HAL
open science

La propriété dans tous ses états

Emmanuel Desmontils, Jean-Yves Martin

To cite this version:

Emmanuel Desmontils, Jean-Yves Martin. La propriété dans tous ses états. [Rapport de recherche] RR-IRIN-148, 1997, pp.34. hal-00816891

HAL Id: hal-00816891

https://inria.hal.science/hal-00816891

Submitted on 23 Apr 2013

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

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

LA PROPRIÉTÉ DANS TOUS SES ÉTATS

Rapport de recherche IRIN - n°148

Emmanuel DESMONTILS
Jean-Yves MARTIN

IRIN - MGII
Unversité de Nantes

NANTES - MARS 1997[1]

[1] E. Desmontils, J.-Y. Martin, "Properties Taxonomy in Declarative Modeling", International Conference on Imaging Science, Systems and Technology (CISST'97), H. R. Arabnia ed., Las Vegas, USA, 30 juin - 3 juillet

PLAN

1. INTRODUCTION 4

2. RAPPELS DES NOTIONS ÉLÉMENTAIRES 4

3. CLASSIFICATION SYNTAXIQUE 5

3.1 PROPRIÉTÉS DESCRIPTIVES 5
3.1.1 Propriétés prédicats 6
3.1.2 Signe d'une propriété 7
3.1.3 Cas des propriétés paramétrées 7
3.1.4 Propriétés prédicats et quantificateurs 8
3.1.4.1 Le quantificateur existentiel et le quantificateur universel 8
3.1.4.2 Les quantifications intermédiaires 8
3.1.4.3 Forme d'une propriété prédicat quantifiée 8
3.1.5 Propriété prédicat globale et propriété prédicat locale 9
3.1.6 Propriétés anonymes et propriétés nommées 9
3.1.7 Propriétés de concepts n-aires 10
3.1.7.1 Propriétés de concepts binaires 10
3.1.7.2 Propriétés quantifiées de concepts binaires 10
3.1.7.3 Propriétés de concepts n-aires 10
3.1.8 Propriétés de relations n-aires 10
3.1.8.1 Propriétés de relation binaires 10
3.1.8.2 Propriétés n-aires (n>2) 11
3.2 AFFIRMATIVE ET NÉGATIVE 11
3.3 PROPRIÉTÉS CONSTRUCTIVES 12
3.4 PROPRIÉTÉS MODIFICATRICES 12
3.4.1 Propriété modificatrice 12
3.4.2 Propriété modificatrice quantifiée et relative 13

4. CLASSIFICATION STRUCTURELLE 13

4.1 RAPPELS SUR LA STRUCTURE D'UNE PROPRIÉTÉ 13
4.2 POSITION DE LA FONCTION D'APPARTENANCE 14
4.2.1 Propriétés à valeur absolue 14
4.2.2 Propriétés à valeur relative 14
4.3 PROPRIÉTÉ MODIFIABLE 14
4.4 FORME 15
4.5 DIMENSIONS DU DOMAINE 15

5. CLASSIFICATION COMPORTEMENTALE 16

5.1 RAPPELS SUR LES MÉTHODES DE GÉNÉRATION 16
5.2 LES PARCOURS D'ARBRES 16
5.2.1 Généralités 16
5.2.2 Quelques définitions utiles 16
5.2.2.1 Définitions à propos des mesures des domaines 16
5.2.2.2 Définitions à propos des fonctions d'appartenance 17
5.2.2.3 Seuil d'acceptation 17
5.2.3 Synthèse de différents types de comportements 17
5.2.4 Les propriétés stables 18
5.2.5 Propriétés quasi-stables 19
5.2.6 Elagage et stérilisation des propriétés 19
5.2.7 Optimisations automatiques 20

5.2.7.1 Hypothèses....20
5.2.7.2 Formes des propriétés et stabilité....20
5.2.7.3 Détections automatiques sur la stabilité d'une propriété....20
5.2.7.4 Notion de valeur critique....21
5.2.7.5 Cas particulier des mesures strictement croissantes ou strictement décroissantes....22
5.2.8 Concepts spécifiques....22
5.2.8.1 Mesures incrémentales....22
5.2.8.2 Propriétés incrémentales quantitatives (quantités ou proportions)....22
5.2.8.3 Propriétés incrémentales statistiques....23
5.3 MOTEURS D'INFÉRENCES....23
5.3.1 Les règles de création....23
5.3.2 Les règles de suppléance....24
5.3.3 Les règles de modification....25
5.3.4 Les règles de vérification....25
5.3.5 Les règles internes....25
5.3.6 Les MétaRègles....26
5.3.7 Ordre d'application des règles....26
5.4 AUTRES TECHNIQUES....27
5.4.1 Réduction d'intervalles....27
5.4.2 Tirages aléatoires....27
5.5 APPRENTISSAGE....28
5.5.1.1 Apprentissage de l'utilisateur....28
5.5.1.2 Apprentissage de l'univers....28
5.6 INTERVENTION DES PROPRIÉTÉS....28
5.6.1 Les propriétés de vérification....29
5.6.2 Les propriétés de génération....29
5.6.3 Les propriétés de modification....29
5.7 RELATIONS ENTRE PROPRIÉTÉS....29
5.7.1 Niveau d'une propriété....29
5.7.2 Dépendance d'instanciation....29
5.7.3 Dépendance sémantique....30
5.7.4 Dépendance de construction....30
5.7.5 Propriétés de composition....30
5.7.5.1 L'opérateur conjonctif "."....31
5.7.5.2 L'opérateur conjonctif "Et"....31
5.7.5.3 L'opérateur disjonctif "Ou"....31
5.7.5.4 L'opérateur disjonctif "Ou Bien"....31
6. CLASSIFICATION SÉMANTIQUE....31
6.1 LES CONCEPTS « COURANTS »....32
6.1.1 Décrire un objet....32
6.1.2 Décrire un ensemble d'objets....32
6.2 LES CONCEPTS CONSTRUITS DE MANIÈRE AUTOMATIQUE....32
7. CONCLUSION....32

1. Introduction

La modélisation déclarative s'attache à construire un ensemble de solutions correspondant à une description effectuée à l'aide de divers médias. Il s'ensuit que la description d'une scène constitue un élément clé du processus déclaratif.

Depuis [LMM89], un certain nombre de projets ont vu le jour et ont permis de mettre en évidence des connaissances et des techniques indispensables à un modeleur déclaratif. Citons par exemple les projets PolyFormes (modélisation déclarative de polyèdres [MaM90]), AutoFormes (modélisation déclarative d'automates cellulaires [Mar90b]), VoluFormes (contrôle spatial déclaratif [Chau94]), CurviFormes (modélisation déclarative de courbes [Dan95]). Ces projets ont tous en commun de traiter un sujet spécifique et ne sont pas directement adaptables à d'autres sujets. Cependant, la plupart d'entre eux mettent en oeuvre des techniques similaires. Nous nous sommes donc intéressés à la mise en oeuvre d'un ensemble d'outils, une plate-forme (le projet CordiFormes), visant à faciliter la mise en oeuvre de futurs modeleurs déclaratifs.

Parmi les outils à mettre en oeuvre, nous nous intéressons entre autres à ceux permettant d'exploiter une description. Celle-ci est généralement vue comme une liste de propriétés. La propriété, trait caractéristique de la scène par rapport à un concept donné, est donc une notion centrale en modélisation déclarative. Notre objectif est d'une part de mieux comprendre le fonctionnement d'une description et d'autre part de proposer des traitements automatiques, en particulier pour une interprétation fine de la description, une gestion plus efficace de la cohérence et une optimisation performante de la génération des solutions.

Pour atteindre cet objectif, nous devons nécessairement comprendre le mécanisme de fonctionnement des propriétés utilisées pour la description. Ce travail demande une étude de toutes les facettes d'une propriété ainsi qu'une formalisation cohérente.

Après un bref rappel des notions élémentaires utilisées, nous présenterons un classement syntaxique des propriétés de la description. Nous aborderons ensuite des éléments de classification structurelle par rapport à la méthode de représentation des propriétés choisie, et une classification comportementale vis à vis de la génération des solutions. Enfin, nous proposerons une première classification du point de vue sémantique.

2. Rappels des notions élémentaires

Une scène est caractérisée par plusieurs <u>concepts</u> C_i ($i \in J \subset N$). Par exemple, un bâtiment est caractérisé par sa hauteur, sa largeur, sa forme, son style, sa couleur, son nombre de fenêtres, son nombre de sorties… Pour le concept C_i, le <u>domaine de description</u> D_i est l'ensemble des valeurs qu'il est susceptible de recevoir. La <u>mesure de D_i</u> est une fonction de U_f (univers des scènes possibles) dans D_i qui à toute scène de U_f associe une valeur de D_i. A chaque concept C_i est associé un ensemble de propriétés P_{ik} (appelées <u>propriétés de base</u>) se référant au domaine D_i. Etant donné que ces propriétés sont souvent de nature imprécises ou vagues, la théorie des sous-ensembles flous ([Zad65]) peut être utilisée pour les représenter.

La Figure 1 présente trois propriétés de base définies sur un domaine D_i d'un concept C_i. Généralement, deux cas se présentent :

- le concept ne contient qu'une seule propriété de base dont le nom est issu de celui du concept (ou par défaut « vérifiée »).
- le concept contient trois propriétés de base : « faible » , « moyen » et « important » qui prend souvent un nom issu de celui du concept.

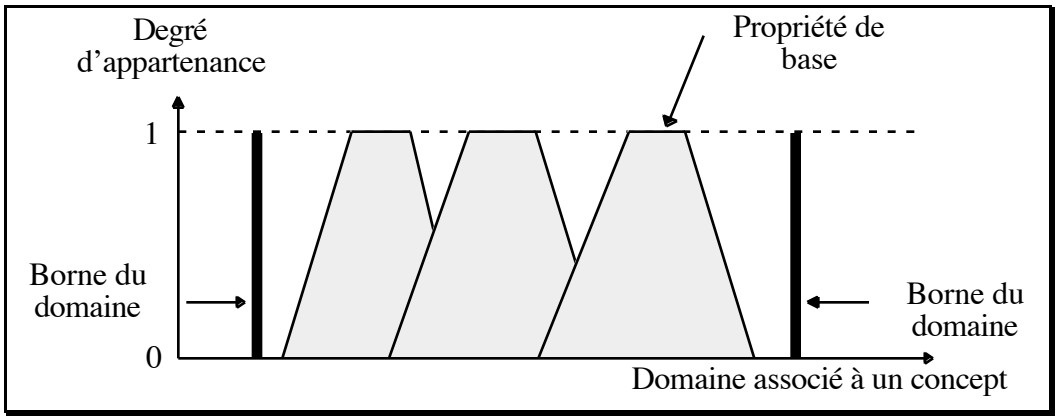

Figure 1 : Domaine et propriétés de bases d'un concept donné

Ces cas sont justifiables au niveau linguistique. [DSc95] présente une notion intéressante : la catégorie sémantique ([Bro43], [Bro48], [Hje35], [Hje37]). Il est possible de définir les catégories sémantiques (pour nous, les concepts) à partir d'une catégorie maximale constituée des unités sémantiques (pour nous, les propriétés) suivantes :

- deux termes, I (positif) et F (négatif), disjoints présentant deux qualités incompatibles,
- un terme neutre M qui indique l'absence de l'une et l'autre de ces qualités (la non-application de la catégorie),
- un terme complexe C qui recouvre à la fois I et F (indique seulement l'application de la catégorie),
- deux termes IM et FM équivalents à C mais avec insistance soit sur I soit sur F.

Toutes les catégories sont construites avec un sous-ensemble des unités sémantiques de cette catégorie maximale avec cependant des contraintes de symétries sur les éléments « signés ».

Du point de vue d'une description, ce seront les éléments I, F et M qui seront le plus souvent présents. Ce sont les termes simples de base. Ce qui revient à pouvoir construire les catégories suivantes : {I}, {F}, {F, I} et {F, M, I}.

Par exemple, avec le concept « longueur d'un segment », nous aurons naturellement les trois propriétés de base « faible » (ou « court »), « moyen » et « important » (ou « long »). De même, avec le concept « densité de population », nous aurons la propriété de base « dense ».

3. Classification syntaxique

Pour construire une scène à l'aide d'un modeleur déclaratif, le concepteur doit fournir une description. Celle-ci est formée d'une liste de propriétés. Nous nous intéressons à la « forme » de ces propriétés. Elles se regroupent en deux grandes catégories : les propriétés permettant de construire la description initiale et celles modifiant la description en cours de génération.

Les schémas standard que nous proposons permettent d'interpréter cette description au niveau syntaxique et sont utilisés pour construire automatiquement le modèle de description nécessaire à la génération.

3.1 Propriétés descriptives

Ces propriétés permettent de construire une description déterminant le sous-univers des formes plausibles. Elles se classent en plusieurs catégories selon leur structure. La première catégorie de propriétés concerne les propriétés descriptives. Elles permettent de construire une description menant à la construction de l'ensemble des formes plausibles.

3.1.1 Propriétés prédicats

Ce sont les propriétés les plus élémentaires. Elles correspondent à des prédicats logiques du premier ordre et portent sur la forme entière, sur une partie de la forme ou sur un élément. Elles correspondent à des énoncés de la forme :

« C_i [de X] est f_α m_β P_{ik} » .

Où X est un objet de la scène ou d'une sous-scène (implicite ou explicite), f_α est un opérateur flou et m_β est un modificateur de la propriété de base P_{ik}. Elles ont été étudiées sous le nom de « propriétés simples » précisément dans [Des95b], [Des96] et [DPa96] (Figure 2). Nous pensons que le terme « propriété prédicat » est plus proche de la notion que cette catégorie de propriétés représente.

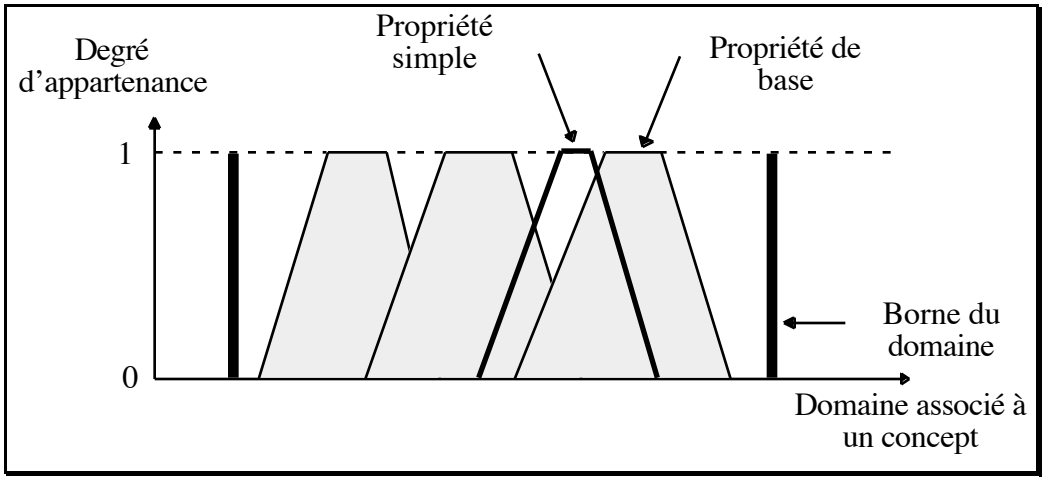

Figure 2 : Propriétés de base et propriétés prédicats

Les modificateurs possibles sont très nombreux. Nous avons restreint notre choix à un ensemble de P modificateurs, noté : $\boldsymbol{M}_P=\{m_\alpha \mid \alpha\in[1..P]\}$, pour lesquels existe la relation d'ordre total suivante : $m_\alpha \leq m_\beta \Leftrightarrow \alpha \leq \beta$. $\boldsymbol{M}_P$ comporte un modificateur particulier « normalement » également noté « Ø ». C'est l'opérateur par défaut ou modificateur vide. Ainsi, un énoncé de la forme « x est P_{ik} » est équivalent à l'énoncé d'une propriété prédicat « x est Ø P_{ik} » ou « x est normalement P_{ik} ». Il est donc implicite dans l'énoncé « x est important ». Dans ce travail, nous avons utilisé l'ensemble suivant : $\boldsymbol{M}_7$={extrêmement peu, très peu, assez peu, normalement, assez, très, extrêmement} (Figure 3a). On peut noter le cas particulier du modificateur « très » qui peut être répété plusieurs fois (deux en général).

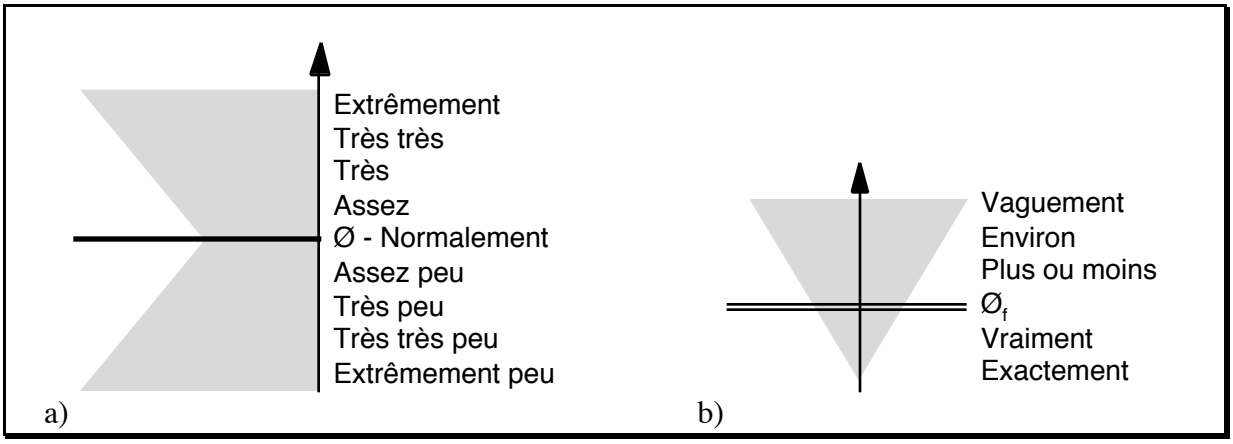

Figure 3 : Modificateurs et opérateurs flous choisis

Les opérateurs flous possibles étant très nombreux, nous avons restreint notre choix à un ensemble noté $\boldsymbol{F}_Q=\{f_\alpha \mid \alpha \in [1..Q]\}$ muni de la relation d'ordre total : $f_\alpha \leq f_\beta \Leftrightarrow \alpha \leq \beta$. $\boldsymbol{F}_Q$ comporte un opérateur par défaut noté « $Ø_f$ ». Nous avons choisi l'ensemble $\boldsymbol{F}_6$={exactement, vraiment, $Ø_f$, plus ou moins, environ, vaguement} (Figure 3b).

Nous remarquons que l'énoncé « C_i [de X] est P_{ik} » est donc équivalent à « C_i [de X] est $Ø_f$ Ø P_{ik} ».

Exemples :

- « Le nombre d'intersections [de la scène] est assez important » où « Le nombre d'intersections » est le concept , « assez » un modificateur et « important » une propriété de base du concept,
- « La longueur du segment est vraiment très faible » où « La longueur du segment » est le concept, « vraiment » un opérateur flou, « très » un modificateur et « faible » une propriété de base du concept.

Les propriétés portant sur l'ensemble de la scène sont parfois appelées « propriétés globales » et celles portant sur une partie de la scène (sous-ensemble d'une scène structurée ou élément de base) « propriétés locales ». Cependant, il semble que cette classification ne soit pas judicieuse. La séparation entre ces deux ensembles n'est absolument pas nette. Une propriété globale est une propriété locale sur l'objet (unique) qu'est la scène.

3.1.2 Signe d'une propriété

Pour manipuler les propriétés prédicats, on a besoin de déterminer son signe. Or celui-ci est fonction de la propriété de base utilisée ainsi que des modificateurs.

Le <u>signe de la propriété de base</u> P_{ik} (+1, -1 ou 0) est celui de la direction suivie par les fonctions d'appartenance de $m_\alpha P_{ik}$ où $m_\alpha \in \boldsymbol{M}_P$ et α allant de 1 à P par rapport à la variable du domaine.

Une propriété P_{ik} est dite <u>positive</u> si son signe est +1. Elle est dite <u>négative</u> si son signe est -1 et <u>neutre</u> si le signe est 0. Nous retrouvons là les notions relatives aux termes linguistiques que l'on a présenté au paragraphe 2.

Le <u>signe d'un modificateur</u> m_α, noté signe(m_α), est défini dans $\boldsymbol{M}_P$ comme-suit : signe(m_a) = 0 si m_a = Ø, signe(m_α) = +1 si $\alpha > a$ et signe(m_α) = -1 si $\alpha < a$.

Par conséquent, le <u>signe d'une propriété prédicat</u> dépend du signe de la propriété de base et de celui du modificateur suivant la formule : signe(m_α) * signe(P_{ik}) $\in \{-1, 0, +1\}$.

3.1.3 Cas des propriétés paramétrées

Parfois, une description utilise une propriété faisant référence à une ou plusieurs valeurs du domaine. A partir des propriétés élémentaires, nous avons alors besoin de construire des propriétés paramétrées par une ou deux valeurs du domaine. La propriété de base P_{ik} est remplacée par une propriété de base paramétrée P_j(U) ou P_j (U,V). Les énoncés de propriétés paramétrées sont de trois formes (Figure 4) :

1) « C_i [de X] est f_α Ø de V » ou « C_i [de X] est f_α Ø entre U et V »,
2) « C_i [de X] est f_α m_β supérieur à V » ou « C_i [de X] est f_α m_β inférieur à V »,
3) « C_i [de X] est $Ø_f$ Ø au moins de V » ou « C_i [de X] est $Ø_f$ Ø au plus de V ».

On note que ces formes se rapprochent de celle des propriétés prédicats. Une propriété sur une valeur précise (« Le longueur de la configuration est environ de 3 unités ») et sur un intervalle classique (« Le nombre de maisons est entre 10 et 20 ») sont donc des propriétés paramétrées de forme 1. L'utilisation d'opérateur flou permet de transformer une valeur précise en valeur floue et un intervalle classique en intervalle flou.

Les énoncés de forme 3 correspondent aussi à des intervalles classiques entre la valeur donnée et une des deux bornes du domaine (« La hauteur de la maison est au moins de 4 mètres »). Ils ne supportent aucun opérateur.

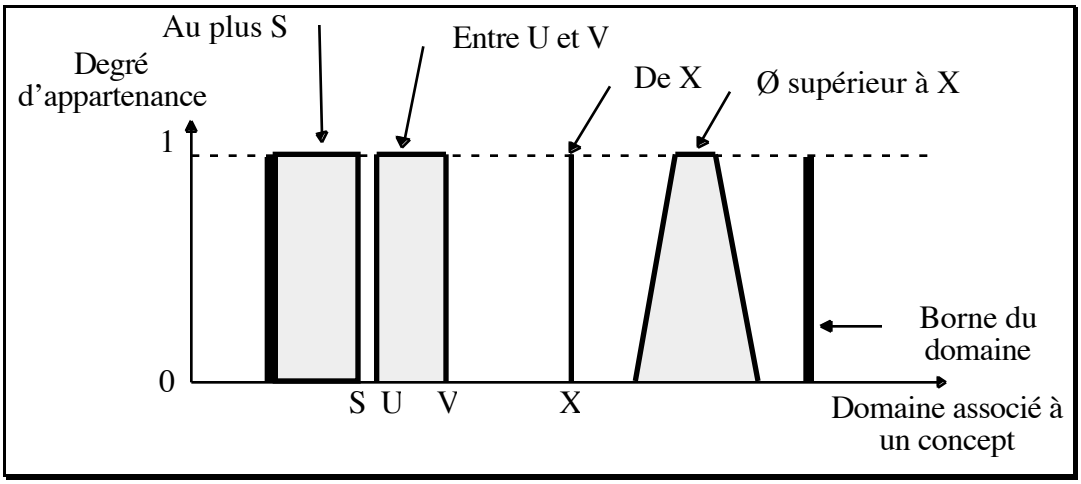

Figure 4 : Exemples de propriétés prédicats paramétrées

Avec les énoncés de forme 2, nous obtenons des intervalles flous associés à des propriétés de base qui ne sont pas explicitement paramétrées (« La surface du terrain est très supérieure à 3 Ha »). Cet intervalle se trouve à une distance plus ou moins importante de la valeur proposée en fonction de l'opérateur et du modificateur utilisé. Cela revient à déterminer une propriété de base sur un sous-domaine construit à partir de la valeur et d'une borne du domaine.

3.1.4 Propriétés prédicats et quantificateurs

Avec certains concepts, la propriété prédicat ne s'applique plus forcément sur un seul objet mais sur tout un ensemble d'objets. Il est alors nécessaire de préciser un quantificateur. Comme en logique du second ordre, il est possible de trouver les deux quantificateurs classiques : « il existe » (∃) et « pour tout » (∀). Entre ces deux extrêmes, toute une gradation est possible.

3.1.4.1 Le quantificateur existentiel et le quantificateur universel

Le quantificateur existentiel permet l'énoncé d'une propriété d'un unique élément (peut être plusieurs si l'on prend le sens « au moins un »). L'utilisation du quantificateur universel permet de décrire une propriété que doivent respecter chacun des objets de la scène (ou d'une partie de la scène).

Exemples :

- « La longueur d'un segment est très importante » est interprétable comme « Il existe un segment dont la longueur est très importante »,
- « La pente de tous les segments est assez faible » est interprétable comme « Pour tout segment, la pente est assez faible ».

3.1.4.2 Les quantifications intermédiaires

Entre les deux quantificateurs classiques existe tout un ensemble de quantificateurs indiquant une quantité ou une proportion plus ou moins importante d'objets vérifiant la propriété. Ces quantificateurs peuvent être précis (« un », « 12 »…), imprécis (« entre 10 et 15 ») ou flou (« une dizaine », « quelques », « beaucoup », « au moins »…).

Exemples : « La hauteur d'une dizaine de maisons est très importante », « La longueur de beaucoup de segments n'est pas très importante ».

3.1.4.3 Forme d'une propriété prédicat quantifiée

Une propriété prédicat quantifiée est un énoncé de la forme :

$$\text{« } C_i \text{ de } Q_t \text{ X est } f_\alpha \, m_\beta \, P_{ik} \text{ ».}$$

où C_i est une concept portant sur une caractéristique d'un objet. Q_t est un quantificateur pris parmi les valeurs suivantes : (∃), « un », « au moins un », …, « une dizaine », …, « le tiers », …, « quelques », …, « beaucoup », …, « tous », (∀).

En fait, les quantificateurs sont de trois catégories :
- les quantificateurs proportionnels qui font référence à une proportion de propriétés vérifiées (« 1/3 », « la moitié », « la plupart », « tous », « aucun »…).
- les quantificateurs énumératifs qui donnent une idée du nombre de propriétés vérifiées (« un », « une douzaine », « au moins trois », « au plus 20 »…).
- les quantificateurs ambigus comme « un tout petit peu », « un peu », « quelques », « Ø », « beaucoup », « énormément » qui, suivant le contexte, sont classables dans l'une ou l'autre des catégories précédentes.

En fait, ces derniers sont de deux sortes : les quantités absolues (« une dizaine ») et les proportions (« le tiers », « beaucoup »).

A noter que les propriétés commençant par « Le nombre de X ayant C_i… » sont une forme de propriété quantifiée.

3.1.5 Propriété prédicat globale et propriété prédicat locale

Ceci nous amène à donner une nouvelle définition pour « locale » et « globale ». Selon le concept, la présence des quantificateurs n'est pas toujours indispensable. Nous proposons donc de baser nos définitions sur cette utilisation des quantificateurs.

Nous appellerons concept local tout concept dont les propriétés utilisent de façon systématique un quantificateur. Une propriété prédicat locale est une propriété d'un concept local. Souvent, un concept local est construit à partir :
- d'un paramètre d'un objet de la scène,
- d'une relation entre deux (ou plus) objets de la scène,
- d'un concept spécifique sur un objet ou un sous-ensemble d'objets.

Par opposition, les propriétés associées à un concept global ne demandent pas de quantificateurs. Une propriété prédicat globale est une propriété d'un concept global. Un concept global peut être :
- une statistique ou une quantification sur une propriété locale,
- le dénombrement d'objets,
- un concept spécifique.

Mais alors, comment classer une propriété telle que : « la longueur des segments est importante » ? Comme elle sous-entend un quantificateur universel, elle est classée comme une propriété locale.

Exemple : Si l'on considère un logement composé d'une maison, d'un garage et d'un jardin, nous pouvons donner la propriété globale suivante : « La hauteur de la maison est assez importante ». Par contre, si l'on considère un lotissement composé d'un certain nombre de maisons, nous pouvons donner la propriété locale suivante : « La hauteur d'une maison est assez importante » étant sous-entendu « Il existe une maison dont la hauteur est importante ».

Il existe un cas particulier de propriété globale : les statistiques sur une propriété locale (« la surface moyenne d'une propriété est très importante »).

3.1.6 Propriétés anonymes et propriétés nommées

Dans certaines descriptions, le concepteur veut pouvoir nommer des objets afin de pouvoir leurs attribuer plusieurs propriétés.

En première approche, il semblerait se dégager trois sous-catégories de propriétés locales : les propriétés anonymes (« La hauteur d'une maison est assez importante »), les propriétés pseudo-nommées (« La maison la plus haute est rouge) ou « La troisième maison [décrite] est longue ») et les propriétés nommées (« La hauteur de la maison M est assez importante »).

Cependant une propriété anonyme ou pseudo-nommée peut être considérée comme une propriété dont l'objet est nommé par défaut (« La hauteur [d'une maison] de la maison M102 est assez importante »).

3.1.7 Propriétés de concepts n-aires

Dans certains cas d'application, il existe des domaines dont les mesures ne se font pas par rapport à un élément mais par rapport à plusieurs éléments. Ces concepts représentent des relations entre des objets Ce sont des concepts n-aires (relation avec n objets). Une propriété relative est une propriété prédicat associée à un concept n-aire. En dehors de la référence aux objets, cette forme de propriété est de même nature que celle des propriétés prédicats.

3.1.7.1 Propriétés de concepts binaires

Les concepts permettant de tenir compte de relations (d'interactions) entre deux objets (concepts binaires) sont les plus courants et les plus facilement manipulables. Par exemple, si on considère des relations spatiales nous aurons des relations de placement relatif (« à gauche de », « devant », « sur »…) et de distance (« loin de », « proche de »…). Nous avons alors des énoncé de la forme :

« C_i entre X_p et X_q est f_α m_β P_{ik} ».

Exemples : « La distance entre la maison M1 et la maison M2 est très faible », « La maison M1 est à gauche de la maison M2 ».

3.1.7.2 Propriétés quantifiées de concepts binaires

A partir de ces propriétés de concepts binaires, on peut construire des propriétés quantifiées faisant intervenir plus de deux objets. On les appelle des propriétés binaires quantifiées. Ces énoncés ont une des deux formes suivantes :
1) « C_i entre X_p et Q_t X est f_α m_β P_{ik} » ou « C_i entre Q_t X et X_p est f_α m_β P_{ik} ».
2) « C_i entre Q_t (X et Y) est f_α m_β P_{ik} ».

Exemples : « La distance entre la maison M1 et les autres maisons est très importante » (cas 1), « La distance entre chaque maison est assez faible » (cas 2).

3.1.7.3 Propriétés de concepts n-aires

Par extension de la notion de concept binaire, on peut imaginer d'avoir des concepts n-aires (n>2). Cependant, à notre connaissance, ce type de concept n'existe pas ou, en tout cas, n'est vraiment pas courant car sans doute trop complexe à manipuler par l'être humain. Ceux qui existent sont souvent reformulables à l'aide de concepts unaires ou binaires. Cependant, ce n'est évidemment pas toujours le cas. Certaines propriétés ne sont pas reformulables (ou très difficilement). Nous aurons par exemple une propriété comme « Quatre maisons forment un carré » (ou « La forme géométrique des maisons M1, M2, M3 et M4 est un carré »).

3.1.8 Propriétés de relations n-aires

Ces propriétés mettent en jeu soit au moins deux objets de la scène par rapport à une propriété soit deux propriétés sur un ou deux objet (ou sur la scène). Les éléments des domaines sont nécessairement comparables au niveau sémantique. Les propriétés utilisées sont des propriétés prédicats ou relatives.

3.1.8.1 Propriétés de relation binaires

Une telle propriété permet de manipuler un ou deux objets de la scène suivant une ou deux propriétés de base pas forcément du même concept. Les concepts manipulés sont unaires. Cette catégorie de propriété est appelée propriété de comparaison. Nous avons des énoncés de la forme :

« C_i de X_p est k_α s_β P_{ik} que C_j de X_q ».

avec i et j quelconques du moment que les valeurs de D_i et D_j sont comparables au niveau arithmétique et sémantique. k_α est l'indicateur de différence qui est flou (« un tout petit peu », « un peu », « Ø », « beaucoup », « extrêmement ») ou précis (« 2 fois », « de V unités », « Entre U et V unités », « d'au moins V unités », « d'au plus V unités ») et s_β indique la direction de comparaison (« moins », « aussi » et « plus ») par rapport à la direction de la propriété de référence P_{ik}.

Par exemple :
- « La hauteur de la maison M1 est beaucoup plus importante que la hauteur de la maison M2 ».
- « La largeur de la maison M1 est plus importante que sa hauteur », « La longueur de S1 est beaucoup moins importante que celle de S2 ».

Habituellement, nous avons les cas suivants:
- (i =j et p≠q) appelée comparaison homogène,
- (i≠j) appelée comparaison hétérogène.

3.1.8.2 Propriétés n-aires (n>2)

Par extension, nous pouvons construire des propriétés mettant en relation un objet avec plusieurs. Ces propriétés sont construites comme une relation binaire à l'exception d'un des deux objets qui devient un groupe d'objets. Nous pouvons alors avoir des énoncés selon les deux formes suivantes :
1 - « C_i de X_p est k_α s_β P_{ik} que C_j de Q_t X ».
2 - « C_i de Q_t X est k_α s_β P_{ik} que C_j de X_p ».

Exemples : « La maison M1 est beaucoup plus grande que les autres » (cas 1), « Une dizaine de maisons ont une largeur plus importante que celle de la maison M1 » (cas 2).

Lorsque la propriété de référence n'est plus associée à un concept unaire mais à un concept binaire, nous aurons des propriétés de comparaison relatives. Ce sont généralement des comparaisons homogènes. Nous aurons des énoncés de la forme :

« C_i entre X_p et X_q est k_α s_β P_{ik} que C_i entre X_r et X_s ».

Par exemple : « La maison M1 est beaucoup plus loin de la maison M2 que la maison M3 » (avec X_q=X_r), « La maison M1 est beaucoup plus loin de la maison M2 que de la maison M3 » (avec X_p=X_r), « La maison M1 est beaucoup plus loin de la maison M2 que la maison M3 de la maison M4 ».

Nous pouvons étendre ces propriétés de comparaison relatives à l'aide de quantificateurs de la même manière qu'avec les propriétés de comparaison simples.

3.2 Affirmative et négative

Bien sûr, l'utilisateur décrit ce qu'il veut obtenir mais aussi ce qu'il ne veut pas. Par conséquent, une description comporte des propriétés affirmatives et négatives. Par exemple, on peut dire « La maison n'est pas très haute » (propriété négative avec « n'est pas ») puis « La maison est rose » (propriété affirmative avec « est »). Nous pouvons noter que l'interprétation de la négation n'est pas triviale. La négation logique classique n'est pas satisfaisante. Il est préférable de s'orienter vers une interprétation linguistique ([Mul91], [DPa96]).

Pour les linguistes, « ne pas être A » n'équivaut pas, en général, à la propriété logique « être non-A » qui en termes de sous-ensembles flous est définie à partir de A. Ceci résulte du fait que la propriété logique « non-A » le plus souvent n'a pas de signification linguistique. Le locuteur veut ainsi traduire le fait qu'un objet satisfait à un certain degré une propriété P du même domaine que A. Lorsqu'un locuteur souhaite donner un énoncé comme « le nombre d'intersections n'est pas très faible » cela peu signifier « le nombre d'intersections est vraiment important ». En d'autres termes, un énoncé de négation équivaut alors à une affirmation se référant à une autre propriété (du même domaine). Cependant, il arrive que le locuteur se con-

tente de fournir une négation sans arrière pensée quant à sa signification. La détermination de la signification d'une négation ne peut pas être entièrement automatique. Le logiciel détermine et trie les propriétés plausibles mais c'est au locuteur de décider du sens définitif.

3.3 Propriétés constructives

Lorsque l'utilisateur d'un modeleur déclaratif désire décrire une scène, il veut pouvoir décrire les composants. Pour cela, il utilise un certain nombre de propriétés spécifiques indiquant de quoi est composée la scène. Ces composants sont les éléments de base de la scène mais aussi des objets « de niveau supérieur » construits à partir d'autres objets. Nous appellerons ces propriétés des propriétés constructives.

Nous pouvons mettre en évidence deux catégories de propriétés constructives :

- les propriétés déterminant des composants de base de la scène selon le schéma « La scène est composée de X, Y… » appelée propriétés de construction descendante,
- les propriétés déterminant les objets de la scène selon le schéma « Les objets X, Y… forment un objet Z » appelée propriétés de construction ascendante.

Nous avons des descriptions comme « La scène est composée de cercles et de rectangles », « Les cercles forment une zone A », « Les rectangles forment une zone B » et « Les zones A et B forment une zone C »…

Ces propriétés ne manipulent pas de concepts mais seulement des éléments qui seront caractérisés par des propriétés descriptives. Nous aurons, par exemple, dans la suite de la description ci-dessus la propriété « La zone C est très large ».

Cette construction hiérarchique de la scène est soit donnée en bloc au début de la génération soit construite en ajoutant au fur et à mesure les éléments et seulement lorsque ceux de niveau supérieur sont convenablement construit. Cette dernière méthode correspond à une construction par ébauches successives.

Pour revenir à notre première description, la méthodes par ébauches construit d'abord une zone schématique C possédant les propriétés voulues. Ensuite, les zones A et B sont construites dans la zone C selon leurs propres caractéristiques. Enfin, les cercles et les rectangles sont placés dans leurs zones respectives en fonction des contraintes qui leurs sont appliquées. Avec une telle technique, chaque ébauche d'un niveau est description pour le niveau en dessous.

Entre les propriétés constructives et les propriétés descriptives, nous trouvons un cas particulier de propriétés descriptives indiquant la quantité d'objets d'un type donné. Nous avons alors un énoncé de la forme : « La nombre d'objets X est f_α m_β P_{ik} ». Par exemple, nous aurons « Le nombre de rectangles est assez important » ou « Le nombre de cercles est très faible ».

3.4 Propriétés modificatrices

Lorsque le concepteur prend connaissance d'une solution à sa description, il peut lui arriver de vouloir l'ajuster en modifiant une propriété. Ces propriétés sont donc relatives à une forme solution donnée.

3.4.1 Propriété modificatrice

Une propriété modificatrice est un énoncé de la forme :

« C_i [de X] est f_α k_β s_δ P_{ik} » .

avec les différents opérateurs définis pour les propriétés de comparaison. L'action de ces propriétés dépend de l'opérateur s_δ. Pour ces opérateurs tels que « plus » et « moins », la solution attendue sera différente de celle proposée. Par contre, avec l'opérateur « aussi », l'utilisateur fige la valeur du concept à la valeur actuelle.

Exemples :
- « La longueur de la rue est beaucoup plus importante »,
- « La hauteur de la maison M est environ de 2m plus importante »,
- « La longueur de la rue est entre 2 et 3 Km plus faible »,
- « La maison M1 est aussi grande » (« La taille attendue de la maison M1 est la même que la taille actuelle » ≡ « ne plus bouger la taille de la maison »).

Ce type de propriété est à rapprocher des propriétés prédicats (ou relatives) paramétrées en considérant le paramètre comme étant la valeur de la scène courante. Ainsi, « La longueur de la rue est beaucoup plus importante » est équivalent à « La longueur attendue de la rue est très supérieure à la longueur actuelle de la rue » (Figure 5). Elles portent principalement sur des concepts unaires ou binaires. Bien sûr ces propriétés sont, elles aussi, quantifiables.

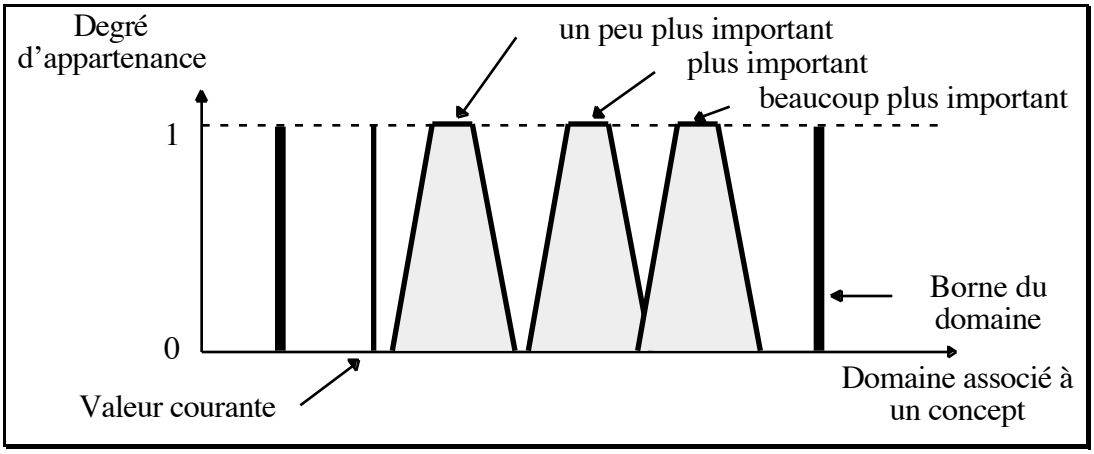

Figure 5 : Exemple de modification d'une description par rapport à un concept

3.4.2 Propriété modificatrice quantifiée et relative

Une propriété modificatrice quantifiée est un énoncé de la forme :

« C_i de Q_t X est f_α k_β s_δ P_{ik} » .

Sauf avec les deux quantificateurs classiques (∃ et ∀), il ne semble pas que ce genre de propriété soit souvent utilisée. Exemples : « La longueur de tout les segments est plus importante », « La pente du segment S est beaucoup moins faible ».

Une propriété modificatrice relative est donc un énoncé de la forme :

« C_i entre X_p et X_q est f_α k_β s_δ P_{ik} » .

Exemple : « La distance entre les maisons A et B est beaucoup plus importante ».

Une propriété modificatrice relative quantifiée est donc un énoncé de la forme :
1) « C_i entre X_p et Q_t X est f_α k_β s_δ P_{ik} » ou « C_i entre Q_t X et X_p est f_α k_β s_δ P_{ik} » .
2) « C_i entre Q_t (X et Y) est f_α k_β s_δ P_{ik} » .

Exemple : « La distance entre chaque maison est beaucoup plus importante ».

4. Classification structurelle

Selon la forme de la fonction d'appartenance d'une propriété, sa manipulation n'est pas toujours la même. Nous nous intéressons donc aux différentes caractéristiques liées à cette forme.

4.1 Rappels sur la structure d'une propriété

A toute propriété prédicat ou relative P_i (et forcément à toute propriété de base) est associée une fonction d'appartenance définie sur le domaine D_i. Cette fonction d'appartenance est construite à l'aide d'une fonction de type LR. L'expression d'une fonction d'appartenance L-R d'une pro-

priété P_i d'un domaine D_i (Figure 6) définie par le quadruplet $<\alpha,a,b,\beta>$ et les fonctions L(x) et R(x) est donnée par :

$$\mu_{<\alpha,a,b,\beta>,\,L,\,R} : \quad D_i \rightarrow [0,1]$$

$$t \rightarrow \begin{cases} 0 & \text{si } t < a-\alpha \\ L((a-t)/\alpha) & \text{si } (\alpha \neq 0) \text{ et } (a-\alpha \leq t < a) \\ 1 & \text{si } a \leq t \leq b \\ R((t-b)/\beta) & \text{si } (\beta \neq 0) \text{ et } (b < t \leq b+\beta) \\ 0 & \text{si } t > b+\beta \end{cases}$$

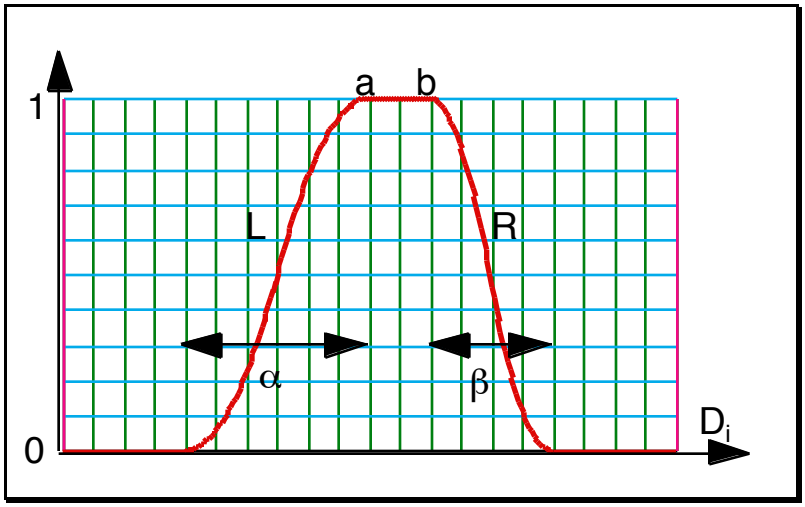

Figure 6 : Fonction d'appartenance d'une propriété

4.2 Position de la fonction d'appartenance

Un des éléments constitutifs d'une propriété prédicat est la propriété de base. La forme de cette propriété est déterminée par la forme intuitive de la propriété prédicat possédant l'opérateur flou par défaut et le modificateur par défaut (forme de P_{ik} = forme de $Ø_f$ Ø P_{ik}). Dans certains cas, il est indispensable de recalculer les bornes du domaine (dépendant par exemple d'un paramètre de l'univers). Il faut donc se poser la question suivante : quelle sera la forme et la position de la propriété de base après cette modification ?

4.2.1 Propriétés à valeur absolue

La première situation concerne les propriétés définies indépendamment des bornes du domaine car construites autour d'une valeur caractéristique. Elles ne sont pas concernées par les modifications des bornes. Par exemple, si l'on considère la taille d'un individu, les tailles minimales et maximales sont susceptibles d'être modifiées plus souvent que la taille moyenne qui servira de référence pour la propriété de base « moyen ». Il en est de même pour les propriétés construites à partir de valeurs paramètres sauf dans le cas particulier de « supérieur à » et « inférieur à » où c'est plutôt la place relative qui importe.

4.2.2 Propriétés à valeur relative

Le second cas porte sur les propriétés dépendantes des bornes. Elles sont relatives à leur position dans le domaine. Dans ce cas, ce n'est plus une valeur qui importe mais la place et la forme de la propriété sur le domaine et, en particulier, par rapport aux bornes. Il en est ainsi des propriétés comme « important » et « faible » dans la plupart des concepts. Les propriétés de cette catégorie sont souvent des propriétés floues (à l'exception parfois de « moyen ») ou des propriétés paramétrées de la forme « inférieur à » et « supérieur à ».

4.3 Propriété modifiable

Le problème qui se pose est le suivant : pourquoi peut on dire « Le nombre de segments est très important » mais pas « Le nombre de segments est très de 10 » ? Il y a donc des propriétés qui sont modifiables à l'aide d'un modificateur autre que « Ø » et d'autres pas.

Les propriétés modifiables, c'est-à-dire acceptant un modificateur, sont associées à une fonction d'appartenance représentant un intervalle flou (Figure 7 : F2b en particulier). L'explication de ce phénomène est que les modificateurs ne peuvent s'appliquer qu'à des propriétés suffisamment floues. C'est justement l'intérêt de ces modificateurs : manipuler une propriété dont on ne connaît pas les valeurs exactes et qui ne sont donc pas manipulables autrement.

Par contre, les opérateurs flous s'appliquent à une propriété quelconque. Sur des propriétés qui ne sont pas modifiables, comme les propriétés paramétrées, ils permettent d'introduire de l'incertain (par exemple : « la hauteur des plafonds est environ de 3 mètres »).

4.4 Forme

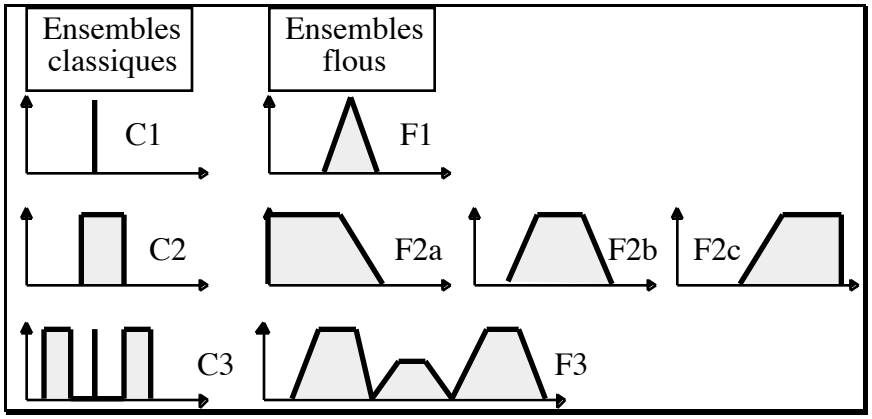

Figure 7 : Classement des différentes formes de fonctions d'appartenance en théorie des ensembles flous

Les fonctions d'appartenance peuvent être classées en deux groupes divisibles eux-mêmes en trois formes :

- les ensembles classiques
  - les valeurs précises (Figure 7 : C1)
  - les intervalles classiques (Figure 7 : C2)
  - les ensembles classiques quelconques (Figure 7 : C3)
- les ensembles flous
  - les valeurs floues (Figure 7 : F1)
  - les intervalles flous de forme Z, Π ou S (Figure 7 : resp. F2a, F2b et F2c)
  - les ensembles flous quelconques (Figure 7 : F3)

4.5 Dimensions du domaine

A chaque concept C_i est associé un domaine D_i regroupant l'ensemble des valeurs que peut prendre C_i vis à vis d'une forme. Ces valeurs sont généralement de dimension 1. Le type des valeurs est a priori quelconque. On a par exemple :

- ***b***, pour les concepts qui n'ont pas de mesure numérique et qui permettent de vérifier une qualité sans pouvoir quantifier (la mesure se contente de signaler si la propriété est ou non présente),
- ***n***, pour les propriétés qui, par exemple, dénombrent une caractéristique,
- ***z***, ***r***, [0,1] très souvent utilisés,
- ou d'autres types comme une énumération d'objets (chaînes de caractères…).

Nous venons de voir des domaines de dimension 1 mais rien n'interdit d'imaginer des concepts avec un domaine de dimension supérieure (en particulier de dimension 2 et 3). Cependant, en dehors des propriétés de comparaison, il nous semble difficile d'envisager simplement de tels concepts. Le problème essentiel n'est pas de trouver une mesure mais surtout de définir aisément des propriétés (précises, des intervalles, de base…) et de les manipuler facilement (et fidèlement à l'intuition s'il y en a une !). Par contre, il est tout à fait envisageable d'utiliser un modeleur déclaratif spécifique permettant de rechercher soit une solution convenable pour

l'utilisateur soit de transformer le domaine en dimension 1 pour pouvoir manipuler plus facilement les différents opérateurs. Cette méthode s'applique pour tout concept trop complexe pour être manipulé directement.

5. Classification comportementale

L'objectif essentiel d'une propriété est d'intervenir dans la génération afin de sélectionner au plus vite toutes les solutions de l'univers qui la vérifient. Pour cela, elle va intervenir à plusieurs niveaux dans le processus déclaratif pour réduire cet univers, orienter les recherches et valider les formes obtenues.

5.1 Rappels sur les méthodes de génération

En modélisation déclarative, le rôle des propriétés et leurs conséquences dépendent des techniques de génération utilisées et des caractéristiques des concepts vis à vis de cette génération. Les premières études sur des classifications de ce type ont été effectuées par [Ple89], [Mar90a], [Col90], [Chau94], [Paj94] et [Pou94].

On distingue trois grandes méthodes de génération :
- L'exploration algorithmique (arbres d'énumération, de construction…),
- Les systèmes à base de règles et de faits,
- La génération aléatoire sous contraintes (systèmes d'intervalles, résolution de contraintes, tirage aléatoire…).

Dans les techniques classiques de modélisation déclarative, on a montré qu'il est possible d'effectuer des optimisations de l'arbre de génération en fonction de caractéristiques des propriétés ([LDe96]). Avec l'utilisation des ensembles flous, la méthode est conservée. Le concepteur peut donc pouvoir décrire les caractéristiques de la propriété et du concept propres à provoquer automatiquement les optimisations de la génération.

Les travaux les plus importants sur le rôle des propriétés en génération a été produit pour les arbres d'exploration ([Elk89], [Col90], [Mar90a], [Col92]) et sur les systèmes experts ([Ple89], [MaM89]). L'étude des comportements des propriétés permet, en particulier, d'améliorer le parcours en provoquant des élagages de branches et en simplifiant les vérifications (« réduction des contrôles » [Col92] ou « stérilisation des propriétés » [Gra96]).

5.2 Les parcours d'arbres

5.2.1 Généralités

Les caractéristiques de comportement dans les arbres de génération ainsi que les techniques d'optimisation sont celles utilisées pour les propriétés à deux états.

Nous allons étudier certaines conditions permettant de déterminer les caractéristiques de comportement d'une propriété prédicat dans un arbre donné. Ces conditions sont généralisables aux autres propriétés (quantifiées, relatives…).

Remarque : toutes les définitions et propriétés que nous énonçons ici dépendent de l'arbre de génération.

5.2.2 Quelques définitions utiles

Le comportement d'une propriété prédicat est fonction du comportement d'une part de la mesure associée au domaine sur lequel elle est définie et d'autre part de sa fonction d'appartenance. Nous allons donc donner un certain nombre de définitions permettant de décrire le comportement de ces deux fonctions.

5.2.2.1 Définitions à propos des mesures des domaines

Une mesure est dite *croissante* dans un arbre donné si, à un niveau i de l'arbre, f_{i-1} et f_i deux formes consécutives dans une descente d'arbre sont telles que $m(f_{i-1}) \leq m(f_i)$. Une mesure est

dite *strictement croissante* dans un arbre donné si, à un niveau i de l'arbre, f_{i-1} et f_i deux formes consécutives dans une descente d'arbre sont telles que $m(f_{i-1}) < m(f_i)$.

Une mesure est dite *décroissante* dans un arbre donné si, à un niveau i de l'arbre, f_{i-1} et f_i deux formes consécutives dans une descente d'arbre sont telles que $m(f_{i-1}) \geq m(f_i)$. Une mesure est dite *strictement décroissante* dans un arbre donné si, à un niveau i de l'arbre, f_{i-1} et f_i deux formes consécutives dans une descente d'arbre sont telles que $m(f_{i-1}) > m(f_i)$.

Une mesure est dite *[strictement] monotone* si elle est [strictement] croissante ou [strictement] décroissante.

5.2.2.2 Définitions à propos des fonctions d'appartenance

Une propriété prédicat P est dite *croissante* si sa fonction d'appartenance μ est croissante sur le domaine D, c'est-à-dire si : $\forall(t,t')\in D\times D : t > t', \mu(t) \geq \mu(t')$. Une propriété prédicat P est dite *strictement croissante* si sa fonction d'appartenance μ est croissante sur le domaine D, c'est-à-dire si : $\forall(t,t')\in D\times D : t > t', \mu(t) > \mu(t')$. C'est le cas des propriétés de forme S (Figure 7-F2c).

Une propriété prédicat P est dite *décroissante* si sa fonction d'appartenance μ est décroissante sur le domaine D, c'est-à-dire si : $\forall(t,t')\in D\times D : t > t', \mu(t) \leq \mu(t')$. Une propriété prédicat P est dite *strictement décroissante* si sa fonction d'appartenance μ est décroissante sur le domaine D, c'est-à-dire si : $\forall(t,t')\in D\times D : t > t', \mu(t) < \mu(t')$. C'est le cas des propriétés de forme Z (Figure 7-F2a).

Une propriété prédicat P est dite *[strictement] monotone* si elle est soit [strictement] croissante soit [strictement] décroissante.

5.2.2.3 Seuil d'acceptation

Nous avons montré dans [Des95b] qu'il est possible de choisir entre plusieurs politiques pour déterminer un *seuil d'acceptation* (seuil à partir duquel la propriété est considérée comme suffisamment vérifiée pour être retenue). La détermination de la forme de la propriété est à la charge du concepteur : d'un bon choix dépendra une production raisonnable de solutions. A priori, dès que le degré d'appartenance de la scène à la propriété n'est plus nul, on peut considérer que cette scène appartient un peu à la propriété. Autrement dit, toute scène dont le degré d'appartenance à la propriété n'est pas nul vérifie (plus ou moins) cette propriété. En conséquence, nous pouvons choisir un seuil d'acceptation qui soit juste au-dessus de 0 (0.01 ou 0.1 suivant la précision des propriétés).

Remarque : les « flouistes » utilisent généralement la valeur 0.2 comme seuil.

5.2.3 Synthèse de différents types de comportements

Jusqu'ici, les propriétés sont évaluées selon la logique booléenne classique. Donc, lors d'un parcours dans une branche d'un arbre, une propriété peut prendre deux états :
- V (la propriété est vérifiée),
- F (la propriété n'est pas vérifiée).

Il y a donc deux situations possibles liées aux deux états :
- E (la propriété est étudiée et est V ou F),
- ? (il n'y a pas de vérifications).

Un parcours de branche s'écrit alors à l'aide de la suite des états de la propriété. Par exemple : FFVVFFVFFVVVFF

Pour synthétiser l'écriture d'un parcours, nous posons les éléments suivants :
- E* : l'état E se répète 0, 1 ou plusieurs fois,
- E+: l'état E se répète 1 ou plusieurs fois,
- E[p] : l'état E se répète p fois,
- (ABCD) : les états A, B, C, D forment un "motif" (qui peut se répéter)

A l'aide de ces conventions, le Tableau 1 effectue une synthèse des classements de propriétés proposées dans les différents travaux en modélisation déclarative. Ces classements ne sont pas complets. Nous allons donc en proposer un nouveau effectuant d'une part une synthèse et d'autre part une adaptation au formalisme flou.

Tableau 1 : Types des propriétés classiques [Des95a]

[Col90] (C), [Elk89] (E) et [Pou94] (P)	[Paj94]	Catégorie selon [Mar90a]	Branche	Remarques
Stable (C, E et P)	Transmissible de type 2	2	F*V+	
Stable (E et P)	Transmissible de type 1	3	V*F+	Extension de Stable pour (C)
p-régulière (C)		5	(VF[p-1])+	
Alterne (C)		5	(VF)+	
		5	F*(VF[p-1])+	Extension de p-régulière de (C)
		6	F*(VE[p-1])+	Extension de p-périodique de (C)
p-périodique (C)		6	(VE[p-1])+	
		1 - 4 - 7	?*E+	Sans caractéristiques générales mais selon le domaine d'application des optimisations sont possibles
	Semi-transmissible de type 1		F*V+F+	
	Semi-transmissible de type 2		V*F+V+	
Pseudo-stable (P)			E*F+V+ E*V+F+	Extension des semi-transmissibles de [Paj94]
			F[p]V[q]F*	Propriétés pq-semi-transmissibles type 1
			V[p]F[q]V*	Propriétés pq-semi-transmissibles type 2

5.2.4 Les propriétés stables

Une *propriété stable* est une propriété prédicat qui une fois vérifiée (resp. non vérifiée) le reste pour tout le sous-arbre. Les propriétés stables sont réparties en deux catégories (Figure 8) :

- les *propriétés V-Stables* qui, une fois vérifiées, le restent définitivement.
- les *propriétés F-Stables* qui, une fois non vérifiées, le restent définitivement.

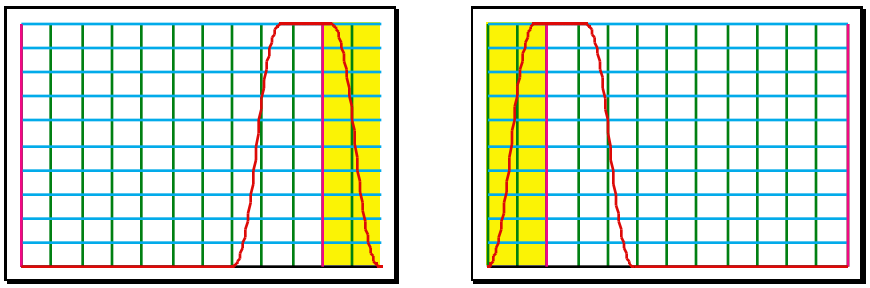

Figure 8 : Exemples de propriétés V-Stable et F-Stable (mesure croissante)

Pour déterminer si une propriété est stable, il faut étudier la mesure du domaine (m) et la fonction d'appartenance à la propriété (μ). Nous allons étudier les conditions pour avoir une propriété V-Stable.

D'une manière générale, une propriété est V-Stable si, à partir d'un niveau de l'arbre, $m \in P_{\mathbf{s}}$ (Coupe de niveau $\mathbf{s} \neq 0$, $\mathbf{s}$ est le seuil défini plus haut) dans tout le sous-arbre. En particulier, c'est le cas dans les situations suivantes :

- $m = Cste$ et $Cste \in P_{\mathbf{s}}$.
- $m \in [a,b]$ et $[a,b]$ contenu dans $P_{\mathbf{s}}$.
- (m croissante et μ croissante) ou (m décroissante et μ décroissante).

D'une manière générale, une propriété est F-Stable si, à partir d'un niveau de l'arbre, m $\notin P_s$ dans tout le sous-arbre. En particulier, c'est le cas dans les situations suivantes :

- m = Cste et Cste $\notin P_s$.
- m $\notin$ [a,b] et [a,b] n'est pas contenu dans P_s.
- (m décroissante et μ croissante) ou (m croissante et μ décroissante).

5.2.5 Propriétés quasi-stables

Les *propriétés quasi-stables* sont des propriétés prédicats qui ne sont vérifiées (resp. non vérifiées) que sur un intervalle de niveaux de l'arbre. Les propriétés quasi-stables sont réparties en deux catégories (Figure 9) :

- les *propriétés quasi-V-Stables* qui sont vérifiées sur un intervalle.
- les *propriétés quasi-F-Stables* qui ne sont pas vérifiées sur un intervalle.

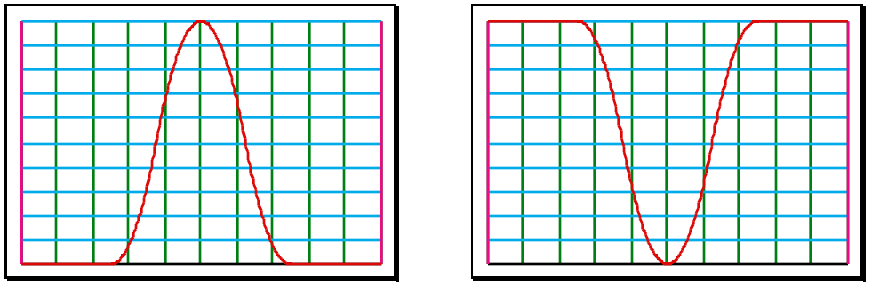

Figure 9 : Exemples de propriétés Quasi-F-Stable et Quasi-V-Stable (mesure monotone)

Cas particulier pour obtenir des propriétés quasi-stables :

- m est monotone et μ de forme Π (Figure 7-F2b),
- m se comporte en deux phases (croissante puis décroissante ou inversement) et le point de changement (maximum ou minimum) se trouve dans P_s.

Remarque : ce cas se retrouve assez souvent dans une description. En effet, on obtient des propriétés de ce style si l'on dit : « L'objet formé est d'environ X unités », « L'objet est entre X et Y », « L'objet est de longueur moyenne »…

5.2.6 Elagage et stérilisation des propriétés

Une propriété provoque un élagage de l'arbre de génération si elle permet d'éviter l'exploration du sous-arbre dont elle est la racine. Il existe deux modes d'élagage ([Col90] et [Elk89]) :

- L'élagage immédiat, appelé aussi élagage par le haut. Lorsqu'une forme ne vérifie pas une propriété F-Stable, cette forme est rejetée et le parcours de cette branche s'arrête là. Cet élagage peut avoir lieu aussi lors d'une vérification définitive de toutes les propriétés (voir la stérilisation des propriétés). Il est donc effectué quand le système est certain de l'échec ou de la réussite de toutes les formes du sous-arbre.
- L'élagage par le bas. Lorsqu'une propriété requise ne peut plus être vérifiée dans le sous-arbre, il n'est plus nécessaire de continuer le parcours de cette branche. Aucune forme issue de ce niveau ne sera solution. On opère donc un élagage sous la forme courante, l'objectif ne pouvant être atteint.

La réduction des contrôles ou stérilisation des propriétés est une diminution du nombre de propriétés à étudier. Elle a lieu quand une propriété V-Stable est vérifiée. Cette propriété est retirée de la liste des propriétés à vérifier pour toutes les branches issues de la forme courante car elle sera toujours vérifiée. Toutes les formes issues du noeud héritent de la propriété.

Lorsqu'on est sûr qu'une propriété ne sera pas vérifiée au moins avant un niveau n du parcours de l'arbre, il n'est pas nécessaire de vérifier les propriétés de la liste avant ce niveau. Il va donc y avoir une stérilisation globale des propriétés : aucune propriété ne sera plus étudiée avant le niveau n. Ce type de stérilisation a lieu éventuellement plusieurs fois dans le parcours d'une branche. On appelle ce niveau le seuil de détection.

5.2.7 Optimisations automatiques

Les optimisations, en particulier les conditions d'élagage, sont généralement à la charge du concepteur. Cependant, il est tout à fait possible de détecter automatiquement des situations d'élagage sans que le concepteur ait à s'en occuper. Nous allons donner quelques situations permettant de provoquer automatiquement des optimisations.

5.2.7.1 Hypothèses

La méthode de génération que nous avons choisie est la génération par arbres d'énumération. Une des caractéristiques essentielles de cette méthode est que, pour un niveau donné, tout fils est construit en ajoutant un chiffre au numéro courant. A partir de cette constatation, il y a souvent des mesures monotones faciles à détecter de la part du concepteur.

Nous allons donc proposer quelques conditions détectables automatiquement, basées sur le comportement des mesures et sur la forme des propriétés.

5.2.7.2 Formes des propriétés et stabilité

Compte tenu du comportement de la mesure et de la forme de la propriété, il est assez facile de déterminer la stabilité de la propriété ou de son complémentaire (avec $^{-1}$) (Tableau 2).

Tableau 2 : Etude de la stabilité d'une propriété en fonction de son domaine et de sa forme

Forme de la propriété	Comportement de la mesure	
	Croissante	Décroissante
Z ou S^{-1}	F-Stable	V-Stable
S ou Z^{-1}	V-Stable	F-Stable
Π	Quasi-F-Stable	Quasi-F-Stable
$Π^{-1}$	Quasi-V-Stable	Quasi-V-Stable

Nous pouvons noter que les formes proposées ici ne sont parfois que partiellement entre les bornes du domaine (Figure 10). Nous pouvons associer dans cette figure les formes A à la forme Π, les formes B au complémentaire d'une forme Π, les formes C1 et D2 à la forme Z et, enfin, les formes C2 et D1 à la forme S. Il arrive donc que l'optimisation potentielle ne soit jamais atteinte.

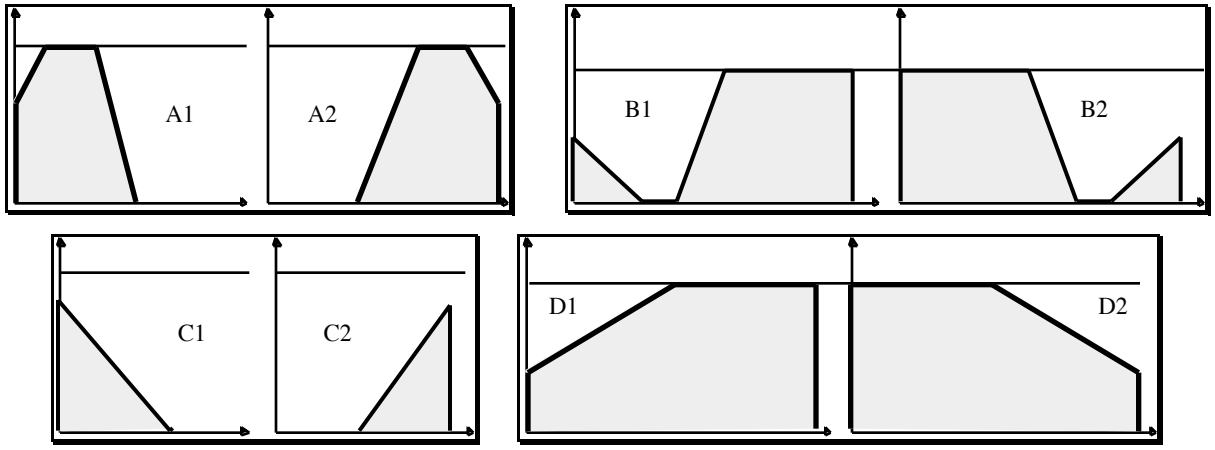

Figure 10 : Exemples de formes classiques « dégénérées »

5.2.7.3 Détections automatiques sur la stabilité d'une propriété

Une fois que l'on connaît la qualité de stabilité de la propriété, il suffit de détecter les valeurs clés ou les conditions spécifiques qui, quand elles apparaissent, provoquent un élagage de l'arbre de génération ou une stérilisation d'une ou plusieurs propriétés (Tableau 3).

Tableau 3 : Situations d'élagage de l'arbre ou de stérilisation d'un propriété

Domaine	F-Stable	Quasi-F-Stable	V-Stable	Quasi-V-Stable
Monotone	Elagage $\mu(m_n) < S$	Elagage $\mu(m_n) < S$ et $\mu(m_{n-1}) \geq S$	Stérilisation $\mu(m_n) = 1$	Stérilisation $\mu(m_n) = 1$ et $\mu(m_{n-1}) < 1$
Strictement monotone	Elagage $\mu(m_n) \leq S$	Elagage $\mu(m_n) \leq S$ et $\mu(m_{n-1}) > S$	Stérilisation $\mu(m_n \pm u) = 1$	Stérilisation $\mu(m_n \pm u) = 1$ et $\mu(m_{n-1}) \leq 1$

Avec :
- S : seuil d'acceptation
- u : unité du domaine (+ dans le cas croissant et - dans le cas décroissant)
- μ : fonction d'appartenance de la propriété
- m_n : mesure au niveau n de l'arbre

5.2.7.4 Notion de valeur critique

Une *valeur critique* est une valeur particulière du domaine permettant, dans certaines situations, de provoquer une optimisation (élagage ou réduction des contrôles).

On peut mettre en évidence deux types de valeurs critiques :
- les stables,
- les instables.

Une *valeur critique stable* est une valeur d'un domaine telle que si, au cours d'une descente dans l'arbre, la mesure l'atteint, tout le sous-arbre aura cette valeur dans ce domaine.

Une *valeur critique instable* est une valeur d'un domaine telle que si, au cours d'une descente dans l'arbre la mesure prend une autre valeur, elle ne sera plus atteinte dans le sous-arbre.

Remarques :
- Ces valeurs sont souvent des extrêmes des domaines.
- Dans certains cas, cette notion de valeur critique peut être étendue à des intervalles critiques. La gestion est alors identique.

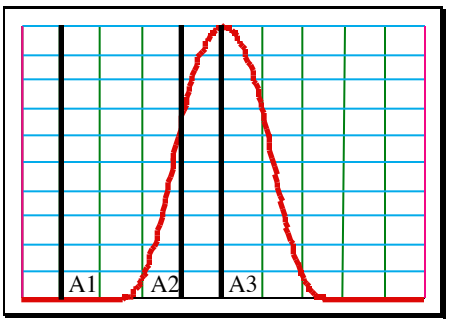

Figure 11 : Cas possible de valeur critique stable

La Figure 11 nous donne trois cas possibles de placement d'une valeur critique stable par rapport à une propriété prédicat. Deux cas sont intéressants : A1 (le degré d'appartenance de la valeur est nul) et A3 (le degré d'appartenance est 1). Si lors d'une exploration, la mesure associée au domaine atteint la valeur A1, tout le sous arbre aura cette valeur. Par conséquent, il y a possibilité d'élagage. Par contre, si l'on arrive à la valeur A3, tout le sous-arbre vérifiera la propriété. Il y a alors possibilité d'une réduction des contrôles.

En ce qui concerne la valeur critique instable, un seul cas est intéressant : la propriété prédicat est une valeur précise égale à la valeur critique instable. Dans ce cas, dès que la mesure ne vérifie plus la propriété nous pouvons effectuer un élagage, car elle ne sera jamais plus vérifiée dans le sous-arbre.

5.2.7.5 Cas particulier des mesures strictement croissantes ou strictement décroissantes

Pour les mesures strictement croissantes ou strictement décroissantes, il est possible d'améliorer le traitement des valeurs critiques. En effet, les élagages sont détectables un niveau plus tôt que pour les autres. Ceci peut faire économiser un nombre d'explorations important (cas où le nombre de fils d'un arbre est très important).

5.2.8 Concepts spécifiques

Outre les optimisations sur la forme des propriétés comme nous venons de le montrer, certains concepts classiques disposent de caractéristiques particulières amenant des situations d'élagage de l'arbre, de stérilisation des propriétés ou d'autres techniques permettant une amélioration de l'exploration.

5.2.8.1 Mesures incrémentales

Pour certains concepts, mesurer une forme solution s'avère très coûteux en temps. Pour optimiser cette évaluation, quand c'est possible, une mesure peut être évaluée de manière incrémentale : toute mesure à un niveau n de l'arbre d'exploration dépend uniquement de la mesure au niveau n-1 et de la forme courante. Ainsi, on bénéficie du travail déjà fait.

A noter que, d'un niveau sur l'autre, on peut conserver d'autres informations que spécifiquement la mesure du domaine. Nous verrons dans ce qui suit qu'il est intéressant, dans le cas de concepts quantitatifs, de garder des calculs annexes (sommes de valeurs…).

5.2.8.2 Propriétés incrémentales quantitatives (quantités ou proportions)

L'idée de base des mesures des domaines de ces concepts est la quantité (absolue pour la quantification et relative pour la proportion). Cette quantité peut être le nombre d'éléments vérifiant une propriété prédicat locale (gestion d'un quantificateur par exemple) ou une mesure d'un concept spécifique.

Nous allons poser les hypothèses suivantes : la mesure est monotone et incrémentale (la quantité courante est la quantité du niveau précédent plus ou moins une valeur calculée sur la forme courante). Dans ce cas, nous sommes capables de calculer l'intervalle D' du domaine dans lequel se trouve la mesure de toute solution issue d'un noeud donné.

Si on arrive à montrer que l'intervalle flou, fonction d'appartenance de la propriété, a son support (intervalle où le degré d'appartenance n'est pas nul) disjoint de D' alors on peut effectuer un élagage de l'arbre. En effet, on est alors certain que le propriété ne sera jamais plus satisfaite.

Si nous avons les données suivantes :
- M la plus grande quantité possible,
- $Dmin_0$ et $Dmax_0$ les bornes du domaine au départ,
- M_i la plus grande valeur à ajouter ou retirer au niveau i,
- m_i la valeur effectivement ajoutée ou retirée au niveau i.

Nous allons chercher à calculer :
- $Smin_i$ la quantité effective au rang i,
- $Smax_i$ la quantité maximale au rang i.
- $Dmin_i$ et $Dmax_i$ les bornes du domaine connaissant le niveau i.

Nous cherchons à calculer $[Dmin_i, Dmax_i]$ cet intervalle D'.

Si la mesure est croissante, nous aurons :
- $Smin_i = Smin_{i-1} + m_i$
- $Smax_i = Smax_{i-1} - M_i + m_i$.

Si la mesure est décroissante, nous aurons :
- $Smin_i = Smin_{i-1} + M_i - m_i$
- $Smax_i = Smax_{i-1} - m_i$.

a) Cas des propriétés associées à un concept de quantité
Nous avons alors $Dmin_0 = 0$, $Dmax_0 = M$, $Smin0 = Dmin_0$ et $Smax_0 = Dmax0$.
Et enfin : $Dmin_i = Smin_i$ et $Dmax_i = Smax_i$.

b) Cas des propriétés associées à un concept de proportion
Nous avons alors $Dmin_0 = 0$, $Dmax_0 = 1$, $Smin_0 = 0$ et $Smax_0 = M$.
Et enfin : $Dmin_i = Smin_i / M$ et $Dmax_i = Smax_i / M$.

A noter qu'il est parfois difficile d'interpréter des propriétés à base de quantités. Il y a ambiguïté entre la quantité « pure » et la proportion. Par exemple « La quantité de segment horizontaux est importante » signifie-t-elle « La quantité de segment horizontaux est importante par rapport à la quantité maximale possible » (quantité « pure ») ou « La quantité de segment horizontaux est importante par rapport à la quantité de segments courante » (proportion).

5.2.8.3 Propriétés incrémentales statistiques

Pour les concepts à base de statistiques et, en particulier, pour les concepts de moyenne, nous allons pouvoir mettre au point une optimisation similaire à celle développée pour les propriétés de quantité.

Nous allons étudier les concepts concernant les moyennes sur une quantité. Ici encore nous avons à calculer une quantité sur la forme courante. Cependant, nous allons poser les hypothèses suivantes : la mesure du domaine est la moyenne des quantités vues jusqu'ici et non directement cette quantité et cette même quantité est incrémentale (la quantité courante est la quantité du niveau précédent plus ou moins une valeur calculée sur la forme courante). Il n'est pas nécessaire de poser une hypothèse sur la monotonie de la mesure.

Comme pour les propriété du paragraphe précédent, nous cherchons l'intervalle D' dans lequel seront toutes les mesures des formes du sous-arbre et qui nous permettra éventuellement d'effectuer des élagages.

Si nous avons les données suivantes :
- N la profondeur de l'arbre,
- $Dmin_0$ et $Dmax_0$ les bornes du domaine au départ,
- c_i la quantité calculée au niveau i.

Nous allons chercher à calculer :
- S_i la somme des quantités au rang i ($S_0 = 0$),
- $Dmin_i$ et $Dmax_i$ les bornes du domaine connaissant le niveau i.

Nous cherchons à calculer $[Dmin_i, Dmax_i]$ cet intervalle D'.

Nous avons : $S_i = S_{i-1} + c_i$
Donc, nous avons :
$Dmin_i = (S_i + (N - i) * Dmin_0) / N$
et $Dmax_i = (S_i + (N - i) * Dmax_0) / N$.

5.3 Moteurs d'inférences

Les techniques à base de moteurs d'inférences génèrent des informations sur l'objet en procédant par déduction sur les informations déjà acquises. Cette technique de génération utilise la notion de faits (ou une notion équivalente) pour matérialiser la connaissance que l'on a de la scène à un instant donné. Le mécanisme de raisonnement est généralement mis en oeuvre à l'aide d'un ensemble de règles et d'un moteur d'inférences.

Les règles utilisées dans les moteurs d'inférences peuvent être regroupées en six catégories.

5.3.1 Les règles de création

Qu'elles soient appelées règle de création, règle de construction, ou toute appellation de même nature (voir [MaM91], [Col90]), le but de ces règles est la production de nouveaux faits. Elles

travaillent à partir des informations contenues dans la base de faits et en déduisent de nouvelles informations. [MaM91]distingue deux classes de règles :

- Des règles de calcul. Elles visent essentiellement à déduire une valeur en la calculant à partir des données fournies par la base de faits. Une règle de ce type commence par vérifier l'existence de faits nécessaires au calcul et, si les conditions sont réunies, elle effectue le calcul et engendre un ou plusieurs faits.
- Des règles de déduction. Ces règles sont des mécanismes de raisonnement sur les faits. Elles analysent les faits existant et en déduisent de nouveaux, ne nécessitant pas forcément des étapes de calcul.

Ces règles sont toutefois de nature identiques : leur seul but est de construire de nouveaux faits à partir de ceux existants. Le fait que ces règles soient purement calculatoire ou non ne change rien à leur nature. Seule la nature de leur implémentation peut expliquer un tel classement.
Remarque : Les faits produits par ces règles peuvent être de nature :

- précise valuée (« la hauteur de la maison est de 5 unités »),
- précise non valuée (« la largeur de la maison est égale à sa longueur »),
- imprécise valuée (« la hauteur de la maison est entre 3 et 5 unités »),
- imprécise non valuée (« la position de la maison A est entre celles des maisons B et C »),
- floue valuée (« la hauteur de la maison est environ 5 unités »),
- floue non valuée (« la longueur de la maison est très supérieure à sa largeur »).

En fonction de la précision que peut fournir une règle lors de la création d'un fait, le moteur va être plus ou moins efficace. Par exemple, les règles de calcul s'appliquent rarement sur des intervalles, mais sur des valeurs fixes. Cette précision est souvent difficile à gérer car elle correspond à des principes inhérent au domaine étudié.

Une règle précise valuée ou non peut engendrer la production immédiate d'un nouveau fait. Une règle imprécise peut également permettre la production d'un fait, mais dont la nature sera moins précise. Une gestion d'intervalle peut dans ce cas se révéler efficace. Une règle floue est plus complexe à gérer.

5.3.2 Les règles de suppléance

Ces règles sont spécifiques à ce type de moteur d'inférences. Elles ne sont déclenchées que lorsque le moteur d'inférences ne peut plus rien déduire de sa base de faits. En effet, le mécanisme de déduction mis en oeuvre dans le moteur d'inférences est de temps en temps bloqué parce qu'aucune règle ne peut fournir de nouveaux faits (toutes les déductions possibles ont été faites).

Ces règles mettent en oeuvre une instanciation forcée de certains paramètres de la scène. En fonction de la nature du domaine auquel appartient le paramètre, le comportement de cette règle peut être très différent :

- Pour un domaine fini, et dont le nombre d'éléments possibles n'est pas trop important, la règle va générer un balayage de l'ensemble de ces éléments. L'ordre dans lequel se fait le balayage peut être fixé à l'avance, tiré au hasard, ou suivre une quelconque règle (préférences utilisateur, optimisation de parcours, ...)
- Pour un domaine dans lequel il est possible de définir un nombre fini, peu important de classes d'éléments, sous réserve que ces classes puissent être représentées dans leur entier (intervalle de valeurs, ...), la règle va engendrer un balayage systématique de ces classes.
- Pour les autres domaines, d'autres mécanismes devront être mis en oeuvre (tirages aléatoires, intervention de l'utilisateur pour effectuer des choix, ...).

Les méthodes appliquées par ces règles de suppléance peuvent conduire à des résultats différents. Une énumération n'implique pas de notion d'ordre sur le parcours des éléments d'un ensemble, ce qui influe sur l'ordre de présentation des résultats (sans remettre en cause leur existence). Un tirage aléatoire n'est généralement pas le même d'une génération à l'autre (sauf

mécanisme de génération aléatoire contrôlé), ce qui peut conduire, pour deux générations identiques, à des résultats complètement différents.

5.3.3 Les règles de modification

Les faits produits par les règles de création conduisent inévitablement à des redondances d'information. Les règles de modification ont pour but de raisonner sur la base de faits pour en limiter le nombre.

[Bea89] et [MaM91] utilisent la notion de "règles de modification", mais dans un sens différent. Ces règles visent à modifier la valeur d'un fait sans remettre en cause son existence. Il est clair que ce type de règle vise essentiellement à optimiser le moteur d'inférence et sera classé dans la catégorie correspondante.

Ces règles ne font que modifier la base de faits, sans changer la nature des faits qu'elle contient. A ce titre, on ne peut donc pas parler d'une quelconque modification du comportement du moteur d'inférences.

5.3.4 Les règles de vérification

Ces règles ont pour but de vérifier la cohérence de la base de faits. Elles permettent d'assurer que la génération ne conduit pas à une impasse. De même que pour les arbres de génération, on va retrouver différents types de règles de vérification:

- Des règles de vérification immédiates. Elles vérifie que les faits énoncés dans la base ne sont pas contradictoires. Par exemple, l'existence simultanée des faits « La longueur du segment est 3 unités » et « la longueur du segment est au moins de 7 unités » conduit à un échec.
- Des règles de vérification et anticipation. Ces propriétés permettent de définir si, en fonction des données de la base de faits, il sera possible de construire la scène. Ceci est à rapprocher du fonctionnement de certaines propriétés dans les arbres de génération. La stabilité de certaines propriétés conduit à la mise en oeuvre de ce type de règles.

Les règles de vérification, même si elles ne conduisent pas l'ordre d'obtention des solutions, agissent énormément sur le comportement du moteur d'inférences. De la qualité des règles mises en oeuvre va dépendre, pour une bonne part, la rapidité d'obtention des solutions. Les plus complexes sont évidemment les règles de vérification et anticipation qui doivent permettre de prévoir, en fonction des faits courants si une solution est possible ou non. C'est pourtant souvent leur intervention qui augmente la rapidité de déduction du moteur d'inférences.

5.3.5 Les règles internes

Ces règles assistent le moteur d'inférence dans sa gestion des faits. Elles peuvent être divisées en plusieurs classes :

- Des règles de manipulation des faits. Les mécanismes de création et suppression de faits sont des exemples typiques de ce type de règles. Les "règles de suppression" de [Col90] entrent également dans cette classe.
- Des règles d'optimisation de parcours. Elles visent à indiquer au moteur d'inférences les éléments à traiter en priorité. Ces règles modifient l'ordre de parcours des éléments. Elles ont une influence sur l'ordre de présentation des solutions et parfois sur les résultats (dans la cas par exemple de l'existence d'une règle limitant le nombre de solutions).
- Des règles d'optimisation du moteur. Ces règles visent à optimiser certaines actions effectuées par le moteur d'inférence. On retrouve dans cette catégorie les "règles de modification" de [Bea89] et [MaM91] dont le but essentiel est la modification de la valeur d'un fait, évitant ainsi de mettre en place un mécanisme de destruction, puis de création du fait avec une autre valeur.

Ces règles ont évidement une influence sur le moteur d'inférences. Les règles de manipulation de faits et les règles d'optimisation du moteur ne font qu'en assurer un fonctionnement plus rapide. Les règles d'optimisation de parcours permettent de structurer les éléments de manière à

obtenir plus rapidement des solutions, laissant les branches ayant moins de chances de comporter des solutions pour la fin de l'exploration.

5.3.6 Les MétaRègles

Les métarègles n'interviennent pas directement sur la génération d'une scène. Leur but est d'enrichir la base de règles du moteur pour en optimiser le comportement. Elles agissent à plusieurs niveaux :

- Raisonnement sur les générations. Il s'agit d'analyser les différentes générations effectuées par le moteur d'inférences pour en déduire de nouvelles règles. Ceci passe donc par la mémorisation des caractéristiques des scènes engendrées et leur analyse a posteriori
- Raisonnement sur les échecs. Les règles de vérification conduisent de temps en temps à un échec car la base de faits est devenue incohérente. Dans certains cas, il est possible de déterminer la raison de cet échec (combinaison de deux règles incompatibles par exemple). L'analyse des échecs permet alors de définir de nouvelles règles qui vont éviter que d'anciennes situations d'échec se reproduisent.
- Raisonnement sur les règles. Il s'agit cette fois d'un raisonnement formel. Le but est de déduire directement des informations à partir des règles présentes dans la base de règles. Aucun travail n'a pour l'instant abordé ce problème.

Les métarègles interviennent en dehors du mécanisme de génération. Elles permettent d'accumuler de la connaissance pour les explorations futurs. Cette connaissance permet d'optimiser les générations effectuées par la suite car les erreurs ne sont plus reproduites.

Actuellement, aucun moteur n'a encore implémenté de façon efficace ce type de règle. Les essais effectués ont souvent conduit à des échecs car le mécanisme d'apprentissage induisait plus souvent un ralentissement du moteur, suite à la gestion de ces nouvelles règles.

5.3.7 Ordre d'application des règles

Le comportement du moteur d'inférences est directement dépendant de l'ordre d'application des différentes règles. L'ordre est généralement le suivant :

1. Les règles de création : Elles sont appliquées en premier car ce sont elles qui permettent de mettre en oeuvre les faits de la base de faits.
2. Les règles de modification : Ces règles ne s'appliquent que lorsque la scène a été créée. Elles ne peuvent donc qu'intervenir qu'après le mécanisme de génération de scène.
3. Les règles de suppléance : Elles ne s'applique que lorsque les règles de création ne sont plus capables de fournir de nouveaux faits.
4. Les règles de vérification : Lorsque de nouveaux faits ont été créés (par les règles de création ou les règles de suppléance), les règles de vérification peuvent s'appliquer.
5. Les MétaRègles : Elles mettent en oeuvre des déduction sur les résultats obtenus et peuvent donc être appliquée en dehors du mécanisme de génération.

Les règles internes ne peuvent être classées dans cette liste car se sont, pour la plupart des règles de base utilisées par les autres règles.

L'ordre indiqué ci-dessus n'a aucun caractère obligatoire. certaines règles peuvent être appliquées avant d'autres en fonction des besoins, ou des desiderata. L'ordre cité ci-dessus a été établit en suivant un processus logique de conception qui n'est pas forcément optimum pour tous. Il est clair également qu'à l'intérieur de chacune des catégories, l'ordre d'application des règles aura une influence. Ainsi, une règle de création peut dépendre d'une autre règle de création, et dès lors, leur ordre d'exécution n'est pas sans conséquences.

5.4 Autres techniques

5.4.1 Réduction d'intervalles

Les mécanismes que nous avons abordés jusqu'à présent permettent de parcourir des ensembles de valeurs. En un certain sens, on peut considérer que pour ces méthodes, on commence par générer des valeurs, puis qu'on vérifié, a posteriori, si ces valeurs sont correctes. Bien sûr, des mécanismes ont été mis en place pour reculer de plus en plus cet état de fait (comme par exemple les règles de vérification et anticipation).

La réduction d'intervalle aspire a un autre but. Si l'expression d'une partie des propriétés de la scène peut être mise sous la forme d'un système d'équations et d'inéquations, la recherche d'une solution s'apparente alors à la recherche d'une solution dans cet ensemble d'équations et d'inéquations. Il est toutefois rare que cet ensemble fournisse une solution évidente. Dès lors, on définit pour chacun des paramètres un domaine de validité. L'ajout d'une équation ou d'une inéquation va conduire à une éventuelle diminution (ou réduction) de cet intervalle de validité. Lorsque toutes les équation et inéquations ont été prises en compte, on dispose alors de l'intervalle de validité de chacun des paramètres, ce qui conduit à l'ensemble des solutions.

Le cas où l'on peut déterminer directement les intervalles de validité des paramètres est malheureusement peut fréquent. L'interdépendance des paramètres bloque de temps en temps la résolution, et l'adjonction d'une méthode plus classique de parcours (parcours d'arbre ou moteur d'inférences) permet de lever momentanément certains problèmes.

Les techniques de réduction d'intervalle doivent donc être vus plus comme un outil complémentaires de techniques habituelles que comme une méthode de parcours. Cet outil permet de cerner assez précisément (et donc a priori) les intervalles de validité des paramètres définissant la scène.

Lors de problème relativement simples, la réduction d'intervalle peut conduire à définir des intervalles de validité des paramètres tels que toute valeur prise dans ces intervalles conduit à une solution. Cette situation permet alors de construire des "solutions génériques". Ces solutions constituent des données intéressantes pour un mécanisme d'apprentissage.

5.4.2 Tirages aléatoires

Le tirage aléatoire est souvent utilisé lors de parcours d'univers dont certains paramètres sont à valeur réelles ou, de façon plus générale, lorsque le domaine de validité de ces paramètres sont infinis. Il s'applique en dernier recours dans les mécanismes de parcours d'arbres et les moteurs d'inférence.

Ces techniques de parcours sont particulièrement adaptés aux univers dont les paramètres sont finis. Lorsque les domaines de validité des paramètres sont infini, ils essaient de repousser le plus tard possible le moment où ils donneront une valeur à ce paramètre. Une réduction d'intervalle est alors un outil efficace. Il arrive toutefois un moment où cette valeur devient essentielle pour continuer la génération. La valeur étant quelconque dans le domaine de validité, un tirage aléatoire fournit alors de quoi continuer la génération. Il est clair que ce choix est complètement arbitraire, qu'il peut ou non conduire à une convergence du processus de génération, mais que, de manière générale, aucune conclusion ne peut être tirée sur des valeurs voisines du paramètre choisi.

La fonction fournissant une valeur aléatoire revêt une grande importance si elle doit être utilisée. En effet, il peut être intéressant de conserver des informations pour effectuer plusieurs fois une génération identique (comme par exemple pour tester l'influence d'une règle d'optimisation sur un moteur d'inférence). Dans ce cas, le mécanisme de génération aléatoire doit fournir les mêmes séquences aléatoires lors de deux générations successives. Ceci est la plupart du temps possible car le mécanisme de génération fournit des valeurs à partir d'un germe qu'il suffit de mémoriser pour obtenir deux fois de suite des séquences identiques. Lorsque ce n'est pas le

cas, il faut implémenter un mécanisme interne de génération aléatoire (basé le plus souvent sur des fonctions trigonométriques) de manière à disposer de ce germe.

5.5 Apprentissage

L'apprentissage est un outil qui peut être vu de deux manières différentes. D'un certain coté, c'est un outil qui permet à un programme de s'améliorer en acquérant des connaissances sur le domaine qu'il traite et en exploitant ces connaissances. Ce type d'apprentissage est en quelque sorte un apprentissage de l'univers de travail du modeleur. D'un autre coté, au contact de l'utilisateur, le programme peut apprendre à anticiper ses réactions parce qu'il a acquis des connaissances sur les méthodes de travail de cet utilisateur. Nous parlerons alors d'un apprentissage de l'utilisateur.

5.5.1.1 Apprentissage de l'utilisateur

Que ce soit lors de la description d'une scène, de sa modification, ou de la prise de connaissance des solutions, le concepteur peut utiliser des termes ambigus. Il utilise cependant ces termes avec la même signification d'une génération à l'autre.

- Apprentissage des pré-requis. Il est fréquent qu'un utilisateur, souvent par habitude, omette de préciser certains éléments de sa description, parce que, pour lui, ils sont sous-jacents. En fonction des utilisateur, certaines propriétés devront être ajoutées à la description, sauf si l'utilisateur précise explicitement le contraire.
- Apprentissage des paramètres. Les termes utilisés pour décrire des scènes, bien qu'utilisant le même langage, ne correspondent pas forcément aux mêmes résultats. Ainsi, suivant les utilisateurs, le terme grand peut recouvrir des notions légèrement différentes : la nation de grandeur est relative. Il en va ainsi pour tous les paramètres flous qui peuvent être utilisés. Chacun, d'eux utilise des paramètres qui permettent d'expliciter une fonction floue désignant le concept. L'apprentissage des paramètre correspond à un apprentissage du réglage de ces paramètres pour un utilisateur donné.

Tous ces mécanismes d'apprentissage participent au confort de l'utilisateur. Suivant les pré-requis et les paramètres, la génération peut être très différente d'un utilisateur à l'autre. La description « le garage est à droite de la maison » correspond soit à un garage inclus dans la maison mais sur la droite, à un garage collé à la maison et à droite, à un garage séparé de la maison et plus ou moins loin à droite. Toutes ces situations répondent à la description initiale mais peuvent être mise en oeuvre au moyen de pré-requis et du paramétrage.

5.5.1.2 Apprentissage de l'univers

D'une génération à l'autre, il est possible de recueillir des informations permettant d'accélérer ultérieurement les mécanismes de parcours. Ceci peut aller de la simple mémorisation de conjonctions de faits à l'analyse de ces conjonctions. Quelle que soit la méthode utilisée, elle conduit à mettre en place de nouvelles propriétés lorsqu'un ensemble donné de propriétés se trouve réunit.

Ces "propriétés d'apprentissage" ne sont pas nécessairement connues de l'utilisateur, mais elles peuvent modifier considérablement le mécanisme de parcours dans la mesure où ce dernier dispose d'informations qu'il ne possédait pas à l'origine.

Notons toutefois qu'un juste milieu doit être trouvé entre le mécanisme de parcours et le mécanisme d'apprentissage. Ce dernier ne doit pas, de par sa mise en oeuvre, induire un comportement plus lent du parcours. Cela sous-entend qu'une informations apprise doit être suffisamment générale pour que le programme ne passe pas son temps à chercher dans une base de connaissance s'il se trouve dans un cas particulier ou non.

5.6 Intervention des propriétés

Dans un système déclaratif, les propriétés interviennent dans différentes conditions en fonction du moment où elles sont utilisées et de l'influence qu'elles ont. Nous pouvons donc les classer

en fonction de leur rôle dans le processus de conception déclarative. Notons que ces modes d'intervention ne sont pas exclusifs.

5.6.1 Les propriétés de vérification

Une propriété de vérification intervient lorsqu'une solution est construite. Elle permet de vérifier a posteriori si la forme répond à la description. C'est l'action la plus courante et la plus naturelle à mettre en oeuvre. Pour ce type de propriétés, nous pouvons différencier deux comportements possibles :

- la propriété ne vérifie que des solutions complètes, ce sont des propriétés terminales,
- elle vérifie aussi les formes partiellement construites.

5.6.2 Les propriétés de génération

Une propriété peut intervenir pour contribuer à optimiser la méthode de génération en dirigeant et en participant à la construction. C'est une propriété de génération. Nous avons, de ce point de vue deux styles de propriétés :

- les propriétés directives qui permettent d'orienter la recherche en organisant l'espace de recherche et son parcours (orientation de la génération pour avoir les solutions le plus vite possible),
- les propriétés constructives qui permettent d'améliorer l'exploration de l'univers des formes en participant à la construction par réduction statique ou dynamique de l'espace de recherche (réduction des solutions à tester) et par aide directe à la construction (en intervenant sur les paramètres de génération).

Parmi les propriétés constructives, nous pouvons mettre en évidence une catégorie de propriétés remarquable permettant de voir de préférence tel ou tel type de solution. Nous les appelons des propriétés d'ordonnancement.

5.6.3 Les propriétés de modification

Enfin, une propriété peut permettre de modifier, d'affiner une description tout au long de la génération et de l'exploration des solutions. Ce sont les propriétés modificatrices que nous avons vu plus haut. A partir d'une solution, elle permettent d'affiner l'exploration.

5.7 Relations entre propriétés

Dans une description, les propriétés énoncées ne sont pas forcément indépendante les unes des autres. Nous allons regarder maintenant les différentes relations possibles entre elles. Cela nous permet d'améliorer la construction du modèle de description et la gestion de la cohérence.

5.7.1 Niveau d'une propriété

Une propriété peut être mise en relation avec un autre concept au niveau syntaxique. Nous aurons trois niveaux de propriétés :

- Niveau 1 : Ce sont des concepts dont la mesure prend son argument dans la scène (objet, ensemble d'objets ou scène entière). Nous avons par exemple : toutes les propriétés prédicats (quantifiées ou non) sur des propriétés de base, les propriétés relatives (quantifiées ou non)…
- Niveau 2 : Cela concerne les concepts prenant comme paramètre la mesure d'autres concepts. Par exemple, nous avons : les propriétés de comparaison, les propriétés concernant des statistiques sur une propriété locale, celles de dénombrement de propriétés locales…
- Niveau 3 : Concerne les propriétés construites à partir de la composition d'autres propriétés. Nous verrons, dans la suite, comment de telles propriétés sont construites.

5.7.2 Dépendance d'instanciation

Dans la description, des propriétés peuvent être liées par la référence à un objet explicitement désigné par un nom. Prenons la description suivante : « La hauteur de la maison M1 est assez importante (P1). La surface de la maison M2 est plus importante que celle de la maison M1

(P2). La couleur de la maison M2 est blanche (P3) ». Les propriétés P1 et P2 sont liées par la référence à l'objet M1. Nous parlerons de dépendance d'instanciation. Cette dépendance peut être étendue à la propriété P3 car les objets M1 et M2 sont explicitement en relation grâce à la propriété P2.

Par contre, si nous avions eu : « La hauteur d'une maison est assez importante (P1). La surface de la maison M2 est plus importante que celle de la maison M3 (P2). La couleur de la maison M1 est blanche (P3) ». Ces trois propriétés sont indépendantes (même s'il peut arriver que, lors d'une évaluation, la maison de P1 soit M1, M2 ou M3).

5.7.3 Dépendance sémantique

Lors d'une description, il peut arriver plusieurs propriétés fassent référence au même concept. Pour s'assurer de la cohérence, il faut s'intéresser aux liens sémantiques entre les propriétés. Prenons par exemple la description suivante : « La hauteur de la maison M1 est assez importante. La hauteur de toutes les maisons n'est pas extrêmement importante… ». Ces deux propriétés sont liées sur la maison M1 qui doit pouvoir vérifier au moins un peu des deux. Tout dépendra de l'interprétation que l'on fera de la négation (comme on l'a déjà vu).

Ce type de lien entre propriétés peut être direct comme nous venons de le voir mais aussi plus ou moins indirect par l'intermédiaire d'autres propriétés.

5.7.4 Dépendance de construction

Généralement, la description n'est pas fournie complètement, d'un seul trait. Les propriétés sont énoncées les unes après les autres. Le traitement d'une propriété donnée ne sera donc pas le même suivant sa place dans la description.

Par exemple, si l'on donne la propriété « Les maisons ne sont pas très larges », l'interprétation linguistique de cette négation sera différente si l'on place avant ou après la propriété « La largeur des maisons est moyenne ». En effet, si elle est placée avant, les propriétés candidates à la négation ne seront que celles ayant une intersection avec elle. Par contre, si elle est placée après, toutes les propriétés suffisamment différentes seront candidates.

Certaines méthodes de construction traitent une description de manière incrémentale. Une solution est produite après chaque ajout de propriété. Ainsi, la propriété « La prochaine maison est à droite de celles déjà faites » dépend de manière évidente de la description précédente.

5.7.5 Propriétés de composition

Parfois, une propriété ne peut pas s'exprimer à l'aide d'une propriété prédicat. Il est alors nécessaire de combiner des propriétés (prédicats, relatives, de comparaison…) à l'aide d'opérateurs de composition. Les propriétés définis à l'aide d'une composition s'appellent des propriétés de composition.

Nous distinguerons deux classes d'opérateurs de composition :

- les opérateurs conjonctifs ("Et", ".") pour obtenir une vérification de chacune des propriétés de la composition ,
- les opérateurs disjonctifs ("Ou", "Ou Bien") pour construire des propriétés alternatives.

Les énoncés permettant de définir une propriété de composition sont de la forme :

« P signifie : "P1" OP "P2" OP "P3"… ».

où OP est un opérateur de composition.

Cette propriété de composition est alors utilisée dans une description comme une propriété prédicat ou relative. Elle permet de construire des énoncés comme « X est f_α m_β P » (il n'y a pas de concept explicitement utilisé avec de telles propriétés). Il est en effet tout à fait possible de lui appliquer un modificateur et un opérateur flou. Cependant, l'application de tels opérateurs ne sera pas identique à celle effectuée sur une propriété de base d'un concept et dépendra de l'opérateur utilisé.

5.7.5.1 L'opérateur conjonctif "."

Une composition de propriétés dont l'opérateur est "." est une conjonction de propriétés. Elles sont considérées comme totalement indépendantes les unes des autres. Cet opérateur permet de construire une énumération de propriétés demandées simultanément, c'est-à-dire qu'elle n'est vérifiée que si toutes ses propriétés le sont.

La description « La maison M1 est très haute. La couleur des maisons est beige. La hauteur des maisons n'est pas extrêmement importante… » comporte un certain nombre de propriétés, éventuellement liées implicitement par une dépendance sémantique, indépendante les unes des autres en particulier par rapport à d'éventuels modificateurs. L'application d'un opérateur à P sera alors équivalente à l'application de l'opérateur à chacune des propriétés composant P (distributivité des opérateurs).

S'il on considère que « Trapu signifie : La hauteur est faible. La largeur est importante. » alors « Très trapu » sera équivalent à « La hauteur est très faible. La largeur est très importante. ».

5.7.5.2 L'opérateur conjonctif "Et"

Une composition de propriétés dont l'opérateur est "Et" est aussi une conjonction de propriétés. Cependant, son utilisation fait intervenir une ou plusieurs relations implicites entre les opérandes de cette conjonction. Ces relations ne sont possibles que si les domaines de ces opérandes sont comparables au niveau sémantique. Dans le cas contraire, "Et" est totalement équivalent à ".".

Les relations qui lient les différentes propriétés sont implicites. Ce qui veut dire qu'il est assez difficile de les déterminer automatiquement. Comme pour la négation, il faut arriver à faire préciser par le locuteur les éventuelles propriétés implicites de la description.

Ainsi, « Trapu signifie : La hauteur est faible Et la largeur est importante. » peut être équivalent à « Trapu signifie : La hauteur est faible. La largeur est importante. La hauteur est beaucoup plus faible que la largeur. ». Par conséquent, « Très Trapu » sera équivalent à « La hauteur est très faible. La largeur est très importante. La hauteur est beaucoup beaucoup plus faible que la largeur. » (« très beaucoup » étant considéré comme équivalent « beaucoup beaucoup »).

5.7.5.3 L'opérateur disjonctif "Ou"

Une composition de propriétés dont l'opérateur est "Ou" est une disjonction de propriétés. Elles sont considérées comme totalement indépendantes les unes des autres. De toute manière, il est difficile d'imaginer une relation, même implicite, entre deux propriétés qui ne sont pas forcément vérifiées en même temps. Cet opérateur permet de construire une énumération de propriétés alternatives. Elles peuvent être éventuellement vérifiées simultanément, c'est-à-dire que elle est vérifiée si une au moins de ses propriétés l'est.

Du fait de l'indépendance totale des propriétés les unes par rapport aux autres, l'application d'un opérateur sur une propriété de composition sera identique à celle effectuée pour l'opérateur ".".

5.7.5.4 L'opérateur disjonctif "Ou Bien"

Une composition de propriétés dont l'opérateur est "Ou Bien" est une disjonction de propriétés. Son utilisation et son comportement sont identique à ceux de l'opérateur "Ou" à ceci près qu'il faut nécessairement qu'une seule propriété soit vérifiée.

6. Classification sémantique

Lorsque l'on parcourt les différents travaux effectués en modélisation déclarative, très vite on se demande s'il ne serait pas possible de disposer d'une bibliothèque de concepts voir de pouvoir en construire certains automatiquement.

6.1 Les concepts « courants »

Comme le domaine privilégié de la modélisation déclarative en synthèse d'image est la modélisation géométrique, et cela à déjà été montré dans [Des95a], certains concepts ou classes de concepts se retrouvent assez souvent dans un modeleur. Plus généralement, il serait intéressant de disposer de tous les concepts faisant partie de l'ontologie en modélisation (voir [CBB96]).

Nous distinguerons deux catégories de concepts :
- ceux liés à un objet,
- ceux liés à un ensemble d'objets.

6.1.1 Décrire un objet

Les concepts permettant de décrire un objet sont des concepts liés essentiellement à sa géométrie (forme géométrique, topologie, dimensions, orientation…), à son placement dans la scène (souvent mesuré par rapport à la Boîte de Manhattan qui lui est associée) et à ses caractéristiques de rendu (couleur, texture, luminosité…).

6.1.2 Décrire un ensemble d'objets

Dans certaines applications, la scène comporte plusieurs objets éventuellement de type différent. Il existe un ensemble de concepts liés aux relations entre ses objets (propriétés inter-objets) avec, en particulier, les concepts de positionnement relatif, de répartition et de dénombrement (nombre d'objet de chaque type).

6.2 Les concepts construits de manière automatique

Nous venons de montrer que certains concepts se retrouvent très souvent dans les modeleurs. En fait, aux différents objets ainsi qu'à la scène, on peut associer des ensembles de paramètres. Il est alors envisageable de construire des concepts se basant sur ces paramètres. Eventuellement, tel ou tel paramètre ne donnant pas de concept « représentatif » peut être éliminé.

Par exemple, si l'on désire décrire des segments dans un plan (projet FiloFormes), la scène sera paramétrée par les dimensions de la grille sur laquelle seront « posés » les segments. Un segment (objet de la scène) se paramètre par son origine, sa pente et sa longueur. Ainsi, nous pouvons construire automatiquement, à la définition de ces paramètres des concepts comme « la taille verticale de la scène », « la pente d'un segment »…

Plus généralement, nous rencontrons deux types de concepts dans une application déclarative :
- les concepts liés à une caractéristique d'un objet (de base ou complexe),
- les concepts liés à des relations entre les objets.

Les premiers servent éventuellement à générer l'objet alors que les seconds se contentent de vérifier la scène construite.

7. Conclusion

Nous avons mis en évidence quelques méthodes de classement des propriétés composant une description. En particulier, trois axes sont ressortis :
- le syntaxique qui permet de comprendre une description et de construire un modèle de description interne cohérent et fidèle pour la phase de génération,
- le structurel pour manipuler efficacement les différentes propriétés,
- le comportemental pour réduire l'espace de recherche et orienter la génération.
- le sémantique pour construire et manipuler automatiquement certains concepts courants.

Le formalisme présenté ainsi que l'utilisation des différentes facettes des propriétés est actuellement en cours d'implémentation et d'évaluation dans le cadre du projet CordiFormes. En particulier, l'implémentation des propriétés prédicats a permis de valider tout ce qui concerne leur formalisation et leur manipulation.

Ce travail ouvre la porte à un vaste champ d'investigation en particulier sur l'axe comportemental. En effet, un certain nombre de travaux ont été effectués sur ce sujet. Cependant, ils ne portent que sur certains points précis et ne sont pas développés dans le cadre général de la modélisation déclarative. De même, il conviendrait d'affiner la formalisation de l'ensemble des classes syntaxiques proposées afin de travailler dans un cadre formel unique et cohérent.

Bibliographie

[Bea89] C. Beasse, "Description formelle de formes", Rapport de DEA, Rennes, 1989, 74 pages

[Bro43] V. Brondal, "*Essais de linguistique générale*", Copenhage, 1943

[Bro48] V. Brondal, "*Les parties du discours : Etude sur les catégories linguistiques*", Copenhage, 1948

[CBB96] J. Charlet, B. Bachimont, J. Bouaud, P. Zweigenbaum, "Ontologie et réutilisabilité : expérience et discussion" dans "*Acquisition et ingénierie des connaissances*", Cépadues-Editions, Toulouse, 1996, pp. 69-87

[Chau94] D. Chauvat, "Le projet VoluFormes : un exemple de modélisation déclarative avec contrôle spatial", Thèse de doctorat, Nantes, Décembre 94, 225 pages

[Col90] C. Colin, "Modélisation déclarative de scènes à base de polyèdres élémentaires", Thèse de doctorat, Rennes, Décembre 1990, 266 pages

[Col92] C. Colin, "Les propriétés dans le cadre d'une modélisation géométrique déclarative", MICAD 92, Paris, 1992, pp 75-94

[Dan95] M. Daniel, "Une première approche de la modélisation déclarative de courbes", Rapport de recherche IRIN-81, Nantes, Janvier 1995, 11 pages

[Des95a] E. Desmontils, "Les modeleurs déclaratifs", Rapport de recherche IRIN-95, Nantes, Septembre 1995

[Des95b] E. Desmontils, "Formalisation des propriétés en modélisation déclarative à l'aide des sous-ensembles flous", Rapport de recherche IRIN-106, Nantes, Décembre 1995, 41 pages

[Des96] E. Desmontils, "Une formalisation des propriétés en modélisation déclarative à l'aide des ensembles flous", 3IA'96, Limoges, 1996, pp 87-105

[DPa96] E. Desmontils, D. Pacholczyk, "Modélisation déclarative en synthèse d'images : traitement semi-qualitatif des propriétés imprécises ou vagues", AFIG'96, Dijon, 1996, pp 173-181

[DSc95] O. Ducrot, J.-M. Schaeffer et al., "*Nouveau dictionnaire encyclopédique des sciences du langage"* , Editions du Seuil, Paris, 1995

[Elk89] G. Elkharroubi, "Etude des propriétés attachées à un arbre de modélisation", Rapport de DEA, Rennes, 1989, 63 pages

[Gra96] O. Grange, "Matériaux pour la modélisation déclarative de graphes", Rapport de recherche IRIN-124, Nantes, Juillet 1996, 76 pages

[Hje35] L. Hjelmslev, "La catégorie des cas (1)", Acta Jutlandica, 1935

[Hje37] L. Hjelmslev, "La catégorie des cas (2)", Acta Jutlandica, 1937

[LDe96] M. Lucas, E. Desmontils, "Les modeleurs déclaratifs", Revue Internationale de CFAO et Infographie, volume 10, n°6/1995, 1996, pp 559-585

[LMM89] M. Lucas, D. Martin, P. Martin et D. Plemenos, "Le projet ExploFormes : quelques pas vers la modélisation déclarative de formes", Journées AFCET-GROPLAN, Strasbourg, 1989, publié dans BIGRE, n° 67, janvier 1990, pp 35-49

[Mar90a] J.-Y. Martin, "Etude de quelques techniques de parcours pour la modélisation déclarative", Rapport de recherche ENSM-LISI-R16, Nantes, Août 1990, 35 pages

[Mar90b] J.-Y. Martin, “Synthèse d’images à l’aide d’automates cellulaires”, Thèse de doctorat, Rennes, Décembre 1990, 228 pages

[MaM89] P. et D. Martin, “Un système de génération déclarative de polyèdres”, Rapport de recherche LIST 89-02, Nantes, Juin 1989, 76 pages

[MaM90] P. et D. Martin, “Systèmes à base de règles pour la génération et l’exploration d’univers de formes”, Rapport LIST 90-12, Nantes, Juin 1990, 22 pages

[MaM91] P. et D. Martin, “Usage des règles de modification pour la génération déclarative de scène”, Rapport LIST 91-08, Nantes, juin 1991, 44 pages

[Mul91] C. Muller, “*La négation en français”*, Publications romanes et françaises, Genève, 1991

[Paj94] L. Pajot-Duval, “Modélisation déclarative de configurations de segments de droite : le projet FiloFormes”, Thèse de doctorat, Nantes, Juin 1994, 142 pages

[Ple89] D. Plemenos, “Contribution à l’étude et au développement des techniques de modélisation, génération et visualisation de scènes : le projet MultiFormes”, Thèse de doctorat d’état, Nantes, 1991, 308 pages

[Pou94] F. Poulet, “Modélisation déclarative de scènes tridimensionnelles par énumération spatiale : le projet SpatioFormes”, Thèse de doctorat, Rennes, Juin 1994, 135 pages

[Zad65] L. A. Zadeh, “Fuzzy Sets, Information and Control”, vol. 8, 1965, pp 338-353