(m_1) < 0 \text{ et } I(m_3) \leq 0) \\ 0 & \text{sinon (si } I(m_1) = 0 \text{ et } I(m_3) = 0) \end{cases}$$

Plus généralement, on définit $\circ$ ainsi :

Définition 7. *Soient deux modificateurs m_1 et m_2 avec, pour intensités, respectivement, $I(m_1)$ et $I(m_2)$. La fonction $\circ(m_1, m_2)$ est définie par l'intermédiaire de l'intensité $I(m_1 \circ m_2)$:*

$$\begin{array}{rcl} \circ \ : \ \mathbb{N} \times \mathbb{N} & \rightarrow & \mathbb{N} \\ (I(m_1), I(m_2)) & \mapsto & I(m_1 \circ m_2) \end{array}$$

avec $I(m_1 \circ m_2) = \pm I(m_1) \pm I(m_2) \pm I(m_0)$, selon les valeurs $I(m_1)$ et $I(m_2)$.

Concernant la propriété des conditions aux bornes, on constate que, dans la définition, les intensités I peuvent sortir des limites admises $[-n/2, n/2]$. Travaillant sur des nombres et non sur des symboles, nous avons volontairement exclu le recours aux opérations min ou max pour pouvoir garder la propriété d'inverse entre les fonctions $\odot$ et $\circ$. Nous avons donc en réalité deux intensités, l'une théorique qui peut sortir des bornes, et l'autre, calculée en utilisant min ou max.

Pour conclure, nous pouvons noter que ces outils donnent une "signification formelle" à des expressions, à la manière des liens signifiant / signifié qui sont étudiés en sémantique des langages de programmation. $\odot$ et $\circ$, inverses l'une de l'autre, définissent les opérations de base pour composer des fragments d'expression afin de traduire ensuite des expressions plus complexes.

Le lecteur intéressé pourra trouver d'autres détails concernant notamment l'implémentation dans l'article suivant : [Truck et Akdag, 2003].

2.4 Conclusion

Lors de notre doctorat, nous avions amorcé des travaux qui ont donné lieu à des outils et à des réflexions sur la gestion des imprécisions (taxonomie des modificateurs flous, mise en place d'outils pour modifier et agréger des termes linguistiques). Par la suite, il nous a semblé nécessaire d'étendre ce travail pour permettre une *approche linguistique de bout en bout*. L'intérêt de cela, outre que la manipulation de termes linguistiques diminue grandement la quantité d'information traitée, est de pouvoir, à tout moment dans la chaîne du raisonnement, s'adresser à des humains.

Modifier ou agréger des termes linguistiques présuppose que le système est en possession de ces termes, c'est-à-dire qu'il les a (i) obtenus par question/réponse ou (ii) calculés automatiquement.

La quête de ces termes par question/réponse est facilitée par les modificateurs GSM qui peuvent être utiles pour préciser l'intention de l'utilisateur, en particulier en utilisant les opérateurs $\circ$ et $\odot$ comme on vient de le voir. Cependant, l'élicitation de ses choix et préférences reste encore un problème important à traiter, surtout s'ils sont liés entre eux. Cela permet de s'adapter aux appréciations personnelles et donc subjectives, relatives à la perception humaine.

Le calcul automatisé de ces termes suppose une capacité à transformer des valeurs numériques captées (d'une façon ou d'une autre) en valeurs linguistiques. On sait que les techniques de partitionnement flou et d'affectation des valeurs aux partitions permettent typiquement cela. Mais des techniques d'automatisation du partitionnement et de l'affectation sont encore à imaginer pour faciliter le passage valeurs numériques $\leftrightarrow$ termes linguistiques.

De plus, une *approche linguistique de bout en bout* implique un traitement sur les données faisant uniquement appel aux termes linguistiques (ou à leurs équivalents). La figure 2.11 montre les points déjà amorcés dans notre doctorat : la légende en haut à droite met en exergue trois avancées (notées MF, MS et AS) et les place sur le schéma général.

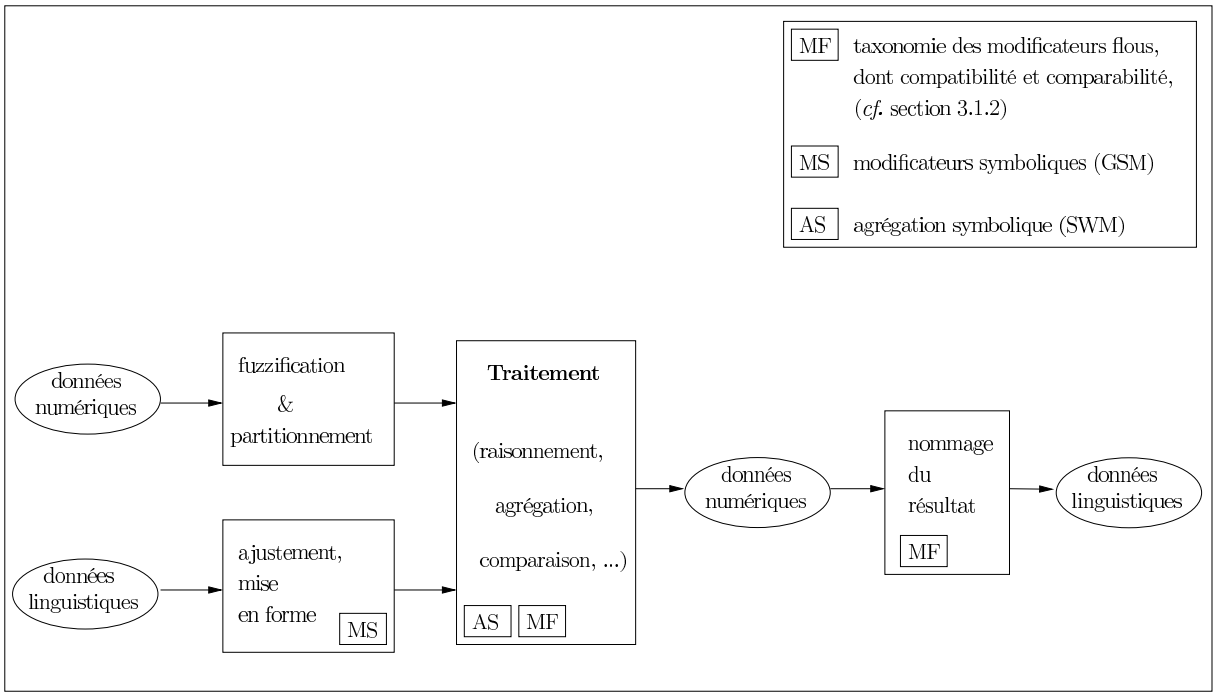

FIGURE 2.11 – Schéma général avec mise en exergue des travaux de thèse ou immédiatement post-thèse.

L'objectif poursuivi après le doctorat a donc été de se donner des bases de travail pour traiter le mieux possible les informations linguistiques afin de compléter les manques soulignés ci-dessus.

On notera par ailleurs que les deux théories (celle de Zadeh et celle de De Glas) qui sous-tendaient notre cadre de travail étaient vues comme deux approches très distinctes pour raisonner, leur (seul) point commun étant la manipulation de données linguistiques. En pratique, elles se sont montrées des alternatives intéressantes et complémentaires : l'une (floue) travaillant sur des univers discrets ou continus, l'autre s'interdisant le continu, s'autorisant seulement l'utilisation d'un ensemble fini de symboles ordonnés sur une échelle, muni principalement de l'opération de négation, du min et du max.

Chapitre 3

Démarche et travaux poursuivis

Sommaire

NOUS AVONS POURSUIVI NOS RECHERCHES dans le domaine de la perception des informations et dans celui de la capture des intentions. En particulier, nous nous sommes intéressée au *perceptual computing*, c'est-à-dire à l'application du *Computing with Words* [Zadeh, 1996] à la notion de perception humaine. C'est la perception visuelle des couleurs qui a d'abord focalisé notre attention. Puis nous avons étendu notre champ d'investigation dans les arts de la scène avec la perception de mouvement et la capture des intentions scéniques. Dernièrement, c'est la capture des intentions des programmeurs, pour aider à la prise de décision qui a été et est toujours au cœur de nos recherches.

3.1 Le calcul à l'aide de mots dans la perception visuelle

D'un côté, nous avons poursuivi nos travaux dont le principal domaine d'application était la colorimétrie. Le but était de mettre la perception des couleurs au centre des recherches et des préoccupations. La couleur est un excellent exemple d'intervalle où les valeurs ne peuvent être placées de manière équidistante si l'on considère la perception humaine. En effet, pour la couleur verte, par exemple, modifier *un peu* la teinte ne donne visuellement aucune différence

alors que modifier *de la même manière* la teinte du jaune donne visuellement une grande différence (le jaune devient orange, voire rouge).

La question de la modélisation de la perception n'est pas nouvelle : par exemple, les transducteurs du colorimètre de Benoît modélisent la couleur dans l'espace RVB (rouge, vert, bleu) et composent ainsi une échelle nominale de huit couleurs. La question de la perception est traitée par un apprentissage de ces huit couleurs caractéristiques [Benoît, 1993].

De notre côté, nous voulions envisager la perception *via* un nouvel espace car on a constaté que les espaces de type UCS (Uniform Color Scale) bien adaptés à la perception humaine sont difficilement manipulables car nécessitant de nombreuses transformations (linéaires ou non) d'un espace vers un autre avant d'être visuellement utilisables. Ainsi, notre travail a commencé par la modélisation floue d'un espace de couleur afin de le rendre "uniforme", c'est-à-dire de rapprocher (respectivement éloigner) artificiellement deux points dont la perception colorimétrique est sensiblement la même (respectivement différente).

En continuant sur cette lancée, nous avons envisagé la classification d'images par *couleur(s) dominante(s)*. Nous procédions à un "vote" des pixels puis à une affectation des images à une classe de couleur (ou sous-classe quand il s'agit d'une couleur *modifiée*, par exemple "rouge foncé").

3.1.1 Profil colorimétrique d'une image

Pour établir le profil colorimétrique d'une image, on s'intéresse d'abord à la question de la représentation de la couleur. L'espace le plus couramment utilisé est appelé RVB. C'est un espace tri-dimensionnel dans lequel les trois couleurs "primaires" évoluent entre 0 et 255 généralement. Le triplet $(0, 0, 0)$ correspond au noir et $(255, 255, 255)$ au blanc. Dans les méthodes de classification, cet espace est souvent utilisé dans les calculs de similarité grâce à un histogramme colorimétrique [Swain et Ballard, 1991] représentant la distribution des pixels sur les trois axes rouge, vert et bleu. Mais, avec ce genre de représentation, il est très difficile de représenter approximativement une couleur (par exemple une gamme de bleus) car la corrélation entre les trois teintes R, V et B ne correspond pas à la perception visuelle. De plus, l'identification d'une couleur passe nécessairement par les trois composantes, ce qui n'est pas très intuitif. Pour faciliter cette identification, on choisit un espace permettant une caractérisation *via* une seule composante : la teinte (*hue*). L'espace HLS (Hue, Lightness, Saturation) permet cela, sauf quand la couleur est trop pâle ou trop sombre. La saturation correspond à la quantité de blanc et la luminance à l'intensité de la lumière dans la couleur. L'identification d'une couleur passe par deux étapes : d'abord H puis L et S. H identifie de manière grossière la couleur, puis L et S de façon plus fine. L'espace HLS est habituellement représenté par un cylindre ou un bi-cône. En pratique, plusieurs modèles de représentation de la couleur utilisent aussi une identification à deux étapes [Couwenbergh, 2003] ou bien définissent de nouveaux espaces constitués seulement de deux composantes, par exemple HS (hue-saturation) [Hildebrand et Reusch, 2000] où une fonction d'appartenance à trois dimensions est définie (*cf.* figure 3.1).

Ainsi, ces auteurs sont capables de modéliser directement des couleurs comme *dark blue*

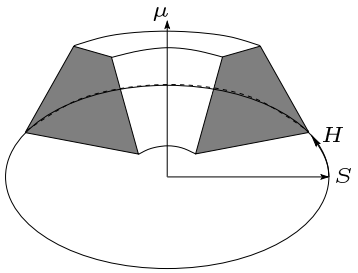

FIGURE 3.1 – Sous-ensemble flou à 3 dimensions défini sur l'espace HS.

(bleu foncé[3]) ou *bright red* (rouge vif). Cependant, ce modèle était insatisfaisant pour nous car nous souhaitions conserver la richesse de l'identification en deux étapes, c'est-à-dire l'identification rapide par H puis affinée par L et S.

Nous avons adopté la stratégie suivante : dans cette étude, on se limite aux neuf couleurs fondamentales définies par l'ensemble $\mathcal{T}$ suivant (dimension H) :

$\mathcal{T} = \{$*red, orange, yellow, green, cyan, blue, purple, magenta, pink*$\}$

Cet ensemble correspond aux sept couleurs de Newton [Roire, 2000] additionné des couleurs *pink* (rose) et *cyan* afin de correspondre aux couleurs intuitives de l'arc-en-ciel. A la manière de Sugano et de son Color Naming System [Sugano, 2001], nous établissons une correspondance entre des termes linguistiques et des valeurs numériques. L'auteur propose une méthode pour transformer une couleur exprimée de façon numérique (avec HLS) en un ensemble de mots (nom de la teinte + modificateur de ton). Il utilise des fonctions d'appartenance floues pour définir L et S pour un H donné. L'espace des teintes est composé de dix teintes des standards industriels japonais (*the Japanese Industrial Standards*). Quant à nous, au contraire, nous partons d'expressions linguistiques des couleurs (d'abord seulement la teinte, puis la teinte + un modificateur de ton) et nous obtenons un ensemble de valeurs numériques représentant la couleur.

Ainsi, HLS est un espace intéressant à utiliser pour établir ce pont "termes linguistiques" $\leftrightarrow$ "triplet numérique", mais c'est un espace non uniforme, non UCS. C'est-à-dire que les valeurs des teintes, notées h ne sont pas uniformément distribuées sur l'axe. Par exemple, l'œil ne perçoit pas les petites variations de teinte lorsque la couleur est le vert ($h = \pm 85$) ou le bleu ($h = \pm 170$) tandis qu'il les perçoit parfaitement quand il s'agit de l'orange ($h = 21$). Une façon de résoudre ce problème est de changer d'espace (d'utiliser CIELab ou CIELuv, par exemple) mais on perd du coup le bénéfice de cette identification à deux étapes, et les diverses transformations que l'on doit faire subir aux pixels (alors que la transformation RGB $\leftrightarrow$ HLS est quasi-directe) rendent le processus trop long et compliqué.

Pour travailler avec des échelles non uniformément distribuées, des auteurs comme Herrera et Martínez proposent d'utiliser des hiérarchies linguistiques floues avec plus ou moins d'étiquettes, en fonction de la granularité désirée [Herrera et Martínez, 2001]. Nous avons, pour notre part, utilisé des sous-ensembles flous, c'est-à-dire que, pour chaque couleur de

3. Nous utilisons désormais l'anglais car tous les travaux ont été réalisés avec des interfaces anglo-saxonnes.

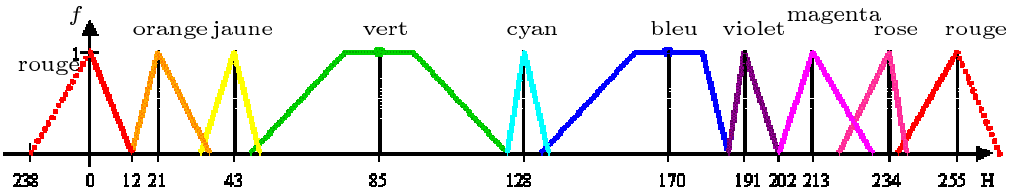

FIGURE 3.2 – La dimension H.

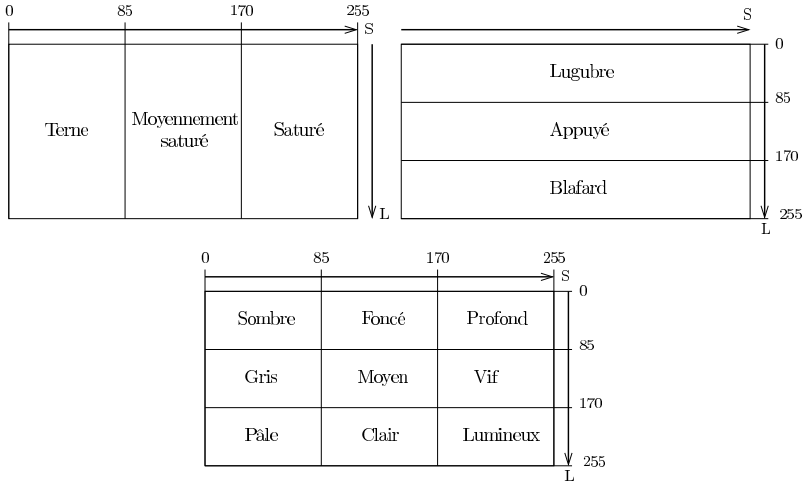

FIGURE 3.3 – Les qualificatifs colorimétriques fondamentaux.

$\mathcal{T}$ on construit une fonction d'appartenance qui varie de 0 à 1 (f_t avec $t \in \mathcal{T}$). Si f_t vaut 1 alors la couleur correspondante est dite "true color" (vraie couleur, *cf.* figure 3.2 où l'on a indiqué les teintes traduites en français).

On constate ainsi que, pour certaines couleurs (vert — "green" et bleu — "blue"), on obtient un sous-ensemble flou (trapézoïdal) dont le noyau n'est pas réduit à un point, mais à un intervalle.

En notant (a, b, α, β) un sous-ensemble flou trapézoïdal [Zadeh, 1965] avec $[a, b]$ le noyau et $[a - \alpha, b + \beta]$ le support, on peut définir la fonction d'appartenance pour chaque couleur t et on la note : $f_t(h), \forall t \in \mathcal{T}$.

La figure 3.3 montre la représentation de L et S. On divise chaque composante en trois intervalles pour ne retenir finalement que neuf qualificatifs fondamentaux, traduits là aussi en français.

On le voit, ces composantes sont liées entre elles, c'est pourquoi on utilise le produit cartésien en définissant les fonctions d'appartenance de L $\times$ S dans $[0, 1]$ (*cf.* figure 3.4).

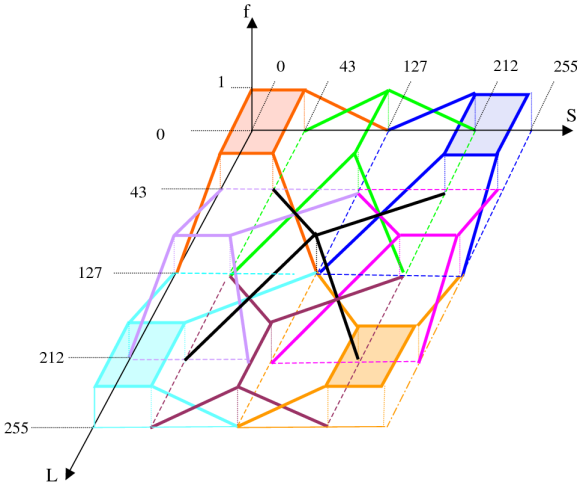

FIGURE 3.4 – Dimensions L et S.

Ainsi, l'ensemble des qualificatifs est noté $\mathcal{Q}$: $\mathcal{Q} = \{$ *somber, dark, deep, gray, medium, bright, pale, light, luminous* $\}$. A chaque élément de $\mathcal{Q}$ on associe une fonction d'appartenance $\tilde{f}_q$ que l'on note $\tilde{f}_q(l,s), \forall q \in \mathcal{Q}$. Chaque $\tilde{f}_q$ est représentée par l'ensemble de valeurs $(a,b,c,d,\alpha,\beta,\gamma,\delta)$ (*cf.* figure 3.5).

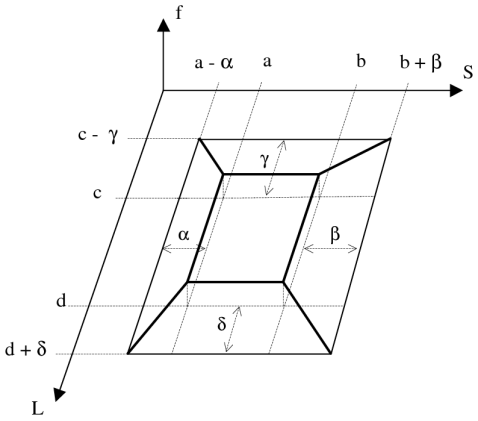

FIGURE 3.5 – Sous-ensemble flou trapézoïdal à trois dimensions.

Les couleurs achromatiques *noir*, *gris* et *blanc* sont, quant à elles, définies uniquement au travers des espaces L et S (f_{black}, f_{white} et f_{gray}). Si la luminance est très faible alors la couleur est *blanc*, tandis qu'une grande valeur de luminance indique le *noir*. Si la saturation est très faible alors la couleur est *gris* et nous définissons pour cette dernière couleur trois qualificatifs : *dark, medium* et *light* associés aux fonctions $\tilde{f}_{dark}$, $\tilde{f}_{medium}$ et $\tilde{f}_{light}$ (*cf.* figure 3.6).

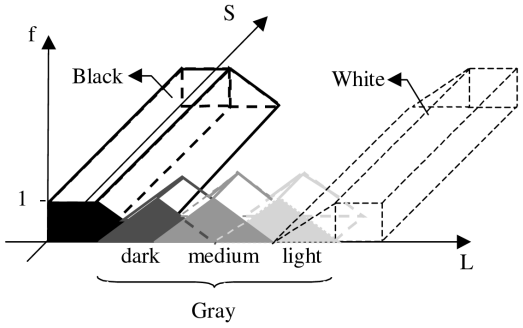

FIGURE 3.6 – Noir, gris et blanc.

Le profil d'une image est constitué des valeurs d'appartenance de l'image à chaque catégorie : les neuf couleurs fondamentales et les neuf qualificatifs. Pour chaque pixel, on peut déterminer les valeurs prises par les différentes fonctions d'appartenance des catégories. Pour chaque catégorie, la valeur obtenue correspond au ratio entre la somme — sur tous les pixels de l'image — des valeurs d'appartenance et le nombre de pixels, ce qui donne une quantité comprise entre 0 et 1. Cette quantité est le degré d'appartenance d'une image à une classe donnée. Le degré d'appartenance d'une image à une certaine classe est défini ainsi :

Soit I une image.

Soit $\mathcal{P}$ l'ensemble des pixels de I.

Chaque élément p de $\mathcal{P}$ est défini par ses coordonnées colorimétriques (h_p, l_p, s_p). p peut être un pixel ou un ensemble de pixels. On calcule les fonctions $f_t(h_p)$, $\tilde{f}_q(l_p, s_p)$ pour $t \in \mathcal{T}$ et $q \in \mathcal{Q}$.

Soient F_t et $\widetilde{F}_{t,q}$ les fonctions qui représentent les degrés d'appartenance de I aux classes t et (t,q). F_t est la valeur moyenne des f_t pour chaque pixel :

$$F_t(I) = \frac{\sum_{p \in \mathcal{P}} f_t(h_p)}{|\mathcal{P}|}, \forall t \in \mathcal{T}$$

Et $\widetilde{F}_{t,q}$ est la valeur moyenne des $\tilde{f}_q$ pour chaque pixel appartenant à une des neuf teintes. On note donc :

$$\widetilde{F}_{t,q}(I) = \frac{\sum_{p \in \mathcal{P}} \tilde{f}_q(l_p, s_p) \times g_t(h_p)}{|\mathcal{P}|}, \forall (t,q) \in \mathcal{T} \times \mathcal{Q}$$

$$\text{avec } g_t(h_p) = \begin{cases} 1 & si\ f_t(h_p) \neq 0 \\ 0 & sinon \end{cases}$$

Exemple 1. *Considérons 2 pixels p_0 et p_1 pour simplifier.*

$$p_0 : \begin{cases} h_{p_0} = 178 \\ l_{p_0} = 50 \\ s_{p_0} = 100 \end{cases}, p_1 : \begin{cases} h_{p_1} = 173 \\ l_{p_1} = 255 \\ s_{p_1} = 128 \end{cases}$$

– $f_{blue}(h_{p_0}) = f_{blue}(178) = 0.9$, $\tilde{f}_{somber}(l_{p_0}, s_{p_0}) = \tilde{f}_{somber}(50, 100) = 0.34$

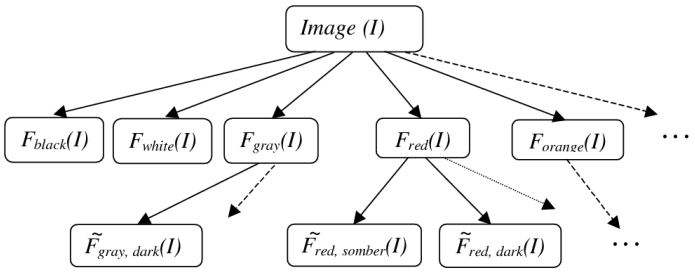

FIGURE 3.7 – Profil représentant une image.

– $f_{blue}(h_{p_1}) = 1,\ \tilde{f}_{somber}(l_{p_1}, s_{p_1}) = 0$

Ainsi, pour la classe "blue", les valeurs obtenues sont :

$F_{blue}(I) = 0.95$

Et pour la classe "somber" issue de "blue", on obtient : $\widetilde{F}_{blue,somber}(I) = 0.17$

Chaque image est donc définie par un profil de 96 éléments ($|\mathcal{T}| + |\mathcal{T} \times \mathcal{Q}| + |\{black, white, gray\}| + |\{gray\} \times \{dark, medium, light\}| = 9 + 81 + 3 + 3$).

Un profil est représenté comme suit : $[F_t(I),\ \widetilde{F}_{t,q}(I)]$

Une image peut être assignée à plusieurs classes. Il y a 96 classes, dont 12 principales : C_t avec $t \in \mathcal{T} \cup \{black, white, gray\}$, et 84 sous-classes qui correspondent à un raffinement de la recherche : $\widetilde{C}_{t,q}$ avec $(t,q) \in \mathcal{T} \times \mathcal{Q} \cup \{gray\} \times \{dark, medium, light\}$. Comme on peut le voir dans la figure 3.7 les classes peuvent être représentées par un arbre. Les classes C_t avec $t \in \mathcal{T}$ sont les classes parents et les classes $\widetilde{C}_{t,q}$ avec $(t,q) \in \mathcal{T} \times \mathcal{Q}$ sont les classes filles. Par exemple, la classe parent de $\widetilde{C}_{red,somber}$ est C_{red}.

On note :

– $F^*(I) = \max_{t \in \mathcal{T}}(F_t(I))$
– $\widetilde{F}^*_t(I) = \max_{q \in \mathcal{Q}}(\widetilde{F}_{t,q}(I))\ \forall t \in \mathcal{T}$, et pour $t = gray,\ q \in \{dark, medium, light\}$

Une image I est assignée :

– aux classes C_t si $F_t(I) \geq F^*(I) - \lambda,\ \forall t \in \mathcal{T} \cup \{black, white, gray\}$, avec λ un seuil de tolérance ;
– aux classes $\widetilde{C}_{t,q}$ si $F_t(I) \geq F^*(I) - \lambda$ et $\widetilde{F}_{t,q}(I) \geq \widetilde{F}^*_t(I) - \lambda,\ \forall (t,q) \in \mathcal{T} \times \mathcal{Q} \cup \{gray\} \times \{dark, medium, light\}$.

Par ailleurs, une image affectée à plusieurs classes ne peut être affectée à une sous-classe que si (et seulement si) elle est aussi affectée à sa classe parente. Par exemple, une image ne peut appartenir à la classe "red, bright" ($\widetilde{C}_{red,bright}$) si elle n'appartient pas déjà à la classe "red" (C_{red}).

Des développements logiciels ont été effectués en C++. On alimente une base de données d'images pour lesquelles on calcule en amont le profil colorimétrique. Par requête, il est

possible de choisir la ou les couleurs dominantes souhaitées, parmi une liste de couleurs et de qualificatifs associés. Quand un couple (couleur, qualificatif) est choisi, le logiciel propose en outre des couleurs exprimées en langage naturel qui correspondent au choix, par exemple "bleu nuit" (*midnight blue*). On établit ainsi un lien entre le couple et une expression linguistique. En guise d'exemple, le tableau 3.1 présente plusieurs termes linguistiques pour le bleu, classifiés selon les neuf qualificatifs qui lui sont attachés. Le dictionnaire qui a été choisi ici est X11Colors (`http://www.w3.org/TR/css3-color/#svg-color`).

somber	**dark**	**deep**
darkslategray	midnightblue	darkblue
gray	**medium**	**bright**
slategray	darkslateblue	mediumblue
cadetblue	slateblue	royalblue
	steelblue	dodgerblue
pale	**light**	**luminous**
silver	lavander	cornflowerblue
lightsteelblue	lightblue	aliceblue
		lightskyblue

TABLE 3.1 – Termes linguistiques pour la couleur bleue (*blue*).

La figure 3.8, quant à elle, montre les images dont la couleur dominante est "bleu lumineux" (*luminous blue*).

La comparaison de notre outil avec d'autres logiciels pour la recherche d'images n'a pas été chose facile. D'abord, un certain nombre de travaux considère des requêtes avec des termes *pondérés*, y compris pondérés à l'aide d'expressions linguistiques (voir [Bordogna et Pasi, 1993], par exemple), alors que nous faisons hypothèse que la requête contient uniquement des termes d'égale importance. Ensuite, beaucoup de travaux opèrent par "requête-image", c'est-à-dire que la requête ne s'exprime pas avec des termes linguistiques mais avec une image-témoin qui va servir de modèle (mêmes formes, mêmes couleurs) [Omhover et Detyniecki, 2004]. Enfin, les bases de données sont rarement les mêmes. Certains utilisent des bases de données de texture [Binaghi et al., 1994], [Frigui, 2001] et d'autres, des bases diverses comme la base Comstock [Vertan et Boujemaa, 2000] ou encore Common Color Dataset [Han et Ma, 2002]. Pour notre part, notre choix s'est porté sur la base ImageBank (`http://creative.gettyimages.com/imagebank/`) qui contient, pour chaque photo, une description par mots-clefs colorimétriques, bien utile pour nos tests.

Concernant la performance de notre outil, nous utilisons les mesures classiques proposées par Salton et McGill [Salton et McGill, 1983] :

1. Temps de réponse : représente le temps écoulé entre la requête utilisateur et la réponse du logiciel. Etant donné que le traitement (le calcul des profils) sur les images est effectué en amont, ce temps est négligeable.
2. Rappel (appelé *recall* en anglais) et précision : les images sont considérées comme

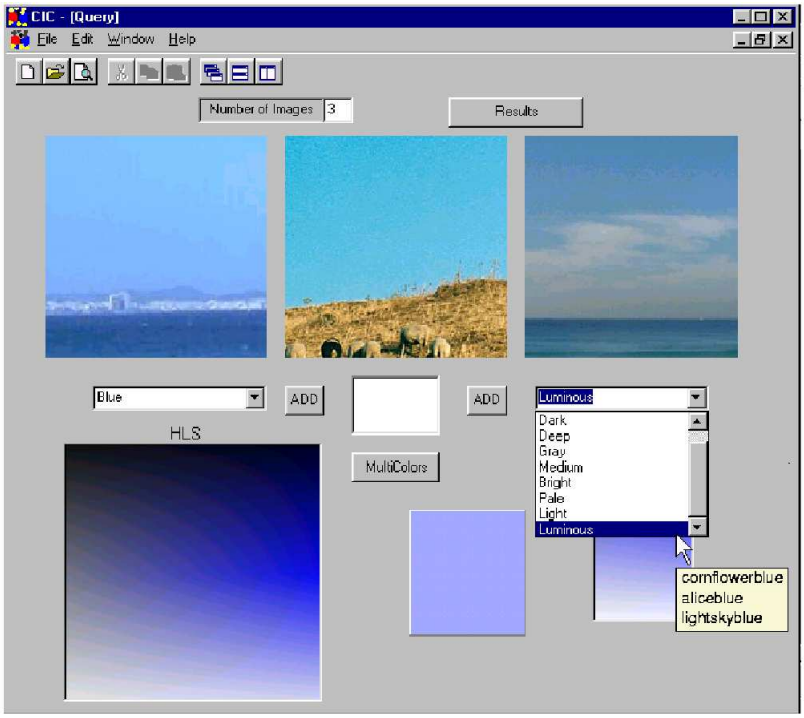

FIGURE 3.8 – Requête avec un couple (couleur, qualificatif).

étant pertinentes ou non, et extraites ou non (*cf.* table 3.2).

	Pertinent	Non Pertinent
Extrait	EP (correctement extrait)	ENP (incorrectement extrait)
Non Extrait	NEP (manqué)	NENP (correctement rejeté)

TABLE 3.2 – Performance dans la recherche d'information.

Le rappel, mesure de sensibilité, informe sur la capacité du logiciel à extraire toutes les images pertinentes :

$$\text{Rappel} = \frac{\text{correctement extrait}}{\text{tous les pertinents}} = \frac{\text{EP}}{\text{EP} + \text{NEP}}$$

et la précision mesure la capacité à extraire seulement les images pertinentes :

$$\text{Précision} = \frac{\text{correctement extrait}}{\text{tous les extraits}} = \frac{\text{EP}}{\text{EP} + \text{ENP}}$$

En utilisant la base ImageBank (IB), sur un total de 1000 images, nous avons obtenu les résultats suivants : 85 % pour le premier taux (rappel) et 89 % pour le second (précision) (*cf.* table 3.3).

	Pertinent	Non Pertinent
Extrait	897	103
Non extrait	164	/

TABLE 3.3 – Mesures de performance avec ImageBank.

La raison pour laquelle 103 images ne sont pas "bien classées" et 164 images pertinentes ne sont pas sélectionnées est à mettre au compte de la subjectivité de la classification des experts d'IB. En fait, notre logiciel permet d'associer davantage de couleurs dominantes que ne le fait IB qui associe, au plus, deux mots-clefs colorimétriques. De plus, le nommage des couleurs est très subjectif. Par exemple, ce qu'un expert peut appeler *dark pink* (rose foncé) peut être appelé *magenta* par un autre. Et cette perception dépend aussi beaucoup des couleurs voisines dans l'image. Par exemple, un *jaune* sur un fond noir ne sera pas perçu de la même manière que le même jaune sur un fond blanc.

En quelque sorte, ces résultats démontrent, voire évaluent, la subjectivité dans la perception des couleurs.

Le lecteur intéressé trouvera davantage de détails, notamment concernant le logiciel développé, dans les articles [Aït Younes et al., 2004a, Aït Younes et al., 2004b, Mamlouk et al., 2007].

La classification par *couleur(s) dominante(s)* ***perçue(s)*** s'est, quant à elle, davantage rapprochée des notions de perception. Le pari était alors : "ayant une vision floue (au sens propre, cette fois !) de l'image, quelle(s) couleur(s) m'apparaît (m'apparaissent) en premier ?" En pratique, nous avons traduit cela par la construction d'un profil colorimétrique associé à l'image, contenant les couleurs dominantes obtenues par détermination de zones dans l'image (algorithmes de type Deriche ou Canny). Un certain nombre de zones sont détectées (une zone étant associée à une seule couleur), selon une politique à seuils et un mécanisme de poids affecté à chacune afin de modifier le rapport *taille_zone / taille_image* selon les zones. Pour cela, nous avons utilisé la théorie des contrastes de quantité d'Itten qui établit des proportions entre couleurs (par exemple, le jaune est trois fois plus lumineux que le violet), nous permettant ainsi de pondérer les zones.

3.1.2 Classification d'images par couleur perçue

Dans le procédé tel que présenté plus haut, un certain nombre de problèmes apparaît à cause de la nature-même des images couleurs. Par exemple, certains pixels, même s'ils participent au réalisme de l'image dans le cas d'une photo, peuvent être considérés comme aberrants selon deux types de problèmes :

1. les pixels près des contours sont souvent aberrants, leurs couleurs sont aléatoires : ils peuvent être considérés comme du bruit (*cf.* figure 3.9) [Boughorbel et al., 2002] ;

FIGURE 3.9 – Détection de contours et pixels bruités autour d'un contour.

2. dans les zones uniformes (en termes de perception de la couleur), on peut trouver quelques pixels aberrants dus à la numérisation de l'image, à la compression, etc. (*cf.* figure 3.10).

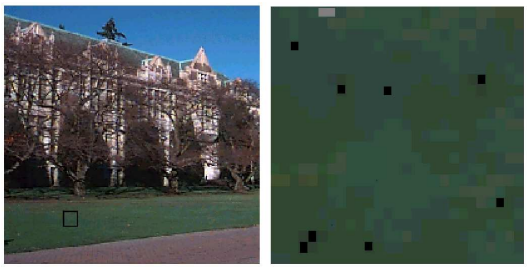

FIGURE 3.10 – Zone uniforme.

Pour résoudre la première difficulté, nous utilisons une méthode de détection de contours pour être en mesure d'éliminer les "pixels-contours" avant les calculs des profils. La méthode choisie est celle du gradient qui cherche les *extrema* de la dérivée première de l'image (algorithme de Canny [Canny, 1986]).

Quant à la deuxième difficulté, nous déterminons toutes les zones uniformes grâce à la détection de contours et nous affectons une couleur dominante à chaque zone. Trois options ont été retenues :

- dans chaque zone, on considère une sous-zone représentative (à la manière d'un centre de gravité). La couleur dominante de la zone est obtenue par la couleur moyenne des pixels appartenant à la sous-zone ;
- dans chaque zone, on choisit aléatoirement deux ou trois sous-zones. La couleur dominante de la zone est obtenue par la moyenne des couleurs de chaque sous-zone. Notons que la notion de "moyenne" de couleur n'a un sens que si les couleurs sont proches, chromatiquement. Ce qui est nécessairement le cas ici, puisqu'on a d'abord procédé à un découpage en zones — contenant des couleurs homogènes, donc) ;
- dans chaque zone, on choisit aléatoirement un sous-ensemble de pixels. Ce sous-ensemble doit représenter un certain pourcentage de la zone (10%, par exemple). La couleur dominante de la zone est donnée par la couleur moyenne du sous-ensemble.

Une autre amélioration (alternative) a aussi été apportée concernant le calcul de la fonction $F_t(I)$. On a remplacé la moyenne équi-pondérée (chaque pixel ou ensemble de

pixels avait la même importance) par une affectation de poids différents, selon leur position (les pixels aberrants ou isolés se voient affectés de poids plus faibles — voire nuls — que les autres) et selon les sept contrastes d'Itten parmi lesquels on trouve notamment le contraste clair-obscur, le contraste chaud-froid, le contraste des complémentaires et le contraste de quantité avec les trois proportions suivantes [Itten, 1961] :

- le jaune est trois fois plus lumineux que le violet ;
- l'orange est deux fois plus lumineux que le bleu ;
- le rouge est autant lumineux que le vert.

Ainsi, les pixels (ou ensembles de pixels) ont une contribution différente les uns par rapport aux autres, un peu comme dans les systèmes de vote avec des grands électeurs.

Par ailleurs, on sait que la couleur est un concept très subjectif [Boust et al., 2003], c'est pourquoi nos efforts se sont portés sur un apprentissage de la perception de l'utilisateur. En fonction de la sensibilité de l'œil, nous modifions la définition des associations teintes et qualificatifs / valeurs H, L et S en changeant le nombre et la forme des sous-ensembles flous à deux et trois dimensions qui codent les t et q. Pour les teintes, changer la cardinalité de $\mathcal{T}$ peut être intéressant si l'on souhaite ajouter une couleur (*moutarde* ajouté entre *vert* et *jaune*) ou en retirer une (*rose* + *magenta* devient *rose*). Quant à la modification des formes des sous-ensembles flous, nous avons repris nos travaux de thèse de doctorat sur la taxonomie des modificateurs flous : des termes linguistiques modificateurs sont proposés à l'utilisateur et des modifications sont appliquées sur les sous-ensembles flous. Alternativement, l'utilisateur peut, s'il le préfère, modifier lui-même les sous-ensembles flous, sous certaines conditions.

Ceci nous a amenée à poursuivre le travail sur les profils d'image en modifiant les profils colorimétriques d'une part, et en proposant des profils utilisateur d'autre part.

Une image est caractérisée comme étant bleue si une de ses couleurs dominantes est le bleu. En d'autres termes, une image est bleue si son degré d'appartenance à la couleur *blue* est suffisamment grand. Mais la même image ne serait peut-être pas considérée comme étant bleue par une autre personne, ou bien, elle ne serait pas cataloguée comme étant bleue si la définition interne de *blue* a été modifiée. Supposons maintenant qu'à l'origine, une image I a un degré d'appartenance $F_t(I)$ à la couleur t. $F_t(I)$ varie dès que les valeurs définissant t varient. Pour obtenir le nouveau degré $F_{t_{new}}(I)$, la seule façon de donner un résultat exact est de recalculer le profil de I, pixel par pixel. Notre approche a été d'éviter ce recalcul en *simulant les variations des positionnements initiaux* des valeurs des couleurs au moyen de la compatibilité (*via* une fonction notée γ) et de la comparabilité (*via* une fonction notée Φ), telles que définies dans [Truck, 2002].

Deux sous-ensembles flous sont dits *compatibles* (selon γ) s'ils sont suffisamment proches l'un de l'autre, c'est-à-dire si le support de l'un a une intersection non nulle avec le support de l'autre. Si c'est le cas, on cherche à quel point ils sont *comparables* (selon Φ).

Soient A et B deux sous-ensembles flous, notés respectivement (a_1, b_1, c_1, d_1) avec $\mathit{Support}(A) = [a_1, d_1]$ et $\mathit{Noyau}(A) = [b_1, c_1]$, et (a_2, b_2, c_2, d_2) avec $\mathit{Support}(B) = [a_2, d_2]$ et $\mathit{Noyau}(B) = [b_2, c_2]$ et tels que A est *avant* B, c'est-à-dire que $a_1 < a_2$.

Le degré de compatibilité de B avec A est défini par :
$\gamma(A, B) = \max(f_A(a_2), f_A(b_2), f_B(c_1), f_B(d_1))$.

γ est une M-mesure de similitude au sens de Bouchon–Meunier *et al.*, puisque $\gamma(A,B)$ est fonction de $A \cap B$, $B-A$ et $A-B$, et que $\gamma(A,B)$ est croissante sur $A \cap B$ et décroissante sur $B-A$ et $A-B$ [Bouchon-Meunier et al., 1996].

L'ensemble de tous les sous-ensembles flous qui sont compatibles (*i.e.* $\gamma > 0$) avec A est noté $\Gamma(A)$. Soit $t_{i_{new}}$ une nouvelle couleur et t_i une couleur initiale, $t_i \in \Gamma(t_{i_{new}})$ si et seulement si $\gamma(t_{i_{new}}, t_i) > 0$. Sauf problème dans la construction, une couleur est toujours compatible avec ses couleurs adjacentes.

Deux sous-ensembles flous compatibles sont dits *comparables* si le support — et respectivement le noyau — de l'un est suffisamment proche du support — et respectivement du noyau — de l'autre.

Le degré de comparabilité entre A et B est défini par : $\Phi(A,B) = \frac{1}{8}\sum_{i=1}^{8} \mu_i$

où $\mu_1 = \mu(a_1, A, B), \mu_2 = \mu(b_1, A, B), \ldots, \mu_4 = \mu(d_1, A, B)$,
$\mu_5 = \mu(a_2, B, A), \mu_6 = \mu(b_2, B, A), \ldots, \mu_8 = \mu(d_2, B, A)$

et avec $\forall x \in \{a_1, b_1, \ldots, d_2\}$, $\mu(x, A, B) = \begin{cases} f_B(x) & si\ x \in Noyau(A) \\ 1 - f_B(x) & sinon \end{cases}$

La raison pour laquelle on complémente la valeur de $f_B(x)$ lorsque x n'appartient pas à *Noyau*(A) est que Φ doit appartenir à $[0,1]$, où 0 signifie que "A et B ne sont pas comparables" et 1 que "A et B sont totalement comparables". Quand x n'appartient pas à *Noyau*(A), il appartient nécessairement à *Support*(A), par définition, et donc l'appartenance recherchée est 0 et non pas 1. Ainsi, si $B = A$, chaque $\mu(x, A, B) = \mu(x, A, A) = 1$ et donc $\Phi(A,B) = \Phi(A,A) = 1$.

Φ est une M-mesure de ressemblance, toujours au sens de Bouchon–Meunier *et al.*, c'est-à-dire une M-mesure de similitude satisfaisant les propriétés de réflexivité et de symétrie.

Le fait que Φ soit symétrique est une condition indispensable pour obtenir des résultats visuellement cohérents. En effet, la place des sous-ensembles flous sur l'ensemble de définition ne doit pas avoir d'importance. Par exemple, si le sous-ensemble flou $rouge_1$ est comparable à $rouge_2$ avec un certain degré, alors $rouge_2$ doit aussi être comparable à $rouge_1$ avec le même degré.

Pour chaque t, on sélectionne les sous-ensembles flous (les autres couleurs) dont le degré de compatibilité avec t est suffisamment grand (politique à seuils). Dans ce cas, on calcule alors leur degré de comparabilité.

Ainsi, le nouveau profil de l'image est recalculé et les couleurs adjacentes à t_i sont aussi modifiées. Si, par exemple, on a un changement pour la couleur bleue, de sorte que le sous-ensemble flou du nouveau bleu se situe entre l'ancien bleu et le vert (non modifié), alors l'appartenance de l'image à la nouvelle couleur bleue est : $F_{B_{new}}(I) = (\Phi(B_{new}, B_{init}) \times F_{B_{init}}(I) + \Phi(B_{new}, G_{init}) \times F_{G_{init}}(I))/2$ où B_{init} est le sous-ensemble flou initial de *blue*, G_{init} le sous-ensemble flou initial de *green* et B_{new} le nouveau sous-ensemble flou attaché à *blue*.

De façon générale, $F_{t_{i_{new}}}(I) = \dfrac{\sum_{t_i \in \Gamma(t_{i_{new}})} \Phi(t_{i_{new}}, t_i) \times F_{t_i}(I)}{\#\Gamma(t_{i_{new}}) + 1}$

Il en va de même pour les qualificatifs, c'est-à-dire que :

$$\forall (t_i, q_j) \in \mathcal{T} \times \mathcal{Q}, \widetilde{F}_{t_{inew},q_j}(I) = \frac{\sum_{t_i \in \Gamma(t_{inew})} \Phi(t_{inew}, t_i) \times \widetilde{F}_{t_i,q_j}(I)}{\#\Gamma(t_{inew}) + 1}$$

On obtient donc un profil *modifié* de l'image d'une part, et un profil utilisateur, d'autre part, construit pendant l'apprentissage (*cf.* figure 3.11).

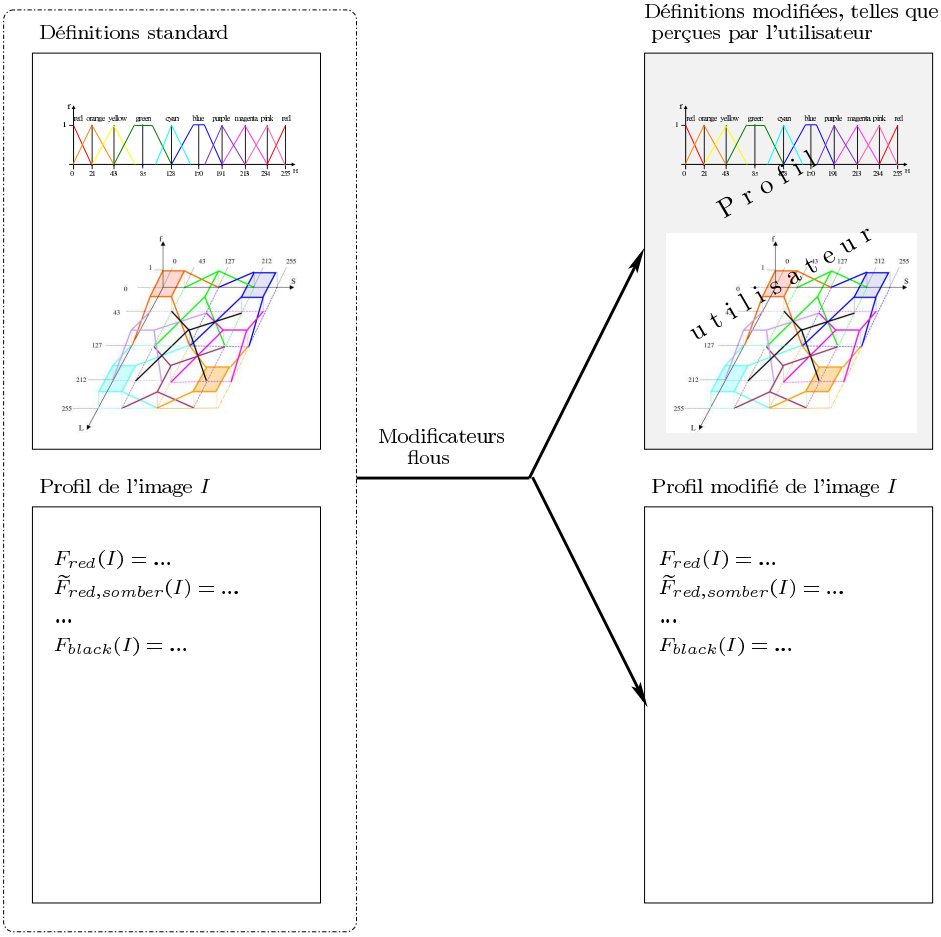

FIGURE 3.11 – Les profils obtenus.

Ainsi, cette approche permet de mesurer le biais des utilisateurs dans l'expression de leurs requêtes et de le corriger lors de l'interrogation de la base. A la limite, cette modélisation *ad hoc* des couleurs pourrait gérer des dyschromatopsies, telle le daltonisme.

Pour plus de détails, le lecteur intéressé pourra se reporter aux trois articles de conférence et revues suivants : [Aït Younes et al., 2005, Aït Younes et al., 2007, El-Zakhem et al., 2007].

Ces recherches ont permis de proposer une classification originale, entièrement dirigée par la perception oculaire et non pas par la sémantique, comme habituellement. L'intérêt était notamment la recherche indirecte d'harmonies dans une image, mais aussi entre plusieurs images, avec comparaison des profils colorimétriques des images.

Ces travaux ont été réalisés en collaboration avec nos collègues Amine Aït Younes et Herman Akdag. Amine Aït Younes, d'abord post-doctorant, est devenu depuis maître de conférences. Nous avons encadré deux étudiants : Mohammed Chokri Mamlouk et Imad El-Zakhem.

3.2 Le calcul à l'aide de mots dans la perception pour les arts de la scène

D'un autre côté, nous avons entamé des travaux dont le principal domaine d'application était les arts de la scène (opéra, théâtre, etc.). Nous avons commencé par définir un *assistant virtuel de* performer *(acteur, chanteur, ...)*, où la machine analysait la performance (analyse de geste, de mouvement, de vitesse de déplacement, etc.) et cherchait à la qualifier, en fonction des consignes du metteur en scène. Autrement dit, la machine se faisait miroir (non déformant !) de l'acteur, en fournissant une analyse se voulant qualitative, mais systématique. Là encore, on voit clairement que ce domaine a toute sa place dans le *perceptual computing* : ce qui est perçu par l'acteur, par le metteur en scène et ce que "perçoit" la machine.

En outre, nous avons, dans ce cadre, proposé un algorithme de partitionnement flou (en classes) par apprentissage, dépendant de la distribution des données et inspiré par les partitionnements non uniformes de Martínez *et al.* Les tests grandeur nature (plusieurs spectacles ont servi pour les tests) ont montré que l'algorithme était robuste et que l'analyse de la performance était pertinente (tant pour l'acteur que pour le metteur en scène).

3.2.1 Un assistant virtuel de *performer*

A l'opéra, au théâtre ou ailleurs, la direction d'acteur est une tâche difficile, notamment dans les productions comportant une part d'improvisation. Mais improvisation ne signifie pas absence de règles pour autant. C'est pourquoi il est utile de concevoir des outils qui peuvent aider le directeur artistique dans sa tâche de supervision de la performance d'un acteur. L'outil que nous proposons est un genre d'assistant qui donne une représentation d'un ensemble de données complexes pour l'exploration et l'observation d'un spectacle. A la manière des sportifs de haut niveau qui ont des outils pour analyser, corriger et améliorer leurs performances, notre outil constitue aussi un moyen de mieux comprendre l'exécution de la performance de l'acteur.

Les assistants de *performer* ne sont pas véritablement nouveaux. Depuis longtemps, il existe divers systèmes de notation (Feuillet, Laban, Benesh, ...) qui fournissent au travers un score les causes du phénomène (la pièce, le morceau, l'air, ... à jouer) pour tenter d'accéder aux intentions de l'auteur (de la pièce, de l'opéra, de la chorégraphie, etc.). Souvent, les scores encodent des descriptions implicites (il faut jouer un do, mais sans explication sur le comment) ou bien explicites (croisement des mains en jouant du piano), et de ces indications structurées, le danseur ou le musicien tente d'inférer les intentions de l'auteur [Kahol et al., 2004]. Bien sûr, avec l'apparition des capteurs de toutes sortes (dont les caméras), on a imaginé des analyseurs de mouvements, de gestuelle, etc. Il y a quelques

années, Camurri et son équipe ont développé *EyesWeb* la première plateforme robuste pour l'analyse de geste [Camurri et al., 1999]. Elle fonctionne selon une simple capture vidéo, plus conviviale qu'un système avec des capteurs posés sur le corps. Elle est fondée sur un langage graphique qui implémente des descripteurs de geste ; par exemple, la quantité de mouvement, la stabilité, etc. Des chercheurs comme Friberg ont utilisé cette plateforme : ils ont conçu, dans le cadre d'un jeu, un algorithme temps réel pour analyser l'expression émotionnelle dans la performance sonore et dans les mouvements du corps [Friberg, 2005]. La démarche est cependant très différente de la nôtre car ils souhaitent un retour immédiat, c'est-à-dire que le joueur bouge, parle et chante tant que le logiciel n'a pas réagi conformément à ses souhaits. Notre but est, au contraire, que l'assistant s'adapte au *performer* et non l'inverse.

Nous avons choisi de travailler sur l'opéra virtuel *Alma Sola* écrit par Bonardi et Zeppenfeld dans lequel un *performer* joue (chante et danse) différents blocs de différents *univers* (comme le Prologue, l'Amour, le Plaisir, etc.). *Alma Sola* est une forme ouverte d'opéra [Bonardi et Rousseaux, 2004]. Le *performer* incarne un personnage de Faust féminin et se promène dans les différents *univers* découpés en blocs. Elle interprète une liste d'airs qu'elle choisit en direct, de façon improvisée, pendant le spectacle. Par exemple, une performance peut être : Amour-3, Opulence-5, puis Plaisir-3, etc. L'ordinateur offre des prolongements au *performer* en suggérant le prochain bloc à exécuter (*cf.* figure 3.12).

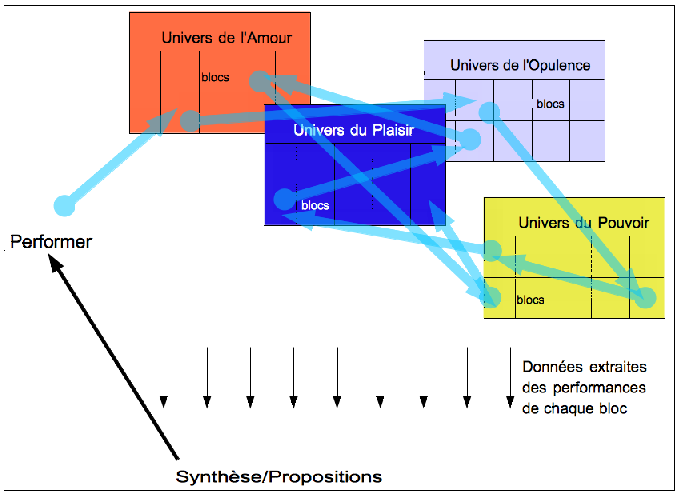

FIGURE 3.12 Séquences dans *Alma Sola*.

Nous avons enregistré la performance dans des fichiers vidéo puis deux scènes très différentes ont été extraites :

- le Prologue (qui est improvisé, presque non écrit)
- et l'univers de l'Amour (qui est complètement écrit, sans improvisation).

Le *performer* est filmé par une caméra statique, en plan large (*cf.* figure 3.13, à gauche) et sa voix est enregistrée dans un fichier à part pour pouvoir gérer les deux sources de données séparément.

Descripteur	**Intervalle**	**Description intuitive**
Durée du mouvement	$[0, +\infty[$ en ms	
Durée de la pause	$[0, +\infty[$ en ms	
Surface de contraction	[0,nombre de pixels dans la vidéo[	Quantité de pixels representant le *performer* dans l'image
Index de contraction	$[0, 1]$	Grand : la posture du *performer* est ouverte Moyen : la posture du *performer* est normale Petit : la posture du *performer* est fermée
Coordonnées de la matrice de contraction	$x, y \in$ [0,nombre de pixels dans la vidéo[	Emplacement du *performer* dans l'image
Stabilité	$[0, 2.34]$	Basse : le *performer* est près du sol, les jambes écartées Haute : le *performer* est debout, les jambes tendues
Quantité de mouvement	$[0, 1]$ 0 : pas de mouvement 1 : toutes les parties de la silhouette ont bougé	Vitesse de mouvement et masse déplacée
Centre de gravité	$x, y \in$ [0,nombre de pixels dans la vidéo[	

TABLE 3.4 – Définition des descripteurs.

FIGURE 3.13 – A gauche, un extrait du fichier vidéo ; à droite, la boîte englobante de la silhouette.

EyesWeb a été utilisé (par l'intermédiaire d'un *patch*) pour analyser, comprendre et exploiter les gestes expressifs non verbaux. En effet, plusieurs paramètres peuvent être extraits du fichier vidéo : la quantité de mouvement, la stabilité, la durée du mouvement, la durée de la pause dans la scène courante, l'accélération, etc. La construction du *patch* a nécessité des heures de tests avec des vidéos exemples, et nous avons finalement choisi d'extraire une dizaine de paramètres qui nous ont semblé pertinents. Le tableau 3.4 en montre quelques uns.

On peut ensuite agréger temporellement les valeurs de ces descripteurs afin d'obtenir des résumés pour chacun d'entre eux, pour la scène considérée. Ils vont servir à alimenter une base de connaissances. Il serait trop hasardeux (ou prétentieux) de croire que l'on peut utiliser les données récoltées pour générer automatiquement les règles. L'humain a une part

Emotions	**Ag$_1$**	**Ag$_2$**	**Ag$_3$**	**Ag$_4$**	**Ag$_5$**	**Ag$_6$**	**Ag$_7$**	**Ag$_8$**	**Ag$_9$**
CroitEnLAmour	*L*	*A*	*L*	*L*	*A*	*L*	*L*	*A*	*L*
Endormi	*H*	*L*	*H*	*H*	*L*	*H*	*H*	*L*	*H*
Fâché	*A*	*H*	*L*	*A*	*A*	*L*	*A*	*H*	*L*
Heureux	*L*	*H*	*L*	*L*	*H*	*L*	*L*	*H*	*L*
Neutre	*H*	*L*	*L*	*H*	*L*	*L*	*H*	*L*	*L*

TABLE 3.5 – Exemple de classification pour l'univers Amour.

très importante ici et il nous semble inconcevable que ce ne soit pas un expert (*performer*, directeur artistique, metteur en scène...) qui écrive lui-même les règles. En revanche, les données récoltées vont servir au partitionnement en classes pour les prémisses des règles. En effet, les règles sont du type "Si le descripteur agrégé 1 est Moyen et ... et le descripteur agrégé n est Faible alors l'émotion transmise par le *performer* est Tristesse". Il est donc nécessaire de définir correctement le nombre de classes et le partitionnement pour chaque descripteur. Mais ce travail n'est pas à faire une fois pour toutes, tant les données peuvent changer en fonction des acteurs. C'est pourquoi nous avons envisagé un algorithme pour apprendre ce partitionnement.

La partition floue proposée n'est pas nécessairement uniforme sur l'axe et dépend de la distribution des données captées. On considère d'abord par défaut cinq classes de très petit à très grand : Very Low, Low, Average, High et Very High (notées *VL*, *L*, *A*, *H*, *VH*). Dans un premier temps, pour le calcul, ces classes sont des intervalles. Pour chaque agrégateur et pour tous les enregistrements de chaque scène, la construction de la classe Low est calculée dynamiquement : la borne inférieure de l'intervalle est la valeur minimum de l'ensemble des données initiales, notée x_0. Ainsi, en analysant un nouvel enregistrement de la même scène, s'il y a des valeurs plus petites que x_0, elles seront catégorisées comme des valeurs Very Low. De la même façon, pour chaque agrégateur et pour tous les enregistrements de chaque scène, la classe High est construite dynamiquement : la borne supérieure de l'intervalle est la valeur maximum de l'ensemble des données initiales, notée x_2. Puis, pour chaque agrégateur et pour tous les enregistrements de chaque scène, la valeur moyenne de l'ensemble des données initiales est notée x_1. Les deux intervalles $[x_0, x_1]$ et $[x_1, x_2]$ sont divisés chacun en trois sous-intervalles égaux. Finalement, l'intervalle pour la classe L est la concaténation des deux premiers sous-intervalles, l'intervalle pour la classe A est la concaténation des deux sous-intervalles suivants et l'intervalle pour la classe H est la concaténation des deux derniers sous-intervalles. La figure 3.14 montre le découpage des intervalles.

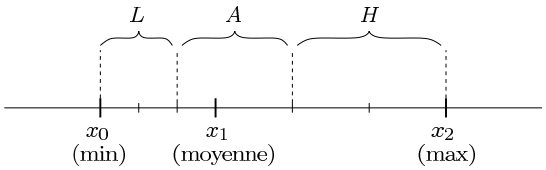

FIGURE 3.14 – Division des intervalles.

Ainsi, il est possible de classifier les valeurs agrégées pour les cinq émotions, comme montré dans le tableau 3.5 où Ag$_i$ est le i^e agrégateur, dans le cas de l'univers Amour.

La dernière étape consiste à construire les sous-ensembles flous représentant les classes pour les règles. La figure 3.15 montre le procédé. La plus petite classe-intervalle, L dans notre exemple, est définie comme un nombre flou de type L-R (un trapèze, en pratique) et détermine l'unité (à la manière d'un vecteur unitaire) pour les autres intervalles. Si l'on note (a_F, b_F, c_F, d_F) le sous-ensemble flou F (avec Support$(F) = [a_F, d_F]$ et Noyau$(F) = [b_F, c_F]$), alors la construction des cinq classes dans l'exemple de la figure 3.15 se fait ainsi :

$$VL \ : \ \begin{cases} a_{VL} = b_{VL} = \inf_{x \in \mathrm{Ag}_i} x \\ c_{VL} = 2x_0 - x_{0,1} \\ d_{VL} = x_{0,1} \end{cases}$$

$$L \ : \ \begin{cases} a_L = 2x_0 - x_{0,1} = c_{VL} \\ b_L = c_L = x_{0,1} = d_{VL} \\ d_L = x_1 \end{cases}$$

$$A \ : \ \begin{cases} a_A = x_{0,1} = b_L \\ b_A = x_1 = d_L \\ c_A = x_{0,2} + x_{1,1} - x_1 \\ d_A = 2x_{1,1} - c_A \end{cases}$$

$$H \ : \ \begin{cases} a_H = c_A \\ b_H = d_A \\ c_H = 2x_{1,2} - b_H \\ d_H = 2x_2 - c_H \end{cases}$$

$$VH \ : \ \begin{cases} a_{VH} = c_H \\ b_{VH} = d_H \\ c_{VH} = d_{VH} = \sup_{x \in \mathrm{Ag}_i} x \end{cases}$$

De cette façon, la plus petite classe-intervalle, c'est-à-dire L ou H, est un nombre flou de type L-R (un "triangle", en pratique). De façon générale, au moins l'une des deux classes L ou H est un nombre flou de type L-R. Néanmoins, A et H (ou respectivement A et L, en fonction de la classe qui contient le plus petit sous-intervalle) peuvent être soit des nombres flous, soit des intervalles flous de type L-R. Elles sont en fait des nombres flous sous une certaine condition dépendant d'un paramètre ε :

$\forall X \in \{L, A, H\}$, on note X^p la classe prédécesseur dans l'univers. Après avoir calculé a_X, b_X, c_X, d_X comme dans l'exemple de la figure 3.15 pour la classe A ou H, si $c_X - b_X < \varepsilon$ alors c_X et b_X doivent être recalculés, i.e. $b'_X = c'_X = 1/2\ (b_X + c_X)$

La valeur prise par ε détermine le degré d'appartenance maximal v avec lequel les classes se chevauchent. En particulier, si $\varepsilon < c_X - b_X$ alors $v = 0.5$ et X est un intervalle flou de type L-R, sinon (i.e. $\varepsilon \geq c_X - b_X$) $v = \dfrac{1}{a_X - b'_X} \left(\dfrac{a_X c_{X^p} - b'_X d_{X^p}}{-a_X + b'_X - c_{X^p} + d_{X^p}} + a_X \right)$ et X est un nombre flou de type L-R.

Finalement, il devient facile d'établir les règles floues selon le tableau 3.5, en écrivant

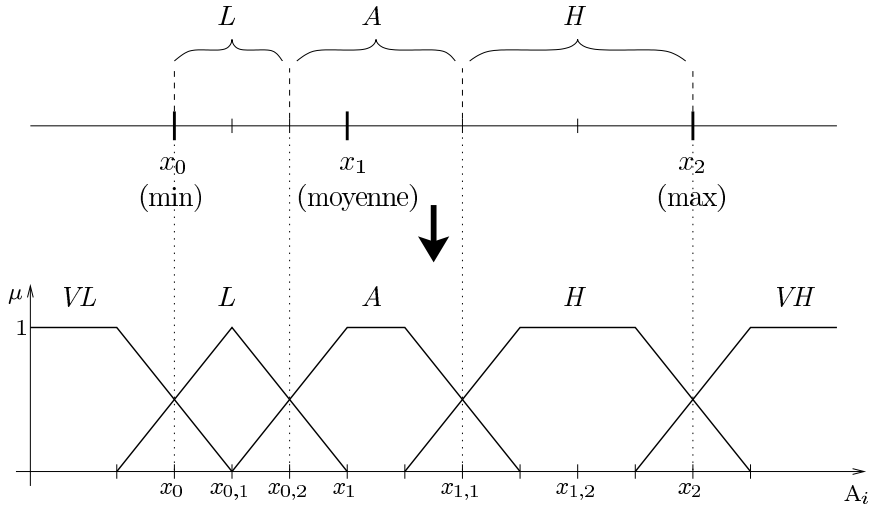

FIGURE 3.15 – Construction des classes.

une règle par ligne. Voici une règle en exemple pour l'univers Amour :

```
Règle 1 :

Si Ag1 est L & Ag2 est A & Ag3 est L &
   Ag4 est L & Ag5 est A & Ag6 est L &
   Ag7 est A & Ag8 est L & Ag9 est L
Alors l'émotion transmise est CroitEnLAmour
```

Ces recherches ont donné lieu à de nombreux développements ainsi qu'à une multitude de tests. Elles ont bénéficié d'un financement dans le cadre d'une action concertée incitative (ACI) à la Maison des Sciences de l'Homme – Paris-Nord pendant deux ans (2004 à 2006). Le lecteur intéressé pourra trouver davantage de détails dans [Bonardi et al., 2006a, Bonardi et al., 2006b, Bonardi et Truck, 2006].

Ces travaux ont été réalisés en collaboration avec notre collègue Alain Bonardi. Nous avons encadré trois étudiants de master ou d'école d'ingénieur : Nicolas Lehallier, Murat Göksedef et Kenji Yamashita.

Toujours dans ce même cadre, nous avons entamé une collaboration avec l'Ircam, et donc, nous sommes tournée davantage vers les questions de son. Nous avons imaginé une extension au logiciel Max/MSP (pour la synthèse, l'analyse, l'enregistrement et le traitement sonore) pour assouplir la gestion des objets en permettant l'utilisation d'objets flous. Nous avons ainsi proposé une librairie (la *FuzzyLib* disponible en ligne) qui implémente la plupart des concepts flous dans Max/MSP. Une telle librairie n'existait pas, du fait de la conception sous Max/MSP très particulière de la notion d'objets, de fonctions, de variables, etc. Elle a notamment permis de coupler des travaux existants sur la reconnaissance et le suivi de geste utilisant les modèles de Markov cachés avec les techniques floues ; en particulier, il est possible désormais de reconnaître graduellement et de façon floue un geste au fur et à

mesure de son exécution.

3.2.2 Une librairie floue pour Max/MSP

Dans les années 1980, le premier environnement pour le traitement sonore temps réel est apparu : *Patcher*. Son but était de créer des interactions musicales intéressantes et *en direct* entre les musiciens et les instruments électroniques de transformation du son. Plus tard, il a donné naissance au logiciel de référence *Max* (qui est devenu Max/MSP/Jitter) et, plus tard, à *PureData*. Depuis, beaucoup de personnes l'ont utilisé et une réelle communauté est née avec son lot de contributeurs à Max/MSP. Ces environnements ont grandi rapidement et ont incorporé des traitements de haut niveau, notamment des modules incluant de l'intelligence artificielle (réseaux de neurones, modèles de Markov cachés, réseaux Bayésiens, etc.) et ont inclus de nombreux protocoles pour gérer les données provenant des *performers via* différents capteurs. Max/MSP est ainsi devenu un standard mondial dans l'analyse sonore en temps réel et est très utilisé dans tous les arts de la scène (musique, danse, opéra, théâtre...).

C'est dans ce cadre que nous est venue l'idée de fournir un outil pour manipuler des données linguistiques dans Max/MSP afin d'être en mesure de traiter, analyser, voire synthétiser les performances humaines. Max/MSP a été conçu pour pouvoir donner une représentation minimale du signal : détection très simple du ton, du timbre, du début, etc. d'un signal. En revanche, tous les concepts comme la mélodie, le pizzicato, le legato, le staccato, etc. lui sont difficiles à détecter.

Mais implémenter des concepts de haut niveau dans un environnement plutôt bas niveau n'est pas trivial. En effet, envisagés comme des langages graphiques, ces environnements nécessitent des *patchs* et donc une certaine forme de programmation, un peu comme l'environnement LabView avec son langage G. Ces *patchs* typiquement conçus pour les électroniciens simulent en fait du matériel, des périphériques comme un métronome, une boîte à rythmes, etc. Ils proposent une sorte de métaphore de laboratoire physique, où les objets représentent des processus qui sont liés virtuellement entre eux. Ainsi, Max impose une "programmation orientée *patch*". Tout doit être vu comme un problème de traitement du signal, c'est-à-dire avec des boîtes reliées entre elles par des connecteurs, chaque boîte représentant un traitement à appliquer au signal — qui est une donnée à une (son) ou deux dimensions (image). Chaque connecteur porte le signal en tant qu'entrée ou bien sortie.

On imagine bien ici qu'une fois encore, le paradigme du *Computing with Words* (CW) va pouvoir apporter une dimension très intéressante à ces environnements, grâce à la possibilité d'exprimer des perceptions, impressions, émotions, etc. [Zadeh, 1971, Zadeh, 2002, Zadeh, 2005]. Finalement, on peut dire que Max aspire à calquer des concepts sémantiques sur le traitement du signal, tandis que le CW offre des concepts sémantiques, mais sans aucun lien vers le traitement du signal.

Pour permettre ce lien, nous proposons donc une bibliothèque pour Max qui implémente des *patchs* dédiés au CW. Un des intérêts du CW dans le champ de la création numérique provient de l'observation que les dispositifs pour créer ou manipuler du contenu numérique en relation avec les performances artistiques manquent de ductilité, contrairement aux instruments réels. De plus, établir l'adéquation entre la perception humaine et les paramètres de contrôle n'est pas évident. Les valeurs numériques entières souvent

utilisées n'offrent pas la granularité nécessaire pour satisfaire l'utilisateur. Pour amener davantage de souplesse dans les traitements, on a donc implémenté une partie des outils offerts par la logique floue [Kaufmann, A., 1975], mais aussi quelques algorithmes plus récents comme le partitionnement non uniformément distribué avec les *fuzzy 2-tuples* de Martínez *et al.* [Herrera et Martínez, 2000], ou encore la méthode de partitionnement par apprentissage que nous avons développée (*cf.* plus haut, et notamment la figure 3.15 qui résume le procédé).

Ces outils pourront sans aucun doute être adaptés pour des performances réelles, dans les cas où les *performers* interprètent des éléments artistiques ou scénographiques en temps réel [Camurri et al., 1995].

De nombreuses bibliothèques sont disponibles en ligne et ont été écrites dans plusieurs langages, notamment Java, C++ ou MATLAB. Par exemple, *jFuzzyLogic* [jFuzzyLogic, 2008] offre des fonctions d'appartenance continues, discrètes ou personnalisées, des défuzzifications, les méthodes min et produit pour les implications dans les règles, etc. *FuzzyJ*, quant à lui, est un ensemble de classes Java pour la gestion du raisonnement et des concepts flous [FuzzyJ, 2006]. La *Free Fuzzy Logic Library* [FFLL, 2003] est une bibliothèque de classes en libre accès et elle fournit une interface de programmation (API) en C++. La *Fuzzy Toolbox*™ a été écrite pour MATLAB (MATrix LABoratory), un environnement de développement pour le calcul numérique [Fuzzy Toolbox, 2007]. Cette boîte à outils est constituée de deux blocs : un éditeur pour le système d'inférence flou (SIF) et un contrôleur flou. Elle implémente les concepts de contrôle flou classiques, mais offre seulement les systèmes d'inférence de Mamdani et Sugeno.

Malgré cette offre variée d'implémentations, aucune ne peut être utilisée pour le traitement du signal temps réel. Cependant, plusieurs auteurs ont proposé des *patchs* pour permettre l'utilisation de certaines techniques d'intelligence artificielle. Par exemple, Eigenfeldt décrit et propose des méthodes pour créer des multi-agents en réseau au sein de Max/MSP ainsi que des notations floues pour un groupe de percussions [Eigenfeldt, 2007]. Douze ans plus tôt, Elsea expliquait comment les concepts-clefs de la logique floue pouvaient être appliqués à des problèmes communs dans l'analyse musicale ou la composition [Elsea, 1995]. Par exemple, le "compteur flou" (*fuzzy counter*) autorisait le concept d' "attendre un peu". D'autres *patchs* ont aussi été proposés (citons le "Fuzzy Harmony patcher") mais ils étaient trop spécifiques pour une réutilisation dans Max et ne constituaient pas, de toutes façons, une bibliothèque générique.

Si les outils du CW sont plutôt simples à implémenter lorsqu'on utilise des variables, des fonctions et des méthodes dans des langages classiques comme le C++ ou le Java, c'est moins vrai lorsqu'on utilise Max. Parmi les différences notables entre une conception objet (orientée C++) et une conception "orientée Max/MSP", on trouve :

- les variables : en C++, on manipule des variables et des pointeurs tandis qu'avec Max, il n'existe pas réellement de variables, seulement des boîtes qui peuvent sauvegarder des valeurs. Le principe est celui du "passage par valeur" sans rien stocker explicitement. Il n'est en effet pas aisé d'adresser les boîtes explicitement. Par exemple, les registres à décalage ne sont pas des variables, mais des valeurs à récupérer ;
- les méthodes : en C++, on manipule des méthodes tandis qu'avec Max, les objets représentent à la fois les "variables" et les méthodes. En fait, les "méthodes" doivent

être regroupées fonctionnellement dans des objets ou des *patchs* :
- on utilisera des *patchs* si le calcul doit être isolé (dans l'idée de le réutiliser dans un autre contexte) ;
- on utilisera des objets sinon. Plusieurs objets peuvent faire partie d'un même *patch*. Ces objets doivent embarquer du code source (Java, Javascript ou C).

Ainsi, une méthode, au sens C++ peut être implémentée par un *patch* (de façon graphique ou en programmant) ou bien par un objet. Le choix dépend des préférences du programmeur et de l'importance du temps dans le calcul : si les calculs sont tous synchrones, il est nécessaire d'utiliser un *patch*.

La bibliothèque développée pour Max/MSP que l'on a appelée *FuzzyLib* est téléchargeable à l'adresse `http://imtr.ircam.fr`. Trois objets, au sens Max, ont été implémentés :

- un objet "variable linguistique"[4] appelé *lv1* pour "linguistic variable version 1" (*cf.* figure 3.16). Le rôle de cet objet est la fuzzification d'un phénomène représenté par une valeur numérique ;
- un objet "application du modus ponens généralisé" appelé *gmpa1*, pour "generalized modus ponens application version 1". L'objet *gmpa* reçoit d'abord des données des objets *lv*, autrement dit des changements des valeurs de variables en temps réel, des ajouts de nouvelles variables linguistiques, ou des retraits de variables précédentes. Il reçoit ensuite des règles floues qui connectent les *lv* en entrée aux *lv* en sortie. Les règles sont exprimées comme des *messages* Max/MSP. L'objet fournit aussi des méthodes de défuzzification, plusieurs implications floues (Reichenbach, Łukasiewicz, Larsen, Mamdani, etc.) et quelques t-normes et t-conormes ;
- un objet interface nommé *ruleComposer*, associé à l'objet *gmpa* pour aider l'utilisateur à écrire ses règles floues et éviter les erreurs de syntaxe. Les noms des variables déclarées dans les objets *lv* sont automatiquement collectés et une aide est fournie pour la génération des règles. L'opérateur 'Not' (en plus du 'And' et du 'Or') a aussi été implémenté dans les règles car il peut parfois être plus pertinent d'exprimer un fait par la négative.

FuzzyLib a été testée en grandeur nature pour la pièce de théâtre *Les petites absences* de Marco Bataille-Testu, créée en décembre 2008 à *La Comédie de Caen* par *Le Théâtre du Signe*. Toute l'équipe et en particulier le directeur artistique a utilisé la bibliothèque avant et pendant le spectacle. Trois sous-tâches ont été identifiées : l'acquisition et la qualification de l'observation du *performer*, et la modélisation sémantique de l'interaction homme-machine en direct (lors du spectacle).

Pour l'acquisition de la performance, l'idée est d'extraire des descripteurs de bas et haut niveau pour qualifier le geste et la voix. On a utilisé pour cela le *patch cv.jit* issu de la *Computer Vision library* qui a permis une captation temps réel. Parfois les données ont été directement injectées dans notre *patch lv*, parfois on a, en plus, en amont injecté des données pour l'apprentissage des classes du partitionnement. Ces données étaient issues des précédentes représentations. Les descripteurs vidéo étaient la quantité de mouvement, la surface de contraction, etc. (*cf.* nos travaux sur cette question évoqués plus haut) et les descripteurs audio étaient le timbre, le ton fondamental, la brillance, le bruit, la durée entre deux attaques, etc.

4. La bibliothèque a été entièrement rédigée en anglais.

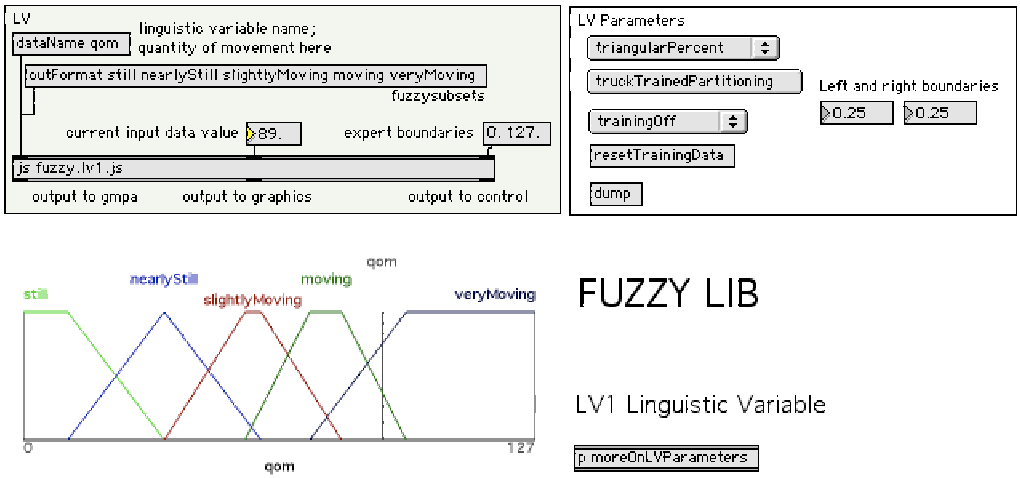

FIGURE 3.16 – Capture d'écran du *patch* d'aide de l'objet *lv*.

Pour la qualification de la performance, les ensembles de descripteurs et d'agrégateurs sont envoyés en entrée de *lv* à la fois pendant la phase d'apprentissage et pendant le spectacle en direct. L'objet *gmpa* peut ensuite indiquer linguistiquement une réponse qualifiant l'activité du *performer*.

La réponse donnée par le logiciel a été utilisée principalement pour contrôler un générateur de sons de foule qui s'adapte au comportement de l'acteur. C'est le directeur artistique qui a lui-même écrit les règles floues.

On le voit, les règles floues sont utilisées ici pour diriger des scénographies, en particulier interactives. Etant donné que l'utilisateur peut saisir son propre vocabulaire, cela fournit des moyens originaux pour les artistes de spécifier des interactions dans leurs spectacles. Alors que les interactions sont habituellement implémentées en définissant des fonctions entre des entrées et des sorties, le CW permet d'écrire ces associations avec des mots.

Le bilan est très positif puisque la pièce a été jouée plus de trente fois depuis décembre 2008, avec ce dispositif, et les acteurs comme le directeur artistique sont très satisfaits. Cependant, on a pu noter lors des expériences que la principale difficulté concerne la compréhension pour l'utilisateur Max des règles floues, et plus précisément des implications floues. En effet, ce n'est pas évident pour un non averti que le résultat d'une implication floue va donner un sous-ensemble flou et non une unique valeur numérique. D'où l'intérêt de travailler le plus possible avec les mots, dans ce contexte. De plus, l'utilisateur non averti utilise souvent une sorte de réciprocité implicite dans les règles. Par exemple, dire "si le mouvement est à peu près stable alors la fréquence est élevée" implique implicitement que "si le mouvement *n'*est *pas* à peu près stable alors la fréquence *n'*est *pas* élevée". Si l'on regarde toutes les implications floues (exceptées celles de Mamdani et Larsen que l'on ne devrait d'ailleurs pas, théoriquement, considérer comme des implications), dire "$A \Rightarrow B$ est équivalent à dire que $\neg A \Rightarrow \neg B$" signifie que l'on considère que "$A \Rightarrow B \equiv A \Leftrightarrow B$". Ce qui est évidemment *a priori* totalement faux. On soulève ici la fameuse différence entre l'implication matérielle et l'implication stricte. L'implication matérielle correspond au conditionnel : "si A alors B", où $A \Rightarrow B$ peut être vrai, même si A est faux (de même, du moment

que B est vrai, quel que soit A, $A \Rightarrow B$ est toujours vrai), tandis que l'implication stricte correspond à un énoncé où la prémisse A est soit vraie, soit fausse, mais ne peut être fausse systématiquement (comme par exemple $1 + 1 = 1$).

Une première façon, simple, consiste à *évacuer* ce problème en considérant comme opérateurs d'"implication" ceux de Mamdani ou Larsen qui, avec leur propriété de commutativité, peuvent correspondre à cette façon de penser les règles : $\min(A, B) = \min(B, A)$ et $A.B = B.A$. Cependant, dans le cas Mamdani, $\min(A, B) \neq \min(\neg A, \neg B)$, donc il est préférable de prendre un opérateur d'équivalence floue comme $\min(A \Rightarrow B, B \Rightarrow A)$.

Une autre façon consiste à tenter de *résoudre* le problème en se penchant davantage sur les règles elles-mêmes. D'après Dubois et Prade, une règle du type "si X est A_i alors Y est B_i" peut exprimer deux types d'information [Dubois et Prade, 2003]. Elle peut exprimer que les situations "(X, Y) *est dans* $(A_i,$ *non* $B_i)$" sont impossibles. Dans ce cas, les règles sont représentées de manière disjonctive, sous la forme *non* A_i *ou* B_i. Mais cette règle peut aussi exprimer que les situations "(X, Y) *est dans* (A_i, B_i)" sont garanties possibles. Et dans ce cas, les règles sont représentées de manière conjonctive, sous la forme A_i *et* B_i. On a donc des situations *garanties possibles* et des situations *non impossibles*. On considère ainsi les informations comme étant *positives* ou *négatives* (logique bipolaire). Pour les informations positives, on utilise les exemples de la base de données pour l'apprentissage. Pour les informations négatives, on utilise les contre-exemples de la base. C'est-à-dire que l'on croit vrai ce que l'on observe souvent, et faux ce que l'on n'observe jamais [Dubois et Prade, 2003].

Une autre utilisation de notre *FuzzyLib* a été amorcée dans le suivi et la reconnaissance de geste à l'aide de modèles de Markov cachés : à l'Ircam, Bevilacqua et son équipe ont en effet développé des *patchs* Max à cet effet [Bevilacqua et al., 2007] mais dont les résultats ne les satisfont pas pleinement. Un travail est en cours sur l'utilisation de notre bibliothèque au sein de leurs *patchs*. Aujourd'hui, il a déjà été montré qu'on pouvait grandement améliorer le système en proposant une reconnaissance "floue" du geste, c'est-à-dire que la reconnaissance peut fluctuer dans le temps et indiquer, soit une réponse globale incluant les imprécisions successives (par exemple, "alors que le geste était effectué à 5 %, il a été reconnu comme étant 52 % du geste A, 25 % du geste B et 23 % du geste C"), soit indiquer au plus vite une réponse (par exemple, "le geste actuellement reconnu est A"), lorsque le système requiert une réponse immédiate, en temps réel.

Deux publications [Bonardi et Truck, 2009, Bonardi et Truck, 2010] présentent toutes ces recherches en donnant notamment quelques détails supplémentaires sur les *patchs* développés dans la *FuzzyLib*.

Ces travaux ont été réalisés en collaboration avec notre collègue Alain Bonardi.

On l'a vu, la capture des intentions dans les arts de la scène grâce au CW est tout à fait pertinente. C'est pourquoi nous avons poursuivi nos réflexions dans le cadre de la capture des intentions du programmeur. En particulier, l'intérêt d'utiliser des approches linguistiques dans le domaine de l'*autonomic computing* (calcul auto-régulé) semble de plus en plus évident. L'*autonomic computing*, qui est une branche du génie logiciel apparue ré-

cemment (2003) sous l'impulsion d'IBM, a pour objectif général d'automatiser au maximum la gestion des applications, tâche habituellement dévolue aux administrateurs systèmes ou aux programmeurs. L'idée est donc que l'informatique soit mise au service du contrôle et de l'optimisation des applications en créant des systèmes auto-adaptables.

Si l'on considère le "modèle" général client / fournisseur, où le client a un besoin que le fournisseur peut lui offrir, on trouve plusieurs thématiques parmi lesquelles les architectures orientées services, l'informatique dématérialisée (mieux connue sous le nom de *Cloud Computing*) ou encore les systèmes à composants. Toutes ces thématiques impliquent des interactions client / fournisseur nécessitant un traitement des imprécisions sous forme linguistique puisque, inévitablement, l'humain est toujours derrière une demande (et une offre), à un moment ou à un autre. On constate, une fois de plus, qu'ici encore, le *Computing with Words* a toute sa place.

3.3 Le calcul à l'aide de mots pour capturer les intentions du programmeur

Dans les architectures fondées sur les services (*Service-oriented architectures* ou SOA), les logiciels sont conçus comme des processus métiers qui vont, le plus possible, utiliser des services existants pour réaliser leurs calculs. Dans ce contexte, les services apparaissent et disparaissent régulièrement et ceux qui réalisent un même calcul mais à des qualités de service et des coûts différents, se retrouvent en compétition entre eux. Ainsi, un problème crucial est celui de la *sélection* des services à utiliser. De nombreux travaux s'intéressent donc à cette problématique. Pour notre part, pour garantir au mieux la fraîcheur de l'information, nous cherchons à repousser au dernier moment la liaison entre service demandé et service offert. Ainsi, on est sûr d'avoir l'information la plus récente pour satisfaire le client et lier sa demande au meilleur service disponible. Pour réaliser cela, il faut être capable, une fois que les contraintes fonctionnelles du service offert (c'est-à-dire le type de service rendu) sont respectées — cela revient à procéder à un premier tri, grossier — de faire coïncider les contraintes non fonctionnelles des services restant en lice à ce moment précis avec les demandes du client, autrement dit, ses préférences — cela revient à affiner le tri. A ce moment-là, il est important d'être suffisamment souple dans la mise en relation entre l'offre et la demande, afin d'éviter de ne faire ressortir aucun service. C'est pourquoi une approche linguistique a été proposée : le client exprime en amont ses préférences de façon imprécise et la liaison peut se faire au dernier moment en affectant une utilité à chaque service candidat encore en lice. Nous avons donc proposé un modèle inspiré des *Conditional Preference networks* de Brafman et Domshlak, graphes dans lesquels les nœuds sont les propriétés et les arcs les préférences conditionnelles (ou non conditionnelles) entre ces nœuds. La gestion de l'imprécision se fait au niveau des nœuds et des arcs en leur associant des tables de préférences linguistiques.

3.3.1 Modélisation de préférences conditionnelles

Les SOA tentent de répondre au besoin croissant pour les applications distribuées d'être en mesure d'évoluer continuellement pendant leur exécution. Dans le contexte du Web, les

fournisseurs proposent des fonctionnalités ou "caractéristiques atomiques" (*atomic features*) qui sont appelées "services Web" et qui peuvent apparaître ou disparaître au cours du temps. L'approche SSOA (*Semantic* SOA) permet de rendre plus riche la publication des services offerts (*via* des annuaires dédiés) en ajoutant des métadonnées sémantiques dans la définition de l'offre. Un résumé des caractéristiques offertes avec leurs propriétés est donc disponible pour chaque service Web. Ainsi, il est possible pour un client, littéralement "abonné" (*consumer*), de *composer* et d'*orchestrer* en temps réel plusieurs services provenant de différents fournisseurs. Un des problèmes qu'il faut traiter est celui de la qualité de cette orchestration et, plus précisément, de la qualité du choix de tel ou tel service. Pour réussir la composition, les services doivent être disponibles au moment de leur appel. On reporte donc au *dernier* moment (*i.e.* à l'exécution) la question du lien (abonné du service) $\leftrightarrow$ (fournisseur du service) : c'est la liaison tardive. De plus, la qualité du choix des services peut être améliorée en offrant la possibilité au client d'exprimer ses besoins sous la forme de contraintes fonctionnelles (par exemple, "je veux un service de caméra vidéo") et non fonctionnelles (par exemple, "la caméra doit fournir une image de résolution 800*600 pixels, en 16 millions de couleurs et avec une latence de 40 ms"). Ces contraintes non fonctionnelles sont généralement appelées "qualité de service" (QoS).

Un des points essentiels à aborder, lorsque l'on traite la QoS, est l'importance de chaque dimension. Par exemple, il faut connaître l'importance pour l'abonné de la latence, de la précision, du coût, etc. souhaités pour pouvoir sélectionner les services qui satisferont la demande. Mais il est souvent rare de disposer d'offres qui correspondent exactement aux souhaits dans chaque dimension de QoS. C'est pourquoi les préférences dans les demandes sont nécessaires pour être à même de sélectionner les meilleures offres et de les ordonner.

L'élicitation et l'expression des préférences ont fait l'objet de beaucoup d'attention depuis de nombreuses années et plusieurs formalismes ont vu le jour. Dans le contexte des SOA, un bon formalisme doit obéir à plusieurs exigences, parmi lesquelles l'utilisation par des non spécialistes, tels que des programmeurs de "processus métier" (*business process*), est de première importance. Une autre exigence est le besoin d'exprimer les concessions ou plutôt compromis (par exemple, "j'accepte un service de caméra vidéo avec une résolution moindre à condition que la latence soit extrêmement faible"). C'est pourquoi nous nous sommes tournée vers des modèles graphiques capables d'exprimer de façon compacte des préférences conditionnelles. Parmi les modèles compacts (modèles d'utilité factorisés), il existe les modèles *additifs* dans lesquels on fait hypothèse que les préférences sur les valeurs d'un attribut peuvent être exprimées *indépendamment* des valeurs des autres attributs [Braziunas et Boutilier, 2006]. Les *Conditional Preference networks* (ou CP-nets) furent les premiers formalismes graphiques d'une famille que l'on peut noter de façon générique *CP-nets qui propose de mettre en réseau les propriétés souhaitées et les préférences entre les souhaits. Nous les avons choisis car ils nous permettent de bénéficier de nombreux avantages comme la facilité d'utilisation pour des non avertis, le coût relativement faible en temps de calcul, une structure mathématique solide et formellement définie et la relative facilité d'ajouter des propriétés additionnelles en fonction des besoins. D'autres formalismes existent dans la littérature, comme les réseaux GAI (Generalized Additive Independence) [Gonzales et al., 2008], davantage inspirés des réseaux bayésiens, mais ces approches demandent un gros effort pour l'élicitation des préférences. Elles sont en fait très utiles lorsque le problème est de discriminer parmi un très grand nombre de possibilités.

Mais ce n'est pas notre cas.

Un CP-net est une représentation compacte graphique des préférences d'utilisateurs [Boutilier et al., 2004]. Il est assez intuitif et possède trois éléments : des *nœuds* qui représentent les variables du problème, des *arcs* (encore appelés cp-arcs) qui portent les préférences parmi ces variables pour différentes valeurs données et des tables de préférences conditionnelles notées CPT pour *Conditional Preference Tables*. Les CPT expriment les préférences sur les valeurs prises par les variables. Les CP-nets permettent la modélisation de souhaits comme "pour la propriété X, je préfère la valeur V_1 à la valeur V_2 si les propriétés Y égale V_Y et Z égale V_Z". En fait, cette représentation graphique permet d'exprimer la dépendance des CPT connectées. Ainsi, les préférences peuvent être exprimées conditionnellement aux valeurs prises par leurs nœuds parents dans le graphe, mais sans tenir compte des valeurs prises par les autres nœuds : c'est la propriété du *ceteris paribus*, centrale aux CP-nets, qui signifie "*toutes choses étant égales par ailleurs*". Il existe aussi une notion de *préférence relative* entre les préférences elles-mêmes : une CPT associée à un certain nœud a une priorité plus forte que les CPT de ses descendants. Cette notion est prise en compte lorsque les affectations complètes sont comparées. Une affectation complète est un tuple de valeurs pour toutes les variables du graphe. Un des intérêts des CP-nets est la possibilité d'approximer facilement les préférences avec des règles d'inférence qui ne seront rien d'autre que les CPT. Cependant, il faut noter que le CP-net doit obéir à certaines restrictions pour pouvoir autoriser, d'un point de vue algorithmique, la plupart des calculs d'inférence pour le raisonnement. La première concerne les graphes eux-mêmes qui doivent être acycliques, la deuxième implique une utilisation raisonnable de la relation d'indifférence dans les préférences et ainsi la définition systématique de pré-ordres totaux dans les CPT pour chaque nœud parent [Boutilier et al., 2004].

Les *Utility CP-nets* (ou UCP-nets) [Boutilier et al., 2001], diffèrent des CP-nets en remplaçant la définition de la relation binaire $\succ$ ("est préféré à") entre deux valeurs de nœuds dans les CPT par des valeurs numériques (des facteurs d'utilité). Ainsi, les valeurs des nœuds elles-mêmes conservent la même forme que dans les LCP-nets, seules les préférences sont quantifiées avec des valeurs d'utilité (numériques). Ce changement a été motivé par le fait que la précision d'une fonction d'utilité (par opposition à un simple ordre de préférence) est souvent nécessaire dans les contextes de prise de décision. Il a aussi été motivé par le fait qu'un CP-net n'autorise ni une comparaison ni un ordre dans les alternatives données en solution au problème. En quantifiant les préférences, ce souci est atténué [Boubekeur et Tamine-Lechani, 2006]. Un facteur d'utilité est un nombre réel associé à une affectation d'un nœud X du graphe, étant donné une certaine affectation à ses nœuds parents. Il exprime un degré de préférence entre différentes affectations. Aussi, les programmeurs qui modélisent les préférences utilisent les utilités des CPT pour indiquer les choix locaux à un nœud, mais l'utilité globale, pour chaque affectation complète, sera calculée par l'UCP-net et permettra ainsi d'ordonner, sans ambiguïté, les solutions. En effet, un UCP-net définit un ordre total sur les affectations, chaque affectation ayant une utilité globale définie dans $\mathbb{R}$.

Une autre extension des CP-nets, les *Tradeoffs-enhanced CP-nets* (ou TCP-nets), gère les compromis dans l'expression des préférences [Brafman et Domshlak, 2002]. Les TCP-nets autorisent des expressions de la forme : "une meilleure affectation à X est plus importante

qu'une meilleure affectation à Y", encore appelées *préférences de relative importance*. De surcroît, ils autorisent logiquement les *préférences conditionnelles de relative importance* : "une meilleure affectation à X est plus importante qu'une meilleure affectation à Y si $Z = z$". Ainsi, de nouvelles tables de préférences sont introduites : les CIT (*Conditional Importance Tables*) ainsi que deux nouveaux types d'arcs : les i-arcs et les ci-arcs. Ces arcs permettent, respectivement, de modéliser des propositions d'importance relative *basiques* et *conditionnelles*. Ce dernier modèle de la famille des *CP-nets correspond bien à nos besoins : si peu de services Web sont disponibles, l'abonné devra sans doute faire des compromis dans ses choix et concéder certaines propriétés de QoS.

L'idée de mélanger les CP-nets avec les propriétés non fonctionnelles des services avait déjà germé chez Schröpfer *et al.* qui ont défini des préférences au travers des CP-nets pour sélectionner le meilleur service dans une architecture de type SOA [Schröpfer et al., 2007]. Mais ils n'ont ni considéré la question des préférences linguistiques, ni véritablement celle des domaines continus de variables — exclus des *CP-nets — et pourtant indispensable dans notre cadre. Il s'agit en fait d'une simple utilisation d'un modèle existant dans leur problème, à ceci près qu'ils ont utilisé une discrétisation simple pour le problème des domaines continus.

Les *CP-nets affichent deux limitations importantes. La première concerne la non gestion de la continuité des domaines. Nous proposons de discrétiser les domaines continus en utilisant des termes linguistiques associés à des sous-ensembles flous ou à des 2-tuples linguistiques. Dans un contexte où les utilisateurs doivent exprimer leurs préférences parmi des valeurs de domaines continus, il est évident que le recours aux sous-ensembles flous ou à un formalisme acceptant des transitions douces permettra de mieux capter les intentions des utilisateurs, et d'améliorer le classement des services à proposer à l'abonné. En effet, si deux services ont des valeurs de QoS à peu près similaires, une utilisation de domaines discrets avec une granularité grossière les placera au même rang dans le classement final. A moins, bien sûr, d'augmenter la granularité et de forcer (artificiellement, comme cela est précisément conseillé par les auteurs des UCP-nets) les différences dans les préférences sur les valeurs. La seconde limitation des *CP-nets concerne l'élicitation et la difficulté d'obtenir des valeurs d'utilité précises de la part des demandeurs des services. Donner une valeur numérique réelle pour exprimer une perception ou une préférence est difficile pour un non spécialiste. En conséquence, les modèles *CP-nets actuels fournissent finalement deux alternatives : l'utilisation des CP-nets qui encodent la relation d'ordre $\succ$ plus réaliste que les utilités, mais qui souffrent de performances basses (longs temps de calcul) pour comparer deux affectations ; l'utilisation des UCP-nets qui arborent de meilleures performances mais avec des valeurs numériques difficiles à obtenir.

Nous avons donc imaginé un nouveau formalisme de modélisation des préférences, en nous fondant sur la famille des *CP-nets, capable de traiter des propositions linguistiques, de considérer des variables sur des univers continus et de faciliter l'élicitation.

Tout d'abord, des descripteurs linguistiques appropriés doivent être choisis pour former l'ensemble de termes attaché à chaque univers de discours. Comme c'est souvent l'usage, on prendra des ensembles de termes à cardinalité impaire (5, 7 ou 9) [Delgado et al., 1993, Herrera et Martínez, 2000] pour être en mesure de représenter un terme "milieu". Par exemple, on considère l'ensemble T constitué des cinq termes sui-

vants[5] : $T = \{s_0 : \textit{very low}, s_1 : \textit{low}, s_2 : \textit{medium}, s_3 : \textit{high}, s_4 : \textit{very high}\}$. Il est aussi habituellement requis qu'il y ait les trois opérateurs usuels *Neg*, *max* et *min* définis sur T [Herrera et Martínez, 2000] :

1. $\text{Neg}(s_i) = s_j$ tel que $j = g - i$ (avec $g + 1$ la cardinalité),
2. $\max(s_i, s_j) = s_i$ si $s_i \geq s_j$,
3. $\min(s_i, s_j) = s_i$ si $s_i \leq s_j$.

Dans le modèle des 2-tuples linguistiques de Martínez *et al.*, les fonctions d'appartenance trapézoïdales ou triangulaires sont suffisantes pour capturer l'imprécision des propositions. Etant donné un terme linguistique, le formalisme 2-tuple fournit un couple (sous-ensemble flou, translation symbolique) $= (s_i, \alpha)$ avec $\alpha \in [-0.5, 0.5[$ comme on peut le voir sur la figure 3.17 où le 2-tuple obtenu est $(s_2, -0.3)$.

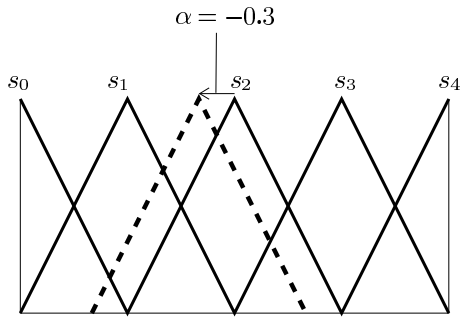

FIGURE 3.17 – Déplacement latéral d'une étiquette linguistique $\Rightarrow$ le 2-tuple $(s_2, -0.3)$.

La translation α peut d'ailleurs être vue comme un modificateur affaiblissant du terme linguistique s_2. Ainsi, en utilisant ce modèle pour faire des calculs et exprimer leur résultat, on affiche une valeur solution égale à l'un des éléments de l'ensemble de départ affecté d'une certaine modification α.

Le modèle computationnel fondé sur les 2-tuples linguistiques a pour caractéristique de faire des "calculs à l'aide de mots" sans perte d'information.

Notre proposition est ainsi d'exprimer les valeurs d'utilité de façon linguistique, en utilisant soit des sous-ensemble flous, soit les 2-tuples linguistiques. Comparé aux précédents *CP-nets, le nouveau formalisme que nous proposons permet l'expression de propositions telles que : "pour la propriété X, je préfère la valeur *à peu près* V_1 à *exactement* la valeur V_2 si les propriétés Y égale *approximativement* V_Y et Z égale *un peu plus que* V_Z". Ces propositions, qui ressemblent à des règles floues élaborées, doivent en outre être interprétées dans un contexte où la préférence globale sur X devra prendre en compte chaque préférence qui s'applique, à un certain degré, à la valeur de Y. Finalement, cela revient à proposer un modèle souple et intuitif pour exprimer des ensembles de règles floues compliqués pouvant potentiellement être interdépendants.

Ce modèle a été nommé LCP-nets (pour *Linguistic CP-nets*). Comme dans les autres modèles, le réseau est constitué de nœuds qui correspondent aux variables du problème dont les domaines continus sont discrétisés en ensembles de termes linguistiques. Pour résumer, les

5. Ici encore, nous laissons l'anglais, langue dans laquelle tous les développements ont été réalisés.

LCP-nets permettent aux utilisateurs d'exprimer des importances relatives, conditionnelles ou non, et des compromis parmi les variables en utilisant les i-arcs ou les ci-arcs des TCP-nets, en plus des cp-arcs des CP-nets. Ils comprennent des CPT similaires à celles des UCP-nets mais exprimant des utilités avec des termes linguistiques. La figure 3.18 résume ces différentes possibilités : il s'agit de préférences utilisateur pour un service de caméra (par exemple, une caméra de surveillance). Le but général est d'obtenir les images captées aussi vite que possible. Trois propriétés de QoS sont utilisées : la sécurité (*security* S), la bande passante (*bandwidth* B) et la résolution de l'image (*image resolution* R). L'utilisateur préfère toujours privilégier B par rapport à S, et si la bande passante est faible, il préfère des images de basse résolution pour les obtenir aussi vite que possible.

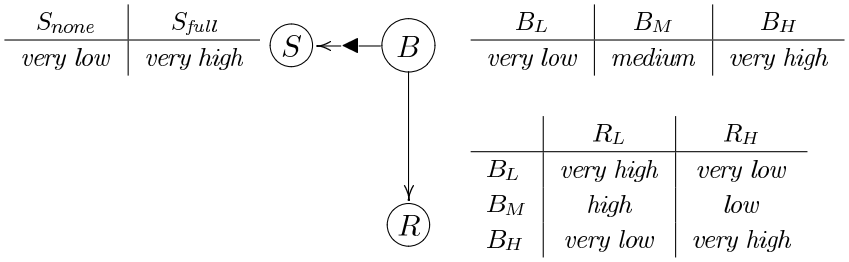

FIGURE 3.18 – Un exemple de préférences pour un service offrant une fonctionnalité de caméra, en utilisant les LCP-nets.

Dans le contexte des SSOA, les étapes qui sont détaillées ci-dessous sont réalisées séquentiellement pour implémenter le processus : la sélection dynamique de services Web fondée sur les préférences non fonctionnelles de l'abonné et sur les valeurs les plus récentes de QoS des services contrôlés. Ces étapes ont été implémentées en Java en utilisant notamment la bibliothèque *jFuzzyLogic* pour gérer la partie purement floue et un *framework* support de LCP-nets a été mis à disposition à l'adresse `http://code.google.com/p/lcp-nets/`. Elles sont au nombre de trois :

- l'étape d'élicitation. Elle est faite en amont, avant l'exécution ;
- l'étape de *traduction* du modèle de préférences en une représentation efficace utilisable à l'exécution. Dans le processus, chaque CPT est traduite en un système d'inférence avec une règle par ligne de la table ;
- l'étape d'évaluation du modèle de préférences. D'abord, pendant la phase d'injection de valeurs de QoS, de nombreuses valeurs de QoS sont récoltées et subissent une fuzzification vers le formalisme flou choisi (sous-ensembles flous ou 2-tuples linguistiques). Puis le système procède au calcul des utilités locales à chaque nœud grâce à un système d'inférence implémentant le modus ponens généralisé avec défuzzification pour obtenir des valeurs numériques. Pour finir, l'utilité globale est obtenue par agrégation des utilités locales.

L'agrégateur à utiliser dans la troisième étape ne peut être une simple moyenne équipondérée. En effet, il faut tenir compte du fait que les arcs donnent l'importance relative des nœuds (et donc des propriétés de QoS) qu'ils interconnectent. Cette importance relative *implicite* due à la position des nœuds doit être répercutée d'une façon ou d'une autre afin d'être prise en compte dans le calcul final de l'utilité globale. Par exemple, pour le service

Web de caméra de surveillance, la bande passante B se trouvant au "sommet" du graphe est nécessairement plus importante que la sécurité et la résolution. Par contre, S et R, situés au même niveau de profondeur, sont d'importance égale. Nous attachons donc un poids à chaque nœud, et ainsi à chaque utilité (*cf.* figure 3.19). Il faut noter que les UCP-nets n'utilisent pas de poids sur les nœuds car les utilités sont déjà censées porter cette information. En effet, le formalisme exige des écarts suffisamment importants entre les facteurs d'utilité pour qu'ils portent implicitement la notion de préférence relative.

Ainsi, des poids sont distribués aux nœuds en tenant compte de leur profondeur — et en commençant par le nœud racine — *via* une fonction monotone, strictement décroissante. On considère des poids valués sur [0,1] dont la somme égale 1. Ce type de fonction correspond tout à fait à une fonction BUM (basic unit-interval monotonic) [Yager, 2007]. Nous donnons un certain nombre de détails à ce sujet et notamment sur le calcul de la mesure de OU ou ET (*orness, andness*) d'un opérateur d'agrégation dans l'article soumis à une revue et ajouté en annexe, page 79.

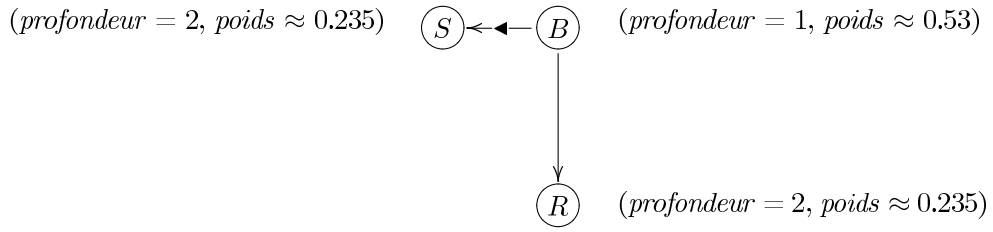

FIGURE 3.19 – De la profondeur des nœuds vers le poids des nœuds.

L'utilité globale ainsi obtenue permet de comparer très simplement (puisqu'il s'agit de nombres) les différentes affectations pour un service donné et autorise donc un classement précis. Si l'on souhaite que le choix final du meilleur service à sélectionner soit fait par un humain (on sort donc du cadre de la liaison tardive des services Web), il peut être utile de donner une réponse linguistique pour les utilités (locales comme globale). Mais cela impose idéalement de ne jamais défuzzifier. En particulier, les 2-tuples linguistiques de Martínez *et al.* pourraient porter l'information des préférences et des utilités de bout en bout. Cette question est discutée dans les perspectives, en section 4.1.4, page 65.

Nous terminons cette partie par un exemple d'utilisation de nos LCP-nets. La figure 3.20 résume le processus global effectué dans le cas précédemment évoqué d'un service Web de caméra de vidéosurveillance.

D'abord, on partitionne les univers associés aux propriétés des QoS. On obtient des termes linguistiques qui serviront à la définition des préférences de l'abonné (étape 1). L'élicitation se fait ensuite *via* la construction des arcs interconnectant les nœuds et des tables de préférences (étape 2). Ici, la préférence de la bande passante sur la sécurité est exprimée par un i-arc de B vers S (préférence inconditionnelle). B est discrétisée avec trois variables linguistiques B_L (*low*), B_M (*medium*) et B_H (*high*). Les préférences parmi ces valeurs sont données par la CPT se trouvant à côté de B qui exprime une préférence très faible (*very low*) pour une bande passante égale à B_L, moyenne (*medium*) pour une valeur B_M et très forte (*very high*) pour une valeur B_H. Il en va de même pour S, discrétisée avec deux va-

riables. La résolution de l'image fournie par la caméra est, quant à elle, discrétisée avec deux variables : R_L (*low*) et R_H (*high*). Les préférences parmi ces valeurs sont conditionnées par la bande passante et symbolisées par un cp-arc. La CPT se trouvant à côté de R montre les choix de l'abonné.

Ensuite (étape 3a), le modèle LCP-net est "compilé" afin de fournir les jeux de règles qui vont servir aux systèmes d'inférence floue. Sur la figure, cela est symbolisé par l'extrait de programme écrit en FCL[6]. En parallèle, les poids attachés aux nœuds peuvent être calculés (étape 3b). Les valeurs courantes des propriétés de QoS sont injectées dans le modèle de préférence compilé (étape 4) et chaque nœud se voit affecté d'une utilité locale (étape 5). Les poids et les utilités locales étant obtenus, il est finalement procédé au calcul de l'utilité globale avec un agrégateur noté Δ dans la figure (étape 6). Dans l'exemple, la valeur finale est 0.47 (les facteurs d'utilité prennent leurs valeurs dans $[0, 1]$).

Notre approche permet d'avoir un formalisme qui construit une fonction d'utilité globale progressive et continue, là où des approches usuelles, comme celle de Schröpfer *et al.* précédemment citée, donnent des fonctions d'utilité globales constantes par morceaux, limitant nettement leur pouvoir de discrimination entre les alternatives. Ainsi, les LCP-nets autorisent une bonne différenciation des choix possibles tout en maintenant un coût d'élicitation et une complexité des préférences faibles.

Quatre publications [Châtel et al., 2008, Le Duc et al., 2009a, Châtel et al., 2010a, Châtel et al., 2010b] ont été écrites sur ce sujet. Le lecteur pourra s'y reporter, notamment pour approfondir les questions, peu abordées ici, des architectures (SOA, SSOA) et de l'implémentation des LCP-nets.

Ces travaux ont été réalisés en collaboration avec notre collègue Jacques Malenfant. Nous avons co-encadré la thèse CIFRE (Thales/LIP6) de ***Pierre Châtel*** *intitulée "Une approche qualitative pour la prise de décision sous contraintes non-fonctionnelles dans le cadre d'une composition agile de services" soutenue en mai 2010, avec la mention* très honorable.

3.3.2 Modélisation des besoins et des offres pour la définition *ad hoc* de politiques

On vient de voir l'intérêt de l'application de la théorie des sous-ensembles flous à la prise de la décision, idée qui vient de Bellman et Zadeh dans un (des nombreux !) article(s) fondateur(s) [Bellman et Zadeh, 1970], ce qui nous a amenée à poursuivre dans cette voie, dans ce domaine des *architectures logicielles auto-adaptatives* et du *calcul auto-régulé*.

Le domaine du *Cloud Computing* est de plus en plus au centre des préoccupations. La dématérialisation a commencé depuis plusieurs années déjà avec la notion de bureau virtuel, lecteur réseau, etc. Pour mettre en œuvre de tels systèmes, il faut être capable de mettre en adéquation les ressources allouées au besoin des applications. Sachant que ces applications s'exécutent sur de longues périodes avec des charges très fluctuantes, l'adaptation dynamique des ressources est une nécessité.

6. FCL qui signifie "Fuzzy Control Language", est un standard pour le contrôle flou (Fuzzy Control Programming) publié par l'International Electrotechnical Commission [IEC, 2001].

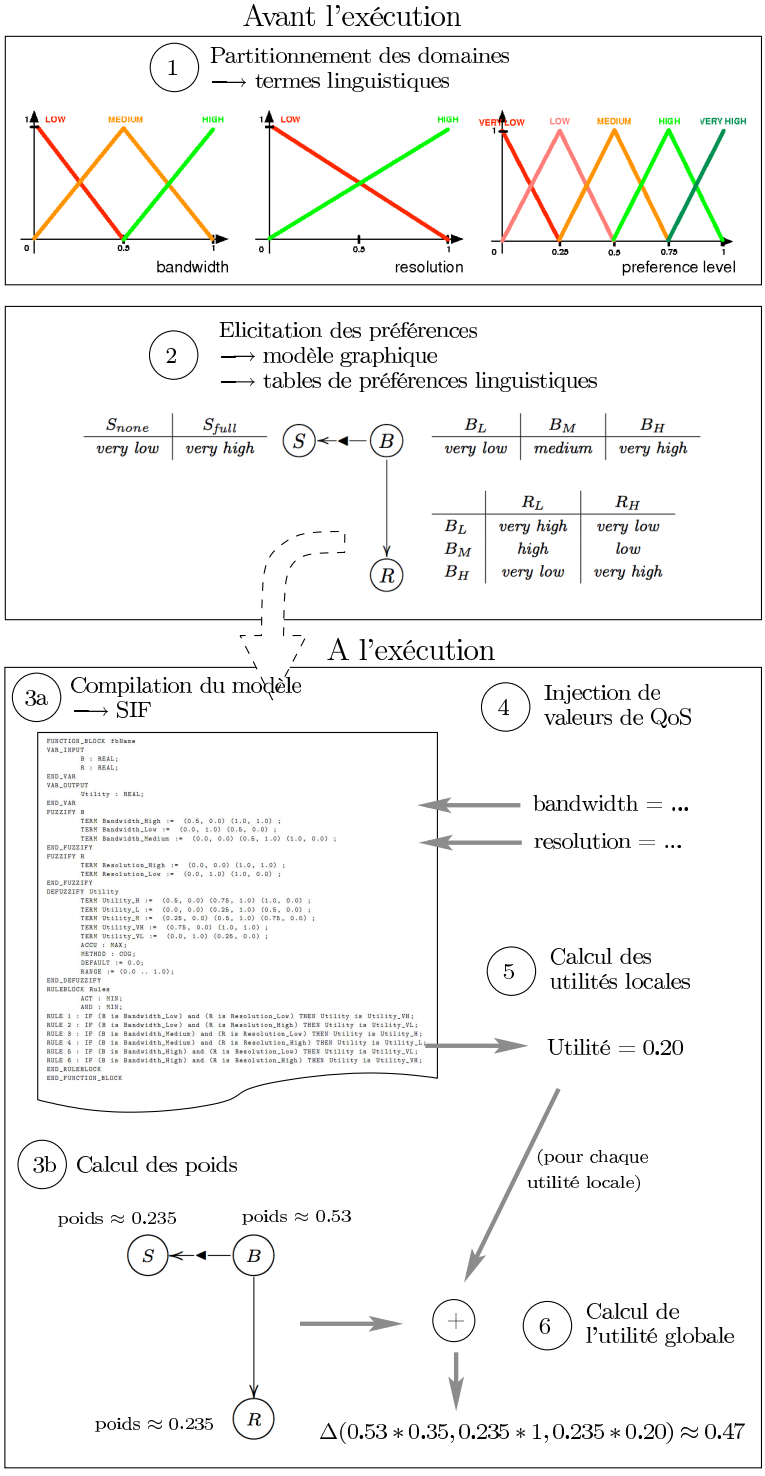

FIGURE 3.20 – Utilisation des LCP-nets pour un service Web de caméra de surveillance.

Actuellement, un certain nombre d'offres existe sur le marché. Par exemple, l'*AppEngine*[7]

7. `http ://code.google.com/intl/fr/appengine`

de Google propose du redimensionnement automatique (*auto-scaling*) dans un environnement de type PaaS (*Platform as a Service*), c'est-à-dire dans le cadre d'une plateforme de conception et d'hébergement d'applications Web. Mais les applications doivent être développées spécifiquement pour ces plateformes. Les modèles de type IaaS (*Infrastructure as a Service*) sont davantage flexibles puisque les utilisateurs ont un libre accès aux ressources physiques virtualisées provenant d'*Amazon EC2*[8] ou d'*OpenNebula*[9], etc. Mais des problèmes demeurent comme le manque de standards (quels contrats mettre en place permettant une bonne expression de l'offre et de la demande) ou l'automatisation elle-même du redimensionnement (ce sont les utilisateurs — ou un tiers — qui doivent administrer et paramétrer les infrastructures auxquelles ils ont accès).

Cette automatisation souhaitée nécessite un module de décision et la définition de politiques. Ces points sont, pour l'instant, relativement sous-estimés et donc trop peu traités dans la littérature [Tesauro et al., 2006, Xu et al., 2007, Kalyvianaki et al., 2009, Lim et al., 2009, Moreno-Vozmediano et al., 2009].

Les politiques doivent être capables de capturer la connaissance des experts et de les améliorer par de l'apprentissage en ligne. Le potentiel des approches floues pour ce faire est énorme. Déjà, nos travaux sur les LCP-nets et l'élicitation linguistique des préférences à l'aide d'un modèle simple et accessible aux non-experts (les ingénieurs logiciels) procèdent de cette idée. Un grand nombre de politiques d'allocation de ressources par exemple, nécessite de déterminer des valeurs pour des paramètres numériques que l'ingénieur logiciel n'est pas en mesure d'estimer précisément. Procéder par une approche linguistique, affiner cela par de l'apprentissage en ligne, pour obtenir enfin des paramètres numériques par défuzzification apparaît comme une démarche de choix pour un très large spectre d'applications en *autonomic computing*.

Dans les centres de calcul, la problématique principale est, pour un client demandeur, de lui fournir une ou plusieurs machines virtuelles (VM pour *Virtual Machine*) afin de lui offrir la ressource dont il a besoin pour faire fonctionner son application. Du côté fournisseur, le problème est de :

- satisfaire le client (atteindre les objectifs négociés (SLO) selon des accords-contrats (SLA)) en lui donnant la ressource *nécessaire* ;
- satisfaire le maximum de clients en dépensant le moins possible, c'est-à-dire en donnant au maximum de clients la ressource *suffisante*.

Cette recherche de compromis est un problème de décision dans lequel beaucoup de variables peuvent jouer un rôle : il est évident que, à chaque ajout d'entrée ou de paramètre, la combinatoire augmente et finit par exploser. Il faut donc sélectionner les plus pertinents.

- Parmi les entrées déterminantes, on trouve le trafic ou charge entrant(e) (*workload*) qui correspond au nombre de requêtes reçues par seconde, le nombre de VM déployées pour ce client (*resource usage*), le nombre de VM disponibles et la performance (nombre de requêtes satisfaites ou plus précisément la durée et le niveau de violation des SLA, sachant qu'un SLA est pour nous le temps de réponse maximal accepté par le client).
- Et parmi les paramètres, on cherchera à satisfaire les clients les moins exigeants à coût

8. `http ://aws.amazon.com/ec2`
9. `http ://www.opennebula.org/start`

fournisseur constant, clients que l'on peut aussi voir comme les plus "rémunérateurs" (avec les SLA les plus facilement atteignables). D'autres paramètres découlent aussi directement des causes d'instabilité du système. Par exemple, la latence, c'est-à-dire le temps qu'une ressource nouvellement ajoutée met à être opérationnelle, amène, si elle n'est pas prise en compte, à prendre trop rapidement une décision (qui devient donc mauvaise). La puissance du contrôle est aussi un point important, c'est-à-dire qu'il faut prendre garde à l'impact du contrôle sur les sorties (ne pas générer un contrôle dont l'amplitude minimum produit des valeurs en sortie telles que l'on sort immédiatement des objectifs fixés). Enfin, on peut noter le phénomène d'oscillation dans les entrées qui se produit si le *pattern* de charge est trop (rapidement) fluctuant. Cela devient un problème lorsque le système ne sait pas s'adapter à la vitesse d'oscillation.

Tout ceci nous a amenée à considérer deux dimensions dans la scalabilité des ressources afin de les rendre davantage "élastiques" : la dimension horizontale qui correspond à la *quantité* de VM, la dimension verticale qui correspond à la *qualité* des VM. En effet, on peut décider d'allouer des VM différentes (plus ou moins de mémoire vive, une plus ou moins grande taille de disque dur, un CPU plus ou moins performant, une bande passante réseau plus ou moins grande, etc.), c'est-à-dire personnalisées selon les besoins. Et cette scalabilité diagonale (horizontale *et* verticale) nous amènera, nous l'espérons, à réduire la latence — ou en tous cas à mieux la maîtriser. En effet, dans certains cas, il peut être inopportun (ou excessif) d'allouer une VM — même de faible capacité — mais, en revanche, judicieux d'augmenter la RAM des (ou de certaines) VM existantes.

Notre système se propose donc de fonctionner en deux temps : (i) recherche du nombre de VM à allouer, en les considérant toutes comme identiques (solution grossière) ; puis (ii) affinage en ajoutant des paramètres comme le nombre de VM disponibles, et obtention d'un *pool* de VM personnalisées.

Ainsi, nous proposons deux avancées dans ce domaine :
- la définition d'un modèle flou de VM, en utilisant des variables linguistiques comme "machine très rapide", "machine sûre", "machine stable", "disque dur de très grande capacité", etc. Le passage par les variables linguistiques est d'un grand intérêt dès lors que le fournisseur propose cette possibilité *via* une IHM, le système de décision permettant ensuite, de façon transparente, d'affiner la (les) VM(s) proposée(s) ;
- la définition d'un modèle de contrat (SLA) par (i) l'expression des besoins et des offres de façon imprécise (mais pas nécessairement) avec une mise en correspondance entre les deux, et par (ii) l'expression des préférences conditionnelles (notamment pour gérer les compromis) traitées comme des LCP-nets.

A terme, ce travail pourra être vu comme une décomposition en une décision séquentielle et un contrôle flou, puis un couplage entre les deux. Nous pourrions faire collaborer une approche séquentielle, par des algorithmes d'apprentissage par renforcement comme le *Q-learning*, et une approche non-séquentielle fondée sur le contrôle flou et plus précisément l'approche neuro-floue par apprentissage également. Les contraintes étant très fortes, nous espérons, avec ce découplage, arriver à ce que les coûts en calcul pour trouver une solution s'additionnent alors que l'intégration des deux aspects dans une unique approche séquentielle produirait plutôt une combinatoire des deux.

Ces recherches sont en cours et les pistes annoncées en train d'être développées [Dutreilh et al., 2010, Dutreilh et al., 2011].

Ces travaux ont été réalisés en collaboration avec notre collègue Jacques Malenfant. Nous co-encadrons la thèse CIFRE (Orange/LIASD/LIP6) de ***Xavier Dutreilh*** *depuis le 1er octobre 2009.*

3.3.3 Modélisation de besoins pour l'auto-adaptation

On l'a vu, ces différentes problématiques nous amènent à travailler autour des *architectures* et notamment des SOA et des PaaS.

C'est ainsi que nous nous sommes intéressée aux systèmes à composants grande échelle, dans le cadre du projet financé par l'ANR[10] : SALTY (*Self-Adaptive very Large disTributed sYstems* ou "très grands systèmes répartis auto-adaptatifs"). SALTY, projet dont nous sommes responsable pour le partenaire Paris 8/LIASD, est exemplaire du type de situations décrites plus haut. L'objectif central de ce projet est de produire une architecture d'*autonomic computing* capable de gérer et de passer à l'échelle de systèmes répartis de très grande taille.

L'intérêt des traitements linguistiques est, pour nous, d'être capable d'éliciter ce que veut le programmeur ou l'administrateur pour écrire des règles de paramétrage des systèmes. De surcroît, on est en mesure d'exprimer aussi ses préférences, ce qui est évidemment fondamental pour les questions de décision multicritères (quels critères promouvoir et quels critères inhiber ?).

Cas d'utilisation. Pratiquement, le cas d'utilisation qui nous intéresse plus particulièrement est la géolocalisation, où l'objectif général est d'adapter la fréquence à laquelle les mobiles géolocalisés vont signaler leur position de manière à trouver un équilibre entre le coût de ces remontées de position et la précision avec laquelle sont atteints les objectifs métiers des entreprises, comme la vérification du suivi d'un corridor pendant tout le trajet. La fréquence optimale de remontée dépend de nombreux paramètres, comme la proximité du mobile des frontières du corridor, la tolérance accordée dans la détection du franchissement du corridor, le niveau de la batterie du GPS, la précision des mesures de position à cet instant précis, etc. Pour les opérateurs logistiques, l'appréciation linguistique et nuancée de ces éléments dans la décision sur la fréquence est une nécessité, dans la mesure où ils n'ont souvent ni la connaissance ni les moyens d'en déterminer des paramètres numériques précis.

Ainsi, de nombreux *scenarii* sont envisagés, notamment le suivi d'une flotte de camions (i) devant rester dans un corridor et (ii) devant traverser des points de passage (pour péage ou livraison). Deux principaux types d'adaptation sont à considérer :

- les adaptations à bas niveau (par exemple changement dans la fréquence de remontée des informations depuis le boîtier, ou bien changement de moyen de localisation),

10. Agence Nationale de la Recherche.

– et les adaptations à un plus haut niveau (adaptation du modèle de l'application qui capture la dynamique du système en termes d'états, de décisions, de transitions et de fonctions de coût).

A bas niveau, nous conjecturons que l'utilisation de données linguistiques dans les boîtiers programmables comme dans la plateforme de géolocalisation permettra de traiter simultanément un ensemble de cas (à cause du passage d'un univers continu de valeurs à un univers partitionné en intervalles linguistiques) et ainsi, sur ce point, d'aider à passer à l'échelle. Et en délocalisant les décisions dans les dispositifs de plus bas niveau, il est probable que nous obtiendrons des temps suffisamment courts pour gérer de grosses flottes de véhicules (100000 camions).

Nous prévoyons donc de déployer une boucle d'adaptation (appelée "boucle MAPE" pour *manage-analyze-plan-execute*) par objectif métier (par exemple le suivi dans un corridor, la vérification de points de passage, la notification d'arrivée aux points de livraison, etc.) et par camion.

De plus, pour s'affranchir des paramétrages souvent complexes de ces boîtiers, nous proposons une abstraction de l'interface de configuration en offrant une interface homme-machine pour capturer l'expression des besoins *via* une interface de dialogue permettant une gestion de bout en bout des termes linguistiques. Nous pensons ici à l'utilisation de techniques floues — modélisation sous forme de sous-ensembles flous ou de *linguistic 2-tuples* — couplées avec du traitement automatique de la langue. Ainsi, les utilisateurs pourront éliciter leurs buts à la fois au niveau application (par exemple, notifier l'application quand un camion est à 30 minutes d'un point de passage) et au niveau adaptation (par exemple, minimiser le nombre de positions transmises dès lors que la notification est assurée à plus ou moins 5 minutes).

Une telle interface en langage naturel permettra de générer automatiquement de nouvelles interfaces métier, pas nécessairement envisagées à l'avance. Par exemple, un utilisateur peut exprimer un besoin non prévu (suivi d'enfants à la sortie de l'école), qui génèrera ainsi une application métier dédiée (au suivi d'enfants), capable de dialoguer avec la plateforme de géolocalisation et de s'autoconfigurer, de sorte à remonter les informations nécessaires à des fréquences adéquates pour assurer le rôle demandé.

Composants autonomiques. Du point de vue plus architectural cette fois, les applications dont nous parlons doivent, pour être performantes, s'adapter à leur contexte d'exécution qui est souvent sujet à des changements imprévisibles (par exemple des pannes, des fautes, des ruptures de lien réseau, etc.) [Salehie et Tahvildari, 2005]. L'adaptation à ces changements soulève principalement deux problèmes : *comment* l'application doit-elle s'adapter et *quand* doit-elle le faire ?

La question revient donc à écrire un modèle pertinent pour définir une application auto-adaptable, c'est-à-dire une application malléable. Pour ce faire, il existe les approches dites *réflexives* à base de composants réflexifs, c'est-à-dire qui utilisent des langages et des systèmes offrant une *représentation d'eux-mêmes* et des *moyens de manipuler* cette représentation à l'exécution de l'application.

Ces composants autonomiques peuvent donc, par exemple et selon certaines politiques,

s'échanger de la ressource (CPU, disque dur, RAM, etc.), ou plus exactement, peuvent s'échanger des *tickets* d'utilisation de la ressource.

En fonction des situations, différentes politiques peuvent être appliquées : le choix de la meilleure à tout instant nécessite une détection des changements dans le système. C'est cette adaptation aux changements qui est au cœur de nos préoccupations dans ce travail.

Ces recherches ont été commencées il y a un an et demi, le projet SALTY ayant démarré à la fin de l'année 2009. Une page Web est maintenue, indiquant l'avancement des travaux : `http://salty.unice.fr`
Les *scenarii* d'adaptation ont été rédigés (notamment dans le cadre de livrables pour l'ANR), des premiers éléments ont été programmés (des schémas XML ou RNC — Relax NG), des tests sur la plateforme de géolocalisation ont été mis en place, mais il reste le point crucial à traiter : la définition de la boucle d'adaptation, boucle dans laquelle on aimerait que nos hypothèses se réalisent. Pour le moment, nous avons co-écrit un premier article qui pose les bases et les hypothèses de travail [Melekhova et al., 2010].

Des développements logiciels ont été aussi entamés, en particulier, l'ajout de "briques" dans la librairie *jFuzzyLogic* concernant la prise en compte des 2-tuples uniformément (*cf.* page 42) et non uniformément distribués de Martínez *et al.* dans les calculs, et en particulier un travail important sur le passage d'une hiérarchie linguistique floue (*cf.* page 23) à une autre ainsi qu'un partitionnement *ad hoc* pour lequel nous proposons notre propre algorithme, plus générique, et ainsi mieux adapté à notre problème, *i.e.* aux données que nous injectons. Nous avons fait écrire un premier article (accepté à une conférence internationale pour étudiants) sur ce sujet à l'un de nos doctorants [Abchir, 2011]. Par ailleurs, après contacts avec Pablo Cingolani, l'auteur de *jFuzzyLogic*, nous avons amorcé une contribution à la librairie en question, qui inclura donc des algorithmes permettant de manipuler des 2-tuples linguistiques uniformément et non-uniformément distribués, mais aussi une contribution au langage FCL, dans lequel on propose de définir ce nouveau type de données (les 2-tuples en question).

Nous réfléchissons aussi à des partitionnements "intelligents" qui choisiraient automatiquement la meilleure hiérarchie — c'est-à-dire la meilleure cardinalité de étiquettes — lorsque les termes linguistiques issus de l'interface en langage naturel comportent beaucoup (ou aucun) synonyme(s). Ces recherches feront bien entendu l'objet de publications par la suite.

Ces travaux ont été réalisés en collaboration avec nos collègues Jacques Malenfant et Anna Pappa.

Nous co-encadrons avec Anna Pappa la thèse CIFRE (LIASD/Deveryware) de ***Mohammed-Amine Abchir*** *qui travaille surtout sur le cas d'utilisation et nous co-encadrons avec Jacques Malenfant la thèse (LIP6, allocation de l'ANR* via *le projet SALTY) d'****Olga Melekhova*** *qui travaille surtout sur les composants autonomiques. Les deux étudiants sont inscrits en thèse depuis le 1er novembre 2009.*

Chapitre 4

Pistes de réflexion en cours

Sommaire

L'HEURE EST MAINTENANT AU BILAN, mais au sens des travaux en cours et des perspectives de recherche à court terme que ces années de réflexion ont ouverts.

Outre les nombreuses pistes — précédemment évoquées — à explorer avec nos collègues et nos doctorants dans la colorimétrie, les harmonies, les arts de la scène et surtout l'*autonomic computing* (*cloud computing* et auto-adaptation), nous voudrions présenter ici nos dernières avancées et réflexions, non toutes publiées, dans le domaine de la modélisation des préférences (avancées sur les LCP-nets) et dans celui, plus général des théories modélisant les imprécisions (réflexions sur une unification de plusieurs de ces théories).

4.1 Sur le calcul à l'aide de mots dans l'*autonomic computing*

Cette section est un complément au projet d'article présenté en annexe, page 79.

On l'a vu, les LCP-nets peuvent être considérés comme une représentation graphique des préférences conditionnelles mélangeant les UCP-nets (préférences exprimées sous forme d'*utilités*) et les TCP-nets (préférences exprimées sous forme d'*ordres* entre elles et avec des types d'arcs permettant d'exprimer des relations d'importance relative basiques et conditionnelles). Ce travail a donné lieu à une implémentation des LCP-nets (en Java) sous forme de *fragments*. Ces fragments ont été imaginés pour éviter des répétitions systématiques, lors de l'exécution. Les préférences communes ont donc été factorisées dans un LCP-net partagé,

et ce LCP-net subit des variations pour qu'à chaque niveau où c'est nécessaire, on obtienne le LCP-net effectif à utiliser lors de la liaison tardive de services. Ce qui signifie que les préférences peuvent être définies de manière incrémentale, *via* des extensions ou modifications locales, d'où la notion de fragment de préférences qui sont des entités, construites sur la base d'un LCP-net existant, et qui décrivent de manière non ambiguë des ajouts, suppressions, ou modifications de tout élément sur leur LCP-net de base.

Cependant plusieurs questions n'ont pas encore été traitées.

4.1.1 Sur la construction de LCP-nets valides

Les opérations proposées et implémentées ne garantissent pas l'obtention de LCP-nets valides. Par exemple, l'ajout d'un fragment de LCP-net à un LCP-net existant ne manipule pas uniquement des LCP-nets valides. Vérifier *a posteriori* la validité des LCP-nets n'est pas trivial. Une approche alternative consisterait à définir un nouvel ensemble d'opérateurs élémentaires qui, ne manipulant et retournant que des LCP-nets valides, garantirait la construction de LCP-nets finaux toujours valides. Ainsi, un LCP-net atomique valide serait constitué seulement d'un nœud avec sa CPT (*conditional preference table*) associée. Puis, nous définirions les opérateurs élémentaires pour lesquels on préciserait des préconditions, postconditions et des invariants.

Invariants

Les invariants sont les faits suivants :

- le nombre total d'arcs (en incluant les cp-, i- et ci-arcs) doit être cohérent, c'est-à-dire ne dépasse pas le nombre total de liaisons possibles inter-nœud. ;
- il y a exclusion mutuelle des trois types d'arcs ;
- le nombre de dimensions de la CPT associée à un nœud est égal à 1 + le nombre de cp-arcs entrant dans le nœud ;
- la taille de chaque dimension d'une CPT est inférieure ou égale au nombre de valeurs du domaine de la variable linguistique associée ;
- il n'y a pas de cycles conditionnels dans le graphe ;
- il y a au moins un nœud (donc au moins une CPT) ;
- il y a au moins autant de CPT que de nœuds.

Sous ces hypothèses, on peut regarder les différentes opérations qui semblent possibles sur un nœud, un arc et une CPT.

Addition d'un nœud

Implémenter l'opération basique d'addition pour un nœud revient à ajouter un nouvel élément à prendre en compte dans le graphe de préférences.

Formellement, ajouter un nœud correspond à trois actions : créer un nœud, créer un ou plusieurs arcs et créer la CPT associée.

Les préconditions sont les suivantes :

- un nouveau nœud à ajouter ;
- il y a présence d'au moins un LCP-net (ou 2, ou plus, si l'on se sert du nœud pour relier 2 ou plusieurs LCP-nets).

Les postconditions sont les suivantes :
- une nouvelle CPT est attachée au nœud. Cette table est à une ou plusieurs entrée(s) selon le type d'arc associé et selon que le nouveau nœud est origine ou destination de l'arc ;
- (au moins) un nouvel arc apparaît ;
- le ou les LCP-net(s) est (sont) modifié(s).

Autres opérations

La même démarche peut être faite pour définir l'opération basique d'addition d'un arc.

De même, nous avons commencé à définir :
- la soustraction :
  - d'un nœud ;
  - d'un arc ;
  - d'une ligne/colonne (enrichissement ou appauvrissement du choix de qualité de la variable linguistique) dans une CPT ;
- la modification :
  - d'un nœud ;
  - d'un arc ;
  - d'un élément (changement d'avis sur la préférence de la variable linguistique) dans une CPT.

Ce travail pourra nous amener, à plus long terme, à définir une sorte d'*algèbre des LCP-nets*, avec des opérandes et des opérateurs pour manipuler des LCP-nets.

4.1.2 Sur une algèbre des LCP-nets

Une algèbre des LCP-nets permettrait de reprendre cette formalisation dans un cadre théorique plus large et complètement indépendant du cas d'utilisation.

Dans cette algèbre, l'ensemble considéré serait celui des LCP-nets valides (atomiques ou non), et un tel travail nous amènerait à définir un groupe abélien $(SL, *)$ avec SL l'ensemble des LCP-nets valides et $*$ une loi de composition interne (commutative). $*$ serait définie grâce à des opérations (des lois de composition externes) faisant appel à d'autres ensembles que SL, tels que l'ensemble des nœuds, l'ensemble des arcs et l'ensemble des CPT. Ces opérations seraient celles dont nous venons de parler, c'est-à-dire : l'ajout d'un arc à un LCP-net, la soustraction d'un arc, la modification d'un arc, l'ajout d'un nœud, la soustraction d'un nœud, la modification d'un nœud, l'ajout d'une ligne/colonne d'une CPT, etc.

L'*élément neutre* admis par la loi $*$ du groupe abélien pourrait être un LCP-net atomique, tel qu'évoqué plus haut, constitué d'un seul nœud, mais pour lequel aucune préférence n'est

définie. C'est-à-dire avec une CPT à une dimension, dans laquelle les valeurs d'utilité sont toutes identiques.

Pour un LCP-net $\mathcal{L}$, l'*élément symétrique* $\bar{\mathcal{L}}$ admis par $*$ pourrait être le LCP-net $\mathcal{L}$ "inversé", c'est-à-dire dont le nœud racine est le dernier nœud de $\mathcal{L}$ (nœud le plus bas), et ainsi de suite pour les autres nœuds, et dont les valeurs d'utilité dans les CPT sont toutes diagonalement inversées.

Plus concrètement maintenant, la composition (par $*$) de deux LCP-nets serait pertinente dans le cas où l'on souhaite combiner deux (ou plus) réseaux de préférences concernant la même décision. Par exemple, deux utilisateurs doivent utiliser le même service à choisir parmi un ensemble, avec des préférences communes : il serait nécessaire, dans ce cas, de combiner les deux LCP-nets définis par les deux clients afin d'arriver à une décision conjointe.

4.1.3 Sur la question de l'*optimization query*

Pour compléter la définition des LCP-nets, il est nécessaire de considérer la question de l'*optimization query* (ou requête d'optimisation) qui interroge sur les valeurs des entrées à injecter afin d'avoir la plus forte utilité en sortie. Ceci revient à se poser la question suivante : "Etant donné un LCP-net $\mathcal{L}$, est-il possible de déterminer, même virtuellement, une affectation idéale qui serait la meilleure parmi toutes les affectations classées qui satisfont $\mathcal{L}$?"

De façon pratique, répondre à cette question est particulièrement utile dans le cas d'une prise de décision multicritère *séquentielle* où l'on construit itérativement la fonction de coût — à l'aide, donc, d'un LCP-net. A chaque itération, il est nécessaire de connaître les affectations qui optimisent la sortie du LCP-net.

La solution n'est sûrement pas unique et ce problème est loin d'être trivial. En effet, en cherchant à optimiser l'utilité globale d'un LCP-net, on doit optimiser les utilités locales à chaque nœud, mais en prenant en compte la distribution des poids. En faisant l'hypothèse que l'opérateur d'agrégation des poids est une simple moyenne pondérée, et donc que toutes les utilités locales sont égales à 1, un second problème apparaît à cause des inférences dans chaque nœud. On doit inverser les inférences pour remonter vers les valeurs observées pour chaque nœud (c'est-à-dire vers les valeurs des attributs de l'affectation idéale). Evidemment, il n'y a aucune raison pour que les meilleures valeurs des attributs donnent l'affectation idéale.

Cette inversion d'inférence revient à un problème d'abduction. Peirce (1839–1914), célèbre logicien, définissait l'abduction ainsi : dans le cas où "C est vrai si A est vrai", si C est observé (C est appelé la *manifestation*), alors il y a des raisons de penser que A peut être vrai. Depuis, beaucoup de travaux se sont intéressés à ce problème. Miyata *et al.* définissent des relations *cause-et-effet*. Ils essaient de donner une définition de sous-ensembles flous maximum et minimum qui peuvent expliquer les manifestations [Miyata et al., 1995]. Dans notre cas, la manifestation est la meilleure affectation. Revault d'Allonnes *et al.* ont aussi essayé de construire un ensemble d'explications probables pour une manifestation donnée [Revault d'Allonnes et al., 2007], mais ils ont constaté qu'il est très difficile d'étendre les résultats de l'abduction floue formelle à différentes classes d'implication. Un ensemble

d'explications peut être construit seulement pour des implications *cohérentes en termes de déduction*, mais pas pour toutes les implications [Revault d'Allonnes et al., 2009].

Toutes ces études montrent qu'il n'est pas possible de prouver la requête d'optimisation de l'affectation, sans fixer un grand nombre de conditions qui sont les suivantes :

- hypothèses sur les formes de tous les sous-ensembles flous (d'ailleurs, considérer seulement des 2-tuples linguistiques (de Martínez *et al.*) permettrait une grande simplification) ;
- hypothèses sur les choix des opérateurs d'implication ;
- hypothèses sur les choix des opérateurs qui permettent d'agréger les conclusions des règles ;
- hypothèses sur les choix de l'opérateur Δ qui agrège les utilités locales.

Dans notre cas particulier, cette question de la requête d'optimisation permettrait donc de définir les services idéaux, avec les meilleures QoS, permettant d'obtenir la plus grande utilité.

4.1.4 Sur un traitement linguistique de bout en bout

Par ailleurs, si le mécanisme d'inférence ici-présenté consomme des valeurs numériques précises et produit des utilités locales sous forme réelle, une alternative serait d'effectuer une inférence linguistique de bout en bout : un processus entièrement linguistique pourrait en effet être en adéquation plus fine avec les intentions initiales de l'utilisateur qui ont été retranscrites de manière linguistique lors de l'expression des LCP-nets.

Discussion préalable

Les 2-tuples de Martínez *et al.* permettent justement un traitement automatique des déclarations linguistiques théoriquement sans perte d'information.

Le fait qu'il n'y ait pas de perte d'information est dû, bien entendu, au recours à la translation symbolique α déjà évoquée qui impose de concevoir le domaine de définition des termes s_i comme un intervalle de nombres réels, donc ordonné et continu. Ceci permet ainsi, à partir de données linguistiques, de procéder à des calculs numériques et de redonner une réponse linguistique. Dans notre contexte, ceci est tout à fait approprié.

Cependant, un traitement linguistique de bout en bout serait "davantage linguistique" s'il ne faisait jamais intervenir de calculs sur des nombres. On a donc deux façons de traiter le problème.

Un traitement purement symbolique, comme le montre le schéma de la figure 4.1 avec les flèches et rectangles en pointillés gras, serait de n'avoir jamais recours à une échelle de nombres, un peu comme nous avons tenté de le faire avec notre travail sur l'agrégateur symbolique SWM (voir sa définition page 7) qui n'utilise que des compositions de modificateurs symboliques GSM. Mais une supposition forte est tout de même admise dans ce cas : les termes linguistiques sont ordonnés. En revanche, on n'impose pas véritablement l'équidistance entre les termes puisque l'on autorise une subdivision à l'infini de l'échelle, *via* les GSMs. En considérant plusieurs échelles à la fois, on permet la définition de termes

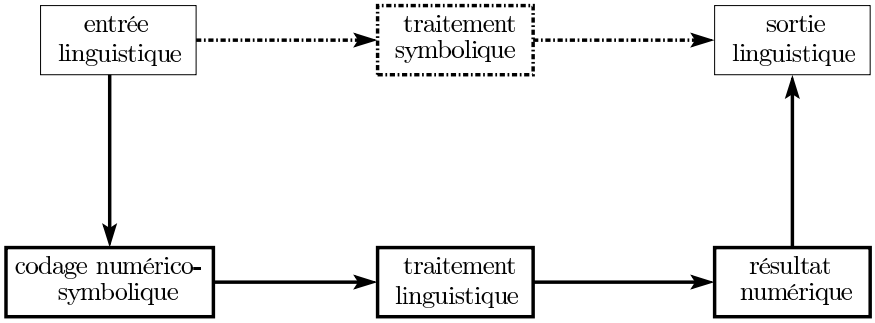

FIGURE 4.1 – Traitement des données entièrement linguistique *versus* traitement des données numérico-symbolique.

non équidistants entre eux — ce qui permet d'ailleurs d'enrichir la variété des données traitables — exactement comme l'ont proposé Martínez *et al.* avec leur modèle des 2-tuples non uniformément distribués sur leur axe de définition (*cf.* la figure 4.2 où trois hiérarchies à 3, 5 et 9 étiquettes sont utilisées).

Avec un traitement numérico-symbolique (flèches et rectangles pleins et gras dans la figure 4.1), qui correspond donc à la méthode utilisant les 2-tuples, on obtient un résultat qui est nécessairement plus précis puisqu'il ne nécessite pas les approximations successives pouvant générer du biais et de l'imprécision qu'imposerait un traitement purement symbolique. Par ailleurs, tant qu'on n'a pas de véritable raisonnement *ad hoc* pour les 2-tuples [Alcalá et al., 2007] (de modus ponens généralisé pour 2-tuples), le résultat sera conforme à celui où l'on applique le modus ponens généralisé de Zadeh en considérant uniquement des sous-ensembles triangulaires.

En fait, les 2-tuples, lorsqu'ils expriment une variable V, expriment un *décalage sur l'axe des abscisses*, alors que l'inférence de Zadeh permet, à chaque règle (pour chaque conclusion), d'exprimer un sous-ensemble flou *non normalisé*, qui vaut exactement V, mais avec une valeur de vérité affaiblie ($\forall x \in X, f_V(x) \leq 1$). Ainsi, une "vraie" inférence de 2-tuples générera, pour chaque règle, une conclusion "décalée" ($\alpha \in [-0.5, 0.5[$) et non pas "affaiblie". Puis, en cherchant à exprimer la solution finale de l'inférence (agrégation des conclusions des règles), on procédera à une agrégation de valeurs toutes aussi vraies les unes que les autres.

Au fond, l'inférence de Zadeh affirme que, pour chaque règle, la conclusion est éventuellement "un peu vraie", mais sans préjuger des valeurs des conclusions voisines. Tandis que l'inférence 2-tuples, pour être cohérente avec la définition-même des 2-tuples, doit nécessairement faire ce préjugé en affectant une valeur non nulle à α. Car c'est là le seul moyen d'exprimer que la valeur n'est pas exactement V.

Notons ici que nous ne nous plaçons pas dans une définition purement symbolique ou purement qualitative, comme le proposent Dubois *et al.* en revisitant l'approche de Savage [Savage, 1954], approche qui fait l'hypothèse que le classement des alternatives (encore appelées "actes") doit être fait en fonction de l'utilité attendue des conséquences des alternatives en question [Dubois et al., 2002, Dubois et al., 2003]. Nous nous plaçons au contraire dans l'hypothèse où l'utilité et l'incertitude sont proportionnées, c'est-à-dire que les deux

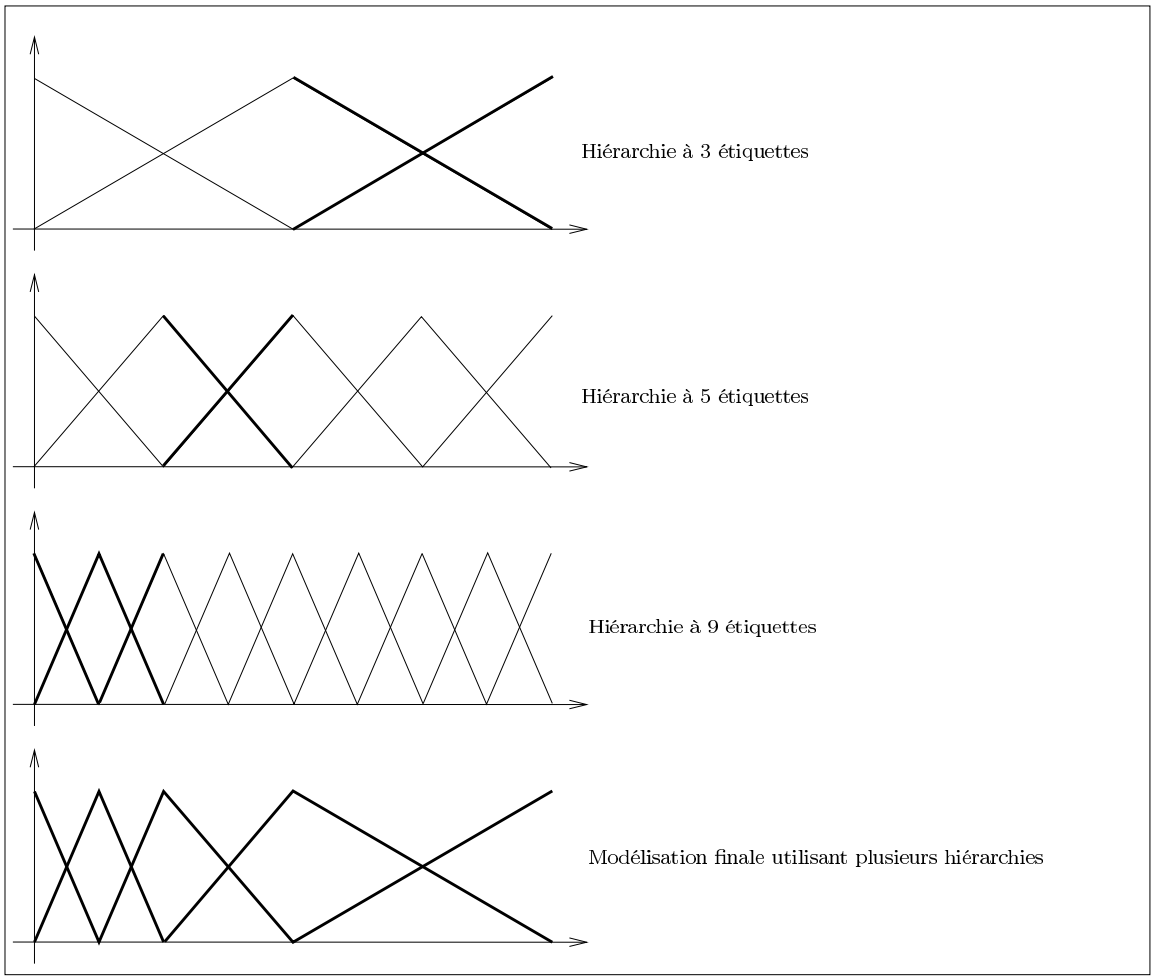

FIGURE 4.2 – Expression de 2-tuples non uniformément distribués.

échelles qui leur sont associées font partie d'une échelle plus grande, et les degrés d'incertitude peuvent être comparés aux degrés de préférence.

Vers une réelle inférence avec des 2-tuples linguistiques ?

Ainsi, si l'on considère les valeurs des propriétés de QoS des services comme fournies sous la forme de 2-tuples, un mécanisme *ad hoc* d'inférence pour 2-tuples tel que nous venons de l'évoquer pourrait être étudié et implanté.

Il nécessiterait notamment l'utilisation de plusieurs opérateurs spécifiques d'agrégation linguistique [Xu, 2008], fondés sur les opérateurs OWA de Yager [Yager, 1988]. Il faudrait que ces opérateurs ne travaillent que sur les valeurs s_i et α, donc avec des valeurs de l'axe des abscisses uniquement.

Ce travail rejoint en partie la problématique de la thèse de Mohammed-Amine Abchir, où un traitement intégralement linguistique des données, couplé à une IHM en langage

naturel est envisagé (cf. section 3.3.3). Dans ce cadre et pour approfondir cette question de l'inférence avec 2-tuples, nous prévoyons de recevoir en 2011/2012, comme chercheur invité, Rocio de Andrés Calle, maître de conférences à l'université de Salamanque en Espagne, qui a travaillé avec Martínez et son équipe.

Une autre piste intéressante est de comparer la rapidité des calculs si l'on considère uniquement des 2-tuples au sein des LCP-nets (donc avec une inférence 2-tuples), c'est-à-dire avec un traitement linguistique de bout en bout. En effet, lors de l'utilisation des 2-tuples (que ce soit le formalisme d'Herrera et Martínez, de Wang et Hao ou de Truck et Akdag), on s'affranchit, en tous cas, dans les calculs, des sous-ensembles flous, tout en gardant une sémantique linguistique attachée aux objets manipulés. Il faudrait donc voir si ces calculs numériques sont moins lourds que ceux associés aux sous-ensembles flous. Cela semble probable vu que seuls des calculs sur des indices, ou bien des implications de type $min(1-x+y,1)$ sont réalisés, alors que le traitement avec manipulation de sous-ensembles flous de bout en bout implique des calculs d'intersection de droites (si les sous-ensembles flous sont trapézoïdaux), qui peuvent être considérés, dans l'absolu, comme plus longs.

Si cette piste s'avère intéressante, elle sera sans doute exploitée aussi dans le cadre des thèses d'Olga Melekhova et de Xavier Dutreilh.

4.1.5 Conclusion

Les dernières collaborations (autour des architectures logicielles, en particulier) que nous avons été amenée à mettre en place nous ont ouverte à des préoccupations que l'on pourrait peut-être qualifier de pragmatiques (nous pensons notamment aux contraintes de mise en œuvre du modèle), mais qui nous obligent en fait à revenir sur la définition-même du modèle. Ainsi, en implémentant les LCP-nets, nous nous sommes heurtée au problème de réutilisation de LCP-net existant : en effet, il était important de considérer le LCP-net comme une structure de données modulable, et non pas comme un bloc indissociable, afin de pouvoir en dupliquer éventuellement certaines parties. C'est pourquoi nous avons envisagé la notion de *fragments* de LCP-nets évoquée à la section 4.1. Mais cette façon d'envisager le "découpage" d'un LCP-net a nourri une autre réflexion : celle autour de la définition de LCP-nets valides et de l'algèbre des LCP-nets. Ainsi, deux visions différentes ont été mises en évidence pour un même problème :

- la vision "fragments", permettant une implémentation efficace du problème et correspondant à une décomposition temporelle des préférences ;
- la vision "algèbre", permettant une vue tournée "opérandes et opérateurs" et correspondant à une décomposition spatiale car les données du problème (ici, les préférences) doivent être utilisées conjointement pour prendre une décision.

La figure 4.3 montre ces deux visions en procédant par analogie : dans le premier cas, on compose un objet (ici, un véhicule militaire) avec un fragment d'objet (ici, une arme) et l'on obtient un nouvel objet (ici, un véhicule militaire équipé). Dans le deuxième cas, on compose un objet avec un autre objet et l'on obtient un nouvel objet ayant certaines caractéristiques du premier objet et certaines caractéristiques du deuxième objet.

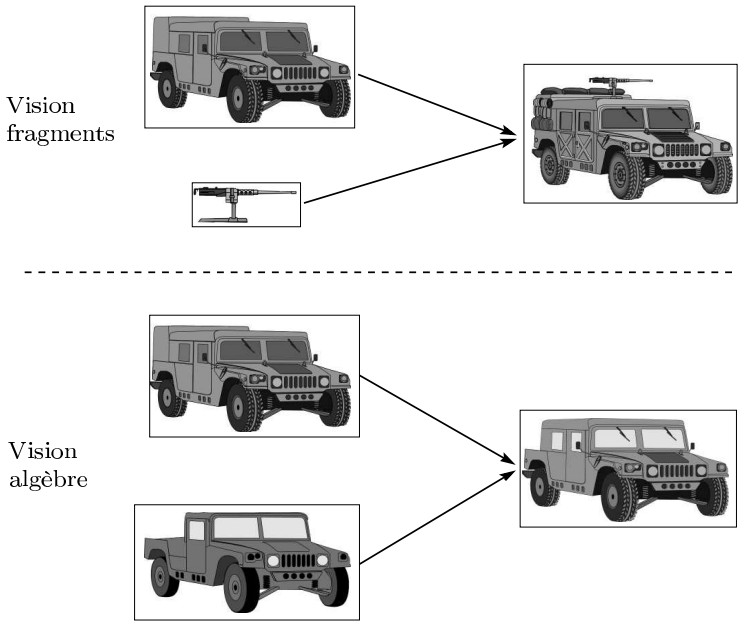

FIGURE 4.3 – Deux visions, selon le point de vue envisagé.

Concernant le travail de thèse de Xavier Dutreilh sur la définition d'un modèle (flou) de machine virtuelle (cf. page 56), en privilégiant la vision "algèbre", on pourra "additionner" une machine virtuelle (plus exactement, ses caractéristiques) très performante pour un certain type de tâche avec une machine virtuelle très performante pour un autre type de tâche. Et en privilégiant la vision "fragments", on pourra "modifier" une machine virtuelle avec des "fragments" de machine pour obtenir la machine virtuelle recherchée.

Concernant le travail de thèse d'Olga Melekhova sur la mise en place d'un modèle d'application malléable, on pourra là encore employer la vision "algèbre" en considérant comme objets à manipuler les (caractéristiques des) composants de l'application, ou bien la vision "fragments" en considérant l'application recherchée comme étant la "modification" d'une application de base (adjonction de composants *ad hoc*, etc.).

4.2 De l'unification des modèles linguistiques ?

Après plus de dix ans de recherche dans ces différents domaines, il apparaît comme évident que certains des modèles linguistiques (en tous cas, leur cadre théorique) avec lesquels nous avons travaillé peuvent être comparés, voire unifiés.

4.2.1 Une caractéristique commune

D'abord, toutes ces notations partagent une caractéristique essentielle qui consiste à réussir à exprimer les sous-ensembles flous du domaine du discours en fonction des sous-ensembles flous de l'utilisateur, ce qui est fondamental pour le retour vers l'utilisateur pour

lui donner une appréciation qualitative des résultats des calculs, donnant ainsi tout son sens à une approche linguistique de bout en bout. Compte tenu de la caractéristique importante de ces modèles, une approche unifiée peut en exploiter les complémentarités à au moins deux titres : pour exprimer qualitativement les résultats des calculs, mais aussi pour passer facilement d'un modèle à l'autre, puisqu'il peut être, au cours des calculs, plus simple d'utiliser un certain modèle plutôt qu'un autre.

Concrètement, si l'on considère la modélisation par 2-tuples linguistiques (Herrera & Martínez), ou par 2-tuples proportionnels (Wang & Hao), ou encore la nôtre (modélisation par GSM), on remarque qu'elles ont toutes trois un point commun : celui d'être exprimable sous forme vectorielle comme une base $(\alpha\overrightarrow{v} + \beta\overrightarrow{u})$. Le vecteur $\alpha\overrightarrow{v}$ peut être considéré comme une indication grossière de la valeur linguistique à représenter, et le vecteur $\beta\overrightarrow{u}$ comme un raffinement de cette première évaluation (cf. figure 4.4). $\overrightarrow{u}$ est un vecteur unitaire.

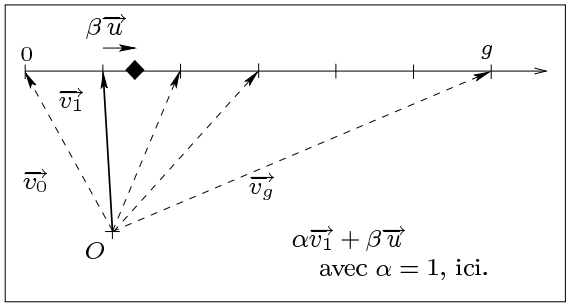

FIGURE 4.4 – Représentation vectorielle proposée pour l'unification.

Nous avons publié un article sur ce sujet [Truck et Malenfant, 2010] qui est inséré en annexe, page 95, dans lequel nous redéfinissons chacun de ces trois modèles selon notre notation vectorielle. Nous utilisons cette même base que nous contraignons d'une façon ou d'une autre selon le modèle exprimé. Cela revient à imposer des conditions sur α, β, $\overrightarrow{v}$ et $\overrightarrow{u}$. Nous obtenons ainsi trois bases contraintes.

4.2.2 Correspondance entre les valeurs exprimées dans les différents modèles

Il reste ensuite la question (non publiée) de la correspondance entre les valeurs exprimées dans les différents modèles. En effet, pour que l'unification fasse sens, il est nécessaire de pouvoir passer d'un modèle à l'autre, donc d'exprimer une même valeur dans les trois modèles. Pour ce faire, parmi d'autres solutions (comme celle de l'utilisation d'un univers "pivot" — $\mathbb{R}$ chez Wang & Hao), on peut penser à l'utilisation — et d'abord l'établissement — de *matrices de passage* pour les changements de base. Pour pouvoir passer d'une base contrainte à une autre, il faut que la matrice contienne en colonnes les coordonnées des vecteurs de la base cible exprimées dans la base source. On construit ces matrices en leur injectant des expressions redondantes de la base source dans la base cible pour outrepasser les contraintes quand c'est nécessaire, mais il faut aussi mettre des contraintes croisées additionnelles pour forcer l'unicité de la représentation dans la base cible.

On le voit, le sujet est loin d'être épuisé, puisqu'il ne s'agit pour l'instant que de balbutiements. En effet, le lien entre les trois modélisations a été établi, mais il reste en particulier la définition des agrégateurs à unifier (seul un agrégateur — le nôtre, la médiane symbolique pondérée — a été réécrit dans cette base vectorielle). Et peut-être ces travaux pourraient-ils mener à une axiomatique de ces trois modélisations unifiée ?

Chapitre 5

Conclusion et perspectives

A L'ISSUE de ces années de recherche, on peut maintenant dresser un bilan sous forme de résumé des intentions qui nous ont toujours animée.

Bilan

Le *Computing with Words*, bien que directement issu de la théorie des sous-ensembles flous qui a maintenant plus de 40 ans d'existence, est un domaine toujours très largement ouvert à la recherche. En effet, un des rêves des êtres humains depuis bien longtemps est de créer des machines *véritablement* dotées d'intelligence. Le formidable engouement pour cet axe de recherche est retombé depuis les échecs[11] de l'intelligence artificielle dans les années 70-80, notamment marqués par le livre de Hubert Dreyfus *What Computers Can't Do* en 1972, et l'on sait maintenant que le robot "copie conforme de l'homme" n'est pas pour demain. En revanche, aider l'humain dans sa prise de décision, simuler son raisonnement ponctuellement, dans des contextes bien définis, est une branche qui a prouvé sa faisabilité et son efficacité, et a donc suscité de nouveaux engouements. La théorie des sous-ensembles flous puis le *Computing with Words* avec ses diverses représentations des données linguistiques, a largement contribué à relancer cette branche de l'intelligence artificielle.

Dans ce mémoire, nous avons vu comment le "calcul à l'aide de mots" peut intervenir à différents niveaux.

Du point de vue pratique, beaucoup d'applications ont bénéficié et peuvent encore bénéficier des approches linguistiques. Citons pour mémoire — et pour ne citer que ce que nous avons personnellement abordé — la colorimétrie, les arts de la scène et l'*autonomic computing* dans lesquels nous avons pu déjà expérimenter et observer des résultats très satisfaisants.

Du point de vue théorique, nous avons vu deux axes de travail :
- celui qui consiste à formaliser (pour tenter de généraliser et donc de réutiliser) de nouveaux concepts issus de l'introduction des approches linguistiques dans des appli-

11. *Cf.* par exemple l'échec du *General Problem Resolver* en 1967, et, plus tard, du système expert MYCIN, qui avait pourtant suscité beaucoup d'intérêt.

cations (par exemple, formaliser les LCP-nets), autrement dit, celui qui *se nourrit* des applications,
- et celui qui consiste à mieux asseoir des concepts théoriques existants (par exemple, imaginer un modus ponens généralisé, des implications, des t-normes, t-conormes, etc. pour 2-tuples linguistiques) ou à en inventer de nouveaux (par exemple, imaginer une unification de plusieurs modèles linguistiques *via* une représentation vectorielle), autrement dit, celui qui *nourrit* des applications.

Nous avons donc montré dans ce rapport l'articulation entre théorie et pratique autour des approches linguistiques pour la modélisation et le raisonnement en milieu imprécis, l'une et l'autre se nourrissant mutuellement.

En faisant maintenant le bilan du point de vue du *perceptual computing*, nous avons montré qu'il peut bénéficier des quatre axes de travail suivants, qui restent encore, au moins partiellement, des objectifs :

- la mise en place de traitements linguistiques de bout en bout,
- l'utilisation de différents modèles de représentation — comme les sous-ensembles flous, les 2-tuples linguistiques d'Herrera & Martínez, les 2-tuples proportionnels de Wang & Hao ou encore les degrés linguistiques de Truck & Akdag — qui permettent de capturer les nuances et les résultats de calculs sur la base de domaines de définition linguistiques de cardinalité limitée et proche de la perception des utilisateurs,
- la mise en œuvre d'outils linguistiques allant *au-delà* de l'inférence floue bien connue, les LCP-nets avec leur gestion des préférences conditionnelles présentant un excellent exemple à cet égard, et
- l'intégration de ces outils au sein de processus d'élicitation de la connaissance et de manipulation de ces processus, notamment pour les composer, de manière à supporter des raisonnements plus complexes comme l'inférence conjointe dans le cas de décision collaborative.

En écho au schéma de la figure 2.11, page 20, nous indiquons pour finir les différentes avancées notables de notre travail (*cf.* Figure 5.1). PF fait allusion au partitionnement expliqué en page 38 ; P2t évoque l'algorithme de partitionnement pour 2-tuples dont nous parlons rapidement page 59 ; M2t, l'inférence pour 2-tuples, apparaît en pointillés car ce travail n'est pas encore réalisé, même si une discussion est amorcée en page 67 ; L fait référence aux LCP-nets et U à l'unification amorcée et expliquée en page 69. On constate donc que l'on s'est efforcé de proposer des contributions dans toute la chaîne de traitement.

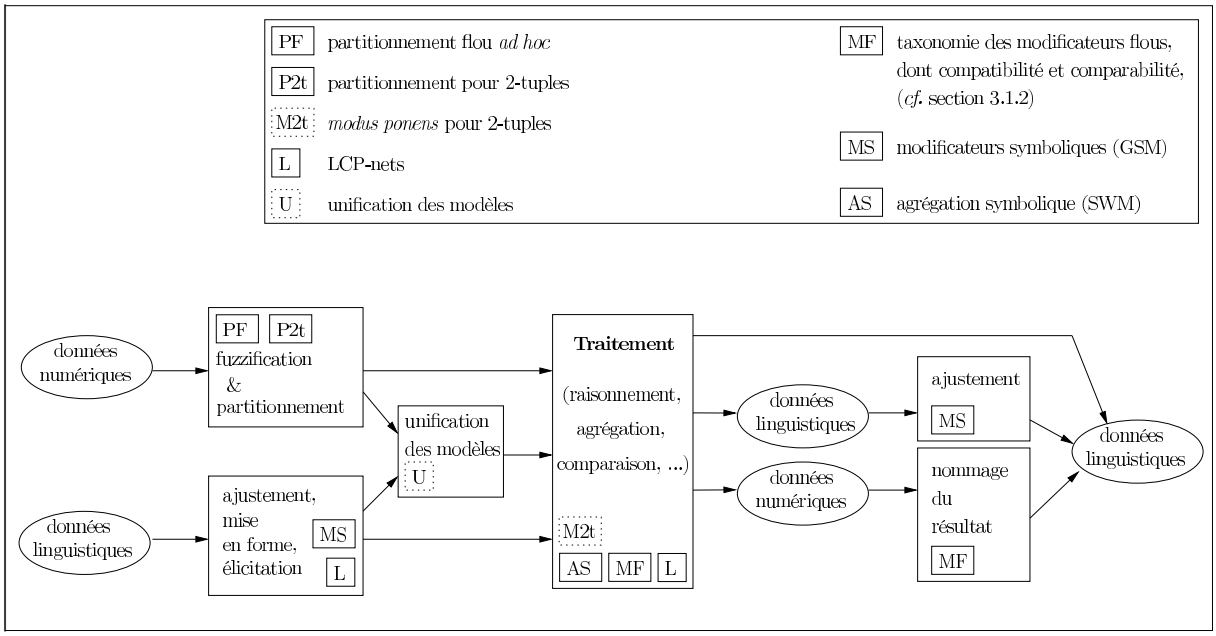

FIGURE 5.1 – Schéma général contenant les travaux réalisés jusqu'alors ainsi que ceux en cours (rectangles en pointillés).

Perspectives

Que ce soit dans le domaine de la colorimétrie, dans celui des arts de la scène, ou plus récemment dans celui du calcul auto-régulé, s'expriment des besoins auxquels le "calcul à l'aide de mots" pourra répondre dans la mesure où les objectifs précédemment évoqués seront pleinement atteints. Combinant processus de décision avec des systèmes répartis complexes, de grande taille (thèse de Xavier Dutreilh), voire à grande échelle (thèses d'Olga Melekhova et de Mohammed-Amine Abchir), le calcul auto-régulé pose en effet de nouveaux défis à la fois en décision et en génie logiciel.

Du point de vue décision dans lequel les approches linguistiques jouent un grand rôle, notons les questions suivantes :

- comment intégrer de manière simple et efficace les critères et objectifs métiers dans la *définition* ou le *calcul des politiques de décision* ? Par exemple, étant par nature multicritères, les problèmes de décision en calcul auto-régulé nécessitent la prise en compte des préférences métier à l'aide de formalismes à la portée des utilisateurs de ces systèmes. La thèse de Pierre Châtel a constitué un premier travail dans ce sens qu'il faut poursuivre ;
- comment *prendre en compte les paramètres effectifs* du comportement des applications dans la détermination des politiques de décision ? Par exemple, le contexte de déploiement des applications n'étant connu que tardivement, le calcul auto-régulé est le domaine privilégié des approches d'apprentissage de politiques en ligne (apprentissage par renforcement, contrôle neuro-flou, ...) ;

- comment s'assurer, à l'exécution, de la *qualité du processus de décision*, qui doit converger, autant que possible, vers une politique optimale et stable ? Par exemple, des expérimentations menées dans le cadre de la thèse de Xavier Dutreilh ont confirmé la grande sensibilité de la stabilité des politiques de décision aux paramètres de latence de la performance dans la prise en compte de nouvelles ressources allouées aux applications. Un défi reconnu du domaine est de détecter et corriger automatiquement de tels effets ;
- enfin, le comportement des systèmes répartis évoluant constamment, le processus de décision doit non seulement calculer une politique optimale mais également la faire *évoluer de concert avec les évolutions du système contrôlé*.

Ces problèmes sont abordés dans un contexte où il est impératif de prendre en compte les problématiques spécifiques du génie logiciel à définir pour la mise en œuvre des applications auto-régulées :

- comment *intégrer* les processus de décision dans une architecture logicielle claire, facile à programmer, à mettre au point et à maintenir ?
- comment *exprimer* les processus de décision sous la forme d'entités logicielles, par exemple des composants autonomiques qui soient *réutilisables*, *composables* et *interchangeables* ?
- comment *tenir compte des contraintes d'exécution*, comme les latences de communication entre les composants contrôlés et contrôlants (ceux qui appliquent les processus de décision) ?

Loin de n'être que des considérations de mise en œuvre, ces problématiques induisent en réalité une véritable dialectique entre les deux domaines — comme c'est le cas avec de nombreux autres domaines — telle solution logicielle répondant au besoin de voir différentes approches de décision comme interchangeables en fonction des besoins, alors que telle contrainte d'exécution exige d'être prise en compte dans le processus de décision. Par exemple, la composabilité des composants autonomiques (ou contrôlants) ne saurait être réalisée sans arriver à *composer les processus de décision* qu'ils représentent, nécessitant donc la définition de tels mécanismes de composition au niveau des modèles de décision.

Devant un aussi vaste contexte, il n'est bien sûr pas question de tout couvrir. Cependant, nos travaux et leurs perspectives permettent de répondre à certaines de ces problématiques. Le développement d'une algèbre de composition des LCP-nets doit en effet permettre de combiner les préférences métier de différents composants autonomiques contrôlant des composants applicatifs (gérés) appartenant à deux utilisateurs mais qui coopèrent, peut-être momentanément, à la réalisation d'une certaine tâche. Combiner les préférences permet ici une décision qui optimise conjointement la décision en fonction des préférences des deux utilisateurs.

Nos perspectives de recherche trouvent de nombreuses justifications dans ces besoins. Les avantages d'une approche linguistique de raisonnement fondée sur les perceptions des utilisateurs ne sont plus à démontrer en ce qui concerne l'expression des connaissances et la capacité d'expliquer les raisonnements obtenus grâce aux "calculs à l'aide de mots". Néanmoins, assurer un traitement linguistique de bout en bout permet d'*étendre* ces avantages à *l'ensemble du processus*, y compris l'interprétation de tous les résultats intermédiaires et finaux. L'utilisation des représentations de type 2-tuples (linguistiques, proportionnels ou

degrés linguistiques) paraît à ce titre une bonne approche, en permettant de garder tout au long des raisonnements la *référence aux termes définis par l'utilisateur*. L'inférence à partir de sous-ensembles et de règles flous permet déjà cela, c'est pourquoi il est souhaitable que l'inférence plus complexe à partir d'outils comme les LCP-nets le permette aussi. D'où notre objectif de compléter ces derniers pour assurer un traitement linguistique intégral sur la base de 2-tuples.

Mais nous sommes également convaincue que cette famille de représentations 2-tuples permettra aussi d'aborder de manière productive les problématiques de réutilisation et de composition. Par exemple, l'utilisation d'une approche avec 2-tuples multi-granulaires (nous pensons ici aux 2-tuples non uniformément distribués, ou encore à nos degrés linguistiques munis de leurs modificateurs symboliques généralisés) ouvre la possibilité de réutiliser des bases de règles pour augmenter (respectivement diminuer) leur précision en introduisant de nouveaux termes tout en préservant les anciens (respectivement en supprimant certains termes parmi ceux qui sont disponibles).

Par ailleurs, le travail amorcé sur l'unification des différents modèles linguistiques (que nous avons donc tous baptisés "2-tuples" afin d'insister davantage encore sur leurs ressemblances) et exposé en section 4.2, doit permettre d'enrichir les possibilités de traitement de l'information multi-sources. En effet, en proposant une interprétation vectorielle de ces modèles et une conversion des valeurs entre modèles par une approche de changement de base, nous permettrons en particulier la composition de base de règles et de LCP-nets définis à partir de représentations 2-tuples différentes.

Ce travail peut être considéré comme original en ce sens qu'en unifiant les différents modèles, on ne cherche pas, comme cela s'est déjà fait dans d'autres travaux, à définir un dénominateur commun et à ramener tous les modèles à celui-ci. Nous souhaitons au contraire conserver l'hétérogénéité (et la richesse) des modèles et cherchons à établir des ponts d'un modèle à l'autre. Ainsi, nous permettons de choisir le meilleur (le plus simple, le plus adéquat) modèle pour les calculs, tout en étant capable, à tout moment au cours de la phase de calculs, de revenir à l'un des modèles, selon le choix effectué.

Nous pensons, que de cette façon, les prises de décision seront enrichies car pouvant s'affranchir de la différence des modèles utilisés, permettant même de *composer les processus de décision* en dépit du fait que ceux-ci utilisent des modèles différents.

Annexe A

Article en cours de soumission

Ce qui suit est un article soumis à la revue IJAMS (International Journal of Applied Management Science) sur la formalisation de nos LCP-nets. Nous définissons formellement les objets manipulés (nœuds, arcs et tables de préférences) et explicitons les règles de calcul pour la génération des utilités locales puis globales.

De surcroît, nous discutons des propriétés souhaitables pour un LCP-net. D'abord, nous définissons le test de dominance pour les LCP-nets et le prouvons. Ensuite, nous évoquons la propriété d'optimisation de l'affectation (affectation idéale). Cette dernière propriété nécessite de nombreuses hypothèses pour pouvoir être prouvée afin de réduire le champ des possibles.

Towards a formalization of the LCP-Nets

Mars 2011

NB : Ceci est un projet d'article sur une formalisation des LCP-nets soumis à la revue IJAMS (International Journal of Applied Management Science).

Abstract

In recent works, we have proposed a graphical model to represent linguistic preferences called LCP-nets. LCP-nets have been implemented and used in a specific use case of industrial engineering. In this paper, we consolidate this contribution in formalizing it through a set of notations and computation rules in order to guarantee its durability and its reusability to other multi-criteria decision contexts. A search algorithm is given and a discussion is proposed about the classical properties that such graphical models must have: dominance testing and optimization queries.

Keywords. Preference networks; fuzzy preferences; CP-nets; LCP-nets; multi-criteria decision making; dominance testing query; optimization query.

1 Introduction

Optimizing complex systems is one of the main points of industrial engineering. This optimization requires lots of tools from various disciplines such as mathematics, management science or artificial intelligence, and decision making is often seen as a central problem to be solved.

In service-oriented architectures (SOA), softwares are conceived as business processes that use available services to perform their computations. In this context, services appear and disappear regularly and those that perform a same computation but with a different quality of service (QoS) and at a different cost, are in competition. So the main problem is to select the best service at any time. To do so, services are first chosen according to their offer and, at the very last moment, according to their QoS and to the preferences. This is called the *late binding*. So the decision making in this context is a multi-criteria optimization problem that has to satisfy the users preferences.

The preference modeling and their elicitation have attracted widespread attention for many years. Several formalisms have been proposed to express the choices or wishes. Among them are the *factored models* that decompose preferences. *Additive models* are a subclass of factored models. Their principle is that the preference values on attributes may be expressed independently of the others. It is the *ceteris paribus* principle [Braziunas and Boutilier, 2006]. This avoids to ask the user for the comparison of all the attributes which would require the joint instantiation of all attributes.

The *generalized additive independence model* (GAI) allow for an additive decomposition (preferences are *added* instead of being multiplied) of a utility function. In [Boutilier et al., 2004], the authors proposed a graphical and compact additive model called *CP-nets* (Conditional Preference networks) that links wished attributes to preferences. This model is a graph and is quite intuitive.

It has three elements: *nodes* that represent the problem variables, *arcs* (or *cp-arcs*) that carry preferences among these variables for various given values, and *conditional preference tables* (CPTs) that express preferences on values taken by the variables.

The CP-nets permit to model wishes such as "for property X, I prefer the V_1 value instead of V_2 if the property Y equals to V_Y and the property Z equals to V_Z". There exists also a notion of *relative preference* between the preferences themselves: a CPT associated to a node has a higher priority than the CPTs of its descendants. This notion is taken into account when the full outcomes are compared. A full outcome is a tuple of values for all the variables of the graph. One of the interests of the CP-nets is their ability to approximate preferences easily with inference rules that will be nothing else than the CPTs. But the CP-net must obey some restrictions in order to allow ("algorithmically" speaking) the inference computations. The first restriction is that the graphs must be acyclic. The second one implies a "reasonable" use of the indifference relation between the preferences and so it implies total preorders in the CPTs for each parent node [Boutilier et al., 2004].

The *Utility CP-nets* (UCP-nets) [Boutilier et al., 2001] are inspired by the CP-nets but the definition of the binary relation $\succ$ ("is preferred to") between two values of nodes in the CPTs is replaced by numerical values (utility factors). Thus the CPTs contain numerical values. The recourse to values instead of order relations has been motivated by the fact that, in a CP-net, it was not possible to establish a comparison nor an order between the alternatives given as solutions to the problem (when there was more than a unique solution). By quantifying preferences, this problem becomes less important [Boubekeur and Tamine-Lechani, 2006]. A utility factor is real number associated to a node affectation given the affectation of its parent nodes. It expresses a preference degree between several affectations. There are *local* utility factors that indicate choices local to a node and *global* utility factors computed for each full outcome that permit to order the solutions without any ambiguity. A UCP-net indeed defines a total order on the outcomes.

Another model inspired by the CP-nets is the *Tradeoffs-enhanced CP-nets* (TCP-nets). It allows to manage tradehoffs in the expression of the preferences [Brafman and Domshlak, 2002]. TCP-nets deal with linguistic expressions such as:"a better affectation for X is more important than a better affectation to Y". These are called *relative importance preferences*. Moreover, TCP-nets deal with *conditional* relative importance preferences: "a better affectation to X is more important than a better affectation to Y if $Z = z$". Thus, new elements are introduced in the model: the *Conditional Importance Tables* (CITs) and two new kinds of arcs: *i-arcs* and *ci-arcs*. These arcs permit to model *basic* and *conditional* clauses of relative importance.

However these models (CP-nets, UCP-nets, TCP-nets) have two important restrictions. The first one concerns the continuity of the variable definition domains. Only discrete domains are handled. The second one is about the difficulty to obtain precise utility values from the users.

That is why we proposed in recent works an alternative to these models, the *linguistic CP-nets* (LCP-nets) that can deal with linguistic clauses and that can take into account values defined on continuous domains [Châtel et al., 2008, Châtel et al., 2010b]. This model has been used in a specific use case to perform late binding between services consumers and service providers. But in order to make this model generic, a formalization is needed.

This paper lays the foundations of a formal definition of the LCP-nets. In section 2, we recall our tool and then give a concrete example in the SOA domain. Section 3 exhibits the preliminary notations of the LCP-nets and the way to compute the global utility factors is detailed in section 4. Thus, some interesting properties are explained and, in particular, dominance testing property is proved in section 5 while section 6 concludes this study.

2 LCP-nets as a tool for service selection

In the LCP-nets, we partition the continuous domains using linguistic terms associated to fuzzy subsets [Zadeh, 1965] or to linguistic 2-tuples [Herrera and Martínez, 2000]. Thus the utility factors are *words*. This allows for an easier way to capture the consumer wishes and for a better ordering of the services that can be proposed to the consumer. Indeed if two services have more or less the same QoS values, a use of discrete coarse-grained domains will prevent a ranking between them. Except if the granularity is increased and if the differences between the preferences are high enough — this is actually an explicit condition in the UCP-nets — to allow for a discrimination.

LCP-nets, as the other models from the "CP-net family", are acyclic graphs with nodes, arcs and preference tables. Linguistic descriptors must first be chosen to describe the term sets on each universe of discourse. As usual we take term sets with an odd cardinality (5, 7 or 9) [Delgado et al., 1993] in order to have a mid-term. For example, a term set T is: $T = \{s_0 : \textit{very low}, s_1 : \textit{low}, s_2 : \textit{medium}, s_3 : \textit{high}, s_4 : \textit{very high}\}$. It is also required to have the three following operators:

1. $\mathrm{Neg}(s_i) = s_j$ such that $j = g - i$ (with $g + 1$ being the cardinality),
2. $\max(s_i, s_j) = s_i$ if $s_i \geq s_j$,
3. $\min(s_i, s_j) = s_i$ if $s_i \leq s_j$.

In the linguistic 2-tuple model of Herrera and Martínez, trapezoidal or triangular fuzzy subsets are enough to express the imprecision of the clauses. Given a linguistic term, the 2-tuple formalism provides for a pair (fuzzy set, symbolic translation) $= (s_i, \alpha)$ with $\alpha \in [-0.5, 0.5[$ as can be seen in Figure 1 where the obtained 2-tuple is $(s_2, -0.3)$.

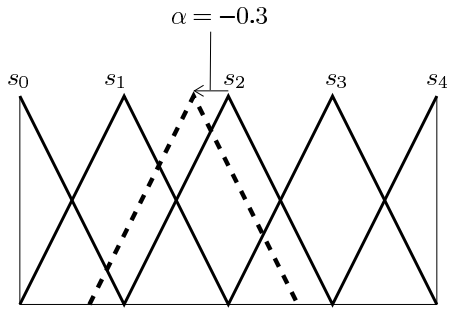

Figure 1: Lateral displacement of a linguistic label $\Rightarrow (s_2, -0.3)$ 2-tuple.

In this example, the α translation can be seen as a weakening modifier of the linguistic term s_2. Thus, using this model for the computations one can give a result more or less equals to one of the elements *from the original term set*, *i.e.* the same linguistic term set can be kept during the whole process.

Compared to the CP-, TCP- or UCP-nets, LCP-nets allow to deal with clauses such as "I prefer the *more or less* V_1 value for property X over *exactly* V_2 if properties Y equals *approximately* V_Y and Z equals *a bit more than* V_Z". These statements that resemble improved fuzzy rules must be interpreted in a context where the global preference on X has to take into account each preference to be applied to Y to a certain degree.

Actually this is equivalent to propose a flexible and intuitive model to express complicated sets of fuzzy rules that can be potentially interdependent.

The LCP-nets allow the consumers to express relative importances (conditional or not) and tradeoffs among the variables in using i-arcs or ci-arcs from the TCP-nets in addition to the cp-arcs from the CP-nets. They include CPTs similar to those from the UCP-nets but with linguistic utility factors. Figure 2 sums these possibilities up: the service proposed is a camera (for example a surveillance camera). The consumer preferences are to obtain the images from the camera as fast as possible. Three QoS properties are used: *security* S, *bandwidth* B and *image resolution* R. The consumer always prefers to optimize B in comparison to S and, if the bandwidth is low, he prefers low-resolution images in order to obtain them as fast as possible.

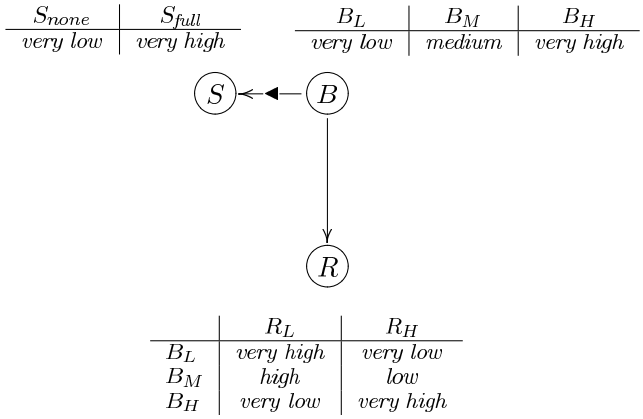

Figure 2: Example of preferences for a service offering a camera fonctionality.

In the SOA context three important steps are accomplished sequentially and have been implemented in Java using the fuzzy library *jfuzzylogic* (which has been improved with the consideration of linguistic 2-tuples by a member of our team [Abchir, 2011]). This work is available at the following Internet address: `http://code.google.com/p/lcp-nets/`. The implementation of LCP-nets in EMF (Eclipse Modeling Framework) has been used in a new programming abstraction implemented in BPEL (Business Process Execution Language) for the French ANR project SemEUsE (07-TLOG-018) [Châtel et al., 2010a].

The three steps are the following. First one is the elicitation process. It is performed *before* execution. Second step is the translation of the preference model into an efficient representation that can be used *during* execution. Each CPT is translated into an inference system with a rule per table line. Last step is the preference model evaluation that corresponds to the search algorithm, see subsection 4.2. At runtime, current QoS values are injected after a fuzzification phase (into fuzzy sets or into linguistic 2-tuples). Then the system computes local utilities for each node thanks to the CPTs with the inference systems. Finally the global utility is obtained by aggregating local utilities.

The aggregation operator cannot be a simple weighted mean. Indeed we must take into account the fact that the arcs give the relative importance of the nodes they interconnect (and so, of the QoS properties). This *implicite* relative importance due to the node position in the graph must be reflected in order to be considered while computing the global utility factor. For example, for the surveillance camera service, bandwidth B that is the higher vertex of the graph is necessarily more

important than security and resolution. On the contrary, S and R that have the same depth are of equal importance. A weight is thus attached to each node *i.e.* to each utility factor (*cf.* figure 3).

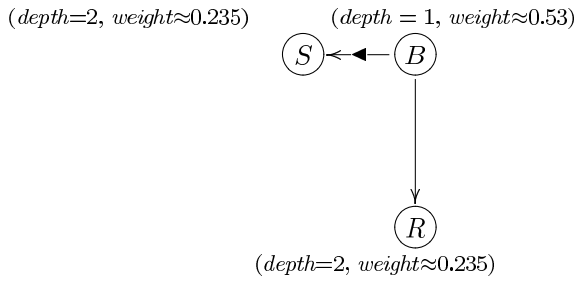

Figure 3: From node depth to node weights.

It has to be noted that UCP-nets don't use any weights on nodes because utility factors are supposed to carry this information already. Indeed, the UCP-nets formalism requires gaps between utility values big enough to carry this information implicitely.

That way, weights are given to the nodes of the LCP-net, taking into account their depth and beginning by the root node. The affectation of the weights needs a monotonous and strictly decreasing function. Weights are between 0 and 1 and sum to 1. This point is discussed and detailed in section 4.

The global utility obtained allows for a simple and quick comparison (since we have real numbers) of the various outcomes for a given service and permits a precise ranking. If we want to give the final proposed choices of services to a human, it can be useful to give a linguistic answer for the utilities (and give instead linguistic 2-tuples answers).

Now we introduce all the preliminary notations needed to define the LCP-nets formally and to guarantee their reusability in another context than service selection and SOA.

3 Preliminary notations

Following TCP-net definitions, we define our LCP-nets in a formal manner. To illustrate the definitions, we take as an example the same than the one described in section 2, *i.e.* a surveillance camera service.

Let:

- V_i be a variable ($i \in \{1, \ldots, p\}$): e.g. security,
- D_i be the definition domain of V_i: e.g. $[0, 100]$,
- T_{V_i} be the linguistic term set associated to V_i: e.g. $\{S_{none}, S_{full}\}$,
- LV (a linguistic variable) be the following triplet: $LV = \langle V, D, T_V \rangle$:
  e.g. $\langle$security,$[0, 100]$,$\{S_{none}, S_{full}\}\rangle$.

As in the UCP-net formalism, preferences are expressed through utilities in our framework. But they are expressed through linguistic variables, as the other variables. They take their values in the single triplet $\langle V_U, D_U, T_{V_U} \rangle$ defined once for all:
e.g. $\langle$utility,$[0, 1]$,{very_low,low,medium,high,very_high}$\rangle$.

One utility is a triplet $LV_U = \langle V_U, D_U, S_{V_U} \rangle$, with $S_{V_U} \in T_{V_U}$, e.g. $\langle$utility,$[0,1]$,low$\rangle$.

A conditional preference table $CPT(LV)$ associates preferences over D for every possible value assignment to the parents of LV denoted $Pa(LV)$. In addition, as in the TCP-nets formalism, each undirected edge is annotated with a conditional importance table $CIT(LV)$. A CIT associated with such an edge (LV_i, LV_j) describes the relative importance of LV_i and LV_j given the value of the corresponding importance-conditioning linguistic variable LV_k.

Graphically, a preference table (CPT or CIT) is a tuple of triplets, i.e. a table with N dimensions. N is the number of the linguistic variables interrelated with LV ($N = |Pa(LV)|$) and a utility S_{V_U} is defined in each case.

Thus a preference table may be represented by the tuple
$\langle LV_i, LV_{i'}, \ldots, LV_{i''\cdots'}, LV_{U_1}, LV_{U_2}, \ldots, LV_{U_\eta} \rangle$
with $\eta \in \{2 * N, \ldots, H\}$
and $H = |T_{V_i}| \times |T_{V_{i'}}| \times \ldots \times |T_{V_{i''\cdots'}}|$.

For example, a preference table is the tuple:
$\langle\langle$resolution,$[320, 1024]$,$\{R_L, R_H\}\rangle$,
$\langle$bandwidth,$[56, 4096]$,$\{B_L, B_M, B_H\}\rangle$,
$\langle$utility,$[0,1]$,very_high$\rangle$,
$\langle$utility,$[0,1]$,very_low$\rangle$,
$\langle$utility,$[0,1]$,high$\rangle$,
$\langle$utility,$[0,1]$,low$\rangle$,
$\langle$utility,$[0,1]$,very_low$\rangle$,
$\langle$utility,$[0,1]$,very_high$\rangle\rangle$.

More precisely, a preference table is equal to:

$$
\begin{array}{l}
\langle\ \langle S_{V_{i_1}}, S_{V_{i'_1}}, \ldots, S_{V_{i''_1\cdots'}}, S_{V_{U_1}} \rangle \\
\quad \langle S_{V_{i_2}}, S_{V_{i'_2}}, \ldots, S_{V_{i''_2\cdots'}}, S_{V_{U_2}} \rangle, \\
\quad \ldots \\
\quad \langle S_{V_{i_\eta}}, S_{V_{i'_\eta}}, \ldots, S_{V_{i''_\eta\cdots'}}, S_{V_{U_\eta}} \rangle \\
\rangle
\end{array}
$$

So we get η tuples $\langle S_{V_i}, S_{V_{i'}}, \ldots, S_{V_{i''\cdots'}}, S_{V_U} \rangle$ with $\min(\eta) = 2 * N$ and $\max(\eta) = H$. The reason why the minimum for H is equal to $2 \times N$ is because it is necessary that $|T_V| \geq 2$ in order to be able to express a preference (!).

Following the same example and considering that resolution and bandwidth are interrelated, the associated preference table can be defined as these six ($\eta = 6$) tuples:
$\langle\langle R_L, B_L, very\_high \rangle$,
$\langle R_L, B_M, very\_low \rangle$,
$\langle R_L, B_H, high \rangle$,
$\langle R_H, B_L, low \rangle$,
$\langle R_H, B_M, very\_low \rangle$,
$\langle R_H, B_H, very\_high \rangle\rangle$.

This is to be read as six rules implying two different linguistic variables, i.e. six triplets $\langle V, D, S_V \rangle$ and six triplets $\langle V_U, D_U, S_{V_U} \rangle$:

R1. If we have
$\langle$resolution,$[320, 1024]$,$R_L\rangle$ and
$\langle$bandwidth,$[56, 4096]$,$B_L\rangle$
then we have
$\langle$utility,$[0, 1]$,$very\_high\rangle$;

R2. ...

...

R6. If we have
$\langle$resolution,$[320, 1024]$,$R_H\rangle$ and
$\langle$bandwidth,$[56, 4096]$,$B_H\rangle$
then we have $\langle$utility,$[0, 1]$,$very\_high\rangle$.

4 LCP-net formal definitions

Definition 1. *An LCP-net over variables $\{LV_1, \ldots, LV_p\}$ is a directed graph over $\{LV_1, \ldots, LV_p\}$ whose nodes are annotated with conditional preference tables $CPT(LV_i)$ and with conditional importance tables $CIT(LV_i)$ for $i \in \{1, \ldots, p\}$.*

Thus an LCP-net $\mathcal{L}$ is a tuple $\langle SL$,cp,i,ci,cpt,cit$\rangle$ where:

- SL is a set of linguistic variables $\{LV_1, \ldots, LV_p\}$, e.g. $SL = \{\langle$security,$[0, 100]$,$\{S_{none}, S_{full}\}\rangle$, $\langle$bandwidth,$[56, 4096]$,$\{B_L, B_M, B_H\}\rangle$, $\langle$resolution,$[320, 1024]$,$\{R_L, R_H\}\rangle\}$,
- cp is a set of directed cp-arcs. A cp-arc $\langle\overrightarrow{LV_i, LV_j}\rangle$ is in $\mathcal{L}$ iff the preferences over the values of LV_j depend on the actual value of LV_i. For each $LV \in SL$, $Pa(LV) = \{LV' | \langle\overrightarrow{LV', LV}\rangle \in$ cp$\}$,
- i is a set of directed i-arcs. An i-arc $(\overrightarrow{LV_i, LV_j})$ is in $\mathcal{L}$ iff $LV_i \triangleright LV_j$,
- ci is a set of undirected ci-arcs. A ci-arc (LV_i, LV_j) is in $\mathcal{L}$ iff we have $\mathcal{RI}(LV_i, LV_j | LV_k)$, i.e. iff the relative importance of LV_i and LV_j is conditioned on LV_k, with $LV_k \in SL \setminus \{LV_i, LV_j\}$. We call LV_k the *selector set* of (LV_i, LV_j) and denote it by $\mathcal{S}(LV_i, LV_j)$,
- cpt associates a CPT with every linguistic variable $LV \in SL$, where $CPT(LV)$ is a mapping from $D_{Pa(LV)}$ (i.e., assignments to LV's parent linguistic variables) to D_U,
- cit associates with every ci-arc between LV_i and LV_j a CIT from $D_{\mathcal{S}(LV_i, LV_j))}$ to orders over the set $\{LV_i, LV_j\}$.

The *CPT* (attached to an *LV*) supplies a local utility for this *LV*. This local utility denoted by *lu* is computed thanks to an inference engine using the aforementioned rules.

So we get on the one hand an LCP-net that defines the preferences and on the other hand p local utilities denoted by the set:

$$LU = \bigcup_{i=1}^{p} \{lu_i\}$$

Then each node of $\mathcal{L}$ is associated with a weight w, i.e. we obtain a weight vector W:

$$W = \bigcup_{i=1}^{p} \{w_i\}$$

$\mathcal{L}$ can now be represented as the tuple $\langle SL$,cp,i,ci,cpt,cit,$W\rangle$ assuming that values taken by W depend on the node depth.

4.1 Computing the node weights

The algorithm for computing W can be based on a BUM (Basic Unit-interval Monotonic) family function [Yager, 2007]. A BUM function f_{BUM} is a mapping from $[0,1]$ to $[0,1]$ and assumes the following properties:

- $f_{BUM}(0) = 0$
- $f_{BUM}(1) = 1$
- f_{BUM} is increasing (i.e. if $x > y$ then $f_{BUM}(x) \geq f_{BUM}(y)$)

So weights W are computed thanks to f_{BUM} in the following manner:

$$w_i = f_{BUM}(i/p) - f_{BUM}((i-1)/p)$$

The chosen f_{BUM} function can be $f_{BUM}(x) = x$ (in this case, all weights equal $(1/p)$ with p the number of nodes) ; or $f_{BUM}(x) = x^3$ (in that case, w_1 is very small compared to w_p) ; or $f_{BUM}(x) = \sqrt{x}$ (in that case, w_1 is the greater weight). To be able to analyze the choice of f_{BUM}, we can compute a measure of *orness* on this weight vector [Yager, 1988]:

$$orness(W) = \frac{1}{p-1} \sum_{i=1}^{p} (p-i) w_i$$

This measure, between 0 and 1, allows us to express to which extent the aggregator using these weights resembles an OR. For example, when $f_{BUM}(x) = x$, $orness(W) = 0.5$. But when w_1 is much bigger than the "following" weights, $orness(W)$ tends towards 1.

As in the CP-nets, the deeper we go, the smaller the weights: we will then choose a vector W whose measure *orness* is between 0.5 and 1[1], i.e. $f_{BUM}(x) = \sqrt{x}$ or $\sqrt[3]{x}$, etc.

Assigning weights to nodes of a graph is slightly different from a classical weight assignment to values. The difference is in the order of the values. In an LCP-net graph several nodes can have the same depth, so the order is not total. That is why assigning w only thanks to a BUM function, even

[1] In our implementation, the obtained weight vector verifies this criterion.

appropriately chosen, doesn't permit to completely answer our problem, since nodes of the same depth will be discriminated.

We apply a BUM function such as the associated w be decreasing ($w_i > w_{i+1}$, with $i \in [1,p]$). Then for every node of the same depth, we sum their associated weights and make an equirepartition of the obtained sum between these nodes.

Thus, every constraint is fulfilled, by constructing weights through f_{BUM}:

- $\sum_{i=1}^{p} w_{i,l_i} = 1$ with l_i the depth of node i, $l_i \in [1,L]$ and $L \leq p$
- $\forall i \in [1,p], \forall l_i \in [1,L], \begin{cases} w_{i,l_i} > w_{i+1,l_{i+1}} & \text{if } l_i \neq l_{i+1} \\ w_{i,l_i} = w_{i+1,l_{i+1}} & \text{otherwise} \end{cases}$

W is combined with LU to obtain the global utility associated to an outcome o denoted GU_o.

$GU_o = \Delta(LU_o, W)$, with Δ a weighted aggregator such as an OWA operator (for example a simple weighted average aggregation).

A local utility is either a linguistic term or a number corresponding to the defuzzification (through operator d) of the subset: $lu = f_{S_{V_U}}$ or $lu = d(f_{S_{V_U}})$ with:

$$f_{S_{V_U}} = f_{S_{V_U}}(y) = \begin{cases} \perp(f_{S^1_{V_U}}(y), \ldots, f_{S^\eta_{V_U}}(y)) \\ \quad \text{if the } \eta \text{ rules are independent} \\ \top(f_{S^1_{V_U}}(y), \ldots, f_{S^\eta_{V_U}}(y)) \\ \quad \text{otherwise} \end{cases}$$

with $y \in D_U$, $\perp$ a triangular conorm and $\top$ a triangular norm.

For sake of simplicity, we assume that $lu_i = f_{S_{V_{U_i}}}(y)$. GU_o is thereof either a linguistic term or a number. We assume that in the case where it is a linguistic term, it is always possible to find a defuzzification operator d that provides for a number.

Considering that the rules are independent and applying the generalized modus ponens,

$$f_{S_{V_U}}(y) = \sup_{(x_1,\ldots,x_N) \in D_1 \times \cdots \times D_N} \Big\{ \top \Big[g(f_{S_{V'_1}}(x_1), \ldots, f_{S_{V'_N}}(x_N), \Phi(g(f_{S^1_{V_1}}(x_1), \ldots, f_{S^1_{V_N}}(x_N)), f_{S^1_{V_U}}(y)) \Big] \\ \vee \ldots \vee \top \Big[g(f_{S_{V'_1}}(x_1), \ldots, f_{S_{V'_N}}(x_N), \Phi(g(f_{S^\eta_{V_1}}(x_1), \ldots, f_{S^\eta_{V_N}}(x_N)), f_{S^\eta_{V_U}}(y)) \Big] \Big\}$$

with $f(x)$ the membership function of element x, Φ any fuzzy implication, V' the real variables observed (retrieved, given by the user), $S_{V'_1}$ the linguistic term associated to the first variable (V'_1) observed and g an aggregation operator such as a triangular norm (min for example). Thus an outcome o is actually a tuple $\langle S_{V'_1}, \ldots, S_{V'_p} \rangle$.

Algorithm 1 Search algorithm

Require: o is a tuple $\langle S_{V_1'}, \ldots, S_{V_p'} \rangle$, SO is the set of o, $\mathcal{L}$ is valid
1: **for** $i = 0$ to p **do**
2: compute the weight vector W composed of w_i
3: **end for**
4: **for** each outcome $o \in SO$ **do**
5: **for** each table $CPT(LV_i)$ **do**
6: inject observed values of o and apply an inference such as the generalized modus ponens,
7: compute and retrieve the set of lu_i for this o.
8: **end for**
9: compute and retrieve GU for this o
10: **end for**
11: **return** the set of GU, one per outcome

4.2 Search algorithm

The search algorithm that has to be used to compute the global utility factor for each outcome is defined as follows (see algorithm 1) and uses all the preliminary notations. The notion of *validity* of an LCP-net in the requirements is explained in section 5.2.

5 Properties of LCP-nets

5.1 Dominance testing

A basic query with respect to the LCP-net model is preferential comparison between outcomes. The ordering query is weaker than the dominance query because the last one guarantees that an outcome o is comparable to any other outcomes and that there exists a dominant outcome.

In order to prove the dominance testing property, we shall prove that an outcome o can always be found as being strictly preferred to another outcome o'.

Theorem 1. *Given an LCP-net $\mathcal{L}$ and a pair of outcomes o and o', we have that $\mathcal{L} \models o \prec o'$ iff GU_o is weaker than $GU_{o'}$. We say that o' is preferred o and that o' dominates o with respect to $\mathcal{L}$.*

Proof.

$$\frac{\text{inference from } CPT(LV_1^o) \quad \ldots \quad \text{inference from } CPT(LV_p^o)}{\mathcal{L} \vdash lu_1^o = f_{S_{V_{U,1}^o}} \quad \ldots \quad lu_p^o = f_{S_{V_{U,p}^o}}}$$

$$\frac{\mathcal{L} \vdash lu_1^o = f_{S_{V_{U,1}^o}} \quad \ldots \quad lu_p^o = f_{S_{V_{U,p}^o}}}{\mathcal{L} \vdash LU_o = \{lu_1^o, \ldots, lu_p^o\}}$$

$$\frac{\text{inference from } CPT(LV_1^{o'}) \quad \ldots \quad \text{inference from } CPT(LV_p^{o'})}{\mathcal{L} \vdash lu_1^{o'} = f_{S_{V_{U,1}^{o'}}} \quad \ldots \quad lu_p^{o'} = f_{S_{V_{U,p}^{o'}}}}$$

$$\frac{\mathcal{L} \vdash lu_1^{o'} = f_{S_{V_{U,1}^{o'}}} \quad \ldots \quad lu_p^{o'} = f_{S_{V_{U,p}^{o'}}}}{\mathcal{L} \vdash LU_{o'} = \{lu_1^{o'}, \ldots, lu_p^{o'}\}}$$

$$\frac{\mathcal{L} \vdash LU_o = \{lu_1^o, \ldots, lu_p^o\} \qquad \mathcal{L} \vdash LU_{o'} = \{lu_1^{o'}, \ldots, lu_p^{o'}\}}{\mathcal{L} \vdash \Delta(LU_o, W) < \Delta(LU_{o'}, W)}$$

$$\frac{\mathcal{L} \vdash \Delta(LU_o, W) < \Delta(LU_{o'}, W)}{\mathcal{L} \vdash d(GU_o) < d(GU_{o'})}$$

$$\frac{\mathcal{L} \vdash d(GU_o) < d(GU_{o'})}{\mathcal{L} \vdash GU_o \prec GU_{o'}}$$

$$\frac{\mathcal{L} \vdash GU_o \prec GU_{o'}}{\mathcal{L} \models o \prec o'}$$

□

This means that starting with a *well-formed* LCP-net, it is always possible to infer that an outcome is preferred another one. Of course, that doesn't mean that in all situations, indifference is impossible. Indeed, if two outcomes are very close (granularity would be too coarse to distinguish between them) then both will be chosen as the best ones.

A well-formed LCP-net is a valid LCP-net, *i.e* an acyclic graph, with a number of arcs less or equal to the number of nodes minus 1 ($\sharp nodes - 1$). Other properties are required and detailed below.

5.2 Valid LCP-nets

When implementing the LCP-nets in EMF we construct the graphs incrementally. This is very important to factorize the objects. In particular we define LCP-nets *fragments* that are pieces of graphs (*e.g.* only a node and its CPT).

This way of doing doesn't guarantee the obtention of valid LCP-nets. Verifying validity of LCP-nets *a posteriori* is not trivial. There are a certain number of conditions that have to be fulfilled. We define an *atomic* valid LCP-net (a minimal LCP-net) as an object with only one node and its CPT (or CIT). The elementary operators to manipulate valid LCP-nets are: addition (of a node, of an arc, etc.), subtraction, etc. These operators have invariants, preconditions and postconditions.

In this paper we only focus on invariants.

Let consider the following objects:

- n is a node;
- V is the linguistic variable attached to n;
- SN is the set of nodes;
- an arc is denoted (s, t) with s the source node and t the sink node. In the ci-arcs, s can be exchanged with t;
- SA is the set of arcs (cp, i and ci) : $SA = \{\text{cp,i,ci}\}$.

Invariants that share all the operators on LCP-nets are the following:

- the total number of arcs (cp, i, ci) is not greater than the number of pairs (s, t) where $s, t \in SN$ and $s \neq t$;

- there is mutual exclusion between the kinds of arcs:
  - if $(s,t) \in \mathsf{cp}$ then $(s,t) \notin \mathsf{i}$ and $(s,t) \notin \mathsf{ci}$;
  - if $(s,t) \in \mathsf{i}$ then $(s,t) \notin \mathsf{cp}$ and $(s,t) \notin \mathsf{ci}$;
  - if $(s,t) \in \mathsf{ci}$ then $(s,t) \notin \mathsf{i}$ and $(s,t) \notin \mathsf{cp}$.
- the dimension of the CPT associated to s node is equal to 1 + the number of cp-arcs that are indegrees of s;
- each CPT has a dimension that is less or equal than the number of domain values of the associated linguistic variable;
- there is no conditional cycles in the graph;
- there is at least one node (*i.e.* at least a CPT) and there are from 0 to n arcs;
- there are at least as many CPTs than nodes, *i.e* there are exactly $\sharp$*nodes* CPTs and $\sharp$*ci-arcs* CITs.

Under these conditions, valid LCP-nets can be constructed.

5.3 Optimization query

Another important property is the outcome optimization query. The question is: "Given an LCP-net $\mathcal{L}$, is it possible to determine — even virtually — an ideal outcome that would be the best one among those preference rankings that satisfy $\mathcal{L}$ (*i.e.* an outcome whose $GU\,GU$ equals 1)?" Because of the inference systems, this turns to be a complex task.

First problem is to determine the LU if GU is not computed with a simple aggregation operator Δ. Assuming that Δ is a simple weighted mean (so the set LU contains only values equal to 1) there is a second problem because of the inferences in each node. We have to reverse the inferences in order to obtain the $f_{S^{\eta}_{V_U}}(y)$ for each rule and for each node. This depends on the way (*i.e* with a triangular conorm or norm) the aggregation of the rules is performed. Finally it is necessary to obtain the $f_{S_{V'_i}}(x_i)$ for $i \in \{1,\ldots,N\}$ for each node. In our running example, it should be a service that would propose S_{none} as security, B_H as bandwidth and R_H as resolution.

Of course there is no reason that the best affectations to each node will give the optimal outcome. This reverse inference is an abduction problem. Peirce (1839 – 1914), a famous logician, defined the abduction this way: in case "C is true if A is true", and C is observed (C is called the *manifestation*), then there is some reasons to think A may be true.

Since, many works have focused on this problem. Miyata *et al.* define cause-and-effect relationships. They try to give a definition of maximum and minimum fuzzy sets which can explain the manifestations [Miyata et al., 1995]. In our case, the manifestation is the best outcome. Revault d'Allonnes *et al.* also tried to construct a set of likely explanations for a manifestation [Revault d'Allonnes et al., 2007], but they noticed that it is very hard to extend formal fuzzy abductive results to different classes of implications. A set of explanations can be constructed only for 'deduction-coherent' implications, not for all the implications [Revault d'Allonnes et al., 2009].

All these studies show us that it is not possible to prove the outcome optimization query without fixing a large number of conditions, such as:

- the shapes of all the fuzzy sets (considering only linguistic 2-tuples shall be a great simplification but probably not sufficient);

- the implication operators;
- the operators used to aggregate the rule conclusions;
- the operator Δ that aggregates the local utilities lu.

6 Conclusion

Although it is possible to represent an LCP-net as a set of nodes, arcs and CPTs/CITs, this paper has shown that, mathematically, it's a tuple consisting of linguistic variables LV, three types of arcs, two types of tables and a weight vector. Furthermore, an LCP-net corresponds, after translation, to a set of LV, several inference systems and weights associated with nodes that allow automated reasoning about these preferences and lay the foundations for their implementation.

The wishable properties for an LCP-net are mainly the dominance query and the optimization query. While dominance is not difficult to be proved, optimization query that has to do with abduction, requires many suppositions in order to prove it.

In a future work we will put some restrictive hypotheses about the shapes of the linguistic variables in order to be able to give a restrictive set of explanations of a manifestation.

References

[Abchir, 2011] Abchir, M. (2011). A jFuzzyLogic Extension to Deal With Unbalanced Linguistic Term Sets. In *Proc. of the 12th International Student Conference on Applied Mathematics and Informatics (ISCAMI'11)*, page to appear.

[Boubekeur and Tamine-Lechani, 2006] Boubekeur, F. and Tamine-Lechani, L. (2006). Recherche d'information flexible basée CP-Nets. In *Proc. Conference on Recherche d'Information et Applications (CORIA'06)*, pages 161–167.

[Boutilier et al., 2001] Boutilier, C., Bacchus, F., and Brafman, R. I. (2001). UCP-Networks: A directed graphical representation of conditional utilities. In *Proc. of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 56–64.

[Boutilier et al., 2004] Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., and Poole, D. (2004). CP-nets: A tool for representing and reasoning with conditional *Ceteris Paribus* Preference Statements. *Journal of Artificial Intelligence Research*, 21:135–191.

[Brafman and Domshlak, 2002] Brafman, R. I. and Domshlak, C. (2002). Introducing variable importance tradeoffs into CP-nets. In *Proc. of the Eighteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI'02)*, pages 69–76.

[Braziunas and Boutilier, 2006] Braziunas, D. and Boutilier, C. (2006). Preference elicitation and generalized additive utility (nectar paper). In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI'06)*, Boston, MA.

[Châtel et al., 2010a] Châtel, P., Malenfant, J., and Truck, I. (2010a). QoS-based Late-Binding of Service Invocations in Adaptive Business Processes. In *The 8th International Conference on Web Services (ICWS'10)*, pages 227–234.

[Châtel et al., 2008] Châtel, P., Truck, I., and Malenfant, J. (2008). A linguistic approach for non-functional preferences in a semantic SOA environment. In *The 8th International FLINS Conference on Computational Intelligence in Decision and Control*, pages 889–894.

[Châtel et al., 2010b] Châtel, P., Truck, I., and Malenfant, J. (2010b). LCP-nets: A linguistic approach for non-functional preferences in a semantic SOA environment. *Journal of Universal Computer Science*, pages 198–217.

[Delgado et al., 1993] Delgado, M., Verdegay, J., and Vila, M. (1993). On Aggregation Operations of Linguistic Labels. *International Journal of Intelligent Systems*, 8:351–370.

[Herrera and Martínez, 2000] Herrera, F. and Martínez, L. (2000). A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Trans. Fuzzy Systems*, 8(6):746–752.

[Miyata et al., 1995] Miyata, Y., Furuhashi, T., and Uchikawa, Y. (1995). A study on fuzzy abductive inference. In *Proc. of the International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium*, pages 337–342.

[Revault d'Allonnes et al., 2007] Revault d'Allonnes, A., Akdag, H., and Bouchon-Meunier, B. (2007). Selecting implications in fuzzy abductive problems. In *IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, pages 597–602.

[Revault d'Allonnes et al., 2009] Revault d'Allonnes, A., Akdag, H., and Bouchon-Meunier, B. (2009). For a data-driven interpretation of rules, wrt gma conclusions, in abductive problems. *Journal of Uncetain Systems*, 3(4):280–297.

[Yager, 1988] Yager, R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Syst. Man Cybern.*, 18(1):183–190.

[Yager, 2007] Yager, R. (2007). Using Stress Functions to Obtain OWA Operators. *IEEE Trans. on Fuzzy Systems*, 15(6):1122–1129.

[Zadeh, 1965] Zadeh, L. A. (1965). Fuzzy sets. *Information Control*, 8:338–353.

Annexe B

Article publié (FLINS 2010)

Ce qui suit est un article qui a été publié dans les actes de la conférence internationale FLINS (Fuzzy Logic and Intelligent technologies in Nuclear Science) qui s'est déroulée du 1er au 4 août 2010 à Chengdu, en Chine.

Il traite de la question de l'unification de trois modèles linguistiques en une représentation vectorielle commune, et a pour titre : *Towards A Unification Of Some Linguistic Representation Models : A Vectorial Approach*.

TOWARDS A UNIFICATION OF SOME LINGUISTIC REPRESENTATION MODELS: A VECTORIAL APPROACH

ISIS TRUCK

LIASD – EA 4383, Université Paris 8,
2 rue de la Liberté, Saint-Denis, F-93526 Cedex, France
E-mail: truck@ai.univ-paris8.fr
www.univ-paris8.fr

JACQUES MALENFANT

LIP6 – UMR 7606, Université Pierre et Marie Curie-Paris 6,
104 av. du Président Kennedy, Paris, F-75016, France
E-mail: Jacques.Malenfant@lip6.fr
www.upmc.fr

This paper takes place in the computing with words framework where a unified model of several linguistic representation models is proposed. We consider (i) the *2-tuple fuzzy linguistic representation model*, (ii) the *proportional 2-tuple fuzzy linguistic representation model*, (iii) the *linguistic degrees* and their associated *generalized symbolic modifiers*. Our approach is based on a vectorial representation. Using vectors and scalar multiplications, we translate and then unify these three models into a single one. In this paper, the scope of our research is limited to the definitions of a linguistic value, the attached aggregation operators remaining the next logical step.

Keywords: Linguistic representation models; Unification; Vectorial approach.

1. Introduction

An important part of artificial intelligence research has been and is still dedicated to qualitative information and its representation. Zadeh has introduced the computing with words (CW) paradigm[1] where several linguistic representation models take place.[2–4] A key issue in CW is to relate all the operations and the intermediate results to the original set of linguistic terms defined by the user, as a way to convey an understandable semantics to these results in the terms used by the humans. Since Herrera and Martínez have proposed their 2-tuple models for CW without loss of pre-

cision,[2] other models have emerged towards the same goal. The key idea behind all of these models is to maintain a representation of fuzzy subsets[1] that relates to the original linguistic terms while retaining full precision over intermediate results during all the operations of CW. Although different, all of these models[2–4] appear to share a common conceptual basis. In this paper, we propose that linear algebra, and its concepts of basis, can capture the relationships between these models and provide for means not only to relate the models to each other but also allows CW to use the simplest possible model for its computation, without loss of precision, while being able to come back to any of these models as soon as human-related interventions require results to be expressed in terms of the original linguistic term set. Models could then be used interchangeably, depending on the properties of the linguistic terms set that may simplify the expression using one 2-tuple model rather than another.

2. Conceptual Framework and Related Work

2.1. *Linguistic Representation Models for CW*

To avoid lack of precision and linguistic approximation, Herrera & Martínez consider an ordered linguistic term set $\{s_0, \ldots, s_g\}$ and a symbolic translation α attached to each term that will support the difference between the term and the value to express. They thus obtain 2-tuples noted $(s_i, \alpha), \alpha \in [-.5, .5)$. Wang & Hao's have proposed proportional 2-tuples where the lack of precision is supported by a proportion $\alpha \in [0, 1]$ attached to the linguistic term l_i: they define a proportional 2-tuple as: $(\alpha l_i, (1-\alpha) l_{i+1})$. It is to notice that these models imply the use of a continuous domain. Truck & Akdag, as for them, have considered the case where the domain is discrete. To avoid the loss of information they change the scale granularity in adding or subtracting terms in the original term set thanks to what they call *generalized symbolic modifiers* (GSM). The model they propose is quite simple: the linguistic term set is represented by a scale of b ordered symbolic values. A symbolic value is noted by a and is specified by its position in the scale: $p(a)$, with $p(a) \in \mathbb{N}$.

To combine such linguistic terms,[5] the authors propose GSMs as mappings from an initial 2-tuple (a, b) to a new 2-tuple (a', b'):

Definition 1. Let $\mathcal{L}_b$ be a set of b linguistic terms, with $b \in \mathbb{N}^* \setminus \{1\}$. A GSM m_ρ is defined as: $\begin{array}{c} m_\rho : \mathcal{L}_b \to \mathcal{L}_{b'} \\ a \mapsto a' \end{array}$ with $p(a) < b$, $p(a') < b'$ and $\rho \in \mathbb{N}^*$.
For example the DW GSM is defined as DW(ρ): $p(a') = p(a)$, $b' = b + \rho$

2.2. *Basic Linear Algebra Concepts*

The cornerstone of our proposal is based on the idea that each 2-tuple model defines a vector space, and strives to express all values in this space using a set of independent vectors, chosen to relate to the original linguistic terms set. To be more precise, key definitions from linear algebra are summarized:

Definition 2 (Basis). *For a given vector space V, a basis is a (finite or infinite) set $B = \overrightarrow{v_i}$ $(i \in I)$ of vectors $\overrightarrow{v_i}$ indexed by some index set I that spans the whole vector space, and is minimal with this property.*

Given a finite basis, any vector $\overrightarrow{v}$ can be expressed as a linear combination of the basis elements: $\overrightarrow{v} = a_1\overrightarrow{v_1} + a_2\overrightarrow{v_1} + \cdots + a_{\#I}\overrightarrow{v_{\#I}}$

Basis are unfortunately a too strong concept to deal with 2-tuple models. The need for 2-tuple models to retain a link with the original linguistic terms set, comes at odd with the minimality required for a basis. We therefore define the concept of constrained basis as follows:

Definition 3 (Constrained basis). *For a given vector space V, a constrained basis is a set $B = \overrightarrow{v_i}$ $(i \in I)$ of vectors $\overrightarrow{v_i}$ indexed by some index set I iff for all $\overrightarrow{v} \in V$, there exists $a_1, ..., a_{\#I}$ such that $\overrightarrow{v} = a_1\overrightarrow{v_1} + a_2\overrightarrow{v_1} + \cdots + a_{\#I}\overrightarrow{v_{\#I}}$, and where the a_i are subject to constraints such that $a_i \in A_i \subset \mathbb{R}$.*

The key concept is that a constrained basis may require more vectors to span the whole vector space, given the constraints on the scalars that can be used to combine them linearly. Hence, they are no longer minimal.

3. Proposition

We now show how all three models are actually constrained bases, such as the one of Figure 1.

Conjecture 3.1. *Herrera & Martínez 2-tuple model forms a constrained basis for the space $[0, g]$, where the vectors $\overrightarrow{v_0}, \overrightarrow{v_1}, \ldots, \overrightarrow{v_g}$ represent the values $\{0, 1, \ldots, g\}$ respectively, where the vector $\overrightarrow{u}$ represents a unit vector such that $\overrightarrow{v_i} + \overrightarrow{u} = \overrightarrow{v_{i+1}}, i \in \{0, g-1\}$, and where all values can be expressed as a linear combination: $\overrightarrow{v_i} + \alpha_i\overrightarrow{u}$ subject to the constraints $\alpha_0 \in [0, .5)$, $\alpha_i \in [-.5, .5), i \in \{1, g-1\}$ and $\alpha_g \in [-.5, 0]$.*

Conjecture 3.2. *The set $(\overrightarrow{v_0}, \overrightarrow{u})$ also forms a constrained basis for the space $[0, g]$, where all values can be expressed as a linear combination: $\overrightarrow{v_0} + \alpha\overrightarrow{u}$ subject to the constraints $\alpha \in [0, g]$.*

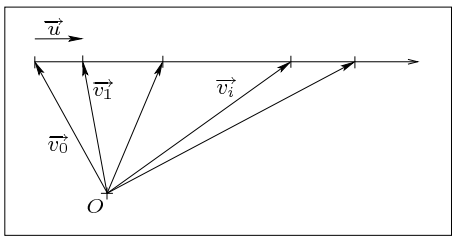

Fig. 1. Vectorial plane.

The interest of this latter basis is that it makes computations very simple. As all values are expressed with the same base vector $\overrightarrow{v_0}$, any aggregator can simply be computed over the α of each unit vector.

Conjecture 3.3. *Wang & Hao 2-tuple model forms a constrained basis for the space* $[0, g]$*, where the vectors* $\overrightarrow{v_0}, \overrightarrow{v_1}, \ldots, \overrightarrow{v_g}$ *represent the values* $\{0, 1, \ldots, g\}$ *respectively, and where all values can be expressed as a linear combination:*

$$\alpha\overrightarrow{v_i} + \beta\overrightarrow{v_{i+1}}, i \in \{0, \ldots, g\}$$

subject to the constraints $\alpha, \beta \geq 0$ *and* $\alpha + \beta = 1$*.*

Wang & Hao show that there are always two ways to write a value x: either through a term and its predecessor or through the same term and its successor: $(\alpha l_i, (1-\alpha)l_{i+1}) = (1-\alpha l_{i-1}, \alpha l_i)$. This equality is easily provable in our model thanks to the parallelogram property: in our formalism, $(\alpha l_i, (1-\alpha)l_{i+1})$ corresponds to $\alpha\overrightarrow{v_i} + \beta\overrightarrow{v_{i+1}}$ and $(1-\alpha l_{i-1}, \alpha l_i)$ to $\alpha\overrightarrow{v_{i-1}} + \beta\overrightarrow{v_i} = \beta\overrightarrow{v_i} + \alpha\overrightarrow{v_{i-1}}$. Consider that $\overrightarrow{v_i}$ has point O as origin, these four vectors form a parallelogram whose opposite points are O and x.

Conjecture 3.4. *Truck & Akdag model* without the recourse of a GSM *forms a constrained basis for the space* $[0, b-1]$*, where the vectors* $\overrightarrow{v_0}, \overrightarrow{v_1}, \ldots, \overrightarrow{v_{b-1}}$ *represent the values* $\{0, 1, \ldots, b-1\}$ *respectively, and where all values can be expressed as a linear combination:*

$$\overrightarrow{v_i} + \beta\overrightarrow{u}, i \in \{0, \ldots, b-1\}$$

subject to the constraints $i = p(a), \beta = 0$*.*

Applying a GSM permits to obtain a pair (a', b') from a pair (a, b).

Conjecture 3.5. *Truck & Akdag model* with *the recourse of a GSM* $m(\rho)$ *forms a constrained basis for the space* $[0, b'-1]$*, where the vectors*

$\overrightarrow{v_0}, \overrightarrow{v_1}, \ldots, \overrightarrow{v_{b'-1}}$ represent the values $\{0, 1, \ldots, b'-1\}$ respectively, and where all values can be expressed as a linear combination:

$$\overrightarrow{v_{i'}} + \beta' \overrightarrow{u}, i' \in \{0, \ldots, b'-1\}$$

subject to the constraints $i' = i = p(a), \beta' = (p(a')(b-1)/(b'-1)) - p(a)$.

NB: $\overrightarrow{v_i}$ and $\overrightarrow{u}$ are unchanged because we intend to express the *modified* pair (a', b') in the same vectorial notation, changing only β.

However, the values obtained for $\overrightarrow{v_{i'}}$ and β' don't permit to write the canonical form of the vectorial notation. Indeed β' is bounded by $(1-b)$ and by $(b-1)$ (knowing that $0 \leq p(a') \leq (b'-1)$ and $(1-b) \leq -p(a) \leq 0$) and for the canonical form it is required that $\beta' \in [-.5, .5)$ as β' is the factor of the unit vector.

Let us take an example. When transforming (a, b) into (a', b') using DW(10) with $p(a) = 3$ and $b = 5$, we obtain $p(a') = p(a) = 3$ and $b' = b + 10 = 15$. Thus $\beta' = (3 * 4/14) - 3 = -2.143$ and the vectorial form is $\overrightarrow{v_3} - 2.143\overrightarrow{u}$. Adding (subtracting in this case) to $\overrightarrow{v_3}$ a vector greater than the unit vector is equivalent to increment (resp. decrement) the subscript of $\overrightarrow{v}$. Hence $\overrightarrow{v_3} - 2\overrightarrow{u}$ is equivalent to exactly $\overrightarrow{v_1}$.

Conjecture 3.6. *We denote by $\widehat{\overrightarrow{v_{i'}}} + \widehat{\beta'}\overrightarrow{u}$ the canonical vectorial form of (a', b') and $[x]$ is the integer part of x.*

if $\beta' < 0$
  then if $\beta' - [\beta'] < -.5$
    then $\widehat{\beta'} = 1 - \beta' - [\beta']$ *;* $\widehat{\overrightarrow{v_{i'}}} = \overrightarrow{v_{i'+[\beta']-1}}$
    else $\widehat{\beta'} = \beta' - [\beta']$ *;* $\widehat{\overrightarrow{v_{i'}}} = \overrightarrow{v_{i'+[\beta']}}$
  else if $\beta' - [\beta'] \geq .5$
    then $\widehat{\beta'} = -1 + \beta' - [\beta']$ *;* $\widehat{\overrightarrow{v_{i'}}} = \overrightarrow{v_{i'+[\beta']+1}}$
    else $\widehat{\beta'} = \beta' - [\beta']$ *;* $\widehat{\overrightarrow{v_{i'}}} = \overrightarrow{v_{i'+[\beta']}}$

This implies that $\widehat{\beta'} \in [-.5, .5)$ as required.

In our example, we obtain $\widehat{\beta'} = -.143$ and $\widehat{\overrightarrow{v_{i'}}} = \overrightarrow{v_1}$. The canonical form is thus $\overrightarrow{v_1} - .143\overrightarrow{u}$.

Figure 2 sums up our proposition for all three 2-tuple models: Herrera & Martínez, Wang & Hao and Truck & Akdag respectively.

4. Conclusion

In this paper we have shown how different linguistic representation models (even over discrete *and* continuous domains) can be unified within a

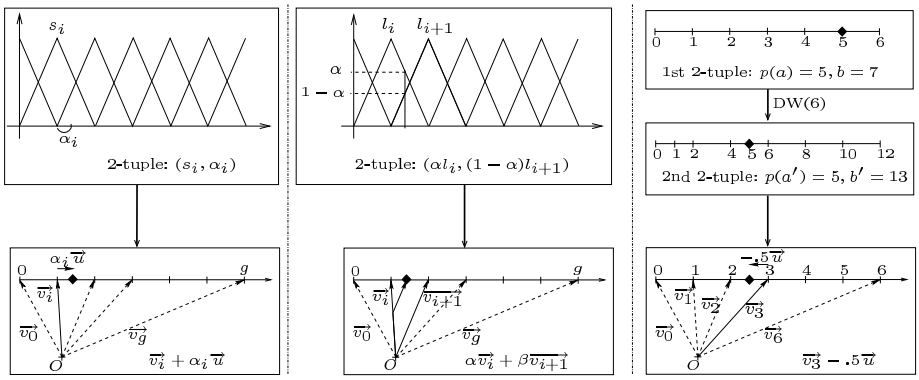

Fig. 2. Vectorial representation of the three considered cases.

single formalism. Fuzzy, proportional and discrete 2-tuples are considered as bases from a vectorial space. Generally speaking, all three models have their values that can be expressed as follows: $\alpha\overrightarrow{v_i} + \beta\overrightarrow{v_j}$ with constraints over i, j, α and β. This approach is advantageous because it is very simple to switch from one model to another, depending on which is preferred. Moreover, and this is future works, for each model it will be of great interest to reconsider the aggregation operators under our "vectorial vision". This will allow us to change the model *even during the calculation process* to match the computational requirements of the operator with the best-suited model.

References

1. L. Zadeh, Fuzzy logic = computing with words, *IEEE Trans. Fuzzy Systems* **4**, 103 (1996).
2. F. Herrera and L. Martínez, A 2-tuple fuzzy linguistic representation model for computing with words, *IEEE Trans. Fuzzy Systems* **8**, 746 (2000).
3. J. Wang and J. Hao, A new version of 2-tuple fuzzy linguistic representation model for computing with words, *IEEE Trans. Fuzzy Systems* **14**, 435 (2006).
4. I. Truck and H. Akdag, Manipulation of Qualitative Degrees to Handle Uncertainty : Formal Methods and Applications, *Knowledge and Information Systems (KAIS)* **9**, 385 (2006).
5. I. Truck and H. Akdag, A tool for aggregation with words, *Information Sciences* **179**, 2317 (2009).

Annexe C

Curriculum Vitæ

Ce qui suit est mon *curriculum vitæ* contenant surtout les activités de recherche ainsi que celles s'y rapportant. On trouvera aussi une liste exhaustive des publications réalisées, classées par type.

Isis Truck

Maître de conférences en Informatique

LIASD – EA 4383 — Université Paris 8
2 rue de la Liberté
93526 SAINT-DENIS Cedex
☎ *01 49 40 64 15*
FAX *01 49 40 67 83*
✉ *truck@ai.univ-paris8.fr*
http://www.ai.univ-paris8.fr/~truck

Etat civil

date de naissance	8 juin 1973
lieu de naissance	Bar-sur-Aube
nationalité	française

Formation

1999–2002 **Doctorat d'informatique**, *Université de Reims*, *Mention Très honorable.*

1998–1999 **DESS d'informatique, spécialité "Communication, Image, Réseaux"**, *Université de Marne-la-Vallée*, *Mention Bien.*

1997–1998 **Maîtrise d'informatique**, *Université de Reims*, *Mention Assez bien.*

Thèse de doctorat

titre	*Approches symbolique et floue des modificateurs linguistiques et leur lien avec l'agrégation*
laboratoire	LERI (Laboratoire d'Etude et de Recherche en Informatique) à Reims
date	13 décembre 2002
jury	Pr. Herman Akdag (directeur), Pr. Salem Benferhat (rapporteur, CR CNRS), Pr. Stéphane Loiseau (rapporteur), Pr. Francis Rousseaux (Président), Pr. Mohamed Quafafou, Dr. Amel Borgi (examinateurs)

Postes occupés

depuis 2003 **Maître de conférences**, *Département MIME (Micro-Informatique, Machines Embarquées)*, Laboratoire d'Informatique Avancée de Saint-Denis (LIASD – EA 4383), Université Paris 8.

2001–2003 **ATER**, *Faculté des Sciences et IUT d'informatique*, Université de Reims.

Activités de recherche

Encadrements

depuis le 01/11/2009 **Co-encadrement avec A. Pappa d'un doctorant**, *M.-A. Abchir*, Université Paris 8, bourse CIFRE Deveryware.
Sujet : "Elicitation et mise en œuvre de modèles de décision pour des applications en géolocalisation".

depuis le 01/11/2009 **Co-encadrement avec J. Malenfant d'une doctorante**, *Olga Melekhova*, Université Paris 6, bourse de l'ANR.
Sujet : "Modèle de décision de composants autonomiques pour systèmes répartis à grande échelle".

depuis le 01/10/2009 **Co-encadrement avec J. Malenfant d'un doctorant**, *Xavier Dutreilh*, Université Paris 8 et Université Paris 6, bourse CIFRE Orange.
Sujet : "Auto-régulation dans l'utilisation des ressources d'un centre de calcul".

2007–2010 **Co-encadrement avec J. Malenfant d'un doctorant**, *Pierre Châtel*, Université Paris 6, bourse CIFRE Thales.
Sujet : "Une approche qualitative pour la prise de décision sous contraintes non-fonctionnelles dans le cadre d'une composition agile de services".
Thèse soutenue le 5 mai 2010 et qualification aux fonctions de maître de conférences par le CNU obtenue en janvier 2011.

2005–2006 **Co-encadrement avec A. Bonardi de plusieurs étudiants de niveau master**, *N. Lehallier, M. Göksedef, K. Yamashita*, Université Paris 8 et Ecole Centrale d'Electronique, bourses ACI, MSH–CNRS.
Sujet : "Assistant virtuel pour metteur en scène et *performer*".

2003 **Co-encadrement avec H. Akdag d'un post-doctorant**, *A. Aït Younes*, Université de Reims.
Sujet : "IHM, Représentation des connaissances et bases de données".

Contrats/Projets

2009–2012 **Projet ANR SALTY (Self-Adaptive very Large distribuTed sYstems : Très grands systèmes répartis auto-adaptatifs)**, *cadre du programme "SEGI" (Systèmes Embarqués et Grandes Infrastructures – ARPEGE).*
Je suis la responsable pour Paris 8
Début le 01/11/2009
Budget de 23 kEUR (versés par l'ANRT)
Projet à 8 partenaires : Deveryware, EBM-Websourcing, INRIA Lille, MAAT-G, Thales Research & Technology, *Université Nice Sophia-Antipolis (porteur)*, Université Paris 6, Université Paris 8.

2009–2012 **Contrat CIFRE avec Deveryware**, *Sous ma responsabilité scientifique (au nom du LIASD).*
Budget de 54 kEUR

2009–2012 **Contrat CIFRE avec Orange**, *Sous ma responsabilité scientifique (au nom du LIASD)*, 3 partenaires : Orange, Université Paris 6, Université Paris 8.
Budget de ? (contrat en attente de signature)

2009 (6 mois) **Contrat de recherche avec Mobiluck**, *Sous ma responsabilité scientifique (travail effectué en collaboration avec mon collègue N. Jouandeau)*, thème "Optimisation de la pertinence des résultats et des revenus publicitaires, pour un service mobile de recherche d'informations locales".
Budget de 24 kEUR

Rayonnement scientifique

relecture **Revues internationales.**
Elsevier :
- Applied Soft Computing, depuis 2009
- Information Fusion, 2008
- Information Sciences, 2005

Wiley :
- Computational Intelligence, 2009

Springer :
- Soft Computing, 2006

Autres :
- Journal of Universal Computer Science, 2009
- Int. J. Uncertainty and Knowledge-Based Systems, 2008

Revue nationale.
TSI (Technique et science informatiques), 2011

Conférences internationales.
Plusieurs conférences, parmi lesquelles :
ICNC-FSKD'11, iiWAS'10, DIMEA'08, FLINS'08, EUROFUSE'07, HIS'05, FLINS'04, JCIS'03, etc.

jurys de thèse

co-directrice, *thèse de P. Châtel*, mai 2010, Université Paris 6.

examinatrice/rapporteur, *thèse de R. de Andrés Calle*, 2009, Université de Valladolid, Espagne.

rapporteur, *thèse de L. G. Pérez Cordón*, 2008, Université de Jaén, Espagne.

comités/groupes

membre du comité de programme, *conférences EUROFUSE'2007, FLINS'2008, ICNC-FSKD'2011, FLINS'2012*, Jaén, Madrid, Shanghai et Istambul.

membre du comité d'organisation et présidente de session, *conférence iiWAS, novembre 2010*, Paris.

membre du groupe SCIP, *(Soft Computing in Image Processing)*, http://www.fuzzy.ugent.be/SCIP.

membre du groupe EUSFLAT, *(European Society for Fuzzy Logic and Technology)*, http://www.eusflat.org.

Tâches collectives

depuis juin 2009 — **élue adjointe à la vice-présidente du Conseil Scientifique (CS)**, *Université Paris 8*.

depuis 2008 — **membre élue au CS**, *Université Paris 8*.

depuis octobre 2008 — **membre du Bureau Exécutif de Cap Digital**, *le pôle de compétitivité des contenus numériques*.

depuis novembre 2007 — **membre élue au CNU**, *27^e section*.

depuis septembre 2007 — **chargée de mission à la taxe d'apprentissage auprès du président**, *Université Paris 8*.

2010 — **membre du comité de sélection (collège B, 27^e section)**, *CNAM Paris*.

2007–2008 — **membre suppléante de la commission de spécialistes (collège B, 27^e section)**, *Université de Reims*.

2006–2008 — **membre élue au CEVU**, *Université Paris 8*.

2005–2006 — **membre élue au Conseil de l'UFR 6**, *Université Paris 8*.

Publications

Revues internationales

[1] Isis Truck et Jacques Malenfant. Towards a formalization of the Linguistic Conditional Preference networks. *International Journal of Applied Management Science, special issue on "Modern Tools of Industrial Engineering: Applications in Decision Sciences"*, to appear.

[2] Pierre Châtel, Isis Truck et Jacques Malenfant. LCP-nets: A linguistic approach for non-functional preferences in a semantic SOA environment. *Journal of Universal Computer Science*, 16(1):198–217, 2010.

[3] Isis Truck et Herman Akdag. A tool for aggregation with words. *International Journal of Information Sciences, Special Issue: Linguistic Decision Making: Tools and Applications*, 179(14):2317–2324, 2009.

[4] Amine Aït Younes, Isis Truck et Herman Akdag. Image retrieval using fuzzy representation of colors. *Soft Computing — A Fusion of Foundations, Methodologies and Applications*, 11(3):287–298, 2007.

[5] Isis Truck et Herman Akdag. Manipulation of qualitative degrees to handle uncertainty : Formal models and applications. *Knowledge and Information Systems*, 9(4):385–411, 2006.

[6] Amine Aït Younes, Isis Truck et Herman Akdag. Color image profiling using fuzzy sets. *Turkish Journal of Electrical Engineering & Computer Sciences*, 13(3):343–359, 2005.

[7] Herman Akdag, Isis Truck, Amel Borgi et Nedra Mellouli. Linguistic modifiers in a symbolic framework. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. Special Issue: Computing with words: foundations and applications*, 9 (supplément):49–62, 2001.

Chapitres de livre

[8] Herman Akdag et Isis Truck. Uncertainty Operators in a Many-valued Logic. In *chapitre de l'Encyclopedia of Data Warehousing and Mining - 2nd Edition, John Wang, ed., Information Science Reference*, pages 1997–2003. 2009.

[9] Isis Truck et Herman Akdag. A Linguistic Approach of the Median Aggregator. In *chapitre du livre Fuzzy Systems Engineering Theory and Practice, Series: Studies in Fuzziness and Soft Computing*, pages 23–51. Springer, 2005.

Conférences internationales

[10] Mohammed-Amine Abchir et Isis Truck. Towards a New Fuzzy Linguistic Preference Modeling Approach for Geolocation Applications. In *The EUROFUSE Workshop on Fuzzy Methods for Knowledge-Based Systems (EUROFUSE'2011)*, to appear, 2011.

[11] Xavier Dutreilh, Sergey Kirgizov, Olga Melekhova, Jacques Malenfant, Nicolas Rivierre et Isis Truck. Using Reinforcement Learning for Autonomic Resource Allocation in Clouds: towards a fully automated workflow. In *The 7th International Conference on Autonomic and Autonomous Systems (ICAS'2011)*, pages 67–74, 2011.

[12] Alain Bonardi et Isis Truck. Introducing Fuzzy Logic And Computing With Words Paradigms In Realtime Processes For Performance Arts. In *The International Computer Music Conference (ICMC'2010)*, pages 474–477 (Poster), 2010.

[13] Pierre Châtel, Jacques Malenfant et Isis Truck. Qos-based late-binding of service invocations in adaptive business processes. In *The 8th International Conference on Web Services (ICWS)*, pages 227–234, 2010.

[14] Xavier Dutreilh, Nicolas Rivierre, Aurélien Moreau, Jacques Malenfant et Isis Truck. From Data Center Resource Allocation to Control Theory and Back. In *The 3rd IEEE International Conference on Cloud Computing (CLOUD'2010)*, pages 410–417, 2010.

[15] Olga Melekhova, Mohammed-Amine Abchir, Pierre Châtel, Jacques Malenfant, Isis Truck et Anna Pappa. Self-Adaptation in Geotracking Applications: Challenges, Opportunities and Models. In *The 2nd International Conference on Adaptive and Self-adaptive Systems and Applications (ADAPTIVE'2010)*, pages 68–77, 2010.

[16] Isis Truck et Jacques Malenfant. Towards A Unification Of Some Linguistic Representation Models: A Vectorial Approach. In *The 9th International FLINS Conference on Computational Intelligence in Decision and Control*, pages 610–615, 2010.

[17] Alain Bonardi et Isis Truck. Designing a Library For Computing [Performances] With Words. In *The 4th International Conference on Intelligent Systems & Knowledge Engineering (ISKE'2009)*, pages 40–45, 2009.

[18] Bao Le Duc, Pierre Châtel, Nicolas Rivierre, Jacques Malenfant, Philippe Collet et Isis Truck. Non-functional Data Collection for Adaptive Business Process and Decision Making. In *The 4th Middleware for Service-Oriented Computing (MW4SOC), Workshop at the ACM/IFIP/ USENIX Middleware Conference*, pages 7–12, 2009.

[19] Pierre Châtel, Isis Truck et Jacques Malenfant. A linguistic approach for non-functional preferences in a semantic SOA environment. In *The 8th International FLINS Conference on Computational Intelligence in Decision and Control*, pages 889–894, 2008.

[20] Imad El-Zakhem, Amine Aït Younes, Isis Truck, Hanna Greige et Herman Akdag. Modeling personal perception into user profile for image retrieving. In *The 8th International FLINS Conference on Computational Intelligence in Decision and Control*, pages 393–398, 2008.

[21] Imad El-Zakhem, Amine Aït Younes, Isis Truck, Hanna Greige et Herman Akdag. Color image profile comparison and computing. In *International Conference on Software and Data Technologies (ICSOFT)*, pages 228–231, 2007.

[22] Alain Bonardi et Isis Truck. First Steps Towards a Digital Assistant for Performers and Stage Directors. In *The Int. Conf. Sound & Music Computing (SMC'2006)*, pages 91–96, 2006.

[23] Alain Bonardi, Isis Truck et Herman Akdag. Building fuzzy rules in an emotion detector. In *The 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'2006)*, pages 540–546, 2006.

[24] Alain Bonardi, Isis Truck et Herman Akdag. Towards a virtual assistant for performers and stage directors. In *6th International Conference on New Interfaces for Musical Expression (NIME)*, pages 326–329 (poster), 2006.

[25] Amine Aït Younes, Isis Truck, Herman Akdag et Yannick Rémion. Image classification according to the dominant colour. In *6th International Conference on Enterprise Information Systems (ICEIS)*, pages 505–510, 2004.

[26] Amine Aït Younes, Isis Truck, Herman Akdag et Yannick Rémion. Images Retrieval Using Linguistic Expressions of Colors. In *The 6th International FLINS Conference on Computational Intelligence in Decision and Control*, pages 250–257, 2004.

[27] Isis Truck et Herman Akdag. Supervised Learning using Modifiers: Application in Colorimetrics. In *Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'03)*, page 116 (7 pages), 2003.

[28] Isis Truck, Herman Akdag et Amel Borgi. A Linguistic Approach of the Median Aggregator. In *Proceedings of the 9th International Conference on Fuzzy Theory and Technology (FT&T'03), part of the 7th Joint Conference on Information Sciences (JCIS)*, pages 147–150, 2003.

[29] Isis Truck, Herman Akdag et Amel Borgi. Comparison of fuzzy subsets: towards a linguistic approach. In *Proceedings of the 8th International Conference on Soft Computing (MENDEL)*, pages 264–269, 2002.

[30] Isis Truck, Herman Akdag et Amel Borgi. Generalized modifiers as an interval scale: towards adaptive colorimetric alterations. In *Proceedings of the 8th Iberoamerican Conference on Artificial Intelligence (IBERAMIA)*, pages 111–120, 2002.

[31] Isis Truck, Herman Akdag et Amel Borgi. A symbolic Approach for Colorimetric Alterations. In *Proceedings of the 2nd International Conference in Fuzzy Logic and Technology (EUSFLAT)*, pages 105–108, 2001.

[32] Isis Truck, Herman Akdag et Amel Borgi. Colorimetric Alterations by way of Linguistic Modifiers: A Fuzzy Approach vs. A symbolic Approach. In *Proceedings of the Symposium in International ICSC-NAISO Congress on Computational Intelligence: Methods and Applications (FLA)*, pages 702–708, 2001.

[33] Isis Truck et Jean-Michel Bazin. Automatic reasoning: Geometrical problem solving. In *The 8th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'2000)*, pages 2002–2005 (Poster), 2000.

Conférences nationales

[34] Mohamed C. Mamlouk, Amine Aït Younes, Herman Akdag et Isis Truck. Extraction des couleurs dominantes d'une image. In *Conférence en Sciences de l'Electronique, Technologies de l'Information et Telecommunications (SETIT'2007)*, page 574 (6 pages), 2007.

[35] Mohamed C. Mamlouk, Amine Aït Younes, Herman Akdag et Isis Truck. Recherche d'images couleurs par le contenu. In *International Conference on Control, Modeling and Diagnosis (ICCMD'2006)*, 2006.

[36] Amine Saïdane, Herman Akdag et Isis Truck. Une approche sma de l'agrégation de la coopération des classifieurs. In *Conférence en Sciences de l'Electronique, Technologies de l'Information et Telecommunications (SETIT'2005)*, page 126 (6 pages), 2005.

[37] Anne Sedes, Benoît Courribet, Jean-Baptiste Thiébaut, Antonio de Sousa-Dias, Alain Bonardi, Isis Truck, Vincent Lesbros et Curtis Roads. Groupe de travail 'visualisation du son'. In *12e Journées d'Informatique Musicale (JIM'2005)*, 2006.

[38] Isis Truck, Herman Akdag et Amel Borgi. Comparaison de sous-ensembles flous : compatibilité et comparabilité. In *Rencontres Francophones sur la Logique Floue et ses Applications (LFA)*, pages 135–142, 2002.

[39] Isis Truck, Francis Rousseaux et Herman Akdag. Un exemple de personnalisation de sites Web utilisant des modificateurs linguistiques. In *Journées Francophones D'Accès Intelligent aux Documents Multimedia sur l'Internet (MediaNet'2002), Hermès*, pages 319–324, 2002.

Bibliographie

[Abchir, 2011] Abchir, M. (2011). A jFuzzyLogic Extension to Deal With Unbalanced Linguistic Term Sets. In *Proc. of the 12th International Student Conference on Applied Mathematics and Informatics (ISCAMI'11)*, page à paraître.

[Abchir et Truck, 2011] Abchir, M. et Truck, I. (2011). Towards a new fuzzy linguistic preference modeling approach for geolocation applications. In *Proc. of the EUROFUSE Workshop on Fuzzy Methods for Knowledge- Based Systems*, page à paraître.

[Aït Younes et al., 2005] Aït Younes, A., Truck, I., et Akdag, H. (2005). Color image profiling using fuzzy sets. *Turkish Journal of Electrical Engineering & Computer Sciences*, 13(3) : 343–359.

[Aït Younes et al., 2007] Aït Younes, A., Truck, I., et Akdag, H. (2007). Image retrieval using fuzzy representation of colors. *Soft Computing — A Fusion of Foundations, Methodologies and Applications*, 11(3) : 287–298.

[Aït Younes et al., 2004a] Aït Younes, A., Truck, I., Akdag, H., et Rémion, Y. (2004a). Image classification according to the dominant colour. In *6th International Conference on Enterprise Information Systems (ICEIS)*, pages 505–510.

[Aït Younes et al., 2004b] Aït Younes, A., Truck, I., Akdag, H., et Rémion, Y. (2004b). Images Retrieval Using Linguistic Expressions of Colors. In *The 6th International FLINS Conference on Computational Intelligence in Decision and Control*, pages 250–257.

[Akdag et Truck, 2009] Akdag, H. et Truck, I. (2009). Uncertainty Operators in a Many-valued Logic. In *chapitre de l'Encyclopedia of Data Warehousing and Mining - 2nd Edition, John Wang, ed., Information Science Reference*, pages 1997–2003.

[Akdag et al., 2001] Akdag, H., Truck, I., Borgi, A., et Mellouli, N. (2001). Linguistic modifiers in a symbolic framework. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. Special Issue : Computing with words : foundations and applications*, 9 (supplément) : 49–62.

[Alcalá et al., 2007] Alcalá, R., Alcalá-Fdez, J., Herrera, F., et Otero, J. (2007). Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation. *Int. J. Approx. Reasoning*, 44(1) : 45–64.

[Bellman et Zadeh, 1970] Bellman, R. E. et Zadeh, L. A. (1970). Decision-Making in a Fuzzy Environment. *Management Science*, 17 : 141–164.

[Benoît, 1993] Benoît, E. (1993). *Capteurs symboliques et capteurs flous : un nouveau pas vers l'intelligence*. Thèse d'université, Université Joseph Fourier, Grenoble I.

[Bevilacqua et al., 2007] Bevilacqua, F., Guédy, F., Fléty, E., Leroy, N., et Schnell, N. (2007). Wireless sensor interface and gesture-follower for music pedagogy. In *Proc. of the Int. Conf. on New Interfaces for Musical Expression*, pages 124–129.

[Binaghi et al., 1994] Binaghi, E., Gagliardi, I., et Schettini, R. (1994). Image Retrieval Using Fuzzy Evaluation of Color Similarity. *Int. J. Pattern Recognition and Artificial Intelligence*, 8(4) : 945–968.

[Bonardi et Rousseaux, 2004] Bonardi, A. et Rousseaux, F. (2004). New Approaches of Theatre and Opera Directly Inspired by Interactive Data-Mining. In *The Int. Conf. Sound & Music Computing (SMC'2004)*, pages 1–4.

[Bonardi et Truck, 2006] Bonardi, A. et Truck, I. (2006). First Steps Towards a Digital Assistant for Performers and Stage Directors. In *The Int. Conf. Sound & Music Computing (SMC'2006)*, pages 91–96.

[Bonardi et Truck, 2009] Bonardi, A. et Truck, I. (2009). Designing a Library For Computing [Performances] With Words. In *The 4th International Conference on Intelligent Systems & Knowledge Engineering (ISKE'2009)*, pages 40–45.

[Bonardi et Truck, 2010] Bonardi, A. et Truck, I. (2010). Introducing Fuzzy Logic And Computing With Words Paradigms In Realtime Processes For Performance Arts. In *The International Computer Music Conference (ICMC'2010)*, pages 474–477 (Poster).

[Bonardi et al., 2006a] Bonardi, A., Truck, I., et Akdag, H. (2006a). Building fuzzy rules in an emotion detector. In *The 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'2006)*, pages 540–546.

[Bonardi et al., 2006b] Bonardi, A., Truck, I., et Akdag, H. (2006b). Towards a virtual assistant for performers and stage directors. In *6th International Conference on New Interfaces for Musical Expression (NIME)*, pages 326–329 (poster).

[Bonissone et Decker, 1986] Bonissone, P. et Decker, K. (1986). *Selecting Uncertainty Calculi and Granularity : An Experiment in Trading-Off Precision and Complexity.* In L.H. Kanal and J.F. Lemmer, Editors., Uncertainty in Artificial Intelligence. North-Holland.

[Bordogna et Pasi, 1993] Bordogna, G. et Pasi, G. (1993). A Fuzzy Linguistic Approach Generalizing Boolean Information Retrieval : A Model and Its Evaluation. *Journal of the American Society for Information Science (JASIS)*, 44(2) : 70–82.

[Boubekeur et Tamine-Lechani, 2006] Boubekeur, F. et Tamine-Lechani, L. (2006). Recherche d'information flexible basée CP-Nets. In *Proc. Conference on Recherche d'Information et Applications (CORIA'06)*, pages 161–167.

[Bouchon-Meunier, 1995] Bouchon-Meunier, B. (1995). *La logique floue et ses applications.* Addison-Wesley France, Paris.

[Bouchon-Meunier et al., 1996] Bouchon-Meunier, B., Rifqi, M., et Bothorel, S. (1996). Towards general measures of comparison of objects. *Fuzzy Sets and Systems*, 84 : 143–153.

[Boughorbel et al., 2002] Boughorbel, S., Boujemaa, N., et Vertan, C. (2002). Histogram-based color signatures for image indexing. In *The 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'2002)*, pages 977–984.

[Boust et al., 2003] Boust, C., Chahine, H., Viénot, F., Brettel, H., Chouikha, M. B., et Alquié, G. (2003). Color correction judgements of digital images by experts and naive observers. In *The PICS Conference, An International Technical Conference on The Science and Systems of Digital Photography*, pages 4–9.

[Boutilier et al., 2001] Boutilier, C., Bacchus, F., et Brafman, R. I. (2001). UCP-Networks : A directed graphical representation of conditional utilities. In *Proc. of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 56–64.

[Boutilier et al., 2004] Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., et Poole, D. (2004). CP-nets : A tool for representing and reasoning with conditional *Ceteris Paribus* Preference Statements. *Journal of Artificial Intelligence Research*, 21 : 135–191.

[Brafman et Domshlak, 2002] Brafman, R. I. et Domshlak, C. (2002). Introducing variable importance tradeoffs into CP-nets. In *Proc. of the Eighteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI'02)*, pages 69–76.

[Braziunas et Boutilier, 2006] Braziunas, D. et Boutilier, C. (2006). Preference elicitation and generalized additive utility. In *Proc. of the Twenty-First National Conference on Artificial Intelligence (AAAI'06)*, pages 1573–1576, Boston, MA.

[Camurri et al., 1995] Camurri, A., Catorcini, A., Innocenti, C., et Massari, A. (1995). Music and multimedia knowledge representation and reasoning : the harp system. *Computer Music Journal*, 19(2) : 34–58.

[Camurri et al., 1999] Camurri, A., Ricchetti, M., et Trocca, R. (1999). EyesWeb – toward gesture and affect recognition in dance/music interactive systems. In *The IEEE International Conference on Multimedia Systems*, pages 643–648.

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6) : 679–698.

[Châtel, 2007a] Châtel, P. (2007a). Toward a Semantic Web service discovery and dynamic orchestration based on the formal specification of functional domain knowledge. In *Proc. of the 20th International Conference & Software & Systems Engineering and their Applications (ICSSEA'07)*.

[Châtel, 2007b] Châtel, P. (2007b). Une architecture pour la découverte et l'orchestration de services Web sémantiques. In *Proc. of the First* Journées Francophones sur les Ontologies *(JFO'07)*, pages 247–264.

[Châtel et al., 2010a] Châtel, P., Malenfant, J., et Truck, I. (2010a). QoS-based Late-Binding of Service Invocations in Adaptive Business Processes. In *The 8th International Conference on Web Services (ICWS)*, pages 227–234.

[Châtel et al., 2008] Châtel, P., Truck, I., et Malenfant, J. (2008). A linguistic approach for non-functional preferences in a semantic SOA environment. In *The 8th International FLINS Conference on Computational Intelligence in Decision and Control*, pages 889–894.

[Châtel et al., 2010b] Châtel, P., Truck, I., et Malenfant, J. (2010b). LCP-nets : A linguistic approach for non-functional preferences in a semantic SOA environment. *Journal of Universal Computer Science*, pages 198–217.

[Couwenbergh, 2003] Couwenbergh, J. (2003). *Guide complet et pratique de la couleur*. Eyrolles, Paris.

[De Glas, 1989] De Glas, M. (1989). Knowledge representation in a fuzzy setting. Rapport interne 89 48, Université Paris 6.

[Degani et Bortolan, 1988] Degani, R. et Bortolan, G. (1988). The Problem of Linguistic Approximation in Clinical Decision Making. *International Journal of Approximate Reasoning*, 2 : 143–162.

[Delgado et al., 1993] Delgado, M., Verdegay, J., et Vila, M. (1993). On Aggregation Operations of Linguistic Labels. *International Journal of Intelligent Systems*, 8 : 351–370.

[Driankov et al., 1993] Driankov, D., Hellendoorn, H., et Reinfrank, M. (1993). *An introduction to fuzzy control*. Springer-Verlag.

[Dubois et al., 2003] Dubois, D., Fargier, H., et Perny, P. (2003). Qualitative decision theory with preference relations and comparative uncertainty : an axiomatic approach. *Artificial Intelligence (Special issue : Fuzzy set and possibility theory-based methods in artificial intelligence)*, 148 : 219–260.

[Dubois et al., 2002] Dubois, D., Fargier, H., Prade, H., et Perny, P. (2002). Qualitative decision theory : from Savage's axioms to nonmonotonic reasoning. *Journal of the ACM*, 49 : 455–495.

[Dubois et Prade, 2003] Dubois, D. et Prade, H. (2003). Informations bipolaires. Une introduction. *Information-Interaction-Intelligence*, 3(1) : 89–106.

[Dutreilh et al., 2011] Dutreilh, X., Kirgizov, S., Melekhova, O., Malenfant, J., Rivierre, N., et Truck, I. (2011). Using Reinforcement Learning for Autonomic Resource Allocation in Clouds : towards a fully automated workflow. In *The 7th International Conference on Autonomic and Autonomous Systems (ICAS'2011)*, page à paraître.

[Dutreilh et al., 2010] Dutreilh, X., Rivierre, N., Moreau, A., Malenfant, J., et Truck, I. (2010). From Data Center Resource Allocation to Control Theory and Back. In *The 3rd IEEE International Conference on Cloud Computing (CLOUD'2010)*, pages 410–417.

[Eigenfeldt, 2007] Eigenfeldt, A. (2007). Drum Circle : Intelligent Agents in Max/MSP. In *Proc. of the International Computer Music Conference*.

[El-Zakhem et al., 2007] El-Zakhem, I., Aït Younes, A., Truck, I., Greige, H., et Akdag, H. (2007). Color image profile comparison and computing. In *International Conference on Software and Data Technologies (ICSOFT)*, pages 228–231.

[El-Zakhem et al., 2008] El-Zakhem, I., Younes, A. A., Truck, I., Greige, H., et Akdag, H. (2008). Modeling personal perception into user profile for image retrieving. In *The 8th International FLINS Conference on Computational Intelligence in Decision and Control*, pages 393–398.

[Elsea, 1995] Elsea, P. (1995). *Fuzzy Logic and Musical Decisions*. University of California, Santa Cruz. `http://arts.ucsc.edu/EMS/Music/research/FuzzyLogicTutor/FuzzyTut.html`.

[FFLL, 2003] FFLL (2003). Free fuzzy logic library. `http://ffll.sourceforge.net/index.html`.

[Fontaine, 1998] Fontaine, G. (1998). *Le décor d'opéra*. Editions Plume, Paris.

[Friberg, 2005] Friberg, A. (2005). A fuzzy analyzer of emotional expression in music performance and body motion. In Brunson, W. et Sundberg, J., éditeurs, *Proceedings of Music and Music Science, Stockholm 2004*.

[Frigui, 2001] Frigui, H. (2001). Interactive image retrieval using fuzzy sets. *Pattern Recognition Letters*, 22(9) : 1021–1031.

[Fuzzy Toolbox, 2007] Fuzzy Toolbox (2007). Fuzzy logic toolbox™ 2.2.9. MATLAB. `http://www.mathworks.com/products/fuzzylogic/`.

[FuzzyJ, 2006] FuzzyJ (2006). The NRC FuzzyJ Toolkit. National Research Council. `http://www.iit.nrc.ca/IR_public/fuzzy/fuzzyJToolkit2.html`.

[Gonzales et al., 2008] Gonzales, C., Perny, P., et Queiroz, S. (2008). GAI-Networks : Optimization, Ranking and Collective Choice in Combinatorial Domains. *Foundations of computing and decision sciences*, 32(4) : 3–24.

[Han et Ma, 2002] Han, J. et Ma, K.-K. (2002). Fuzzy color histogram and its use in color image retrieval. *IEEE Transactions on Image Processing*, 11(8) : 944–952.

[Herrera et al., 2001] Herrera, F., Herrera-Viedma, E., et Martínez, L. (2001). A Hierarchical Ordinal Model for Managing Unbalanced Linguistic Term Sets Based on the Linguistic 2-Tuple Model. In *EUROFUSE Workshop on Preference Modelling and Applications*, pages 201–206.

[Herrera et Martínez, 2000] Herrera, F. et Martínez, L. (2000). A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Trans. Fuzzy Systems*, 8(6) : 746–752.

[Herrera et Martínez, 2001] Herrera, F. et Martínez, L. (2001). A model based on linguistic 2-tuples for dealing with multigranular hierarchical linguistic contexts in multi-expert decision-making. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 31(2) : 227–234.

[Herrera et al., 2002] Herrera, F., Martínez, L., Herrera-Viedma, E., et Chiclana, F. (2002). Fusion of Multigranular Linguistic Information based on the 2-tuple Fuzzy Linguistic Representation Model. In *Proceedings of IPMU 2002*, pages 1155–1162.

[Hildebrand et Reusch, 2000] Hildebrand, L. et Reusch, B. (2000). Fuzzy Color Processing. In *chapitre du livre Studies in Fuziness and Soft Computing, Vol. 52*. Physica-Verlag, Heidelberg.

[IEC, 2001] IEC (2001). IEC 61131-7 Fuzzy Control Programming.

[Itten, 1961] Itten, J. (1961). *The Art of Color*. New York : Reinhold Publishing.

[jFuzzyLogic, 2008] jFuzzyLogic (2008). Open source fuzzy logic library and FCL language implementation. `http://jfuzzylogic.sourceforge.net/html/index.html`.

[Kahol et al., 2004] Kahol, K., Tripathi, P., et Panchanathan, S. (2004). Automated Gesture Segmentation From Dance Sequences. In *The Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'2004)*, pages 883–888.

[Kalyvianaki et al., 2009] Kalyvianaki, E., Charalambous, T., et Hand, S. (2009). Self-adaptive and self-configured cpu resource provisioning for virtualized servers using kalman filters. In *ICAC '09 : Proceedings of the 6th international conference on Autonomic computing*, pages 117–126. ACM.

[Kaufmann, A., 1975] Kaufmann, A. (1975). *Introduction to the Theory of Fuzzy Subsets*. Academic Press, New York.

[Lausen et Innsbruck, 2007] Lausen, H. et Innsbruck, D. (2007). *Semantic Annotations for WSDL and XML Schema*.

[Le Duc et al., 2009a] Le Duc, B., Châtel, P., Rivierre, N., Malenfant, J., Collet, P., et Truck, I. (2009a). Non-functional Data Collection for Adaptive Business Process and Decision Making. In *The 4th Middleware for Service-Oriented Computing (MW4SOC), Workshop at the ACM/IFIP/ USENIX Middleware Conference*, pages 7–12.

[Le Duc et al., 2009b] Le Duc, B., Châtel, P., Rivierre, N., Malenfant, J., Collet, P., et Truck, I. (2009b). Non-functional Data Collection for Adaptive Business Process and Decision Making. In *In proceedings of MW4SOC'09 workshop [to be published].*

[Lim et al., 2009] Lim, H. C., Babu, S., Chase, J. S., et Parekh, S. S. (2009). Automated control in cloud computing : challenges and opportunities. In *ACDC '09 : Proceedings of the 1st workshop on Automated control for datacenters and clouds*, pages 13–18. ACM.

[Łukasiewicz, 1920] Łukasiewicz, J. (1920). O logice trójwartościowej. (Traduction anglaise : On Three-Valued Logic. In : *Jan Łukasiewicz Selected Works*. L. Borkowski, ed., North Holland, 87–88, 1990). *Ruch Filozoczny*, 5 : 169–171.

[Mamlouk et al., 2007] Mamlouk, M. C., Aït Younes, A., Akdag, H., et Truck, I. (2007). Extraction des couleurs dominantes d'une image. In *Conférence en Sciences de l'Electronique, Technologies de l'Information et Telecommunications (SETIT'2007)*, page 574 (6 pages).

[Maurin, 2009] Maurin, M. (2009). Lorsque l'utilité, la gêne ou le confort sont recueillis sur une échelle de catégories : l'interaction dans le contexte multivarié. *Mathématiques et sciences humaines*, 188 : 5–39.

[Melekhova et al., 2010] Melekhova, O., Abchir, M.-A., Châtel, P., Malenfant, J., Truck, I., et Pappa, A. (2010). Self-Adaptation in Geotracking Applications : Challenges, Opportunities and Models. In *The 2nd International Conference on Adaptive and Self-adaptive Systems and Applications (ADAPTIVE'2010)*, pages 68–77. IEEE.

[Miyata et al., 1995] Miyata, Y., Furuhashi, T., et Uchikawa, Y. (1995). A study on fuzzy abductive inference. In *Proc. of the International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium*, pages 337–342.

[Moreno-Vozmediano et al., 2009] Moreno-Vozmediano, R., Montero, R. S., et Llorente, I. M. (2009). Elastic management of cluster-based services in the cloud. In *ACDC '09 : Proceedings of the 1st workshop on Automated control for datacenters and clouds*, pages 19–24. ACM.

[Omhover et Detyniecki, 2004] Omhover, J.-F. et Detyniecki, M. (2004). Strict : An image retrieval platform for queries based on regional content. In *The Third International Conference on Image and Video Retrieval (CIVR)*, pages 473–482.

[Revault d'Allonnes et al., 2007] Revault d'Allonnes, A., Akdag, H., et Bouchon-Meunier, B. (2007). Selecting implications in fuzzy abductive problems. In *IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, pages 597–602.

[Revault d'Allonnes et al., 2009] Revault d'Allonnes, A., Akdag, H., et Bouchon-Meunier, B. (2009). For a data-driven interpretation of rules, wrt gma conclusions, in abductive problems. *Journal of Uncetain Systems*, 3(4) : 280–297.

[Roire, 2000] Roire, J. (2000). Les noms des couleurs. *Pour la Science, Hors série*, 27.

[Salehie et Tahvildari, 2005] Salehie, M. et Tahvildari, L. (2005). A policy-based decision making approach for orchestrating autonomic elements. In *STEP '05 : Proceedings of the 13th IEEE International Workshop on Software Technology and Engineering Practice*, pages 173–181, Washington, DC, USA. IEEE Computer Society.

[Salton et McGill, 1983] Salton, G. et McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company.

[Savage, 1954] Savage, L. J. (1954). *The Foundations of Statistics*. Wiley, New York.

[Schröpfer et al., 2007] Schröpfer, C., Binshtok, M., Shimony, S. E., Dayan, A., Brafman, R., Offermann, P., et Holschke, O. (2007). Introducing preferences over NFPs into service selection in SOA. In *Proc. Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'07)*.

[Sugano, 2001] Sugano, N. (2001). Color-naming system using fuzzy set theoretical approach. In *The 10th IEEE International Conference on Fuzzy Systems*, pages 81–84.

[Swain et Ballard, 1991] Swain, M. J. et Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, 7(1) : 11–32.

[Tesauro et al., 2006] Tesauro, G., Jong, N. K., Das, R., et Bennani, M. N. (2006). A hybrid reinforcement learning approach to autonomic resource allocation. In *ICAC '06 : Proceedings of the 2006 IEEE International Conference on Autonomic Computing*, pages 65–73. IEEE Computer Society.

[Truck, 2002] Truck, I. (2002). *Approches symbolique et floue des modificateurs linguistiques et leur lien avec l'agrégation*. Thèse d'université, Université de Reims.

[Truck et Akdag, 2003] Truck, I. et Akdag, H. (2003). Supervised Learning using Modifiers : Application in Colorimetrics. In *Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'03)*, page 116 (7 pages).

[Truck et Akdag, 2005] Truck, I. et Akdag, H. (2005). A Linguistic Approach of the Median Aggregator. In *chapitre du livre Fuzzy Systems Engineering Theory and Practice, Series : Studies in Fuzziness and Soft Computing*, pages 23–51. Springer.

[Truck et Akdag, 2006] Truck, I. et Akdag, H. (2006). Manipulation of qualitative degrees to handle uncertainty : Formal models and applications. *International Journal of Knowledge and Information Systems*, 9(4) : 385–411.

[Truck et Akdag, 2009] Truck, I. et Akdag, H. (2009). A tool for aggregation with words. *International Journal of Information Sciences, Special Issue : Linguistic Decision Making : Tools and Applications*, 179(14) : 2317–2324.

[Truck et al., 2001a] Truck, I., Akdag, H., et Borgi, A. (2001a). A symbolic Approach for Colorimetric Alterations. In *Proceedings of the 2nd International Conference in Fuzzy Logic and Technology (EUSFLAT)*, pages 105–108.

[Truck et al., 2001b] Truck, I., Akdag, H., et Borgi, A. (2001b). Colorimetric Alterations by way of Linguistic Modifiers : A Fuzzy Approach vs. A symbolic Approach. In *Proceedings of the Symposium in International ICSC-NAISO Congress on Computational Intelligence : Methods and Applications (FLA)*, pages 702–708.

[Truck et al., 2002a] Truck, I., Akdag, H., et Borgi, A. (2002a). Comparaison de sous-ensembles flous : compatibilité et comparabilité. In *Rencontres Francophones sur la Logique Floue et ses Applications (LFA)*, pages 135–142.

[Truck et al., 2002b] Truck, I., Akdag, H., et Borgi, A. (2002b). Comparison of fuzzy subsets : towards a linguistic approach. In *Proceedings of the 8th International Conference on Soft Computing (MENDEL)*, pages 264–269.

[Truck et al., 2002c] Truck, I., Akdag, H., et Borgi, A. (2002c). Generalized modifiers as an interval scale : towards adaptive colorimetric alterations. In *Proceedings of the 8th Iberoamerican Conference on Artificial Intelligence (IBERAMIA)*, pages 111–120.

[Truck et al., 2003] Truck, I., Akdag, H., et Borgi, A. (2003). A Linguistic Approach of the Median Aggregator. In *Proceedings of the 9th International Conference on Fuzzy Theory and Technology (FT&T'03), part of the 7th Joint Conference on Information Sciences (JCIS)*, pages 147–150.

[Truck et Malenfant, 2010] Truck, I. et Malenfant, J. (2010). Towards A Unification Of Some Linguistic Representation Models : A Vectorial Approach. In *The 9th International FLINS Conference on Computational Intelligence in Decision and Control*, pages 610–615.

[Truck et Malenfant, 2011] Truck, I. et Malenfant, J. (2011). Towards a formalization of the LCP-nets. *International Journal of Applied Management Science (Special Issue on Modern Tools of Industrial Engineering : Applications in Decision Sciences)*, page to appear.

[Vertan et Boujemaa, 2000] Vertan, C. et Boujemaa, N. (2000). Using fuzzy histograms and distances for color image retrieval. In *The Challenge of Image Retrieval (CIR'2000), Brighton*, pages 1–6.

[Xu et al., 2007] Xu, J., Zhao, M., Fortes, J., Carpenter, R., et Yousif, M. (2007). On the use of fuzzy modeling in virtualized data center management. In *ICAC '07 : Proceedings of the Fourth International Conference on Autonomic Computing*, page 25. IEEE Computer Society.

[Xu, 2008] Xu, Z. (2008). *Linguistic Aggregation Operators : An Overview*, volume 220, pages 163–181. Springer Verlag. ISBN : 978-3-540-73722-3.

[Yager, 1981] Yager, R. R. (1981). A new methodology for ordinal multiple aspect decisions based on fuzzy sets. *Decision Sciences*, 12(58) : 600.

[Yager, 1986] Yager, R. R. (1986). On the Theory of Bags. *International Journal of General Systems*, 13 : 23–37.

[Yager, 1988] Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Syst. Man Cybern.*, 18(1) : 183–190.

[Yager, 2007] Yager, R. R. (2007). Using Stress Functions to Obtain OWA Operators. *IEEE Trans. on Fuzzy Systems*, 15(6) : 1122–1129.

[Zadeh, 1965] Zadeh, L. A. (1965). Fuzzy sets. *Information Control*, 8 : 338–353.

[Zadeh, 1971] Zadeh, L. A. (1971). Quantitative fuzzy semantics. *Information Sciences*, 3(2) : 159–176.

[Zadeh, 1975] Zadeh, L. A. (1975). The Concept of a Linguistic Variable and Its Applications to Approximate Reasoning. *Information Sciences, Part I, II, III*, 8,8,9 : 199–249, 301–357, 43–80.

[Zadeh, 1978] Zadeh, L. A. (1978). PRUF – a meaning representation language for natural languages. *Int. J. Man-Machine Studies*, 10 : 395–460.

[Zadeh, 1996] Zadeh, L. A. (1996). Fuzzy logic = computing with words. *IEEE Trans. on Fuzzy Systems*, 4 : 103–111.

[Zadeh, 2002] Zadeh, L. A. (2002). From computing with numbers to computing with words : From manipulation of measurements to manipulation of perceptions. *Int. J. of Applied Math and Computer Science*, 12(3) : 307–324.

[Zadeh, 2005] Zadeh, L. A. (2005). Toward a generalized theory of uncertainty (GTU) — an outline. *International Journal of Information Sciences*, 172(1–2) : 1–40.

[Zeng et al., 2004] Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J., et Chang, H. (2004). QoS-aware middleware for web services composition. *IEEE Transactions on software engineering*, 30(5) : 311–327.

[Zhou et al., 2004] Zhou, C., Chia, L., et Lee, B. (2004). DAML-QoS ontology for web services. In *Proceedings of the IEEE international conference on web services*, page 472. IEEE Computer Society.