

Une approche interactive de définition d'ontologies image

An Interactive Approach For Image Ontology Definition

Alexandre NOUVEL

Patrice DALLE

Institut de Recherche en Informatique de Toulouse

IRIT - Université Paul Sabatier
118 route de Narbonne
31062 Toulouse Cedex 4
FRANCE

nouvel@irit.fr

dalle@irit.fr

Résumé :

Nous proposons une approche interactive pour la réalisation d'applications de traitement d'images. Dans cette approche, l'utilisateur associe les résultats d'enchaînements d'opérateurs à des concepts du modèle de la scène, au lieu de procéder par cycles "essai-erreur" d'opérateurs (choix, paramétrage et exécution d'un opérateur, évaluation de ses effets, puis ajustement ou correction). La définition des concepts est conduite par l'utilisateur qui les manipule directement, de même que leurs instances —des entités produites par les opérateurs.

La définition incrémentale des concepts, —par spécification, extraction et nomination d'entités ou par spécialisation et généralisation d'autres concepts— revient à construire une ontologie décrivant le domaine d'application. L'intérêt de cette démarche est de masquer les fonctionnalités du traitement d'images, de raisonner en termes de concepts du domaine d'application, et de se placer dans une logique d'interprétation.

Mots-clés :

traitement d'images, construction d'applications, définition de concepts, extraction d'entités, ontologie

Abstract:

We propose an interactive approach for image processing application design. Within this approach, the user binds results from operator scripts to concepts of the scene model, instead of performing a "trial and error" process with the operators (one chooses an operator, sets its parameters and runs it, then evaluates

its effects, and adjusts or corrects the whole sequence). The concept definition is driven by the user, who handles directly concepts and their instances —entities produced by the operators.

The incremental definition of the concepts, —through entity specification, extraction and naming or through the specialization and generalization of other concepts— can be assimilated to the production of an ontology that describes the application domain. The interest of this approach consists in hiding the image processing functionalities, in being able to reason in terms of the application domain and in viewing the problem in an interpretation way.

Keywords:

image processing, application design, concept definition, entity extraction, ontology

1 Introduction

Nous nous plaçons dans le cadre des recherches visant à faciliter la conception d'applications de Traitement d'Image (TI). Ces recherches étudient des environnements de conception d'applications de TI qui se présentent sous la forme de systèmes de pilotage (Ocapi [13]) ou de planification (Borg [3], ExTI [6]) de bibliothèques d'opérateurs de TI.

Il est cependant possible de réaliser une application de manière incrémentale et interactive, sans passer par des systèmes aussi complexes, mais par essai d'opérateurs de TI, évaluation des résultats et correction si nécessaire. Cette démarche est justifiée lorsque le problème est

simple, mais aussi lorsqu'on veut construire l'ontologie d'un domaine, avant de définir des applications proprement dites exploitant cette ontologie. On peut utiliser pour cela des environnements interactifs de TI dans lesquels l'utilisateur peut accéder aux opérateurs directement, les paramétrer interactivement puis provoquer leur exécution. De tels environnements permettent d'enregistrer l'enchaînement des opérateurs utilisés sous forme de scripts et permettent parfois d'introduire des structures de contrôle dans ces scripts (Khoros [16], Sage [7]). Leur utilisation suppose cependant que l'utilisateur soit suffisamment expert en TI pour savoir manipuler directement les opérateurs et leur paramétrage.

Sans chercher à automatiser ce processus de conception, nous proposons une autre forme de démarche qui permet à l'utilisateur de raisonner en termes d'entités du TI (pixels, points contours, régions...), d'effets de traitements, d'informations et de concepts (« zones claires », « région allongée traversant l'image », « pixels bleus regroupés en haut de l'image »...), plutôt qu'en termes de fonctionnalités ou d'opérateurs. Cette approche originale lui permet de rester dans son domaine d'application, notamment par la définition de concepts basés sur des entités. L'objectif est donc de mettre l'accent sur la mise en relation des effets des opérateurs (production, modification d'entités) avec des concepts du modèle de la scène, en permettant à l'utilisateur de spécifier par des interactions les indices visuels caractéristiques des concepts recherchés. C'est le système qui sélectionnera et qui exécutera les opérateurs correspondants.

Nous décrivons dans cet article la manière dont l'utilisateur construit interactivement et de manière incrémentale les concepts en manipulant et en faisant produire par le système les entités qui lesinstancient (§3). Nous montrons comment le processus de définition des concepts se rapproche de la construction d'une ontologie du domaine d'application (§4). Chaque concept est associé à une expression symbolique ainsi qu'à une chaîne d'opérateurs qui en produit les instances. Enfin, nous donnons une analyse critique du système tel qu'il est défini et nous proposons de nouvelles perspectives pour l'étendre et l'enrichir (§5).

2 État de l'art

En ce qui concerne la conception interactive d'applications de TI, assez peu de travaux sont directement reliés à la formalisation des données, tant du point de vue de leur représentation que des transformations que ces données subissent. Pour ce qui est de la conception proprement dite d'applications, plusieurs pistes sont explorées. Parmi celles qui se sont orientées vers la conception interactive, nous ne considérerons ici que celles qui font appel à une manipulation directe des données plutôt que des opérateurs. Nous présentons dans la dernière partie les systèmes de production d'ontologies.

2.1 Formalisme pour les entités de TI

Le formalisme présenté par Ph. Dejean [10] fournit une description orientée concepts et entités du TI, dans une perspective de planification de chaînes d'opérateurs [6] et de description d'objectifs de TI. Ce formalisme propose d'abord un modèle unifié des données de TI [8] en cinq champs (et quatre champs complémentaires) représentant l'information signal et ses transformées ainsi que l'information géométrique et ses structurations. Le Langage de Description de Données (LDD) permet de décrire les données avec leur structure. Il utilise les catégories syntaxiques du Langage de Description de Concepts (LDC) pour définir cinq catégories syntaxiques correspondant aux cinq champs du modèle de données.

Ce modèle permet de décrire précisément les caractéristiques des données (en termes de transformations subies ou en termes de spécification de transformations à faire subir aux données). Les concepts décrivant les champs d'une donnée sont basés sur des propriétés, représentées par des couples <descripteur de mesure, domaine de valeurs spécifié pour la mesure> [5]. Un concept est défini par un ensemble de propriétés, par une conjonction ou par une disjonction de concepts. Ces propriétés correspondent à des notions de l'utilisateur (position, forme, occupation spatiale, orientation, aspect) à propos des entités. Un concept a pour représentation une expression symbolique dans le langage LDC.

Le modèle d'opérateurs associé a pour but de faire apparaître les transformations que la donnée subit. Il fournit de plus des informations sur les types d'entités admises en entrée et en sortie d'un opérateur. À partir de la représentation unifiée des données, un opérateur est représenté en indiquant comment cinq opérations (nomination, spécialisation, généralisation, structuration et déstructuration) sont réalisées sur les différents champs de la donnée et leur champs complémentaires. Un ensemble de fonctions de transformation élémentaires applicables sur ces champs a été défini à cet effet. Les transformations font intervenir des descripteurs pour expliciter la manière dont l'opérateur mesure et détermine une partition de la donnée lors du traitement.

Un opérateur peut alors être décrit et sélectionné selon les entités qu'il manipule ainsi que par les descripteurs explicitant les transformations qu'il réalise : c'est ainsi que le lien peut être fait avec les concepts dont les entités d'entrée et de sortie sont des instances. Lorsqu'une entité [15] a été produite ou modifiée par un opérateur, on lui attache une description explicitant comment les transformations élémentaires issues du modèle de l'opérateur ont modifié les champs de la donnée associée. Ainsi la représentation d'une entité rend compte à la fois de ce que contient l'entité et des actions effectuées pour l'obtenir. Nous exploitons ces informations dans notre système interactif pour proposer à l'utilisateur des moyens d'intervention pertinents sur les données et pour traduire, de manière transparente, ces interactions en termes de traitements.

2.2 Conception d'applications de TI

V. Ficet a présenté une manière de construire interactivement un modèle conceptuel d'applications de TI [11]. Les différents niveaux d'abstraction des fonctionnalités du TI qui y sont modélisés donnent à l'utilisateur un moyen d'appréhender le TI à un niveau plus explicite et mieux adapté à ses compétences (lorsqu'il n'est pas un expert du TI). L'utilisateur construit ici un plan, en étant aidé par le système pour se diriger dans la hiérarchie des buts, mais il ne manipule pas directement des entités issues des données.

L'extension qui en est faite sous la forme d'une méthodologie de conception d'applications de TI (Athéna, [4]) propose des outils pour la construction et l'exploitation d'un ensemble structuré de tâches que l'on pourrait apparenter à une forme de construction d'une "ontologie de méthode" (au sens de KADS, [18]) propre au TI : il s'agit ici d'une description des stratégies d'utilisation de fonctionnalités d'opérateurs de TI, des méthodes d'utilisation de ces opérateurs, des mises en œuvre des opérateurs et finalement des relations que ces trois niveaux entretiennent. Cependant, dans ce système, les concepts du domaine ne sont toujours pas représentés explicitement.

Dans le système SATI [2], l'accent est mis sur la formulation interactive d'objectifs de TI par la manipulation directe d'instances de résultats. L'exploitation des objectifs de TI formulés consiste à planifier une chaîne d'opérateurs qui réalise ces objectifs. Les objets manipulés sont décrits en tant qu'entités du TI. Ce travail met en lumière la structuration de ces entités. Il décrit également de façon partielle les manipulations et les interactions possibles sur ces entités, en évoquant certains descripteurs. Notre travail se situe dans la continuité de cette approche, mais il n'est pas associé comme dans [2] à un lourd module de planification rendant illusoire une réelle possibilité d'interaction.

2.3 Outils pour la construction d'ontologies

Il existe de nombreux travaux sur des outils de construction d'ontologies se basant sur du texte, que ce soit d'un point de vue de la méthodologie de construction [14], ou du point de vue appliqué, dans des domaines aussi variés que les textes de loi [12] ou le domaine médical [19].

Dans le domaine de l'analyse d'images et des documents multimédia, des recherches sont en cours, en particulier sur les documents vidéo, pour modéliser leur structure [17]. L'objectif est de fournir des outils pour indexer de façon automatique un document en exploitant un modèle de sa structure. La définition de tels modèles de structure et les opérations de construction de documents vidéo reste cependant manuelle, dans la mesure où il n'existe pas d'outil pour manipuler et associer les concepts du modèle dans la phase de définition de celui-ci.

Ces travaux visent à la définition d'une "ontologie d'application" (vue comme la double spécialisation d'une ontologie du domaine et d'une ontologie de méthodes par KADS, [18]) propre aux documents vidéo : les concepts sont connectés par des relations de généralisation et de composition, ils représentent des notions comme un plan, une transition... Par contre, ils ne permettent pas de réaliser une ontologie du domaine pour les informations contenues dans ces documents, comme par exemple un présentateur de journal ou un ballon de football.

3 Construction interactive et incrémentale de concepts

Un concept représente une notion visuelle pour l'utilisateur. Elle peut être liée ou non à des éléments du domaine des connaissances de l'utilisateur sur le contenu des images qu'il cherche à analyser.

Le concept peut représenter une notion visuelle pertinente bien que sans rapport direct avec le contexte de l'application que l'utilisateur cherche à résoudre : par exemple, « objet clair », « objet allongé » ou « objet clair allongé qui traverse l'image ». Nous qualifions un tel concept de "générique".

Un concept peut aussi être associé à une notion du domaine de compétence de l'utilisateur. Ceci peut être fait de deux manières : par renommage d'un concept générique reconnu par l'utilisateur en tant que concept de son domaine (« objet clair allongé qui traverse l'image » pourra être renommé par exemple dans une image de cartes électroniques en « fil traversant l'image », ou bien en « autoroute » dans une image aérienne), ou par construction à partir d'autres concepts étant eux-mêmes génériques ou du domaine de l'utilisateur (par exemple, le concept « lacs » est défini par spécialisation du concept générique « régions de texture ondulée » par adjonction d'une propriété de surface faisant référence au concept du domaine « mer »).

Dans notre système, les données manipulées sont présentées à l'utilisateur sous forme de groupes d'entités. Chaque groupe d'entités est l'instance d'un concept que l'utilisateur peut nommer et réutiliser. Un groupe d'entités est soit le résultat de l'extraction d'informations d'une donnée suite à la définition d'un concept par l'utilisateur (par exemple, une partition en régions de l'image suite à la définition du concept « objets définis par leur texture »), soit le résultat de la sélection de certaines entités dans un autre groupe d'entités, selon une propriété spécifiée par l'utilisateur (par exemple, le sous-ensemble des régions dont le degré de compacité est compris dans un certain intervalle). Dans notre interface, les concepts ne sont pas directement manipulables, sauf par le biais d'entités qui les instancient. Dans la suite de cet article, nous utiliserons le terme "concept" pour désigner indifféremment un concept ou le groupe d'entités associé, sauf mention contraire.

Définir un concept consiste donc pour l'utilisateur à caractériser un ensemble de transformations apportées par une chaîne d'opérateurs sur des données. Il doit indiquer pour cela les propriétés sur lesquelles les transformations doivent porter et les caractéristiques souhaitées pour la donnée à produire. Les chaînes d'opérateurs sont masquées à l'utilisateur, mais peuvent être réutilisées par le système pour instancier des concepts à partir de nouvelles données, afin de tester si une définition de concept correspond bien à la notion que l'utilisateur voulait représenter par ce concept.

Pour produire des entités, l'utilisateur a à sa disposition une liste des concepts déjà produits. Trois catégories d'opérations sont possibles sur un concept :

- faire produire par le système un ensemble de nouvelles entités par structuration ou déstructuration d'un groupe d'entités ;
- modifier la définition du concept en accédant aux propriétés ayant servi à spécifier ses caractéristiques : généralisation ou spécialisation du concept ;
- construire la définition du concept par union (disjonction) de concepts existants ; cette opération n'est pas pour le moment réalisée dans notre système.

Enfin, une opération de nommage de concept est proposée ; elle permet de construire le lexique associé au domaine et de réutiliser par la suite la définition du nom du concept ou des mesures portant sur les entités qui le composent.

3.1 Structuration et déstructuration

Un niveau de structuration est associé à chaque concept, explicitant la nature des entités quiinstancient ce concept (pixels, régions, amas de régions). Le système dispose d'un modèle lui permettant de déterminer les niveaux accessibles à partir de celui de l'entité courante. Par exemple, à partir d'un groupe de régions déjà extraites, il est possible soit de parler individuellement des pixels (en déstructurant les régions), soit de regrouper certaines régions en un amas (structuration). Ceci permet au système de proposer à l'utilisateur des actions de structuration ou de déstructuration sur le concept (figure 1). D'après le modèle des opérateurs et en connaissant le niveau de structuration du concept en entrée ainsi que le niveau de structuration à atteindre, le système détermine une liste d'opérateurs potentiellement déclenchables, mais sans la montrer à l'utilisateur. Plusieurs opérateurs peuvent être sélectionnés par le système à ce stade. Chacun, dans le modèle des opérateurs, est associé à une liste de descripteurs des propriétés qui seront mesurées et transformées par la fonction qu'il réalise. Pour déterminer l'opérateur à appliquer, le système présente à l'utilisateur la liste de tous les descripteurs envisageables compte tenu des opérateurs potentiellement applicables.

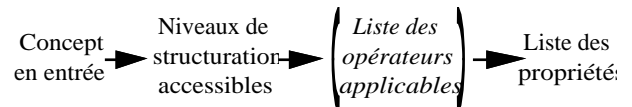


Figure 1 — Du concept aux propriétés permettant de le structurer ou de le déstructurer

L'utilisateur sélectionne un descripteur de propriété sur lequel il désire que le système se base pour effectuer la transformation demandée. Cette sélection déclenche le choix de l'opérateur à mettre en œuvre (figure 2). Pour permettre le réglage des paramètres de l'opérateur, nous lui associons la notion qu'il représente (par exemple un seuil élevé sur un détecteur de contour définit une notion de contraste). À travers cette notion, les paramètres de l'opérateur choisi sont alors présentés à l'utilisateur dans des fenêtres de dialogues indépendantes.

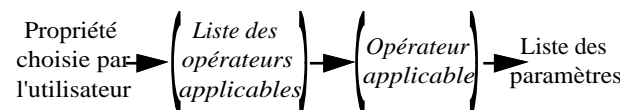


Figure 2 — Des propriétés aux paramètres

Dans chacune de ces fenêtres, l'utilisateur associe un nom de concept, une mesure sur un concept ou bien une valeur brute au paramètre. Une fois que tous les paramètres sont associés à des concepts, l'opérateur est déclenché. Les entités qu'il produit sont ajoutées à la liste des entités disponibles et un nom pour le concept sous-jacent à ces nouvelles entités est demandé à l'utilisateur.

3.2 Spécialisation et généralisation

Il s'agit ici d'opérations de définition d'un concept, par manipulation de la liste des propriétés spécifiant ses caractéristiques. Une propriété d'un concept est ici vue comme une information comportant deux composantes (descripteur, valeur), sur laquelle l'utilisateur peut agir par des interactions (figure 3). La composante valeur est ici une valeur unitaire (c'est à dire un point de l'espace des valeurs possibles), une plage de valeurs unitaires ou une série de plages de valeurs unitaires.

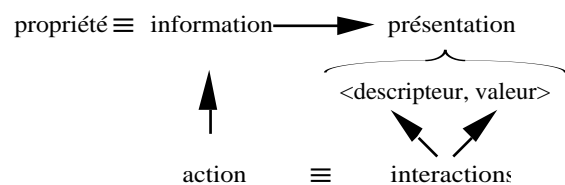


Figure 3 — Couples action-information : interactions sur les composantes d'une propriété

La spécialisation d'un concept est le résultat de l'ajout d'une nouvelle propriété, de la spécialisation du descripteur d'une propriété existante, de la spécialisation de la valeur associée au descripteur d'une propriété

existante, ou de la disjonction de concepts. De façon symétrique, la généralisation d'un concept est le résultat de la suppression d'une propriété, de la généralisation du descripteur d'une propriété existante, de la généralisation de la valeur associée au descripteur d'une propriété existante, ou du retrait d'un concept d'une disjonction de concepts.

Les opérations possibles sur la liste de propriétés d'un concept sont donc :

- l'ajout d'une propriété : cela est associé à une action d'initialisation de la propriété ;
- la modification d'une propriété : cela est associé à une action de correction de la propriété par modification du descripteur ou de la valeur ;
- la suppression d'une propriété : cette opération est triviale au niveau de la construction d'une nouvelle spécification de concept (on revient au concept parent).

Nous définissons dans la figure 4 les différents types d'actions associées aux opérations d'ajout et de modification de propriété :

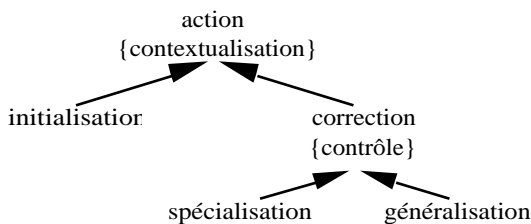


Figure 4 — Les différents types d'actions

Une action peut prendre un statut spécifique (indiqués dans la figure 4 par les termes entre accolades) selon l'intention de l'utilisateur et selon le moment où il va initier cette action dans son processus de conception. Les attributs représentant le statut d'une action sont ici vus au sens "objet", ils sont donc hérités par les actions des niveaux inférieurs.

Le statut de contextualisation est acquis pour une action lorsque l'utilisateur définit une propriété en se basant sur des propriétés ou des mesures en provenance d'une entité ou d'un concept existant.

L'initialisation est par définition l'action qui consiste à effectuer le premier choix d'une valeur par l'utilisateur lors de l'ajout d'une nouvelle propriété dans la liste. La valeur peut se baser sur tout type de valeur source (valeur brute saisie au clavier, valeur issue d'une mesure sur une entité...). Une action est implicitement définie comme une initialisation si elle est déclenchée par le choix d'un nouveau descripteur sur une entité par l'utilisateur.

Une correction est une modification de valeur. Lorsque l'utilisateur effectue des corrections successives, celles-ci peuvent être interprétées par le système comme un moyen d'obtenir une définition de la valeur (par exemple : trouver le plus petit objet ayant une certaine propriété). Ce mécanisme est destiné à générer une structure de contrôle (de type optimisation). Nous étudions les moyens de produire ce fonctionnement d'après les interactions de l'utilisateur. De la même manière que pour l'initialisation, la correction est une manipulation implicite d'une valeur par sa définition (on ne peut corriger qu'une valeur déjà définie).

Lorsque la correction d'une propriété passe par la généralisation du descripteur ou de la valeur, cela revient à parler d'une action de généralisation du concept. De façon symétrique, lorsque la correction d'une propriété passe par la spécialisation du descripteur ou de la valeur, cela revient à parler d'une action de spécialisation.

Dans le système que nous développons, nous n'avons pas différencié ces cas par des méthodes d'interaction séparées. D'une façon générale, l'utilisateur ayant accès à la liste des propriétés d'un concept, il peut soit modifier directement cette liste en ajoutant ou en supprimant une propriété (ce qui revient à spécialiser ou à généraliser ce concept), soit sélectionner une propriété pour en modifier le descripteur ou la valeur. Comme il n'est pas possible a priori de déterminer si l'utilisateur désire créer un nouveau concept à partir de celui qu'il modifie, ou simplement remettre en cause un choix précédent, le système crée systématiquement un nouveau concept à la fin de l'opération et propose à l'utilisateur de lui donner un nom basé sur le nom du concept de départ.

Nous allons maintenant détailler la façon dont l'utilisateur peut initialiser ou modifier les composantes d'une propriété (descripteur, domaine de valeurs).

Les descripteurs sont pour le moment organisés en cinq grandes catégories (position, forme, occupation spatiale, orientation, aspect). Nous étudions les manières de relier les descripteurs, au moins de manière partielle, par des relations de généralisation (par exemple, la largeur est une plus petite dimension selon une direction précise) ainsi que par des relations de dépendance (par exemple, la surface et le périmètre varient ensemble, à forme égale), notamment en nous basant sur le travail entrepris dans [5]. Ceci permettra de proposer des alternatives à un descripteur donné lors d'une interaction de contextualisation, mais, pour le moment, cette fonction n'est pas disponible dans notre système. Dans le cas d'une initialisation, l'utilisateur doit déjà choisir un descripteur en deux étapes :

- choix de la catégorie de descripteur ;
- choix du descripteur. C'est à ce niveau qu'interviendra un raffinement de choix lorsqu'une hiérarchie de descripteurs sera définie.

Pour chaque descripteur, nous avons défini un mode d'interaction correspondant aux types de valeurs que celui-ci accepte. Ceci permet de proposer une zone de dialogue adaptée à la saisie du type de valeur concernée. La saisie d'une valeur peut avoir lieu de plusieurs manières :

- saisie d'une valeur brute : l'ensemble du domaine de valeurs possibles est présenté, l'utilisateur effectue directement sa sélection dans ce domaine. Par exemple, pour la saisie d'une surface, une série de curseurs permettent de définir une valeur numérique, un intervalle de valeurs ou une série d'intervalles de valeurs sur un axe allant de 1 à la surface totale de l'image ;
- référence à la valeur associée à un descripteur équivalent (c'est-à-dire identique ou plus spécifique, au sens d'une hiérarchie des descripteurs) caractérisant un autre concept. Elle est réalisée par une interaction de type glisser-déplacer ou copier-coller. Par exemple, pour saisir une valeur de couleur, l'utilisateur copie, dans la liste des propriétés d'un autre concept, une valeur de couleur ou une valeur de niveau de gris et la colle à l'emplacement "valeur" de la propriété en cours de modification ;
- référence à une valeur mesurée sur un concept qui n'est pas caractérisé par un descripteur équivalent. Elle est réalisée par une interaction de pointage sur le concept de référence. Par exemple, pour saisir une valeur de couleur, l'utilisateur sélectionne simplement un concept sur lequel il veut mesurer cette caractéristique.

Suite aux actions de l'utilisateur sur la liste des propriétés d'un concept (ajout, modification ou suppression d'une propriété), nous définissons la manière dont le système produit la nouvelle instance du concept.

L'ajout d'une propriété va spécialiser le concept, et donc réduire le nombre d'entités quiinstancient ce concept. Cette réduction passe par un opérateur de sélection des entités. Cet opérateur effectue une mesure selon le descripteur de la nouvelle propriété et ne conserve que les entités dont la valeur mesurée appartient au domaine de valeurs de cette nouvelle propriété.

Dans le cas d'une modification ou d'une suppression d'une propriété, et comme nous définissons incrémentalement les concepts (chaque ajout de propriété définit un nouveau concept, qui encapsule la liste des propriétés du concept de départ dans une référence à ce concept de départ), la propriété modifiée ou supprimée est directement reliée avec le dernier opérateur employé. Celui-ci va être appliqué de nouveau sur le concept

initial, ses paramètres étant instanciés avec les valeurs définies lors de son premier appel par le système, à l'exception de celles qui viennent d'être modifiées par l'utilisateur.

3.3 Synthèse

Nous avons vu qu'un opérateur n'est à aucun moment présenté en tant que tel à l'utilisateur, et que ce dernier ne manipule que des entités, les concepts sous-jacents et leurs propriétés : les fonctionnalités des opérateurs et du TI sont donc bien masquées et l'utilisateur est sollicité pour ce qu'il connaît bien : les notions du domaine, via leurs descripteurs ou via des instances. Bien que masqués, les opérateurs restent cependant présents dans le système, de façon sous-jacente, en étant organisés sous formes de graphes d'opérateurs associés aux concepts.

La démarche de construction est incrémentale dans la mesure où chaque apport d'information par l'utilisateur est exploité par le système pour construire un nouveau concept et pour adapter le graphe d'opérateurs permettant d'instancier ce concept. Elle est aussi interactive, par les outils que propose le système pour manipuler les concepts, leurs instances, ainsi que les descripteurs et les valeurs.

4 Graphe de concepts et ontologie

Les concepts dénotent individuellement des notions visuelles pour l'utilisateur ; lorsque celui-ci a défini des concepts correspondant à des notions visuelles de son domaine, ces concepts constituent les mots du vocabulaire de ce domaine. La définition d'un nouveau concept faisant référence à un ou plusieurs autres concepts déjà définis, son ajout à la liste des concepts existants constitue un enrichissement du vocabulaire du domaine.

Les concepts sont associés à une série de transformations sur une donnée, mise en œuvre par un graphe d'opérateurs (*cf.* [9]). En d'autres termes, chacun des mots du vocabulaire définit la manière d'en obtenir une ou plusieurs instances (si elles existent) dans une donnée.

Il est à noter que les termes du vocabulaire ne sont pas, normalement, directement assimilables aux opérateurs. En effet, à partir d'une expression représentant un concept non instancié par des entités, on ne peut pas en déduire les opérateurs sans passer par une phase de planification. La démarche incrémentale de construction de concepts de notre système fait cependant en sorte que chaque concept (défini par la production des entités qui l'instancient), est associé à la chaîne d'opérateurs ayant produit ces entités.

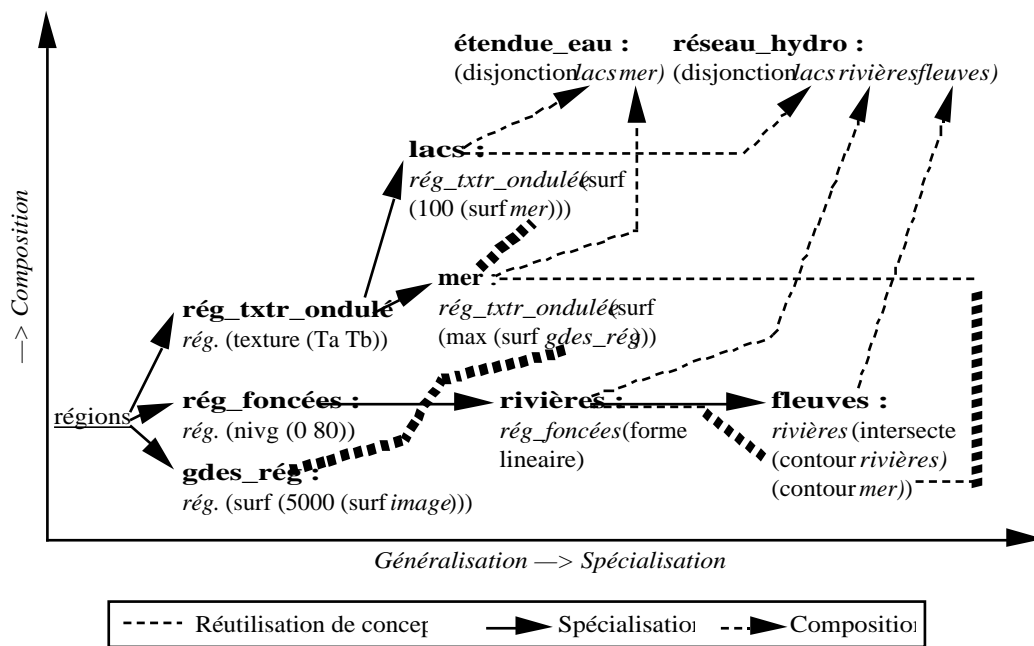


Figure 5 — Exemple de graphe de concepts

La définition d'un concept se fait presque toujours par rapport à un ou plusieurs autres concepts (figure 5), sauf dans le cas des concepts de bas niveau (en général non liés au domaine d'application).

Nous avons détaillé les différentes manières de produire un concept à partir d'un autre (§ 3), par les opérations de structuration/déstructuration d'entités, de généralisation/spécialisation ou de disjonction de concepts, ainsi que celles permettant de caractériser un concept à partir d'un autre, en qualifiant la valeur d'une propriété d'après un concept extérieur. Chacune de ces opérations crée des liens organisant les concepts en graphes, selon trois axes :

- axe de composition/décomposition ;
- axe de spécialisation/généralisation ;
- axe transversal de transmission de propriété, qui vient toujours se greffer sur une relation de l'axe de spécialisation/généralisation.

Les relations sur l'axe transversal sont décrites dans le LDC au niveau de la composante valeur d'une propriété. Dans ce cas de relation transversale, la composante valeur d'une propriété peut être définie par une mesure, par une fonction appliquée à une mesure ou par une fonction reliant deux mesures (comparaison, opération mathématique...).

L'ensemble des relations qu'il est possible d'établir entre les concepts avec notre système s'apparente à celui que l'on retrouve lors de la construction d'une ontologie du domaine. Ceci nous amène à dire que notre système tend

à fournir les outils nécessaires à la création d'une ontologie du domaine d'application de l'utilisateur : les liens de généralisation/spécialisation permettent de construire une hiérarchie de concepts (taxonomie), les liens de composition/décomposition permettent de construire des relations tout-parties (méronymies) et les liens transversaux permettent de comparer et d'organiser les concepts entre eux dans un espace sémantique.

5 Analyse critique de la définition du système

Notre système permet actuellement de masquer les fonctionnalités du TI, en particulier les opérateurs, en ne présentant à l'utilisateur que les groupes d'entités instanciant des concepts et des informations sur ces concepts (propriétés). Il répond en cela à notre objectif de fournir à l'utilisateur des outils lui permettant de parler de son domaine en employant des notions qu'il comprend : les entités et les concepts. Le système propose des mécanismes permettant de définir un nouveau concept par généralisation ou spécialisation d'un concept, et enfin par structuration ou déstructuration d'entités. Il permet également de nommer des concepts.

En ce qui concerne le contrôle nous pouvons proposer une démarche pour le générer : elle consiste à imposer à l'utilisateur d'utiliser explicitement des mesures à chaque fois qu'il évalue une situation. Ceci étant assez contraignant, nous ne l'avons pas encore mis en œuvre, et nous cherchons des techniques plus transparentes pour obtenir le même résultat.

Enfin, notre système est conçu dans une optique incrémentale. Par conséquent, chaque ajout de propriété entraîne la définition d'un nouveau concept. La liste des propriétés d'un concept se résume donc à la plus récente de ces propriétés ; il n'y a pas de possibilité de capitaliser les propriétés ajoutées à un concept si l'on revient sur la définition d'une propriété en amont de ce concept.

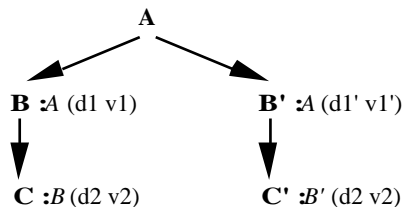


Figure 6 - Encapsulation des propriétés

Dans la figure 6, le concept A est spécialisé en B par l'ajout de la propriété <d1 v1>. B est spécialisé en C par l'ajout d'une deuxième propriété. Si l'utilisateur désire modifier, au niveau de C, la propriété <d1 v1>, il ne peut pas le faire, car les propriétés du concept parent d'un concept édité sont inaccessibles depuis le concept édité (ici, la propriété <d1, v1> n'est accessible que dans la définition de B). Si l'utilisateur veut en modifier une, il devra éditer le concept B et modifier la propriété <d1 v1> en <d1' v1'>, ce qui aura pour conséquence de produire une nouvelle spécialisation B' de A. Après quoi, l'utilisateur sera obligé de spécialiser B' avec les mêmes propriétés qu'il avait utilisé pour spécialiser B en C, afin de produire un concept C' du même niveau de spécialisation que C. Ceci revient à faire des cycles essais-erreurs, mais sur les concepts.

S'il n'y avait pas cette encapsulation de propriétés au niveau de chaque concept, on pourrait bien disposer de la liste complète, dans C, des propriétés ayant permis de spécialiser A en C. Il serait donc possible de modifier la forme extérieure du concept (son expression symbolique) pour produire directement celle de C', mais cela ne permettrait pas pour autant d'effectuer la mise à jour de la chaîne d'opérateurs permettant d'instancier C'. Pour cela, un mécanisme de planification serait nécessaire.

6 Conclusion et perspectives

Dans cet article, nous avons détaillé les mécanismes que nous proposons pour manipuler et caractériser des entités du TI de manière interactive, afin de construire incrémentalement un ensemble de concepts du domaine de l'utilisateur, en vue de la réalisation d'une application de TI. Nous avons présenté les opérations permettant de structurer, déstructurer, spécialiser, généraliser, composer et relier des concepts de manière interactive tout en faisant appel aux connaissances de l'utilisateur sur son domaine. Les descripteurs utilisés pour caractériser les concepts permettent également de rendre compte des transformations apportées aux entités par les opérateurs déclenchés suite aux interactions de l'utilisateur. Enfin,

nous avons présenté un mécanisme permettant de lier les opérateurs de TI aux entités en exploitant des interactions sur les valeurs associées aux descripteurs des entités.

À partir de ces différents éléments, nous analysons l'ensemble des concepts produits par l'utilisateur comme un graphe conceptuel comportant des relations de taxonomie, de méronomie, ainsi que des relations dites transversales, permettant de définir une valeur d'une propriété d'un concept par rapport à un autre concept et d'organiser les concepts d'un point de vue sémantique. Cette analyse nous conduit à voir notre système comme un outil d'aide à la production d'une ontologie du domaine d'application, en vue d'une exploitation de cette ontologie pour de l'interprétation d'images.

Le système que nous développons possède des fonctionnalités permettant de masquer les opérateurs de TI à l'utilisateur. Il lui manque encore un mécanisme pour mettre en œuvre une fonction d'agrégation de concepts, permettant ainsi à l'utilisateur de définir un concept par disjonction de concepts.

D'autre part, il manque également un mécanisme de replanification. Ceci peut être résolu en réutilisant le planificateur décrit dans [6]. Une autre solution est l'intégration des travaux présentés dans cet article au système étudié dans [1]. Ils seront exploités en tant qu'interface entre l'utilisateur et un système multi-agents. Ce dernier, dans lequel chaque agent encapsule un opérateur de TI, possède sa propre autonomie pour construire des concepts, les confronter à ceux de l'utilisateur et évoluer pour parvenir à un accord sur la définition des concepts. La liaison entre la partie interface et la plate-forme multi-agents doit exploiter une mise en œuvre objets du langage LDC, l'expression d'un concept devant être perçue par la société d'agents comme une incitation à produire des entités et des informations sur les transformations ayant abouti à ces entités.

Références

- [1] Y. Abchiche, P. Dalle, Y. Magnien — Construction adaptative de concepts par structuration d'entités de traitement d'images. RFIA 2002 (soumis).
- [2] O. Capdevielle, P. Dalle — Formulation d'objectifs de traitement d'images. IHM'95, 7èmes journées sur l'Ingénierie des Interfaces Homme-Machine, Toulouse, octobre 1995.
- [3] R. Clouard, A. Elmoataz, M. Revenu — Une modélisation explicite et opérationnelle de la connaissance de traitement d'images. 11ème congrès RFIA, T II, pp. 65-74, Clermont-Ferrand, janvier 1998.
- [4] R. Clouard, A. Elmoataz, M. Revenu — Méthodologie de développement d'applications de traitement d'images, 17ème colloque GRETSI, Vannes, pp. 323-326, septembre 1999.

- [5] P. Cros — Typologie des mesures de traitement d'images. DEA Informatique de l'Image et du Langage, Université Paul Sabatier, Toulouse, juin 1997.
- [6] P. Dalle, Ph. Dejean — Planification en traitement d'image : approche basée sur les données. 11ème congrès RFIA, pp. 75-84, Clermont-Ferrand, janvier 1998.
- [7] B. Deen — A graphical user interface for science data processing. Nasa Science Information Newsletter, vol. 2, numéro 39, pp. 22-27, 1996.
- [8] Ph. Dejean, P. Dalle — Modèle symbolique de la donnée de traitement d'images. 10ème congrès RFIA, pp. 746-755, Rennes, janvier 1996.
- [9] Ph. Dejean, P. Dalle — Un langage de description de concepts pour la formulation d'objectifs d'analyse. 5èmes journées ORASIS, Pôle Vision du GDR-PRC 'CHM', pp. 219-224, Clermont-Ferrand, 20 au 24 mai 1996.
- [10] Ph. Dejean, P. Dalle — Image analysis operators as concept constructors. IEEE Southwest Symposium on Image Analysis and Interpretation, San Antonio, Texas (USA), pp. 66-70, avril 1996.
- [11] V. Ficet — Construction interactive d'un modèle conceptuel d'applications de traitement d'image. Proc. Journées jeunes chercheurs en IA, Nantes, 1996.
- [12] G. Lame — Acquisition de connaissances à partir de textes, vers l'élaboration d'une ontologie du droit. RJCIA, Lyon, septembre 2000.
- [13] S. Moisan, R. Vincent, M. Thonnat, V. Clément, J. van den Elst — Manuel de référence du Logiciel OCAPI Version 2.0. Rapport technique INRIA n°183 (RT-0183), novembre 1995.
- [14] J. Nobécourt — A method to build formal ontologies from texts. EKAW'2000, Juan-les-Pins, 2 octobre 2000.
- [15] A. Nouvel, P. Dalle — Description des entités du traitement d'images pour la conception interactive d'applications. ORASIS 2001, pp. 349-358, Cahors, 5 au 8 juin 2001.
- [16] J. Rasure, D. Arjiro, T. Sauer, C. Williams, S. Wilson — Khoros : a visual language and software development environment for image processing. University of New Mexico, Albuquerque, Nouveau Mexique (USA), 1990.
- [17] R. Ruiloba — Description de la structure temporelle de la vidéo : outils d'évaluation. ORASIS 2001, pp. 27-36, Cahors, 5 au 8 juin 2001.
- [18] G. Schreiber, B. Wielinga, J. Breuker — KADS : a principled approach to knowledge-based system development. Academic Press, Londres, Royaume-Uni, 1993.
- [19] P. Zweigenbaum — Encoder l'information médicale : des terminologies aux systèmes de représentation des connaissances. Innovation Stratégique en Information de Santé, (2-3) 27-47, 1999.