Une stratégie de construction de taxonomies dans les objets

Petko Valtchev, Jérôme Euzenat

Une stratégie de construction de taxonomies dans les objets

Petko Valtchev, Jérôme Euzenat
INRIA Rhône-Alpes
655 av. de l'Europe,
38330 Montbonnot Saint-Martin, France
e-mail : {Petko.Valtchev,Jerome.Euzenat}@inrialpes.fr

Résumé

Construire automatiquement une taxonomie de classes à partir d'objets co-définis et indiférenciables n'est pas une tâche aisée. La partition de l'ensemble d'objets en domaines et la hiérarchisation de ces domaines par la relation de composition permettent de différencier les objets et d'éviter certains cycles impliquant une relation de composition. Par ailleurs, l'utilisation d'une dissimilarité bâtie sur les taxonomies de classes existantes dans certains domaines permet d'éviter de traiter d'autres cycles. Il subsiste cependant des références circulaires qui sont alors circonscrites à une partie bien identifiée des domaines.

Mots-clés Classification automatique, Représentation de connaissances, Dissimilarité.

1 Introduction

Notre travail s'inscrit dans un contexte de construction incrémentale de bases de connaissance à objets (il peut aussi s'appliquer au contexte des bases de données à objets, par exemple). La construction d'une telle base est un processus long, le modèle évoluant au fur et à mesure que la connaissance est acquise sur le domaine.

Le travail présenté ici a pour but d'aider un concepteur de base d'objets à développer une taxonomie de classes à partir d'un ensemble d'objets concrets. Pour cela, un outil de classification hiérarchique est mis à sa disposition. Il va l'utiliser soit en tant que "raccourci" (en confiant au système le soin de retrouver et de caractériser des classes qu'il connaît déjà), soit en tant qu'outil d'analyse exploratoire (en lui demandant de proposer des organisations des données qu'il pourra retenir, raffiner ou oublier).

Afin d'opérer le regroupement d'objets au sein de classes, les objets sont comparés entre eux à l'aide d'une mesure de similarité adaptée. Le problème qui sera plus précisément considéré ici est celui d'élaborer une mesure qui tienne compte des particularités des données structurées sous forme d'objets et tout spécialement la composition (le fait qu'un objet puisse être composé d'autres objets) et la co-définition (le fait qu'un groupe d'objets puisse être fortement connecté par des attributs).

2 Objets et classes

On présente ci-dessous le modèle d'objets considéré et l'organisation qu'il impose en particulier en termes de composition et de co-définition. Une stratégie de construction de taxonomies sera ensuite proposée en s'appuyant sur cette organisation. Elle repose sur une mesure de dissimilarité qui tient compte des divers types de dépendance que les relations entre objets induisent.

2.1 Un modèle primitif à objets

Les individus à considérer dans une représentation par objets sont les *objets* eux-mêmes. Ils sont décrits par un ensemble de valeurs d'attributs qui peuvent être des valeurs numériques, nominales ou ordinales, mais elles peuvent surtout être d'autres objets. Deux paramètres peuvent compliquer l'analyse de ces objets :

– les valeurs peuvent être des collections (un ensemble de chaînes de caractères ou une séquence d'objets) ;

– les objets référencés par un attribut sont eux-mêmes décrits par des attributs et peuvent faire référence, directement ou non, à l'objet initial.

Le second point est plus spécifiquement traité ici.

Dans les formalismes à objets, les objets sont naturellement regroupés dans des *classes*. Les classes sont, en général, considérées comme des moules permettant d'engendrer de nouveaux objets en ce sens qu'elles spécifient les attributs partagés par tous les objets qui y sont rattachés et les contraintes pesant sur ces attributs (le type des valeurs ou leur propre classe de rattachement, par exemple).

Les classes elles-mêmes peuvent être organisées en taxonomies, c'est-à-dire structurées par une relation d'ordre, la *spécialisation* (ou sorte-de), de telle sorte que les objets appartenant à une classe spécifique appartiennent aussi à ses super-classes.

Le but de notre travail consiste à aider à la construction de taxonomies dont les classes rassemblent des objets le plus semblable possible. Les classes ne sont plus alors les moules générateurs des objets mais des catégories permettant d'organiser le domaine. Un tel dispositif a malgré tout un rôle en génie logiciel lorsqu'il s'agit de faire du "reengineering" ou, en modélisation, lorsque l'on veut obtenir une vision plus organisée d'un domaine.

Cependant, élaborer une taxonomie à partir d'ensembles d'objets co-définis et indifférenciés n'est pas une tâche aisée. On présente donc ci-dessous une analyse plus élaborée du modèle objet considéré. Ce modèle reprend des caractéristiques du modèle TROEPS que nous avons conçu et implémenté, cependant les caractéristiques présentées sont applicables à n'importe quel modèle (au moins pour les fins poursuivies ici).

2.2 Caractérisation des relations entre objets

La description donnée ci-dessus est complétée sur deux points concernant l'organisation du modèle objet.

Tout d'abord, les objets sont partitionnés en grandes familles nommées *concepts*. Ces concepts et les objets qui en font partie seront désignés comme tels par le concepteur et le but du système sera de construire une taxonomie différente par concept. Les objets d'un même concept, appelés ses *instances*, partagent une structure commune (mêmes attributs — qui peuvent être sans valeur — et type de valeur homogène pour chaque attribut).

Ensuite, les attributs sont distingués en :

propriété lorsque la valeur de l'attribut n'est pas un objet (par exemple l'âge d'une personne),

composant lorsque la valeur est un ou plusieurs autres objets composants (ou parties) du premier (par exemple le moteur et les roues d'une voiture). Cette relation de composition est transitive, anti-symétrique et anti-réflexive : le graphe des attributs composants est, en général, dirigé et sans cycles (DAG). Les composants n'étant pas partageables dans TROEPS, le graphe est même un arbre.

lien lorsque la valeur est un objet avec lequel elle entretient une relation quelconque qui peut dans le cas général être circulaire (par exemple les liens familiaux). C'est sous cette définition qu'entrent des attributs dont le type est le concept considéré.

Cette distinction entre les trois types d'attributs n'a pas besoin d'être connue au préalable, le système est capable de la déduire.

Um attribut relationnel peut être vu comme une application d'un concept, le domaine, vers un autre concept, le co-domaine, et en ce sens il induit une dépendance entre les deux : le sécond sert à définir le premier. En conséquence, l'ensemble des concepts peut être décrit par un graphe orienté, dit *des dépendances*, dont les sommets sont les concepts et les arcs sont les attributs qui les lient (voir figure 1). Ce graphe est ensuite divisé en niveaux dépendants de la composition qui est supposée être la structure qui prime dans les objets. A chaque niveau, des composantes fortement connexes (CFC) se forment en fonction des liens. Les concepts au sein d'une CFC non triviale sont co-définis (par exemple, les concepts `D` et `E`). Globalement, l'ensemble des CFC liés par les attributs forme un hyper-graphe sans circuit, le DAG des CFC.

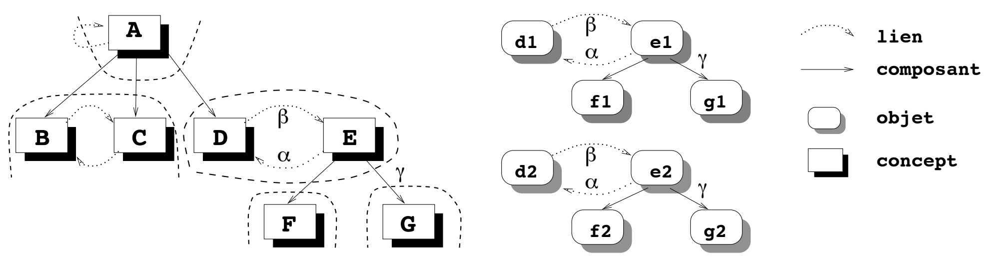

FIG. 1 – ***Exemple d'un graphe des dépendances entre concepts et deux réseaux d'objets :*** *les concepts sont nommés par une lettre en majuscule; les objets instances reprennent le nom du concept respectif en minuscule avec un indice; les attributs relationnels apparaissent comme des arcs; les CFC du graphe sont délimités par des pointillés.*

Les objets sont liés par leurs propres attributs dans des structures de graphes similaires, qu'on appellera *réseaux* (voir figure 1). Les réseaux associés aux instances d'un concept ne peuvent pas dépasser le cadre défini par le graphe des concepts : sur la figure 1, les deux réseaux associés aux objets d_1 et d_2, instances du concept `D`, reflètent la partie du graphe atteignable depuis `D`. En particulier, les couples d'objets d_1, e_1 et d_2, e_2 sont co-définis. Finalement, les classes peuvent également faire référence, moyennant un attribut relationnel, à des classes sur un concept lié. Ainsi, une classe sur `D` de la figure 1 sera décrite en faisant référence à une ou plusieurs classes sur `E`. Dans la suite, les difficultés liées à la construction automatique de classes décrites par d'autres classes sont examinées.

3 Stratégie de construction de classes

Notre but est de pouvoir regrouper, au sein d'une famille, les objets le plus ressemblants dans des classes spécifiques, alors que les objets le plus dissemblants ne se trouveront ensemble que dans une classe très générale. Nous considérons aussi que la description d'une classe est une expression synthétique des ressemblances entre les objets membres : plus les contraintes imposées par les attributs de la classe sont fortes, plus ses membres sont similaires. Les contraintes sur un attribut relationnel étant des classes, leur force est proportionnelle à la spécificité de la classe.

Des méthodes qui permettent de partitionner un ensemble d'individus en groupes tout en maximisant la similarité au sein des groupes et en minimisant celle entre les groupes sont développées en *classification automatique* et en *regroupement conceptuel* [Michalski et Stepp, 1983]. Aux secondes est attribué l'avantage de regrouper les individus en fonction des descriptions intentionnelles que les groupes obtenus admettent. Pour cela, les deux tâches de la construction des classes — la constitution de groupes (regroupement) et l'inférence des descriptions des groupes (caractérisation) — sont accomplies simultanément, ce qui assure leur cohérence.

Dans un contexte où les classes recherchées sont décrites par d'autres classes, une telle démarche nécessite, lorsqu'on traite un concept, de pouvoir assurer l'existence des classes pertinentes sur tous les concepts dont il dépend. Ainsi, la construction de classes sur le concept `E` ci-dessus devrait s'appuyer sur des classes préalablement établies sur `F`, sur `G`, mais aussi sur `D`. Cette condition peut être facilement satisfaite lorsque le graphe des dépendances est un DAG : un parcours post-fixe en profondeur d'abord du graphe permet de ne traiter un concept que lorsque tous les concepts dont il dépend ont déjà été traités (*cf.*[Thompson et Langley, 1991] par exemple). Cette stratégie devient inapplicable dès que le graphe comporte des circuits, c'est-à-dire en cas de co-définition entre concepts (par exemple, les deux concepts `D` et `E`, mais aussi le concept `A`).

La stratégie de construction de classes que nous proposons sépare les étapes de regroupement et de caractérisation. Les classes sont d'abord formées par un algorithme de regroupement classique à partir des dissimilarités entre objets. Ensuite, chaque classe est décrite en intention. Cela permet un traitement simultané des concepts au sein d'une CFC selon le schéma suivant : calcul des dissimilarités, regroupement et constitution des classes, caractérisation. Finalement, les différentes CFC peuvent être traitées dans l'ordre du DAG qu'elles constituent (voir ci-dessus).

La section suivante décrit une mesure de dissimilarité entre objets qui permet de construire des classes compactes sur des concepts co-définis.

4 La dissimilarité sur les objets

4.1 Principes

Afin d'assurer au maximum possible l'adéquation entre les phases de regroupement et caractérisation, nous définissons une mesure de dissimilarité, soit d^p, qui évalue les ressemblances entre deux objets en fonction de la description la plus spécifique que ceux-ci admettent. Pour cela, l'ensemble de leurs valeurs d'attributs sont comparés, moyennant des fonctions d'écart δ^a, selon un principe similaire : on estime que deux valeurs, qu'elles soient simples ou objets, sont d'autant plus proches que la contrainte qu'on peut spécifier sur elles est forte.

Sur un attribut relationnel, cela revient à évaluer la classe commune la plus spécifique au couple d'objets référencés. Par exemple, sur la figure 1, l'apport de l'attribut γ lors du calcul de la dissimilarité entre e_1 et e_2 sera calculé en fonction de la classe la plus spécifique des deux objets référencés, g_1 et g_2 respectivement. Si l'existence d'une telle classe peut être assurée par notre approche, car les concepts `E` et `G` ne font pas partie de la même CFC, cela n'est pas possible pour l'attribut α. Son co-domaine, `D`, étant en co-définition avec `E` via le couple d'attributs α et β, les classes sur les deux concepts ne peuvent être construites que simultanément. Dans ces cas, les objets référencés, ici d_1 et d_2, sont comparés en fonction de leurs valeurs d'attributs. En utilisant pour ce but une mesure de type d^p, cela revient à évaluer leur classe commune la plus spécifique qu'il est *possible* de former.

Avec une telle définition il peut arriver, par un jeux de co-définitions, que les dissimilarités sur un ensemble de couples d'objets dépendent récursivement les unes des autres. Sur l'exemple ci-dessus, les deux couples d_1, d_2 et e_1, e_2 mènent à une telle dépendance. Un exemple de co-définition entre couples d'objets du même concept (`humain` muni d'un attribut `conjoint`) est décrit par la section 5.

Une manière de calculer les valeurs d'une fonction de proximité entre individus qui dépend récursivement d'elle-même peut être tirée des travaux sur le regroupement conceptuel au sein du système KBG [Bisson, 1992]. Le but du système est de construire une hiérarchie de classes d'*entités complexes*, des agrégats composés d'un ensemble d'individus et de relations entre individus. de prédicats. Afin de calculer la similarité entre individus par rapport à leurs propriétés et leurs relations, un système d'équations est posé et résolu.

La nécessité de pouvoir calculer les valeurs de d^o par un système d'équations a déterminé sa forme linéaire.

4.2 Description de la mesure

Soit un concept C, avec un ensemble de n attributs $A = \{a_1, .., a_n\}$. Nous définissons $d^o_C : C \times C \to \mathbb{R}$, la dissimilarité d'un couple d'instances de C comme une combinaison linéaire des écarts δ^a_i sur les attributs a_i. Pour un couple d'objets o et o' (la valeur de a dans o est notée $o.a$),

$$d^o_C(o, o') = \sum_{i=1}^{n} \lambda_i * \delta^a_i(o.a_i, o'.a_i) \tag{1}$$

Ici λ_i sont des poids reflètant l'importance d'un attribut pour le regroupement ($\sum_{i=1}^{n} \lambda_i = 1$). Ils sont calculés en fonction des valeurs fournies par l'utilisateur.

Les écarts δ^a sur les propriétés représentent des fonctions standard pour le type de donnée sous-jacent (soustraction normalisée sur les nombres, par exemple). Sur un attribut relationnel dont le type est un concept C', l'écart δ^a est choisi entre une fonction $d^o_{C'}$ du type décrit ci-dessus et une fonction $\delta^{cl}_{C'}$ qui est calculée sur l'ensemble des classes, sur C' [Valtchev et Euzenat, 1996]. En admettant que chaque écart δ^a_i évalue la compacité de la description la plus spécifique des deux valeurs $o.a_i$ et $o'.a_i$, alors $d^o_C(o, o')$ est une estimation de la compacité de la description la plus spécifique admise par o et o'.

Alors qu'une fonction du type δ^{cl} considère les objets référencés comme entités atomiques, membres de leurs classes, l'utilisation de d^o demande un calcul récursif sur chacun de leurs attributs. Cela revient à une exploration en parallèle des réseaux correspondant aux objets initiaux et si ces réseaux partagent une CFC, alors le calcul revient sur le couple du départ. Dans ce cas, les valeurs des fonctions d^o pour tous les couples d'objets rencontrés sont en dépendent mutuellement. Pour les calculer, un système d'équations est composé à partir des expressions (1). A chaque couple d'objets o_1, o_2 on associe une variable x_i représentant leur dissimilarité, $x_i = d^o_C(o_1, o_2)$. Soit m le nombre total de couples, pour chaque x_i on compose une équation :

$$x_i = b_i + \sum_{j=1, j\neq i}^{m} c_{i,j} * x_j \tag{2}$$

Le membre b_i contient la partie de la dissimilarité selon (1) qui peut être calculée directement, c'est-à-dire la somme pondérée des écarts sur les propriétés, les composants et les liens qui n'appartiennent pas à la CFC commune. Les coefficients $c_{i,j}$ sont des sommes de poids des attributs possédés par le couple d'objets correspondant à x_i et dont les valeurs respectives sont les objets du couple correspondant à x_j (zéro si les deux couples ne sont pas liés). Le système obtenu est de m variables et m équations et possède toujours une solution unique. Celle-ci peut être calculée de manière directe ou itérative.

4.3 Calcul de la dissimilarité

Le calcul de d^o sur un couple d'objets o et o' du concept C nécessite la connaissance de tous les couples d'objets co-définis. Leur ensemble est déterminé en parcourant les réseaux respectifs en parallèle, à la recherche de CFC communes. L'algorithme 1 donne un aperçu des traitements effectués sur un concept C. Tout d'abord, le graphe des dépendances est exploré à partir du concept C, en fixant les CFC de ce graphe. Ensuite, pour chaque objet du concept C, on détecte les CFC dans le réseau associé et on forme le DAG des CFC. Pour tout couple d'objets de C, les deux DAG sont parcourus en parallèle, en post-fixe et en profondeur d'abord. Chaque couple d'objets apparaissant au même rang sur des circuits identiques est retenu. Finalement le système d'équations sur l'ensemble des couples est composé et résolu.

```
décomposition en CFC du graphe des dépendances à partir de C
pour tout objet o de C faire
    décomposition du réseau associé à o en CFC
    construction du DAG des CFC
pour tout couple d'objets o, o′ dans C faire
    pour tout CFC, dans l'ordre des deux DAG, parcours post-fixe en parallèle faire
        parcours des deux CFC en parallèle, détection des couples
        composition et résolution du système d'équations
```

Algorithme 1: Calcul de la dissimilarité sur un concept

5 Exemple

Six objets, instances de `humain`, décrits par trois attributs : `âge`, `salaire` et `conjoint` (voir Table 1) sont à regrouper. L'attribut `âge` est de type entier, avec une étendue `intervalle [20 35]`, `salaire` est de type réel, `intervalle [1.8 3.3]`, il indique le revenu mensuel d'une personne (en Euros). L'attribut `conjoint` est un lien de type `humain`. Les six objets constituent trois couples d'époux : (o_1, o_2), (o_3, o_4) et (o_5, o_6). Les couples forment donc de petits réseaux de deux objets, chaque réseau représentant en soi une CFC.

TAB. 1 – *Objets du concept* `humain`

attribut	o_1	o_2	o_3	o_4	o_5	o_6
âge	26	23	27	30	32	35
salaire	3	2,3	2,2	2,7	2,9	3,2
conjoint	o_2	o_1	o_4	o_3	o_6	o_5

Pour le calcul d^o_h, fixons les poids des attributs à `0,4` pour `conjoint`, `0,3` pour `salaire` et `0,3` pour `âge`. Ainsi, la dissimilarité pour le couple (o_1, o_4), d'après la Formule 1, devient :

$$d^o_h(o_1,\ o_4) = 0,3\ \overline{\delta}^a(26,\ 30) + 0,3\ \overline{\delta}^a(3,\ 2,7) + 0,4\ \overline{\delta}^a(o_2,\ o_3)\ . \tag{3}$$

Aucune structure de classes n'étant disponible sur `humain`, à $\overline{\delta}^a$ sur `conjoint` est substituée la dissimilarité d'objets d^o_h sur $(o_2,\ o_3)$. Celle-ci s'écrit comme suit :

$$d^o_h(o_2,\ o_3) = 0,3\ \overline{\delta}^a(23,\ 27) + 0,3\ \overline{\delta}^a(2,3,\ 2,2) + 0,4\ \overline{\delta}^a(o_1,\ o_4)\ . \tag{4}$$

Avec la même substitution de d^o_h à $\overline{\delta}^a$ sur `conjoint` une dépendance mutuelle entre $d^o_h(o_1,\ o_4)$ et $d^o_h(o_2,\ o_3)$ se constitue. On compose donc un système d'équations où à chacune de ces valeurs une variable est associée, soit x_1 et x_2. Le total des écarts sur les propriétés, étendues et poids compris, est `0,18` pour $(o_1,\ o_4)$ et `0,14` pour $(o_2,\ o_3)$, on a donc $b_1 = 0,18$ et $b_2 = 0,14$:

$$x_1 \ = \ 0,18 \ + \ 0,4x_2 \tag{5}$$

$$x_2 \ = \ 0,14 \ + \ 0,4x_1 \tag{6}$$

Les solutions sont $x_1 \ = \ 0,28$ et $x_2 \ = \ 0,26$. L'ensemble des valeurs de d_h^o est donné dans la Table 2, en dessous de la diagonale principale. En dessus (et en *italique*), une dissimilarité calculée sur les deux propriétés, `salaire` et `âge` pris à poids égaux, notée $\underline{d}_h^o$, est donnée à titre de comparaison. On pourra noter l'égalité des valeurs des deux fonctions sur les couples

TAB. 2 – *Valeurs des dissimilarités d_h^o et $\underline{d}_h^o$ pour les objets de Table 1*

	o_1	o_2	o_3	o_4	o_5	o_6
o_1	0	***0,38***	*0,32*	*0,3*	*0,33*	*0,51*
o_2	**0,38**	0	*0,23*	*0,48*	*0,65*	*0,9*
o_3	0,36	0,26	0	***0,31***	*0,48*	*0,73*
o_4	0,28	0,43	**0,31**	0	*0,17*	*0,42*
o_5	0,5	0,61	0,46	0,33	0	***0,25***
o_6	0,56	0,74	0,57	0,44	**0,25**	0

de conjoints (en gras). En revanche, l'influence mutuelle entre conjoints se traduit par une nette différence entre les ordres sur les couples induits par les valeurs de d_h^o et $\underline{d}_h^o$ (par exemple, le couple o_4,o_5 est de rang 1 selon $\underline{d}_h^o$ et de rang 5 selon d_h^o).

L'effet global de la prise en compte des relations conjugales peut être estimé en comparant les groupes construits par une méthode hiérarchique à partir de d_h^o et $\underline{d}_h^o$ (voir figure 2). Observons,

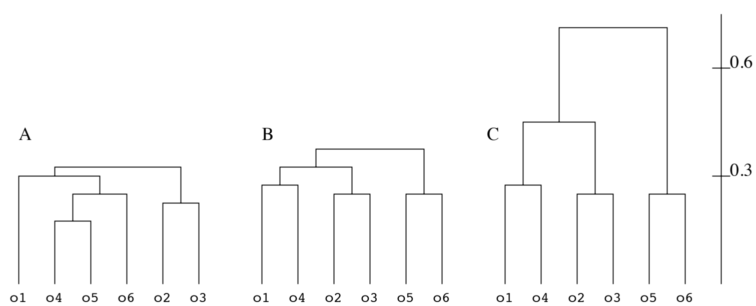

FIG. 2 – *Trois dendrogrammes obtenus par classification hiérarchique sur les objets de l'exemple : A. lien simple avec $\underline{d}_h^o$; B. lien simple avec d_h^o; C. lien complet avec d_h^o.*

par exemple, l'évolution du regroupement des objets o_1 et o_4: dans un groupe avec o_5 et o_6 par rapport à $\underline{d}_h^o$ (fig. 2.A), mais séparés de ceux-ci lorsque d_h^o est utilisée (fig. 2.B et C).

Les groupes retrouvés sont ensuite transformés en classes : par exemple, o_1, o_4 forment la classe `cl#1` et o_2, o_3 la classe `cl#2`. Les quatre objets ensemble forment `cl#3`, la super-classe de `cl#1` et `cl#2` (voir figure 3). Chaque classe est décrite par un ensemble de contraintes sur les attributs synthétisées à partir des descriptions de ses objets membres. Pour un attribut donné, on considère l'ensemble des valeurs dans les objets et on retient la description la plus spécifique que celui-ci admet. Ainsi, pour caractériser `cl#1` selon `conjoint`, on prendra la classe la plus spécifique qui contient les objets référencés par o_1 et o_4, c'est-à-dire o_2 et o_3, donc `cl#2`. Inversement,

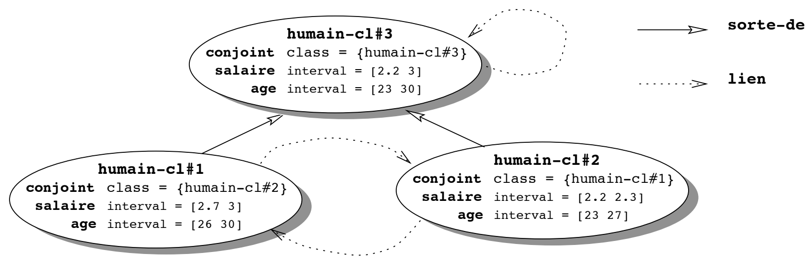

FIG. 3 – *Trois classes obtenues à partir de groupes trouvés à l'aide de* d_h^p.

dans `cl#2` le domaine de `conjoint` sera `cl#1`, ce qui mène à une co-définition. Des classes peuvent être obtenues à partir des groupes selon $\underline{d}_h^p$ en suivant la même procédure. Toutefois, leurs descriptions sont globalement moins compactes sur l'attribut `conjoint`. On en déduit que l'utilisation de la mesure complète permet non seulement de trouver des regroupements nouveaux, mais aussi de construire des classes dont les descriptions sont compactes sur les relations.

6 Conclusion

Une stratégie de construction de classes sur un ensemble de familles d'objets liés par des attributs relationnels a été esquissée. Elle sépare la construction des classes en deux étapes : regroupement des objets selon une mesure de dissimilarité, puis caractérisation des groupes d'objets obtenus. La mesure définie s'applique uniformément sur les propriétés et les relations entre objets, permettant ainsi de construire des classes qui sont compactes sur tous leurs attributs. Le calcul de la dissimilarité sur des couples d'objets co-définis est fait à l'aide d'un système d'équations. Les classes sur les familles d'objets co-définies sont construites simultanément.

Le travail présenté ici vient en complément et reste compatible avec d'autres travaux sur la prise en compte des collections et de multiples taxonomies sur un même concept.

Références

[Bisson, 1992] G. Bisson. Conceptual clustering in a first order logic representation. Dans *Proceedings of the 10th European Conference on Artificial Intelligence, Vienna, Austria*, pages 458–462, 1992.

[Michalski et Stepp, 1983] R. Michalski et R. Stepp. *Machine learning: an Artificial Intelligence approach*, volume I, chapitre Learning from observation: conceptual clustering, pages 331–363. Tioga publishing company, Palo Alto (CA US), 1983.

[Thompson et Langley, 1991] K. Thompson et P. Langley. *Knowledge and experience in unsupervised learning*, chapitre Concept formation in structured domains, pages 127–161. Morgan Kaufman, San Mateo (CA US), 1991.

[Valtchev et Euzenat, 1996] P. Valtchev et J. Euzenat. Classification of concepts through products of concepts and abstract data types. Dans *Ordinal and symbolic data analysis*, rédacteurs Y. Lechevallier E. Diday et Dr O. Opitz, pages 3–12, Heildelberg (DE), 1996. Springer Verlag.