

CAMELIS - MANUEL UTILISATEUR

VÉRONIQUE ABILY , SÉBASTIEN FERRÉ



I R I S A

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES
Campus de Beaulieu – 35042 Rennes Cedex – France
Tél. : (33) 02 99 84 71 00 – Fax : (33) 02 99 84 71 71
<http://www.irisa.fr>

Camelis - Manuel utilisateur

Véronique Abily , Sébastien Ferré

Systèmes symboliques
Projet LIS

Publication interne n ° 1234 — Novembre 2007 — 57 pages

Abstract: Camelis is a logic information system implementation (LIS) which permits to structure and search data on which tags are associated. The data navigation is no longer under a classic hierarchical but by combining requests. The aim of this document is to introduce the Camelis' graphical interface, to show how to give properties to the data, the way to build the requests to perform Research. To illustrate this document, a concrete example is described in the last part where the different processes to sort, search and edit a pictures' collection are explained.

(Résumé : tsvp)



Centre National de la Recherche Scientifique
(UMR 6074) Université de Rennes 1 – Insa de Rennes



Institut National de Recherche en Informatique
et en Automatique – unité de recherche de Rennes

Camelis

Résumé : Camelis est une implémentation de système d'information logique (LIS) permettant de structurer et de rechercher des données auxquelles sont associées des tags. La navigation au sein de ces données ne se fait plus de manière classique hiérarchique mais en combinant des formules logiques sous forme de requêtes. Le but de ce document est de présenter l'interface graphique de Camelis, comment donner des propriétés aux données, la manière de construire les requêtes pour effectuer des recherches. Pour illustrer ce manuel, un exemple concret est décrit en dernière partie où les différentes manipulations pour classer, rechercher et retoucher une collection de photos sont expliquées.

Mots clés : objets,tags,propriétés,logique,navigation,contexte,requête

1 Introduction

Ce logiciel, développé par Sébastien Ferré au sein de l'équipe LIS de l'Irisa, est une implémentation de système d'information logique (LIS). Il permet de structurer et de rechercher des données organisées sous forme d'ensembles d'objets auxquels sont associés des *tags*, encore appelés *propriétés*. La navigation au sein de ces données ne se fait plus de manière classique hiérarchique mais en combinant des *formules logiques* sur les propriétés des objets, sous forme de *requêtes*. Un ensemble d'objets, auquel est associé un ensemble de tags et de formules logiques, s'appelle un *contexte*. Le but de ce manuel est de présenter l'interface utilisateur de l'application Camelis. Nous allons tout d'abord voir comment démarrer l'application, puis nous présenterons globalement l'interface graphique et la navigation au sein de celle-ci, la manière dont sont définis les propriétés et les différents éléments d'un contexte. Ensuite, nous détaillerons les différents menus ainsi que les menus contextuels des fenêtres de l'application. Nous terminerons par des exemples de manipulations courantes sur une collection de photos.

2 Démarrer l'application

Cette application, de nom glis.exe, peut être lancée automatiquement depuis les systèmes d'exploitation Linux ou Windows en double-cliquant sur son icône. L'application, étant basée sur le système multi-fenêtrage X11, elle démarre également, sous Windows, l'émulateur X cygwin afin de permettre son exécution. Elle peut également être lancée en mode commande depuis une fenêtre de commandes X11, afin de préciser certains paramètres. Sa syntaxe est la suivante :

```
glis.exe [-readonly] [<fichier>.lis | <fichier>.ctx ... <fichier>.ctx ]
```

Un ou plusieurs noms de contextes peuvent être précisés, ils serviront de contexte initial sinon celui-ci est vide. Un fichier contexte peut être au format binaire (*<fichier>.lis*) ou au format texte (*<fichier>.ctx*). Au format binaire, il est chargé plus rapidement mais est dépendant de la version de camelis. Au format texte, il contient une liste de commandes camelis qui sont recompilées au chargement du fichier. Il permet d'importer ou d'exporter des éléments d'un contexte à un autre ou d'une version de camelis à une autre. Plusieurs fichiers contextes de type texte peuvent être précisés, ils sont alors chargés dans l'ordre spécifié. Camelis démarre en lecture seule si :

- l'option **-readonly** est présente
- on n'a pas d'accès en écriture sur le contexte courant
- le contexte courant est déjà ouvert en écriture (présence d'un *<fichier>.lck*)

3 Présentation générale

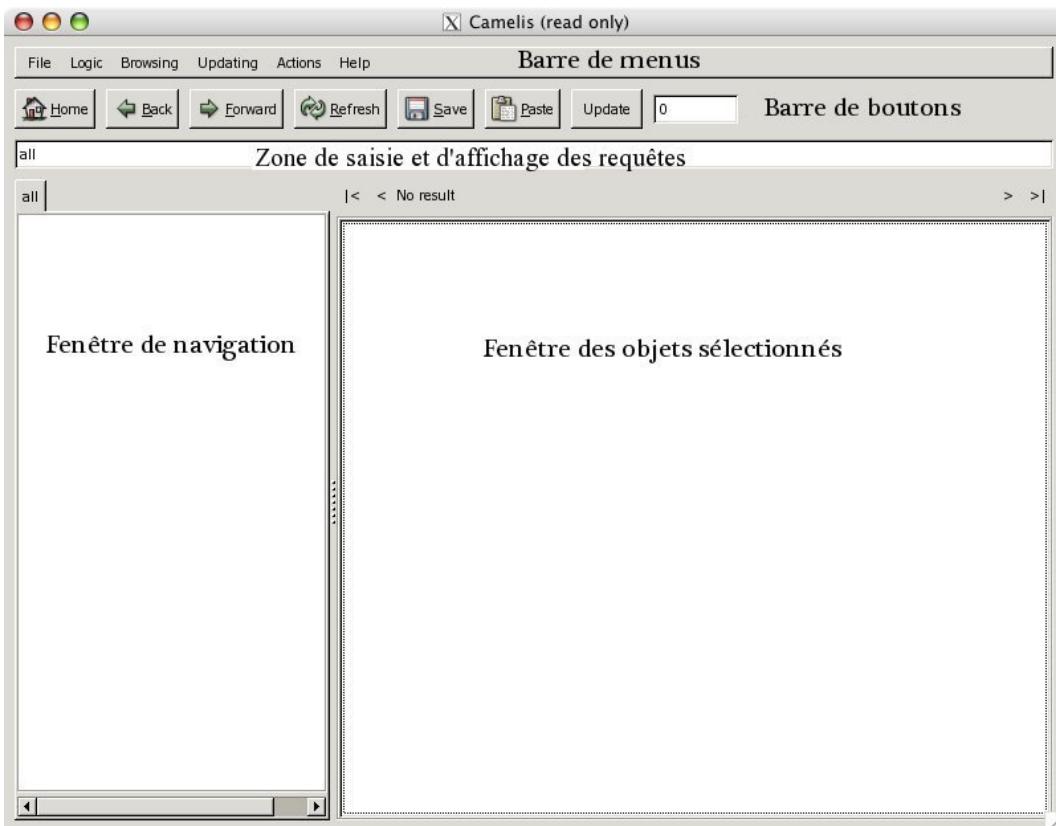


FIG. 1 – Interface graphique de l'application Camelis

L'interface de l'application (cf. figure 1) comprend différentes parties : une barre de menus, une barre de boutons, une zone de saisie et d'affichage des requêtes, une fenêtre de navigation permettant de sélectionner les tags ou propriétés associés aux objets, une fenêtre affichant les objets sélectionnés par la requête courante. Les différentes parties de l'interface sont mises à jour à partir de la requête courante (cf. figure 2).

La zone des requêtes affiche la requête courante et permet de la modifier ou d'en saisir une nouvelle. Une requête permet de sélectionner des objets du contexte. Elle correspond à une simple propriété ou à une combinaison de propriétés. La requête `all` sélectionne l'ensemble des objets du contexte. Une requête se construit de la manière suivante :

`Nouvelle requête = [not] propriété`

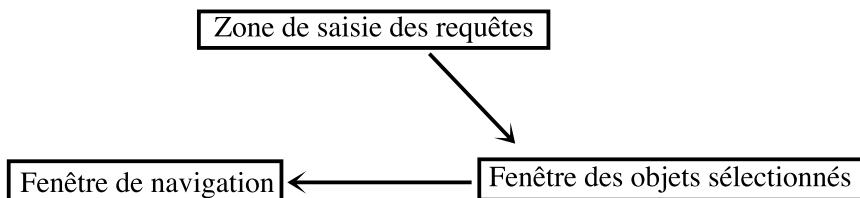


FIG. 2 – Ordre de mise à jour des différentes fenêtres de l'interface

Nouvelle requête = requête courante (and | except | or) (propriété (or propriété)*)

A chaque requête, l'ensemble des objets sélectionnés est mis à jour ainsi qu'un historique des requêtes.

La fenêtre des objets sélectionnés affiche, une fois la requête effectuée, les objets la satisfaisant. Elle est donc mise à jour à chaque requête, le nombre d'objets variant d'une requête à l'autre. Les objets sont affichés page par page. Au-dessus de la fenêtre, s'affichent le nombre d'objets dans la page courante, le nombre total d'objets sélectionnés, des flèches pour passer d'une page à l'autre.

La fenêtre de navigation se met, ensuite, à jour, en affichant les propriétés associées aux objets sélectionnés. Les propriétés communes, à tous ces objets, sont de couleur orange ou rouge si elles appartiennent à la requête courante tandis que les propriétés différentes sont de couleur noire. Plusieurs arborescences de propriétés sont visibles. En effet, les propriétés associées aux objets peuvent être classées dans un certain ordre sous forme d'arbres. Chaque arbre montre les relations entre les propriétés le composant, partant d'une propriété générale à sa racine vers des propriétés plus spécifiques. Un objet possédant une propriété spécifique, possèdera, également, par implication, la propriété plus générale. Dans la figure 3, par exemple, la propriété **Bretagne** dépend de la propriété **France** qui dépend elle-même de la propriété **Europe**. Si un objet possède la propriété **Bretagne**, il est automatiquement associé aux propriétés **France** et **Europe** qui sont plus générales.

Une même propriété peut apparaître plus d'une fois dans les différents arbres si elle dépend de plusieurs propriétés plus globales. Le nombre situé à gauche d'une propriété représente le nombre d'objets sélectionnés la vérifiant. Dans la figure 3, par exemple, 16 objets, de la fenêtre des objets sélectionnés, sont associés à la propriété **Bretagne**.

La barre de boutons, quant à elle, permet d'effectuer certaines actions rapidement :

- **Home** : remise à « all » de la requête courante,
- **Back** : revenir en arrière dans l'historique des requêtes,
- **Forward** : revenir en avant dans l'historique des requêtes,
- **Refresh** : réafficher les fenêtres de navigation et des objets sélectionnés,
- **Save** : sauvegarde du contexte courant (fichier .lis et fichier .ctx associé),

```

16 ▼ date in ...
11   ▷ date = 2006
3    ▷ date = 2003
2    ▷ date = 2005
16 ▼ Europe
16   ▼ France
16     ▼ Bretagne
12       ▷ Finistere
3        ▷ Cote_armor
1        ▷ Ille_et_vilaine
16 ▷ source ?
10 ▷ Mer
8  ▷ Monument
5  ▷ Bateau
2  ▷ Fleur
2  ▷ Port

```

FIG. 3 – Fenêtre de navigation

- **Paste** : associe les objets qui ont été copiés aux propriétés sélectionnées (cf. menu **Update > Paste**, page 29),
- **Update** : met à jour le contexte courant (cf. menu **File > Import**, page 16). Le nombre de mise à jour à effectuer s'affiche à droite du bouton.

4 La navigation

La navigation au sein des fenêtres de navigation et des objets sélectionnés permet, également, de mettre à jour la zone de saisie des requêtes. Il existe, donc, plusieurs manières de construire des requêtes pour effectuer des recherches.

L'interrogation par requête s'effectue en éditant directement la requête dans la zone de saisie des requêtes.

Par exemple, dans la figure 4, si l'on saisit la requête **Bretagne except Finistere**, les objets correspondants s'affichent dans la fenêtre des objets. Ce qui correspond, ici, à des photos prises en Bretagne mais pas dans le Finistère. Il y a 4 objets sélectionnés qui sont tous affichés dans la page courante. Ces objets sont tous associés à la propriété **Bretagne** qui est donc de couleur orange (et également à **France** et à **Europe**), 3 d'entre-eux sont associés à la propriété **Cote_armor** et 1 seul à la propriété **Ille_et_vilaine**.

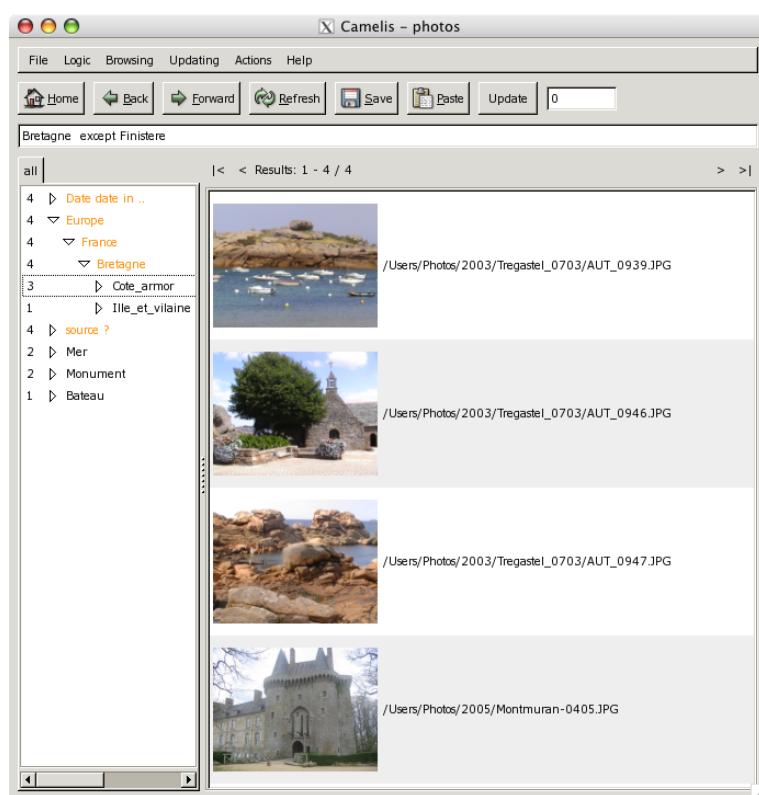


FIG. 4 – Construction d'une requête

L'interrogation par sélection s'effectue en sélectionnant des propriétés dans la fenêtre de navigation.

Les propriétés qui diffèrent selon les objets (de couleur noire) peuvent être sélectionnées pour affiner la recherche, c'est la *navigation descendante*. Par défaut, lorsque l'on double-clique sur une propriété de ce type, celle-ci est rajoutée à la requête courante de manière à ce que les objets sélectionnés la vérifient en plus des propriétés déjà vérifiées. Cela revient à faire un "et logique" ("and") avec la requête courante.

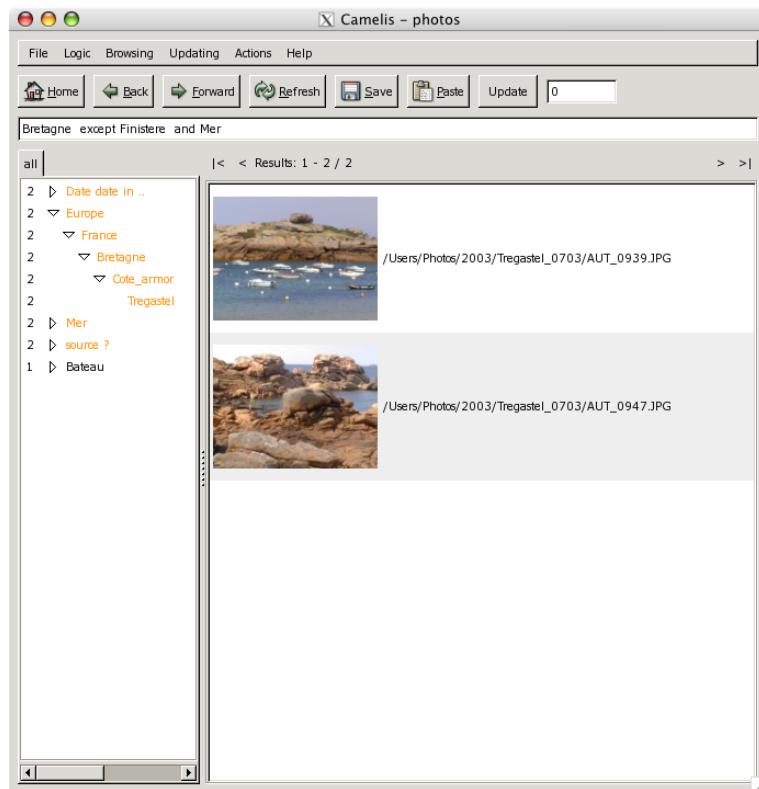


FIG. 5 – Modification de la requête à partir de la fenêtre de navigation

À partir de notre exemple précédent (figure 4), on ne veut garder, parmi les 4 photos déjà sélectionnées, que celles prises au bord de la mer. Pour cela on double-clique sur la propriété **Mer**. Une nouvelle requête est construite automatiquement **Bretagne except Finistere and Mer** mettant à jour les fenêtres de navigation et des objets sélectionnés (cf. figure 5). On constate, qu'à partir de cette nouvelle requête, on peut affiner à nouveau la recherche

en double-cliquant sur la propriété **Bateau** pour sélectionner la photo prise en Bretagne au bord de la mer avec des bateaux mais pas dans le Finistère.

On peut, également, effectuer une *navigation ascendante* en sélectionnant les propriétés communes aux objets (de couleur orange). Si l'on double-clique sur l'une de ces propriétés, elle est supprimée de la requête courante si elle en faisait partie. Elles peuvent être utilisées pour alléger la requête courante, si celle-ci est devenue trop complexe, par exemple. On peut, également, élargir une requête en remplaçant une propriété, faisant partie de la requête courante, par une propriété plus générale appartenant au même arbre de propriétés. Il suffit de double-cliquer sur la propriété générale.

Par exemple, si l'on veut élargir la recherche de notre exemple précédent, à toute l'Europe, il suffit de double-cliquer sur la propriété **Europe** qui remplacera la propriété **Bretagne**, plus spécifique, dans la requête courante.

D'autres exemples de mise à jour des requêtes à partir de la fenêtre de navigation sont détaillés dans la section 9.3.1 page 46.

L'interrogation par l'exemple s'effectue à partir de la fenêtre des objets sélectionnés. En double-cliquant sur un objet, la requête courante est remplacée par la conjonction de toutes les propriétés de l'objet . On sélectionne, alors, tous les objets partageant les propriétés de l'objet sélectionné.

5 Les propriétés

Les propriétés sont définies par différents types d'attributs. Un attribut peut être de type identificateur ou de type terme. Les relations entre les propriétés ne peuvent être définies qu'entre des attributs de type terme. Un identificateur commence par une minuscule. Un terme commence par une majuscule, un souligné ou correspond à une chaîne entre 'quotes'.

Une valeur peut être donnée à un attribut selon la syntaxe suivante :

`(terme | identificateur)+ <mot clé> valeur`

Les mots clés dépendent du type de la valeur qui peut être : une chaîne de caractères, un entier, un réel, une date, une heure ou des droits sur un fichier.

5.1 Chaîne de caractères

Un attribut représentant une chaîne de caractères est défini de la manière suivante :

`(terme | identificateur)+ is "chaîne de caractères"`

L'attribut est exactement la chaîne précisée.

`(terme | identificateur)+ contains "chaîne de caractères"`

L'attribut est une chaîne contenant la sous-chaîne précisée.

`(terme | identificateur)+ beginswith "chaîne de caractères"`

L'attribut est une chaîne commençant par la chaîne précisée.

`(terme | identificateur)+ endswith "chaîne de caractères"`

L'attribut est une chaîne finissant par la chaîne précisée.

`(terme | identificateur)+ match "expression régulière de type unix"`

L'attribut est une chaîne qui correspond à l'expression régulière définie.

Les différents mots clés peuvent être simplifiés en utilisant uniquement leur première lettre.

Exemples de valeurs pour une chaîne :

`c "ab"` représente une chaîne contenant la sous-chaîne "ab"

`beginswith "Il était"` représente une chaîne commençant par "Il était"

`match "[+-] ?[0-9]+"` représente une chaîne exprimant un nombre pouvant être signé

5.2 Entier

Un attribut représentant un nombre entier est défini de la manière suivante :

`terme | identificateur = entier`

`terme | identificateur in intervalle`

On peut donner une approximation du nombre en utilisant un tiret à la place d'un chiffre ou une lettre représentant une unité (k pour kilos, M pour Megas, ...).

Un intervalle est représenté par `[entier] .. [entier]`, les bornes de l'intervalle ne sont pas obligatoires (on peut ainsi exprimer à partir de ou jusqu'à).

Exemples de valeurs pour un entier :

`= 2--` représente un entier compris entre 200 et 299

`= 29-k` représente un entier compris entre 290k et 299k

`in 2000 ..` représente un entier supérieur ou égal à 2000

5.3 Date

Un attribut représentant une date est défini de la manière suivante :

`(terme | identificateur)* date = date`

`(terme | identificateur)* date in intervalle`

Une date peut être absolue ou relative. Elle est, dans ce dernier cas, définie par une expression représentant une période de temps au niveau de la journée, du mois ou de l'année.

```

date = dateabs | daterel
dateabs = jour mois année | mois année | année
jour = 1..31
mois = jan | fev | mar | apr | may | jun | jul | aug | sep | oct | nov | dec
année = nombre représentant l'année
daterel = tomorrow | today | next durée | last durée | n durée ago
durée = day | month | year

```

Un intervalle est représenté par [date] .. [date] , les bornes de l'intervalle ne sont pas obligatoires.

Exemples de valeurs pour une date :

```

date = may 2007 représente une date en mai 2007
date = today représente la date d'aujourd'hui
date in 2000 .. apr 2006 représente une date comprise entre janvier 2000 et avril 2006
date in 2 days ago .. représente « depuis 2 jours »

```

5.4 Heure

Un attribut représentant une heure est défini de la manière suivante :

```
(terme | identificateur)* time = heure
(terme | identificateur)* time in intervalle
```

Une heure peut être absolue ou relative. Elle est, dans ce cas, définie par une expression représentant une période horaire.

```

heure = heureabs | heurerel
heureabs = hh:mm:ss | hh:mm | hh
heurerel = now | next durée | last durée | n durée ago
durée = hour | minute | second

```

Un intervalle est représenté par [heure] .. [heure] , les bornes de l'intervalle ne sont pas obligatoires.

Exemples de valeurs pour une heure :

```

time = 1 hour ago représente « il y a une heure »
time in 1 hour ago .. now représente « dans la dernière heure »
time in .. 13:59 représente « jusqu'à 13 heures 59 minutes »

```

5.5 Droits sur un fichier

Un attribut représentant des droits sur un fichier est défini de la manière suivante :

```
terme | identificateur perm = permission
terme | identificateur perm in intervalle
```

Une permission correspond aux droits d'accès sur un fichier définis sous Unix :

<droits pour le propriétaire> <droits pour le groupe> <droits pour les autres>

Un intervalle est représenté par [permission] .. [permission], les bornes de l'intervalle ne sont pas obligatoires.

Exemple de valeurs correspondant à des droits sur des fichiers :

rwx rx r le propriétaire du fichier peut lire, écrire ou exécuter, le groupe peut lire ou exécuter, les autres ne peuvent que lire

r r r .. au moins tous les droits en lecture

.. rwx r r le groupe et les autres peuvent au plus lire

5.6 Classification des valeurs

Lorsque des valeurs sont définies pour une propriété, qu'elle soit de type terme ou identificateur, ces valeurs sont classées automatiquement les unes par rapport aux autres dans l'arbre de navigation.

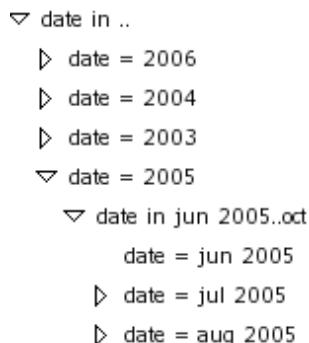


FIG. 6 – Classification des valeurs d'une propriété correspondant à une date

La figure 6 nous montre un exemple de classification de la propriété « Date » correspondant à la date de prise de vue des photos des exemples précédents. Elle est classée par année, puis par intervalle de temps au sein d'une année, puis par mois.

6 Les éléments du contexte

Le contexte représente l'ensemble des objets et l'ensemble de formules logiques associé. Les différents éléments d'un contexte sont répartis en 7 catégories : les sources, les données extrinsèques, les axiomes, les tags, les exports, les règles de mises à jour, les actions.

6.1 Les sources

Ce sont les fichiers et répertoires associés aux objets. Ils permettent d'importer des objets à partir du système de fichiers. Les sources génèrent automatiquement des propriétés sur les objets importés, encore appelées données intrinsèques. Les propriétés créées dépendent du type de source reconnu.

6.1.1 Répertoires

Les répertoires ne produisent pas d'objet en eux-mêmes, mais permettent un parcours récursif sur une arborescence de fichiers lors de la création de sources.

6.1.2 Fichiers et URL

Un objet est produit par fichier ou URL importé. Les propriétés associées automatiquement aux objets sont : le nom, l'extension, le nom de machine (URL), le nom du répertoire dans lequel se trouve le fichier, la taille, la date de dernière modification, etc...

6.1.3 Fichiers BibTEX

Un fichier de type BibTeX (*.bib) produit un objet possédant les propriétés par défaut des objets de type fichier. Si l'option `extract file parts...` a été sélectionnée lors de l'importation du fichier (cf. menu `File > Import > Import file...` page 18), un objet est créé pour chaque référence bibliographique et pour chaque objet un descripteur par champ est défini : référence, auteurs, titre, journal, année de publication.

6.1.4 Fichiers mp3

Un fichier de type mp3 (*.mp3) produit un objet possédant les propriétés par défaut des objets de type fichier. Des propriétés supplémentaires (tags ID3) lui sont également associées : descripteurs, nom de l'artiste, titre.

6.1.5 Fichiers jpg

Un fichier de type jpg (*.jpg, *.jpeg) produit un objet possédant les propriétés par défaut des objets de type fichier. Une petite image est associée à l'objet dans la fenêtre des objets.

6.1.6 Signets Mozilla

Un fichier de type signet Mozilla (bookmarks.html) produit un objet possédant les propriétés par défaut des objets de type fichier. Si l'option `extract file parts...` a été sélectionnée lors de l'importation du fichier, un objet est créé pour chaque URL référencée dans le fichier.

6.1.7 Répertoires Mozilla

Un fichier de type répertoire Mozilla (courrier électronique : *.msf) produit un objet possédant les propriétés par défaut des objets de type fichier. Si l'option `extract file parts...` a été sélectionnée lors de l'importation du fichier, un objet est créé pour chaque email auquel sont associées les propriétés suivantes : sujet, expéditeur, date, destinataires.

6.1.8 Fichiers CSV

Pour les fichiers de type CSV (*.csv), un objet est créé pour chacune des lignes du fichier sauf pour la première qui est utilisée pour générer les descripteurs associés (un par nom de colonne, le séparateur étant la virgule) avec l'option `extract file parts...`.

6.1.9 Fichiers interface ocaml

Un fichier de type interface ocaml (*.mli) produit un objet possédant les propriétés par défaut des objets de type fichier. Si l'option `extract file parts...` a été sélectionnée lors de l'importation du fichier, un objet est créé pour chaque fichier *.mli auquel sont associées les propriétés suivantes : nom, module, type.

6.1.10 Enregistrements DBLP

Un enregistrement DBLP (URL commençant par <http://dbpl.uni-trier.de/rec/bibtex>) produit un fichier BibTeX.

6.1.11 Résultats de recherche DBLP

Le résultat de recherche DBLP (URL commençant par <http://www.informatik.uni-trier.de/>) produit un ensemble d'enregistrements DBLP.

6.2 Les données extrinsèques

Il est également possible de créer des objets qui sont uniquement sous le contrôle de Camelis et indépendants du système de fichiers. Les données extrinsèques représentent les objets et les propriétés définies par l'utilisateur. La manière de les définir est décrite dans la section 9.1.2, page 39.

6.3 Les axiomes

Les axiomes définissent les relations entre les propriétés comme vu précédemment. Ils ne peuvent être établis qu'entre propriétés de type terme. On ne pourra pas créer de relation entre des propriétés définies par des identificateurs.

6.4 Les tags

Les tags correspondent aux propriétés visibles dans la fenêtre de navigation.

6.5 Les exports

À partir d'un ensemble d'objets sélectionnés, il est possible de constituer des fichiers exports listant ces objets. Ces fichiers pourront être réutilisés ultérieurement dans d'autres applications. Par exemple, on peut construire une liste de fichiers sons que l'on pourra utiliser comme entrée dans un lecteur multimédia.

6.6 Les règles de mise à jour

Des règles de mise à jour peuvent être effectuées à partir de requêtes. Elles permettent d'associer des requêtes à des propriétés ou à d'autres requêtes. Ainsi, lorsqu'un objet est créé ou supprimé de l'ensemble des objets satisfaisant la requête, l'ensemble des objets dépendant des propriétés ou des requêtes associées est automatiquement mis à jour par la règle.

Par exemple, sachant qu'un port est toujours au bord de la mer, on peut associer la propriété **Mer** à la requête sélectionnant les objets ayant la propriété **Port**, par une règle de mise à jour. Tout nouvel objet satisfaisant la requête **Port** sera également associé à la propriété **Mer**. De même, tout objet dissocié de la propriété **Port** sera automatiquement dissocié de la propriété **Mer**.

6.7 Les actions

Des actions peuvent être définies sur les objets. Elles permettent l'utilisation d'une application extérieure sur ces derniers. On peut, par exemple, avoir envie de lancer un diaporama à partir d'un ensemble d'objets sélectionnés. Pour cela, on définit une action qui va permettre de lancer la commande système associée.

7 Les différents menus

La barre de menus de l'interface est composée de six menus (cf. figure 1 page 4). Nous allons présenter en détail les cinq menus principaux.

7.1 Le menu « File »

Ce menu (cf. figure 7) correspond aux diverses actions d'entrée/sortie avec le système de fichiers (chargement ou sauvegarde d'un contexte, importation/exportation d'objets depuis/vers le système, ...) . Il permet également d'avoir des statistiques sur le contexte courant.



FIG. 7 – Menu File

7.1.1 Chargement/Sauvegarde de contextes

New... Création d'un nouveau contexte. Une boîte de dialogue apparaît permettant de sélectionner le répertoire et le nom du fichier .lis .

Open... Chargement d'un contexte existant qui remplace le contexte courant. Une boîte de dialogue apparaît permettant de sélectionner le répertoire et le nom du fichier .lis .

Save Sauvegarde du contexte courant (fichier .lis courant et fichier .ctx associé).

Save as... Sauvegarde du contexte courant. (fichier .lis et fichier .ctx associé) Une boîte de dialogue apparaît permettant de sélectionner le répertoire et le nom du fichier .lis .

Close Fermeture du contexte courant.

7.1.2 Importation/Exportations d'objets

Import Le sous-menu "Import" (cf. figure 8) permet de mettre à jour le contexte et de gérer les sources du contexte.

Import context... Importe le contexte à partir d'un fichier contexte au format texte (<fichier>.ctx). Ici, le contexte importé est rajouté au contexte courant. On peut n'importer qu'un sous-ensemble du contexte. Une fenêtre (cf. figure 9) s'affiche pour indiquer les parties du contexte à importer :

- **sources** : sources associées aux objets (avec leurs propriétés associées),

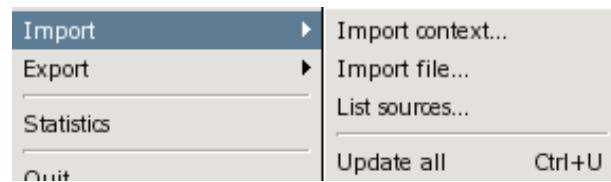


FIG. 8 – Sous-menu File > Import



FIG. 9 – Fenêtre de dialogue pour l'importation/exportation de contexte

- **axiomes** : relations entre les propriétés,
- **extrinsic data** : données définies par l'utilisateur,
- **features** : tags visibles dans la fenêtre de navigation,
- **wells** : les fichiers exports associés,
- **rules** : les règles de mise à jour,
- **actions** : les actions définies sur les objets,
- Un champ de saisie permet de donner le nom du fichier contexte (*.ctx).

Import File Permet de créer des objets à partir de fichiers ou de répertoires du système de fichiers. Une boîte de dialogue apparaît (cf. figure 10).

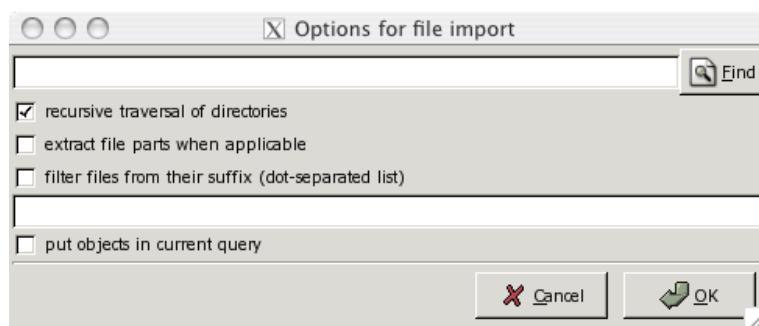


FIG. 10 – Fenêtre de dialogue pour l'importation de fichiers

La première zone permet de saisir ou de sélectionner le répertoire ou le fichier à importer. Cela peut être un nom de fichier ou de répertoire local ou un nom d'URL. Différentes options peuvent également être appliquées :

- **"recursive traversal of directories"** : parcourir récursivement les répertoires.
- **"extract files parts when applicable"** : extraire les différentes parties du fichier en tant qu'objets lorsque cela est possible.
- **"filter files from their suffix"** : filtrer les fichiers à importer à partir de leur suffixe. On peut saisir différents suffixes séparés par le caractère point '.'.
- **"put objects in current query"** : associer les objets importés à la requête courante.

Le nombre de mise à jour à effectuer dans le contexte s'affiche au fur et à mesure dans la case à droite du bouton "update" de la barre de boutons. Les différents fichiers importés sont enregistrés comme "sources" dans le contexte courant. Des propriétés sont automatiquement associées aux objets créés en fonction du type de source importé (cf. section 6.1 "les éléments du contexte" page 13).

List sources... Ouvre une fenêtre permettant la gestion des sources importées. L'arborescence système des sources est affichée (cf. figure 11). Une source est de couleur rouge

si elle n'existe plus dans le système de fichier, elle est de couleur verte sinon. Les répertoires sont de couleur noire.

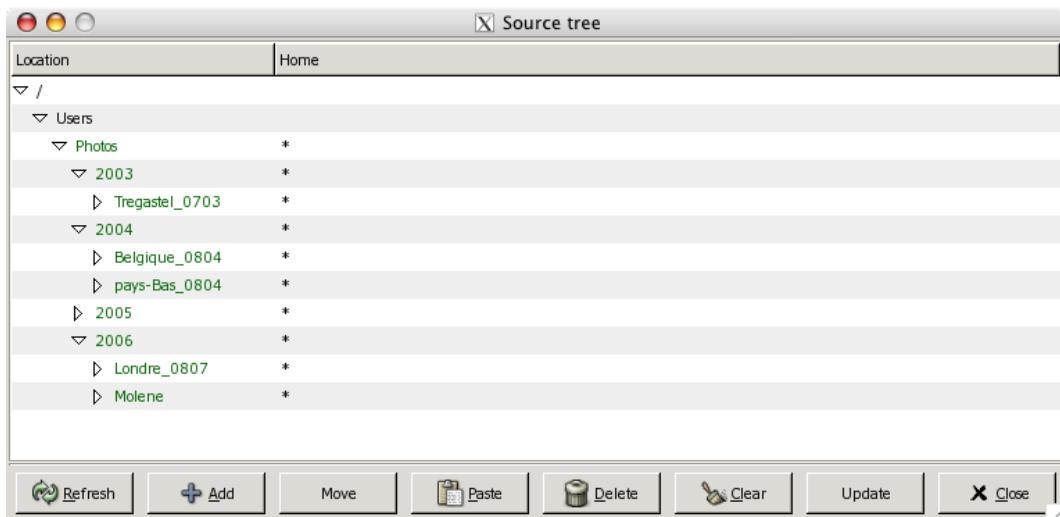


FIG. 11 – Fenêtre de gestion des sources importées

Des boutons permettent d'effectuer un certain nombre d'actions :

- “Refresh” : réaffiche l’arborescence.
- “Add” : rajoute une nouvelle source (équivalent au menu **File > Import > Import file...**, section 7.1.2 page 18).
- “Move” : modifie le chemin de la source sélectionnée dans le système de fichiers,
- “Paste” : associe les propriétés copiées aux sources sélectionnées (équivalent au menu **Logic > Paste**, section 7.4 page 29).
- “Delete” : supprime la source sélectionnée ainsi que les objets qui lui sont associés.
- “Clear” : supprime toutes les sources (et les objets associés) n’ayant plus de chemin valide dans le système de fichiers (celles de couleur rouge).
- “Update” : met à jour le contexte en prenant en compte les modifications des fichiers sources.
- “Close” : ferme la fenêtre.

Update all Met à jour le contexte.

Export Le sous-menu “Export” (cf. figure 12) permet d’exporter les éléments du contexte et des listes d’objets vers le système de fichiers pour pouvoir les réutiliser, ultérieurement,

dans d'autres contextes ou dans des applications externes.



FIG. 12 – Sous-menu File > Export

Export context... Exporte une partie du contexte courant au format texte afin qu'il puisse être réutilisé. La fenêtre (cf. figure 9) de dialogue pour l'exportation de contexte s'affiche pour préciser les parties du contexte à exporter :

- **sources** : sources associées aux objets sélectionnés,
- **axiomes** : relations entre les propriétés,
- **extrinsic data** : données définies par l'utilisateur concernant les objets sélectionnés,
- **features** : tags visibles dans la fenêtre de navigation,
- **wells** : les fichiers exports associés,
- **rules** : les règles de mise à jour,
- **actions** : les actions définies sur les objets.
- Un champ de saisie permet de donner le nom du fichier contexte (*.ctx).

Export file... Exporte l'ensemble des objets sélectionnés dans un fichier. Plusieurs types de formats sont possibles :

- *.gqv : construit une liste de fichiers images pour visionner un diaporama avec le logiciel GQView (Linux, MacOS X).
- *.txt : construit une liste de fichiers images pour visionner un diaporama avec le logiciel Irfranview (Windows).
- *.m3u : construit une liste de fichiers sons pour utilisation sur un lecteur multimédia.
- *.bib : construit un nouveau fichier BibTeX à partir de l'ensemble des objets sélectionnés, issus de sources BibTeX.

Attention, les objets sélectionnés doivent être du bon type (fichiers images, fichiers sons ou fichiers BibTeX). Une boîte de dialogue (cf. figure 13) s'affiche permettant de donner le nom du fichier.

List wells... Ouvre une fenêtre (cf. figure 14) permettant la gestion des fichiers exportés.



FIG. 13 – Fenêtre de dialogue pour l'exportation de fichiers

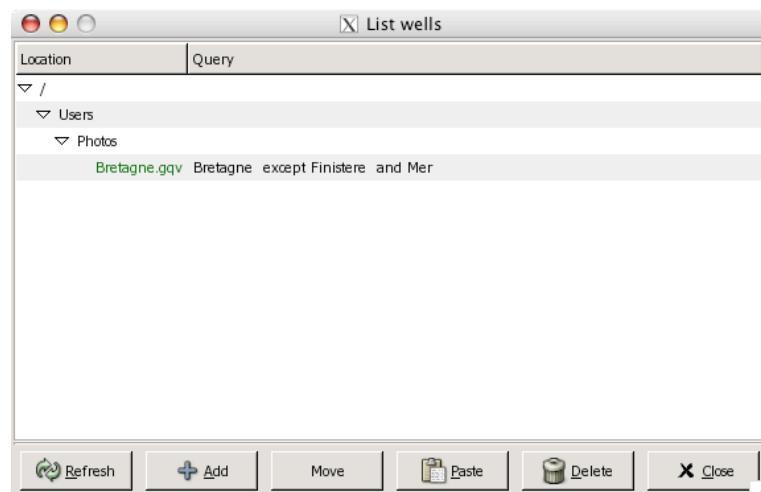


FIG. 14 – Fenêtre de gestion des exports

L’arborescence système des exports est affichée. Pour chaque export, sont précisés son positionnement dans le système de fichiers ainsi que la requête correspondant à la liste des objets sélectionnés pour l’export. Le nom du fichier export est de couleur rouge s’il n’existe plus dans le système de fichier, il est de couleur verte sinon. Les répertoires sont de couleur noire.

En bas de la fenêtre, des boutons permettent d’effectuer un certain nombre d’actions :

- “Refresh” : réaffiche l’arborescence.
- “Add” : rajoute un nouveau export (équivalent au menu **File > Export > Export file...**, section 7.1.2 page 20).
- “Move” : modifie le chemin de l’export sélectionné dans le système de fichiers.
- “Paste” : remplace la requête associée aux exports sélectionnés par la requête courante.
- “Delete” : supprime les exports sélectionnés ainsi qu’éventuellement les fichiers associés.
- “Close” : ferme la fenêtre.

7.1.3 Statistiques et quitter l’application

Statistics Affiche des informations concernant le contexte courant :

- le nombre d’objets,
- le nombre de propriétés,
- la taille du contexte : le nombre d’associations objet-propriété,
- le nombre moyen de propriétés par objet.

La figure 15 nous indique, par exemple, que le contexte courant contient 33 objets, 241 tags (ou propriétés) et 1809 associations objet-propriété. Il y a en moyenne 54 tags par objet.

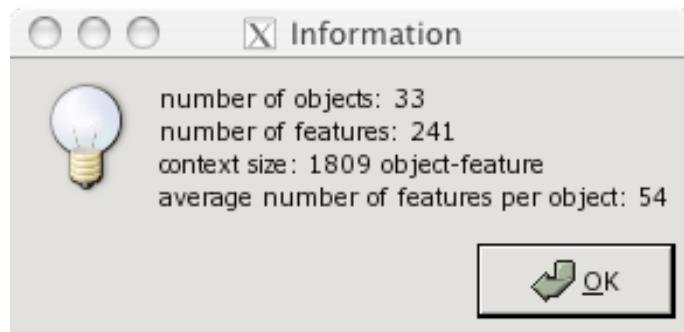


FIG. 15 – Statistiques du contexte courant

Quit Quitte l'application.

7.2 Le menu « Logic »

Ce menu (cf. figure 16) est composé de deux parties, une première concernant la gestion des propriétés dans la fenêtre de navigation et la deuxième concernant la gestion des règles de mise à jour à partir des requêtes.



FIG. 16 – Menu Logic

7.2.1 Gestion des propriétés

Add feature... Ajoute une nouvelle propriété dans la fenêtre de navigation. Cette propriété n'est visible dans la fenêtre de navigation que si au moins un objet lui est associé. Si cette propriété avait été supprimée auparavant, elle retrouve ses positions dans la fenêtre de navigation. Une propriété, une fois créée, n'est jamais réellement supprimée. Elle est juste rendue visible ou non dans la fenêtre de navigation.

Join as... Définit des axiomes en incluant les propriétés sélectionnées dans une propriété plus générale. Une boîte de dialogue apparaît permettant de saisir le nom de la propriété générale. Les propriétés doivent être définies par des termes.

Copy features Copie les propriétés sélectionnées.

Paste Définit des axiomes en incluant les propriétés copiées dans les propriétés sélectionnées. Les propriétés doivent être définies par des termes.

Delete features Supprime les propriétés sélectionnées de la fenêtre de navigation.

7.2.2 Gestion des règles

Copy query Copie la requête courante. On pourra par la suite l'associer à des propriétés pour créer une règle de type `paste <conjonction des propriétés> <requête courante>`.

Delete query Crée la règle `delete <requête courante>` supprimant les objets satisfaisants la requête. Tout nouvel objet satisfaisant cette requête sera automatiquement supprimé.

List rules... Ouvre une fenêtre listant les règles de mises à jour effectuées à partir de requêtes. Une règle est de la forme :

```
<kind><update><query>
<kind> = paste | delete
<update> = propriété(s) associée à tout objet satisfaisant la requête (si <kind> = paste)
<query> = requête correspondante
```

Une règle est de type `paste` si les objets satisfaisant la requête ont été associés à une(des) propriété(s). Une requête est de type `delete` si la suppression d'objets satisfaisants la requête a été effectuée.

La figure 17 nous montre un exemple de listes de règles de mise à jour.

La règle `paste Mer Port` associe la propriété `Mer` à tous les objets satisfaisant la requête `Port`, qui correspond aux objets ayant la propriété `Port`. Ainsi, tout nouvel objet satisfaisant cette requête sera automatiquement associé à la propriété `Mer`.

La règle `delete Mer and Phare` supprime tous les objets satisfaisant la requête `Mer and Phare`, correspondant aux objets ayant la propriété `Mer` et la propriété `Phare`.

En bas de la fenêtre trois boutons permettent d'agir sur cette liste :

- “**Refresh**” : réaffiche la liste des règles de mises à jour.
- “**Delete**” : supprime les règles sélectionnées de la liste, seule la liste est modifiée, il n'y a pas d'effet rétroactif sur les objets concernés.
- “**Close**” : ferme la fenêtre.

Attention, les actions concernent uniquement les règles, et non pas les objets.

7.3 Le menu « Browsing »

Ce menu (cf. figure 18) permet de naviguer dans l'historique des requêtes, de sélectionner les actions de mise à jour de la requête courante ou encore de gérer l'affichage des fenêtres



FIG. 17 – Fenêtre de gestion des règles de mise à jour

de navigation et des objets.

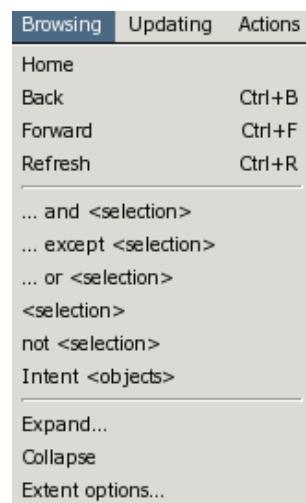


FIG. 18 – Menu Browsing

7.3.1 Navigation dans l'historique des requêtes

Home Réinitialise la zone de requête (remise à all). L'ensemble des objets du contexte courant est, alors, sélectionné.

Back Revient en arrière dans l'historique des requêtes.

Forward Revient en avant dans l'historique des requêtes.

Refresh Réaffiche les fenêtres de navigation et des objets sélectionnés.

7.3.2 Mise à jour de la zone des requêtes

... and <selection> Restreint l'ensemble des objets déjà sélectionnés à ceux vérifiant au moins une des propriétés sélectionnées. Si une seule propriété est sélectionnée, un double-clic sur celle-ci est équivalent.

Nouvelle requête = requête courante and (propriété(s) sélectionnée(s) séparées par 'or')

... except <selection> Restreint l'ensemble des objets déjà sélectionnés à ceux ne vérifiant pas les propriétés sélectionnées.

Nouvelle requête = requête courante except (propriété(s) sélectionnée(s))

... or <selection> Rajoute à l'ensemble des objets déjà sélectionnés, les objets vérifiant les propriétés sélectionnées.

Nouvelle requête = requête courante or (propriété(s) sélectionnée(s))

<selection> Reconstruit l'ensemble des objets à partir des objets vérifiant les propriétés sélectionnées.

Nouvelle requête = propriété(s) sélectionnée(s)

not <selection> Reconstruit l'ensemble des objets à partir des objets ne vérifiant pas les propriétés sélectionnées.

Nouvelle requête = not (propriété(s) sélectionnée(s))

Intent <objects> Sélectionne tous les objets partageant les mêmes propriétés que l(es) objets(s) sélectionné(s). La fenêtre de navigation et la requête courante sont mises à jour. La nouvelle requête est un "et logique" (and) entre toutes les propriétés concernées. Quand un seul objet est sélectionné, un double-clic sur cet objet permet également cette sélection.

7.3.3 Gestion de l'affichage des fenêtre

Expand... Affiche une fenêtre (cf. figure 19) permettant le paramétrage de l'affichage des tags dans la fenêtre de navigation :

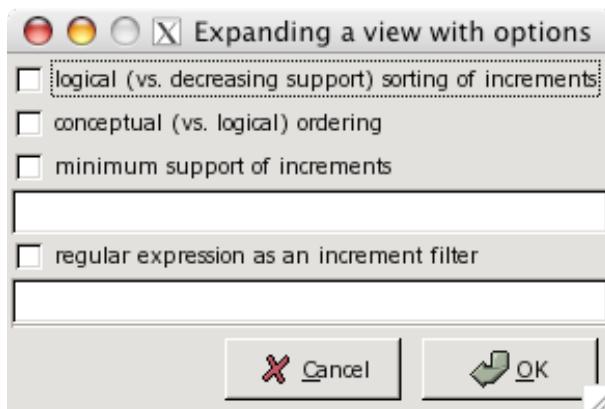


FIG. 19 – Fenêtre de gestion de l'affichage de la fenêtre de navigation

- **logical (vs. decreasing support) sorting of increments** = non sélectionné par défaut. Affiche les propriétés dans l'ordre logique (alphabétique pour les chaînes, chronologique pour les dates, ...). Par défaut, l'ordre d'affichage des propriétés dépend du nombre d'objets associés aux propriétés (ordre décroissant).
- **conceptual (vs. logical) ordering** = non sélectionné par défaut. Affiche les propriétés par rapport aux ensembles d'objets associés plutôt que par rapport à la logique. Si les objets associés à une propriété font partie d'un ensemble d'objets associés à une autre propriété, seule la propriété correspondant à l'ensemble global sera affiché. Par exemple, la propriété `source dir is "Tregastel_0703"` représente le répertoire de stockage des photos prises à Trégastel en juillet 2003. Ce répertoire est un sous répertoire du répertoire `photos` dans le système de fichiers. Les objets associés à cette propriété font également partie de l'ensemble, plus global, des objets associés à la propriété `source dir is "photos"`. Si l'option contextuelle est sélectionnée, seule cette dernière propriété sera affichée. Cela permet de retrouver la structure hiérarchique quand elle existe.
- **minimal support of increments** : Spécifie le nombre minimal d'objets satisfaisant la propriété pour qu'elle soit affichée. Si cette valeur est à 0, aucun objet n'est nécessaire. La valeur par défaut de ce paramètre est calculée automatiquement.
- **regular expression as an increment filter** = non sélectionné par défaut. Restreint la recherche aux propriétés correspondant à l'expression régulière spécifiée.

Ce paramétrage s'applique également aux sous-propriétés (propriétés plus spécifiques) dépendant des propriétés sélectionnées.

Collapse Referme la sous arborescence des propriétés sélectionnées dans la fenêtre de navigation.

Extent options... Affiche une fenêtre (cf. figure 20) paramétrant l'affichage de la fenêtre des objets sélectionnés :



FIG. 20 – Fenêtre de gestion de l'affichage de la fenêtre des objets

- **local objects only** = non sélectionné par défaut. Seuls les objets qui n'ont pas plus de propriétés que celles spécifiées sont affichés.
- **Page start** = 1 par défaut. Numéro du premier objet affiché.
- **Page size** = 10 par défaut. Nombre d'objets affichés par page.

7.4 Le menu « Updating »

Ce menu (cf. figure 21) permet d'effectuer des manipulations sur les objets du contexte. Les objets sélectionnés, par défaut, sont tous ceux satisfaisant la requête courante et non pas, seulement, ceux visibles dans la page courante de la fenêtre des objets. On peut aussi sélectionner des objets à l'aide de la souris à partir de la page courante.

Add object Ajoute un objet. L'objet est associé aux propriétés sélectionnées.

Copy objects Copie les objets sélectionnés. Si aucun objet n'est sélectionné, l'ensemble des objets satisfaisant la requête courante est copié.

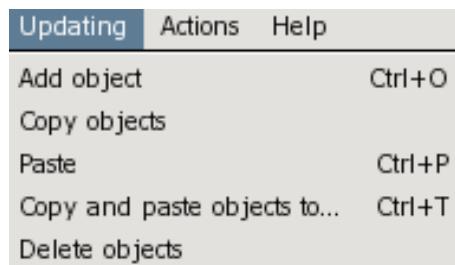


FIG. 21 – Menu Update

Paste Associe les objets copiés aux propriétés sélectionnées.

Copy and paste objects to... Copie les objets sélectionnées et les associe à une(des) propriété(s). Si aucun objet n'est sélectionné, les propriétés seront associées à l'ensemble des objets de la fenêtre des objets. Une boîte de dialogue s'affiche pour saisir le nom des propriétés. On peut combiner plusieurs propriétés à l'aide de **and**. Si une propriété n'existe pas, elle est créée. Si l'on veut supprimer une relation objet-propriété, il faudra passer par l'action **Paste not** du menu contextuel (cf. section 8.1.3, page 35) de la fenêtre de navigation car il n'existe pas d'action spécifique dans le menu **Updating**.

Delete objects Supprime les objets sélectionnés. Les objets sont détruits ainsi que toutes les associations objets-propriétés. Attention, si aucun objet n'est sélectionné, tous les objets satisfaisant la requête courante (objets de la fenêtre des objets sélectionnés) sont détruits.

7.5 Le menu « Actions »

Le menu action (cf. figure 22) permet d'associer aux objets des applications externes pouvant être appelées depuis le contexte courant.

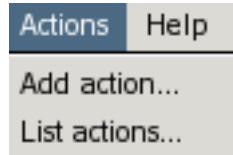


FIG. 22 – Menu Actions

7.5.1 Add action

Associe à un ensemble d'objets une action permettant l'utilisation d'une application externe sur ces objets.

Une boîte de dialogue (cf. figure 23) permettant de définir l'action apparaît :



FIG. 23 – Fenêtre de définition d'une action

- “**Name of the action**” : nom définissant l'action du point de vue utilisateur, qui apparaîtra dans le menu contextuel associés aux objets. Un paramètre de type MIME peut être utilisé pour définir l'action.
`<Nom de l'action> [${<Paramètre MIME>}]`
Dans l'exemple de la figure 23, “*Editer image \${image}*” définit une action de nom “*Editer image*”, en précisant qu'un nom de fichier de type image sera passé en paramètre.
- “**Command to be executed**” : commande qui sera appliquée quand l'action sera sélectionnée.
`<Commande> ["${<Paramètre MIME>}[:win]"]`
`<Commande> = nom de la commande à exécuter`
`<Paramètre MIME> = paramètre de type MIME éventuel`
Le paramètre de type MIME est obligatoire s'il est défini dans le nom de l'action. Lors de l'exécution de la commande, la valeur du paramètre associée à l'objet sera passée en argument, :win permet de convertir le nom du fichier si la commande utilisée est une commande Windows (par défaut, le nom cygwin est utilisé).
Dans l'exemple de la figure 23, “*/Applications/Gimp.app/Contents/MacOS/Gimp "\${image}"*” indique que le logiciel gimp sera lancé avec le nom du fichier image en paramètre.

- **”The query defining the scope of the action”** : requête définissant le domaine d’application sur lequel s’applique l’action.

Dans l’exemple de la figure 23, “**source ext is “jpg”**” précise que les objets concernés par l’action sont ceux pour lesquels l’extension du fichier source est jpg.

L’action définie dans l’exemple de la figure 23 s’applique à un objet donné. On peut, également, définir des actions dont la portée est un ensemble d’objets. Au moment de l’exécution de l’action, un fichier export, comprenant la liste des objets sélectionnés, est créé temporairement. L’action s’applique, alors, sur ce fichier export. Pour cela, on précise au niveau du paramètre de l’action, qu’il s’agit d’une liste, le type des objets concernés et le type d’export à créer. Par exemple, la figure 24 nous montre la définition d’une action permettant de lancer un diaporama via le logiciel gqview à partir d’un ensemble d’objets. Le paramètre MIME de cette action est “\$(list/image/gqv)”.

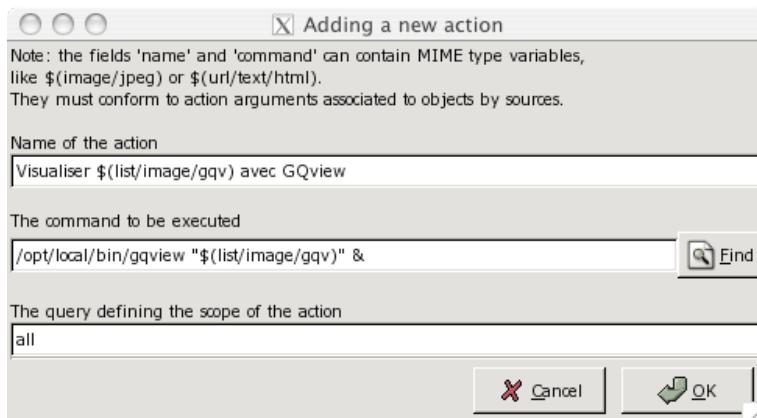


FIG. 24 – Fenêtre de définition d’une action applicable sur un ensemble d’objets

7.5.2 List actions...

Ouvre une fenêtre (cf. figure 25) affichant la liste des actions définies.

Chaque action est de la forme :

<Nom de l’action><Commande><<Requête>

En bas de la fenêtre, cinq boutons permettent d’agir sur cette liste :

- **”Refresh”** : réaffiche la liste des actions.

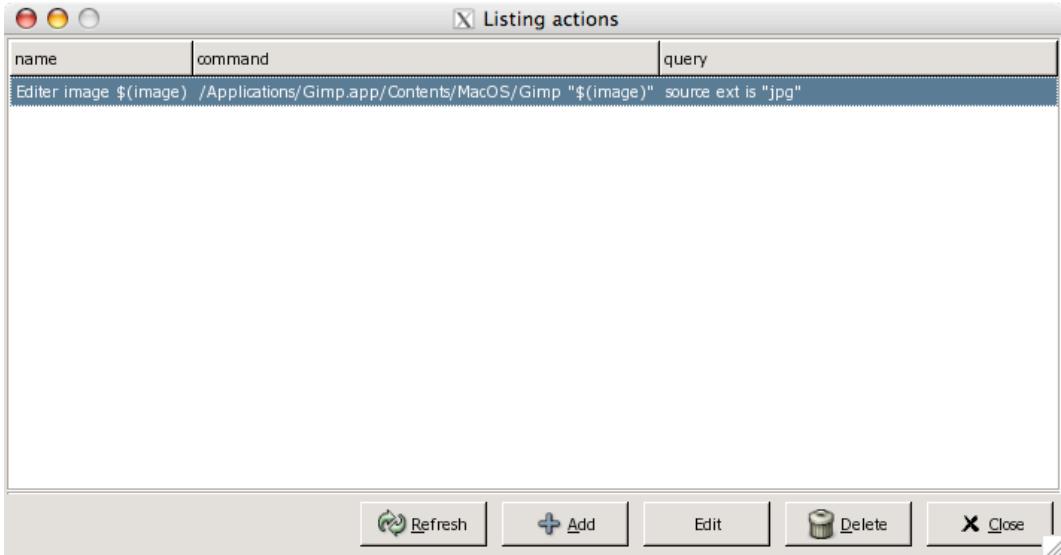


FIG. 25 – Fenêtre de gestion des actions

- “**Add**” : ajoute une action, la boîte de dialogue permettant de définir l’action apparaît (cf. figure 23). Si une action de la liste est sélectionnée, les différents champs sont initialisés avec les valeurs correspondant à celle-ci.
- “**Edit**” : modifie l’action sélectionnée, la boîte de dialogue de définition de l’action s’affiche.
- “**Delete**” : supprime les actions sélectionnées de la liste.
- “**Close**” : ferme la fenêtre.

8 Les menus contextuels

Des menus contextuels sont également accessibles depuis les fenêtres de navigation, la fenêtre des objets sélectionnés et la zone des requêtes. On accède à ces menus par clic droit depuis la zone concernée. Les différentes actions possibles sont, pour la plupart, des raccourcis vers les menus définis dans la section précédente.

8.1 Fenêtre de navigation

Le menu contextuel de la fenêtre de navigation (cf. fig. 26) est composé de quatre parties permettant la mise à jour de la zone des requête à partir des propriétés de la fenêtre de navigation, la gestion de l'affichage et des propriétés de la fenêtre de navigation.

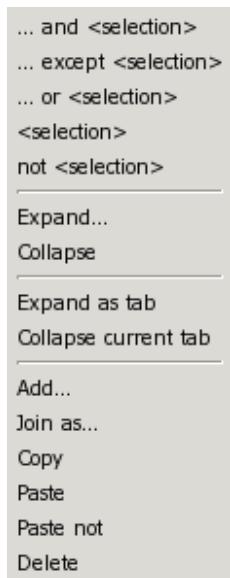


FIG. 26 – Menu contextuel de la fenêtre de navigation

8.1.1 Mise à jour de la zone de requête

Les actions suivantes sont équivalentes à une partie des actions du menu **Browsing** (cf. section 7.3 page 24).

... and <selection> Nouvelle requête = requête courante **and** propriété(s) sélectionnée(s) séparées par **or**

... except <selection> Nouvelle requête = requête courante **except** propriété(s) sélectionnée(s)

... or <selection> Nouvelle requête = requête courante **or** propriété(s) sélectionnée(s)

<selection> Nouvelle requête = propriété(s) sélectionnée(s)

not <selection> Nouvelle requête = **not** propriété(s) sélectionnée(s)

8.1.2 Gestion de l'affichage de la fenêtre

Les deux premières actions sont équivalentes à celles concernant la gestion de l'affichage de la fenêtre de navigation du menu **Browsing** (cf. section 7.3 page 24), les deux autres sont spécifiques au menu contextuel.

Expand ... Une fenêtre (cf. figure 19) s'affiche permettant le paramétrage de l'affichage des propriétés sélectionnées.

Collapse Referme la sous arborescence des propriétés sélectionnées.

Expand as tab Crée un onglet contenant le sous arbre de la propriété sélectionnée. Si plusieurs propriétés sont sélectionnées, un onglet est créé pour chacune d'elles. Cela permet ainsi de créer des raccourcis de navigation.

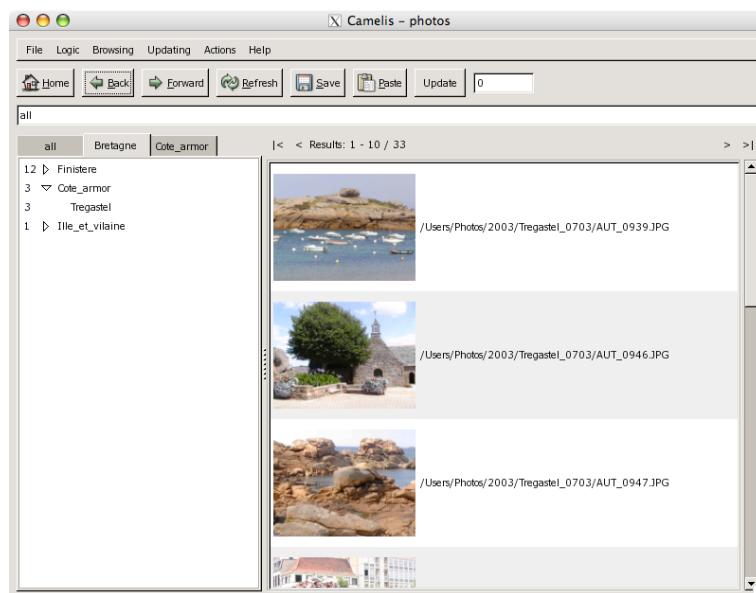


FIG. 27 – Exemple d'onglets dans la fenêtre de navigation

Dans la figure 27, par exemple, les onglets **Bretagne** et **Cote_armor** sont des raccourcis vers les propriétés correspondantes, permettant ainsi un accès direct à celles-ci dans l'arbre de navigation.

Collapse current tab Supprime l'onglet courant.

8.1.3 Gestion des propriétés

Les actions de gestion des propriétés du menu contextuel sont équivalentes aux actions du menu **Logic** (cf. section 7.2.1 page 23). L'action permettant de supprimer une relation entre une propriété et un objet est, cependant, spécifique au menu contextuel et n'a pas d'équivalent dans les différents menus.

Add... Ajoute une nouvelle propriété.

Join as... Inclut les propriétés sélectionnées dans une propriété plus générale.

Copy Copie les propriétés sélectionnées.

Paste Inclut les propriétés copiées dans les propriétés sélectionnées, ou, associe les objets copiés aux propriétés sélectionnées.

Paste not Dissocie les objets copiés des propriétés sélectionnées.

Delete Supprime les propriétés sélectionnées de la fenêtre de navigation.

8.2 Fenêtre des objets sélectionnés

Le menu contextuel de la fenêtre des objets sélectionnés (cf. figure 28) permet d'appliquer des actions ou d'effectuer des manipulations sur les objets, et de gérer le paramétrage de la fenêtre.



FIG. 28 – Menu contextuel de la fenêtre des objets sélectionnés

8.2.1 Appliquer des actions

Les actions définies via le menu **Actions** (cf. 7.5.1 page 30) peuvent être appliquées sur les objets sélectionnés par le menu contextuel.

<Nom action> Action définie par l'utilisateur pouvant être appliquée sur les objets sélectionnés. Les actions sont réparties en deux parties.

La première partie du menu contient la liste des actions s'appliquant à un seul objet. Si un ensemble d'objets est sélectionné le nombre d'objets sur lesquels s'applique l'action apparaît entre parenthèse, l'action sera alors exécutée sur chacun de ces objets. L'application sera, alors, lancée plusieurs fois. Une boîte de dialogue demandant une confirmation à l'utilisateur apparaît. Dans la figure 28, l'action "**Editer image**" , définie dans l'exemple de la figure 23, permet d'ouvrir le logiciel gimp sur les objets de type image.

La seconde partie du menu contient la liste des actions s'appliquant à une liste d'objets regroupés dans un export temporaire. Dans la figure 28, l'action "**Visualiser Diaporama avec GQview**", définie dans l'exemple de la figure 24, permet d'ouvrir le logiciel gqview sur le fichier export temporaire de type gqv créé à partir de l'ensemble des objets sélectionnés. L'action ne sera exécutée qu'une seule fois.

List action arguments... Donne pour l'objet sélectionné les paramètres pouvant être utilisés pour définir une action sur celui-ci :

- **Mime type** : type MIME du paramètre, ce type pourra être utilisé dans la définition du nom ou de la commande associée à une action,
- **Role** : rôle du paramètre (fichier, liste de fichiers, ...),
- **Value** : valeur du paramètre (nom du fichier, de l'URL, ...), cette valeur sera passée en paramètre lors de l'exécution d'une commande associée à une action.

La figure 29 nous donne, par exemple, la liste des arguments pour un fichier image de type jpg.

8.2.2 Manipuler les objets

Les manipulations sur les objets sont équivalentes à celles pouvant être effectuées par le menu **Updating** (cf. section 7.4 page 28).

Add Ajoute un objet.

Copy Copie les objets sélectionnés. Si aucun objet n'est sélectionné, l'ensemble des objets de la fenêtre des objets est copié.

Paste Associe les objets copiés aux propriétés sélectionnées.



FIG. 29 – Fenêtre donnant les paramètres d'un objet utilisables dans une action

Copy an paste to... Copie les objets sélectionnées et les associe à une(des) propriété(s). Si aucun objet n'est sélectionné, tous les objets de la fenêtre des objets sont associés aux propriétés.

Delete Supprime les objets sélectionnés.

Intent Sélectionne tous les objets partageant les mêmes propriétés que l(es) objets(s) sélectionné(s). Cette action est équivalente à l'action "Intent" du menu **Browsing** section 7.3.

8.2.3 gestion de l'affichage de la fenêtre

La gestion de l'affichage de la fenêtre peut également s'effectuer par le menu contextuel de la même manière que par le menu **Browsing** (cf. section 7.3.3 page 28).

Options ... Paramétre l'affichage de la fenêtre.

8.3 Zone de requête

Les actions du menu contextuel de la zone des requêtes (cf. figure 30) sont équivalentes aux actions concernant les règles de mise à jour dans le menu **Logic** (cf. section 7.2.2 page 24).

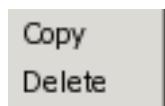


FIG. 30 – Menu contextuel de la zone des requêtes

Copy Copie la requête courante.

Delete Crée la règle `delete <requête courante>` supprimant les objets satisfaisants la requête.

9 Exemples de manipulations courantes

Nous allons, dans cette section, illustrer ce manuel à partir d'un exemple concret. Le but est de classer, de rechercher et, éventuellement, de retoucher des photos prises au cours de voyages de 2003 à 2006. Initialement, le répertoire `/Users/Photos` a été créé afin d'y stocker les différents fichiers jpg au fur et à mesure des prises de vue. A ce stade, les photos ne sont pas classées, le seul classement étant un sous-répertoire par voyage. Nous utiliserons Camelis pour les tagger et les classer.

9.1 Créer des objets

Un objet peut être créé par importation à partir du système de fichiers ou directement à partir de Camelis. Les deux méthodes sont décrites ici, sachant toutefois que seule la première méthode a été utilisée pour créer les objets liés à nos photos.

9.1.1 Objets liés au système de fichiers

Pour créer des objets associés à nos photos, nous allons importer les fichiers sources. Pour cela (cf. figure 31), nous utiliserons le menu `File > Import > Import file...` qui nous ouvre la boîte de dialogue pour l'importation de fichiers. On précise le nom du répertoire (`/Users/Photos`) dans lequel se trouvent nos photos, la manière de le parcourir (récursevement car nos photos sont réparties dans différents sous-répertoires), le suffixe des fichiers (on n'importe que les fichiers jpg). Pour chaque fichier jpg trouvé, Camelis crée un objet. Les données intrinsèques des objets sont positionnées automatiquement et apparaissent sous forme de propriétés dans la fenêtre de navigation. Les objets importés étant de type fichier, les propriétés `source` sont définies pour chaque fichier. Elles représentent : ses permissions Unix, le nom du répertoire dans lequel il se trouve, son extension, le nom de son groupe Unix, la date du dernier accès au fichier, sa dernière date de modification, la date de modification de son statut, le nom du fichier source, le nom absolu du fichier dans l'arborescence

Unix, sa taille, le nom de son propriétaire. De plus, les objets étant de type image, une miniature est affichée dans la fenêtre des objets sélectionnés pour chacun d'entre-eux.

9.1.2 Objets purement Camelis

Nous pouvons, également, créer des objets purement Camelis pour lesquels seules des données définies par l'utilisateur lui sont associées. Pour cela, nous utilisons le menu **Updating > Add object**. Des propriétés initiales peuvent lui être associées si elles ont été sélectionnées au préalable. La manière de définir des propriétés est décrite dans la section suivante.

9.2 Créer des propriétés, les associer aux objets et les classifier

Maintenant que nos objets sont créés, il va falloir les classifier en créant des propriétés (ou tags) caractérisant, par exemple, un lieu géographique, la date de prise de vue, le type d'objets se trouvant sur la photo, un événement particulier, ...

Classiquement, on sélectionne des objets auxquels seront associées de nouvelles propriétés. On peut, toutefois, créer une propriété sans qu'elle soit initialement liée à un objet, l'association propriété-objet pouvant se faire par la suite.

9.2.1 Créeer une propriété initialement associée à un ou plusieurs objets

En parcourant nos photos, nous trouvons plusieurs photos de châteaux que nous voudrions tagger pour pouvoir les retrouver facilement. Pour cela (cf. figure 32), nous sélectionnons le sous-ensemble d'objets représentant des châteaux dans la fenêtre des objets puis, à l'aide de l'action **Copy and paste objetc to...** du menu contextuel de la fenêtre des objets ou à partir du menu **Updating > Copy and paste objetc to...**, nous les associons à une nouvelle propriété. Dans la fenêtre de dialogue qui s'affiche, nous rentrons le nom de la propriété, ici **Chateau**. Celle-ci apparaît alors dans la fenêtre de navigation. Le nombre d'objets qui lui sont associés correspond aux nombres d'objets que nous avions sélectionnés.

9.2.2 Créeer une propriété non initialement liée à un objet

On peut, aussi, créer des propriétés qui, initialement, ne sont pas associées à des objets. Pour cela, nous utilisons l'action **Add...** du menu contextuel de la fenêtre de navigation ou le menu **Logic > add feature.....** La boîte de dialogue demandant le nom de la nouvelle propriété s'affiche. La propriété, une fois créée, n'est, par défaut, pas visible (car aucun objet de lui est associé). Pour la rendre visible, nous utilisons l'action **Expand...** du menu contextuel de la fenêtre de navigation qui permet de modifier la valeur du paramètre "**minimum support of increments**". Nous mettons cette valeur à 0. La nouvelle propriété s'affiche, alors, en vert, dans la fenêtre de navigation.

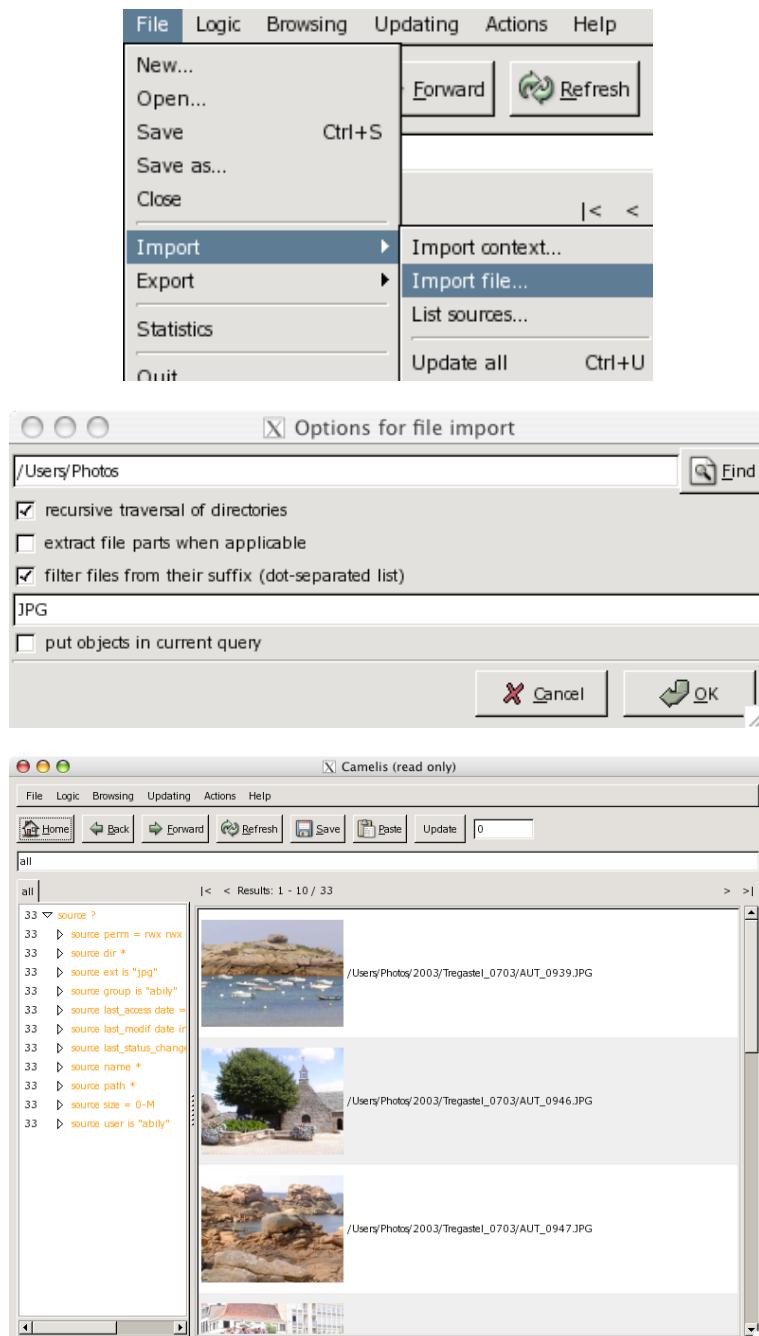


FIG. 31 – Créer des objets par importation de fichiers sources

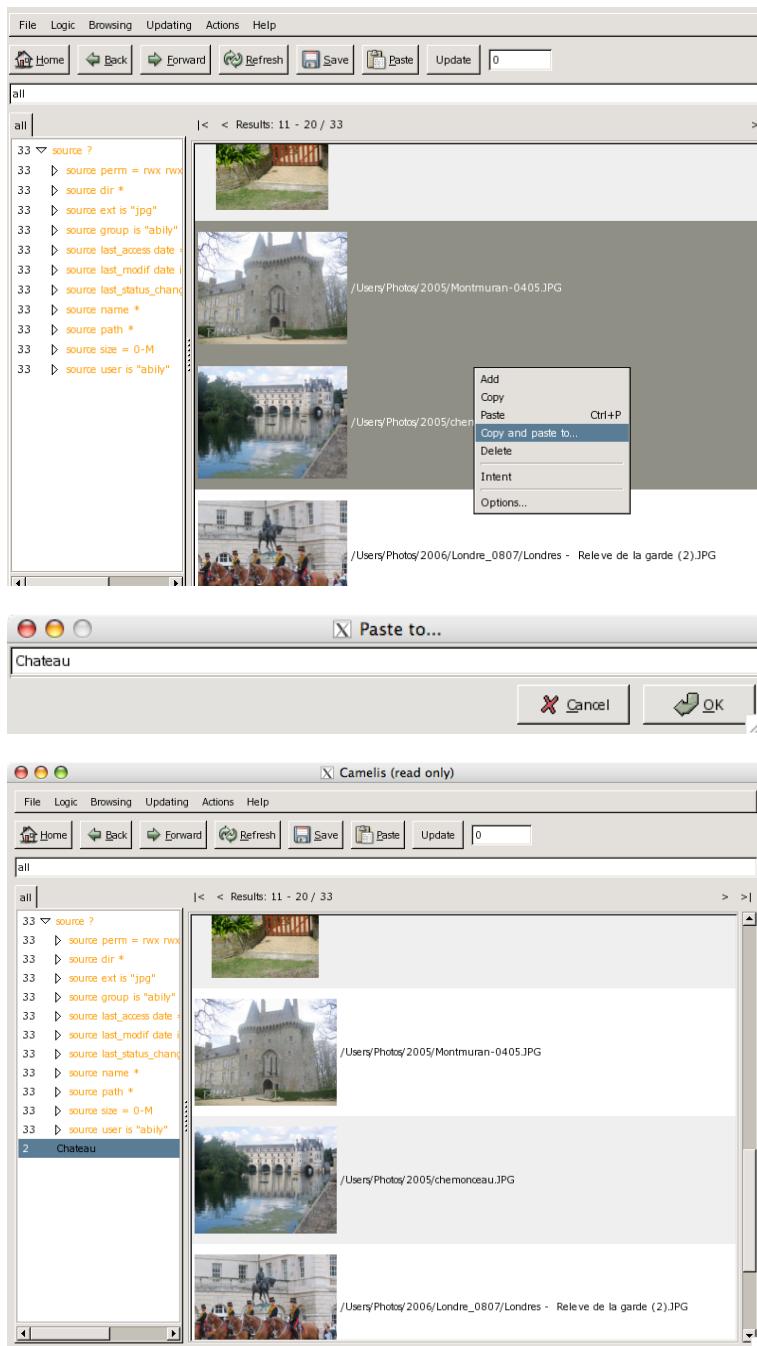


FIG. 32 – Créer une propriété initialement associée à des objets

9.2.3 Crée des associations objet-propriété

Au fur et à mesure du parcours de nos photos, nous allons être amenés à les tagger à partir de propriétés déjà créées. En trouvant, par exemple, une autre photo représentant un château, nous voulons l'associer à la propriété **Chateau** précédemment créée (cf. figure 33). A partir de la fenêtre des objets, nous sélectionnons l'objet représentant la photo, puis nous le copions via l'action **Copy** du menu contextuel de la fenêtre (ou le menu **Updating > Copy objects**). Nous sélectionnons ensuite la propriété que nous voulons attribuer à cet objet, dans notre exemple la propriété **Chateau**. Et, nous les associons par l'action **Paste** du menu contextuel de la fenêtre de navigation (ou le menu **Updating > Paste**, ou encore le bouton **Paste**). Etant donné qu'un seul objet et une seule propriété étaient sélectionnés, nous aurions, aussi, pu effectuer directement un glisser-déposer de l'objet vers la propriété.

9.2.4 Supprimer des associations objet-propriété

Nous pouvons aussi être amenés à dissocier des propriétés sur certains objets si des erreurs d'associations ont été effectuées. Par exemple, si lors de l'association d'un objet à la propriété **Chateau**, nous n'avions pas sélectionné la photo correspondante, l'association aurait été effectuée sur l'ensemble des objets de la fenêtre des objets. Nous nous serions, alors, retrouvés avec tous nos objets taggés en tant que château, ce que nous ne voulons pas.

Pour cela (cf. figure 34), nous copions les objets concernés par le menu **Updating > Copy objects** (ou l'action **Copy** du menu contextuel de la fenêtre des objets), puis nous sélectionnons la propriété **Chateau** et la dissocions des objets par l'action **Paste not** du menu contextuel de la fenêtre de navigation.

9.2.5 Hiérarchiser les propriétés en créant des axiomes

Une fois les différentes propriétés créées, il est intéressant de les classifier en créant des arbres de navigation par le biais d'axiomes. Pour notre propriété **Chateau**, nous voudrions créer une propriété plus générale **Monument** dont dépendraient automatiquement les objets taggés précédemment.

Pour cela (cf. figure 35), nous sélectionnons la propriété **Chateau** dans la fenêtre de navigation, puis nous l'incluons dans une propriété plus générale par l'action **Join as..** du menu contextuel de la fenêtre (ou le menu **Logic > Join as...**). Une fenêtre s'affiche, nous demandant le nom de cette propriété. Nous saisissons le nom **Monument**. L'action s'effectue car nos deux propriétés sont définies par des termes (leur nom commence par une majuscule). La hiérarchie **Monument > Chateau** apparaît dans l'arbre de navigation et nous constatons que nos 3 objets taggés **Chateau** sont automatiquement associés à cette nouvelle propriété.

9.3 Rechercher des objets

Maintenant que nos objets sont créés et associés à des propriétés hiérarchisées, nous pouvons effectuer des recherches en construisant des requêtes. Celles-ci peuvent être effectuées

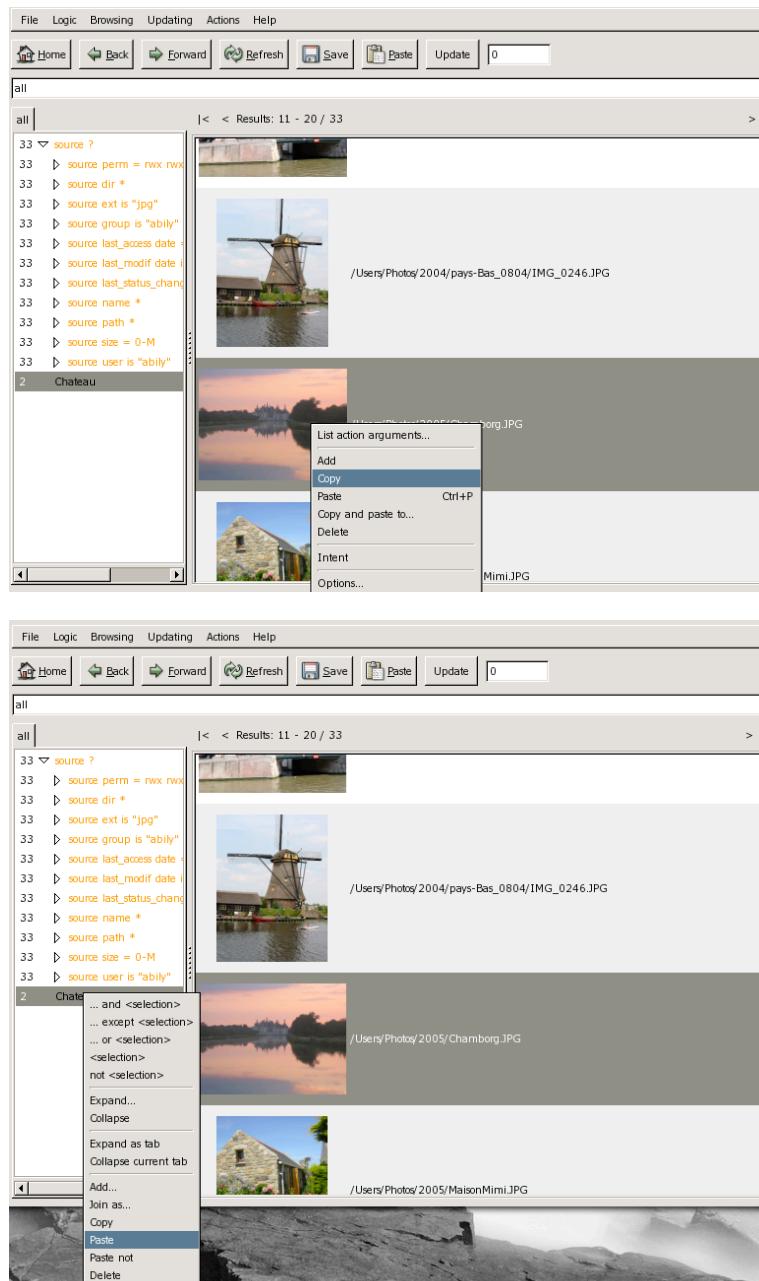


FIG. 33 – Associer un objet à une propriété déjà créée

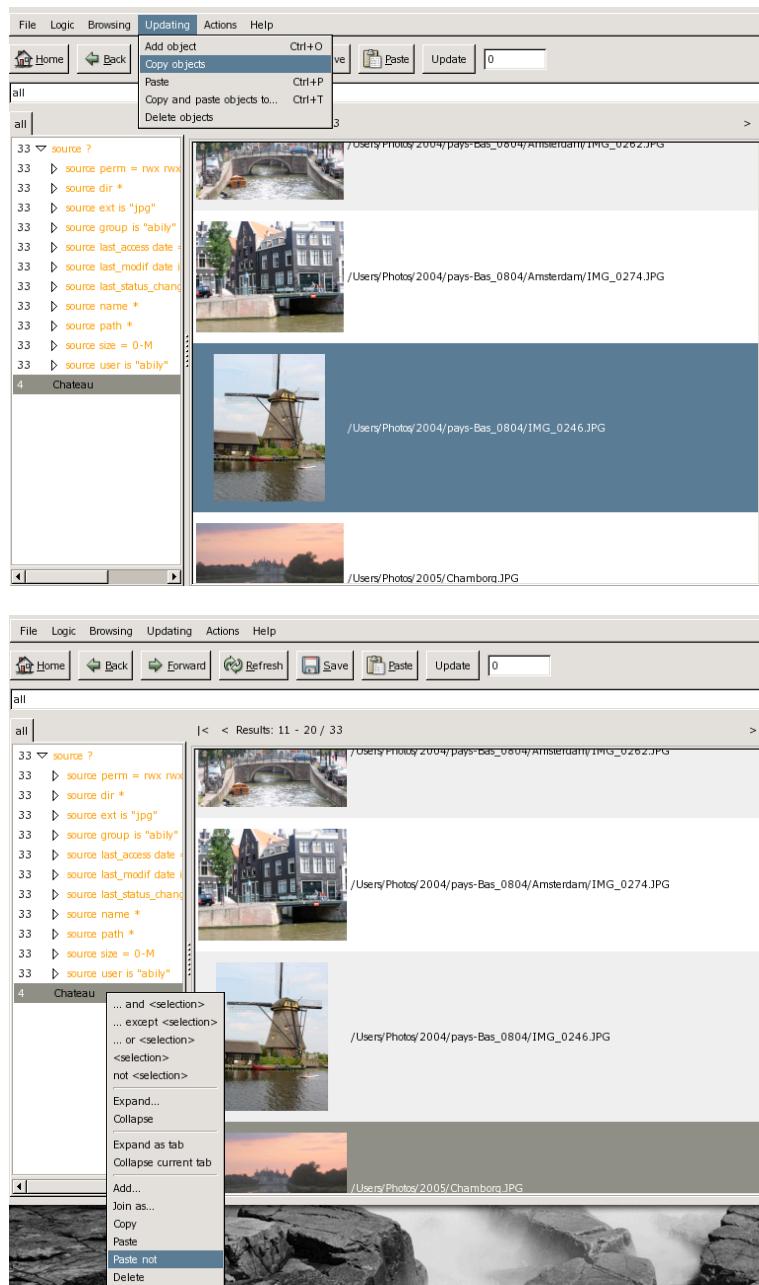
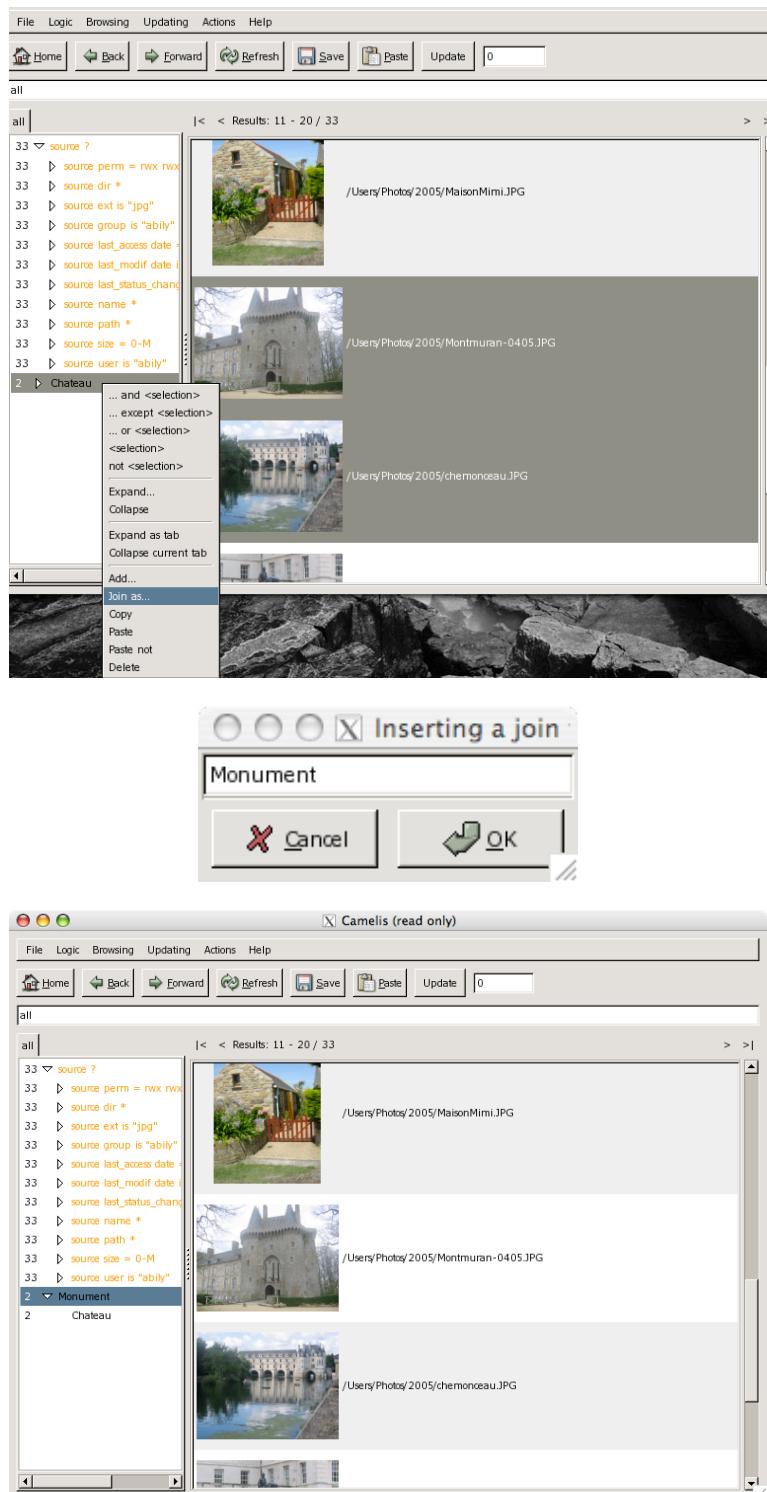


FIG. 34 – Dissocier un objet d'une propriété



PI n° 1234

FIG. 35 – Créer un axiome

de plusieurs manières : à partir de la fenêtre de navigation, en saisissant directement la requête ou à partir de la fenêtre des objets.

9.3.1 Rechercher des objets à partir de la fenêtre de navigation

De manière générale, l'utilisateur préfère ce mode de recherche qui lui permet de construire ses requêtes en naviguant dans les arbres des propriétés.

Pour effectuer une simple requête composée d'une seule propriété, il suffit de double-cliquer sur cette propriété à partir de la fenêtre de navigation. Par exemple (cf. figure 36), en double-cliquant sur la propriété **Bretagne**, la zone de requête passe automatiquement à **Bretagne**, sélectionnant ainsi les 16 objets représentant des photos prises en Bretagne. Ils sont tous associés aux propriétés **Bretagne**, **France** et **Europe**, qui sont maintenant de couleur orange car communes à tous les objets sélectionnés.

Nous pouvons, également, sélectionner plusieurs propriétés à la fois et mettre à jour la zone de requête à partir d'une action du menu contextuel ou du menu **Browsing**. Par exemple, sélectionnons les propriétés **Monument** et **Port** puis appliquons l'action **...and <selection>** du menu **Browsing**, la nouvelle requête devient "**Bretagne and Monument or Port**", affinant, ainsi notre recherche en ne sélectionnant que les photos de monuments ou de ports de pêche prises en Bretagne. Il nous reste 10 photos. Au fur et à mesure de la navigation, si nous sélectionnons des propriétés de couleur noire, qui ne sont donc pas communes à tous les objets, les requêtes se précisent, diminuant le nombre d'objets sélectionnés.

Jusqu'à présent, nous n'avons fait que rajouter de nouvelles propriétés à la requête courante. Nous pouvons avoir envie de supprimer des propriétés de la requête ou encore de remplacer une propriété par une autre.

A partir de notre exemple, supposons que nous voulons visualiser uniquement les photos de ports de pêche de Bretagne. Pour cela, à partir de la requête précédente ("**Bretagne and (Monument or Port)**") , il suffit de double-cliquer sur la propriété **Port** dans la fenêtre de navigation, la nouvelle requête devient alors "**Bretagne and Port**". La propriété **Monument or Port** a été remplacée par la propriété **Port** plus spécifique. Dans la fenêtre de navigation, la propriété **Port** étant, maintenant, commune à tous les objets sélectionnés, passe en orange. Si l'on double-clique une seconde fois sur cette propriété, celle-ci est supprimée de la requête courante qui devient **Bretagne**.

Revenons à la requête "**Bretagne and Port**", pour cela, nous pouvons utiliser le bouton **Back** permettant de remonter dans l'historique des requêtes. Nous voulons maintenant rechercher tous les ports d'Europe, ce qui revient à remplacer la propriété **Bretagne** par la propriété **Europe** dans la requête. Les deux propriétés appartiennent au même arbre de navigation et sont communes aux objets de la fenêtre des objets (et donc de couleur orange). La propriété **Europe** étant plus générale , si l'on double-clique sur celle-ci, elle remplace la propriété **Bretagne** dans la requête courante. Nous avons, ainsi, élargit notre recherche.

Nous pouvons aussi être amené à remplacer une propriété par une autre appartenant à un autre arbre de navigation. A partir de notre requête "**Port and Europe**" construite précédemment, 2 photos sont sélectionnées dont 1 avec des bateaux. Nous voudrions avoir plus de photos (pour faire un diaporama par exemple), aussi nous souhaitons remplacer

la propriété **Port** par la propriété **Bateau** pour voir si les photos de bateaux prises en Europe sont plus nombreuses que les photos de ports. Tout d'abord, nous allons affiner notre recherche en double-cliquant sur la propriété **Bateau**. Il nous reste un seul objet satisfaisant la requête "**Port and Europe and Bateau**". Puis nous élargissons la recherche en double-cliquant sur la propriété **Port** qui est alors supprimée de la requête courante.

9.3.2 Rechercher des objets à partir de la zone de saisie des requêtes

Toutes les requêtes ne peuvent pas être faites via la fenêtre de navigation, aussi, il est parfois nécessaire d'utiliser directement la zone de saisie des requêtes. Il s'agit des recherches faisant intervenir des valeurs dans les propriétés.

Recherchons, dans notre collection, des photos prises entre 2004 et 2006 de taille inférieure à 500k. Pour cela, nous saisissons la requête "**Date date in 2004..2006 and source size in ..500k**". Nous pouvons remarquer que les deux propriétés **Date date in 2004..2006** et **source size in ..500k** sont mémorisées dans la fenêtre de navigation (cf. figure 37). Chaque nouvelle propriété, donnée par l'utilisateur à partir d'une requête, est mémorisée dans les arbres de navigation pour pouvoir être réutilisée ultérieurement à partir de la fenêtre de navigation. Si cette propriété, ne s'avère à l'usage pas utile, nous pouvons la cacher en la supprimant de la fenêtre de navigation par l'action **Delete** du menu contextuel ou à l'aide du menu **Logic > Delete features**.

9.3.3 Rechercher des objets à partir de la fenêtre des objets

Nous pouvons, également, effectuer des recherches à partir de la fenêtre des objets. Partons, par exemple, de la requête "**Bretagne except Phare**" (cf. figure 38). Sélectionnons, à partir de la fenêtre des objets, un sous-ensemble d'objets qui nous semblent intéressants. Puis à partir du menu contextuel, appliquons l'action **Intent** qui sélectionne tous les objets partageant les propriétés des objets sélectionnés et modifie la requête courante en conséquence. A ce stade, les propriétés utilisées sont très spécifiques (**source path ***, **source ext is "jpg"**, ...) rendant la requête difficile à lire. Il est alors intéressant de supprimer de la requête les propriétés ne donnant pas d'informations intéressantes en parcourant les arbres de navigation comme vu précédemment.

9.4 Agir sur les objets ou les supprimer

Une fois que nous avons retrouvé les objets recherchés, nous pouvons avoir envie de les utiliser dans une application externe à Camelis. Pour cela, nous allons définir des actions pouvant s'appliquer aux objets. Il peut, également, être intéressant d'exporter une liste d'objets que nous pourrons utiliser ultérieurement dans une application, de type diaporama par exemple.

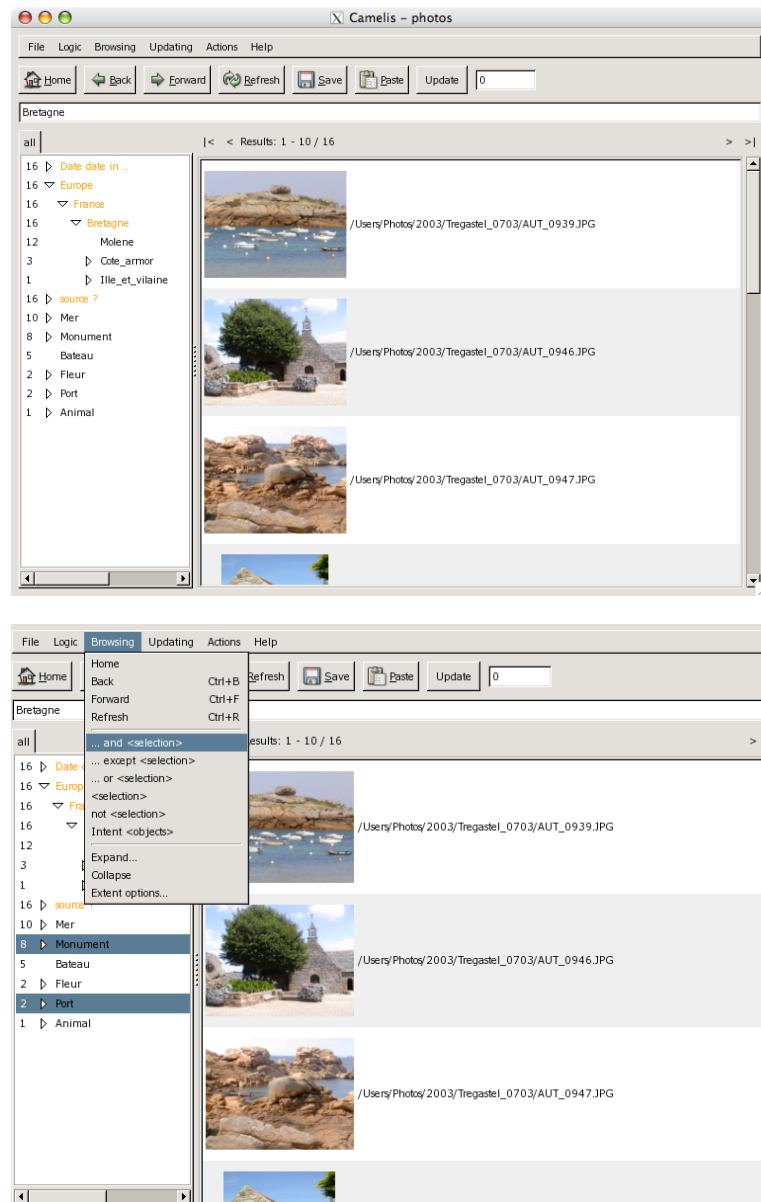


FIG. 36 – Rechercher des objets à partir de la fenêtre de navigation

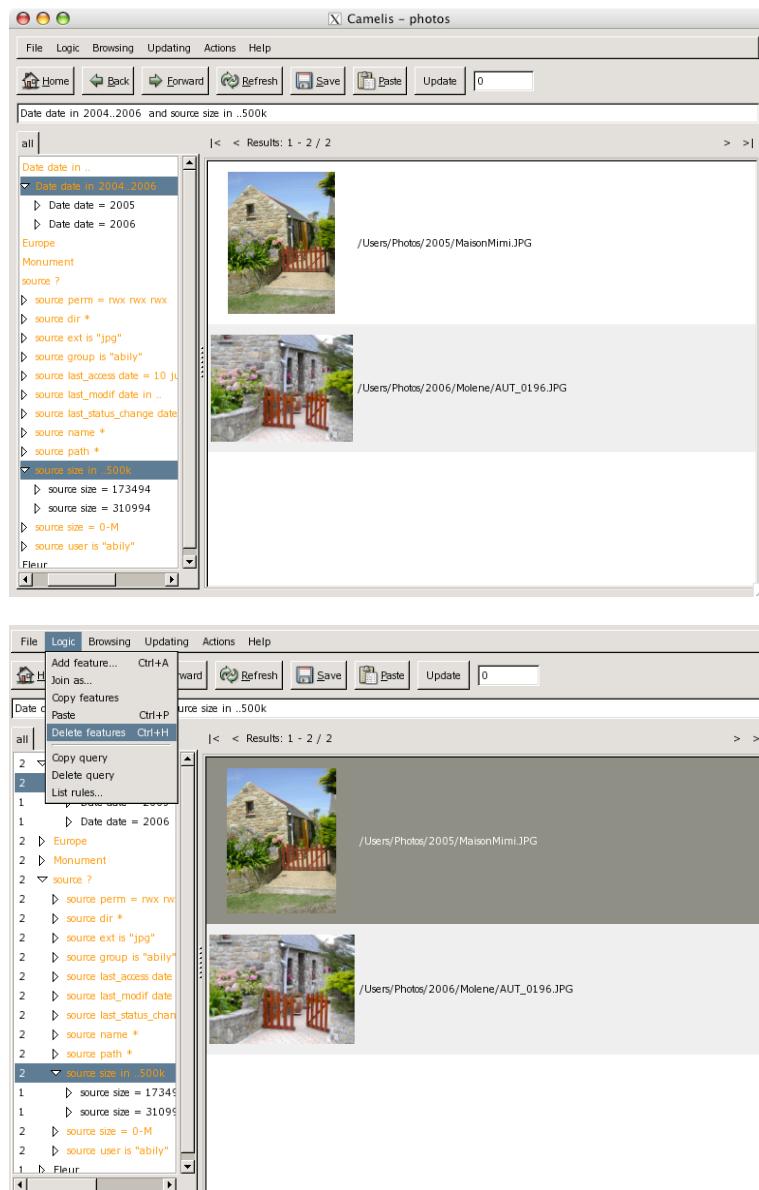


FIG. 37 – Rechercher des objets à partir d'une requête - Cacher des propriétés

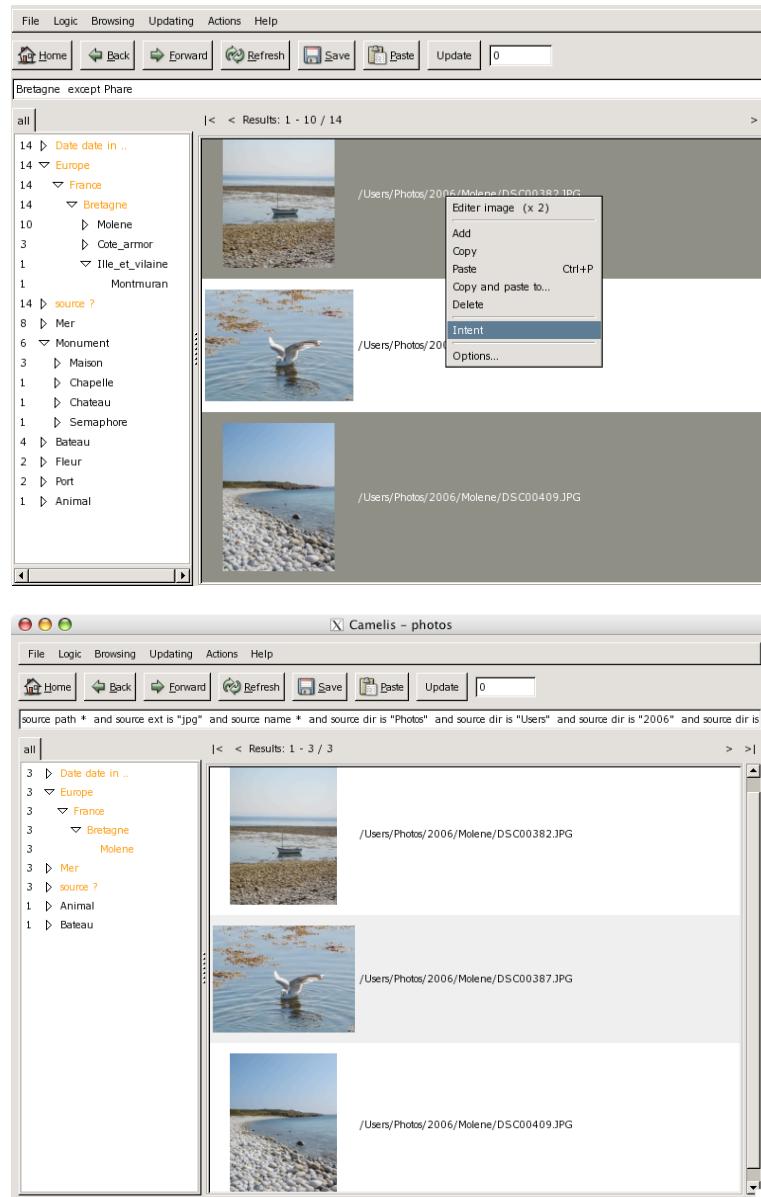


FIG. 38 – Rechercher des objets à partir de la fenêtre des objets

9.4.1 Définir une action pour un type d'objets donnés

Pour définir une action, nous utilisons le menu **Actions > Add action...** (cf. figure 39). Si par exemple, nous voulons faire du ménage dans nos photos, nous définissons l'action **Supprimer source** qui supprime le fichier source du système de fichiers en appliquant la commande `/bin/rm "$(image)" &`.

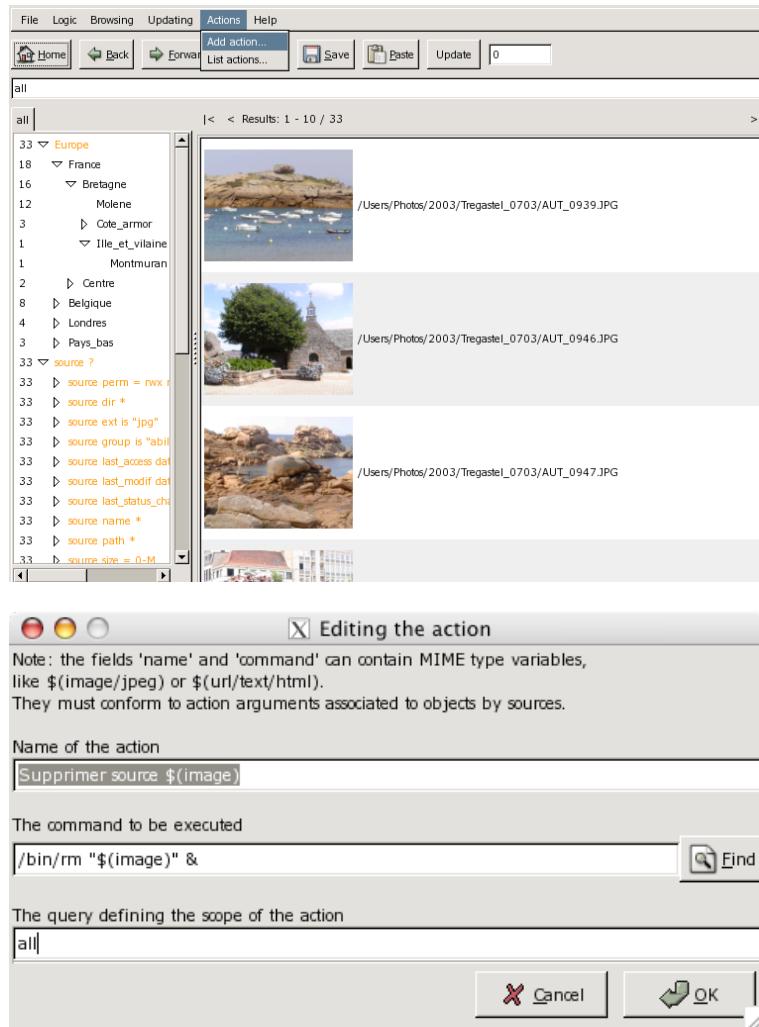


FIG. 39 – Définir une action

9.4.2 Appliquer une action sur un objet

Pour appliquer l'action précédente sur nos objets, nous passons par l'action **Supprimer source** qui, une fois définie, s'est rajoutée au menu contextuel des objets (cf. figure 40). Attention de bien sélectionner l'objet dont nous voulons supprimer le source, car si aucun objet n'est sélectionné avant d'appliquer l'action, tous les sources correspondants aux objets de la fenêtre seront supprimés, l'action s'appliquant sur tous les objets.

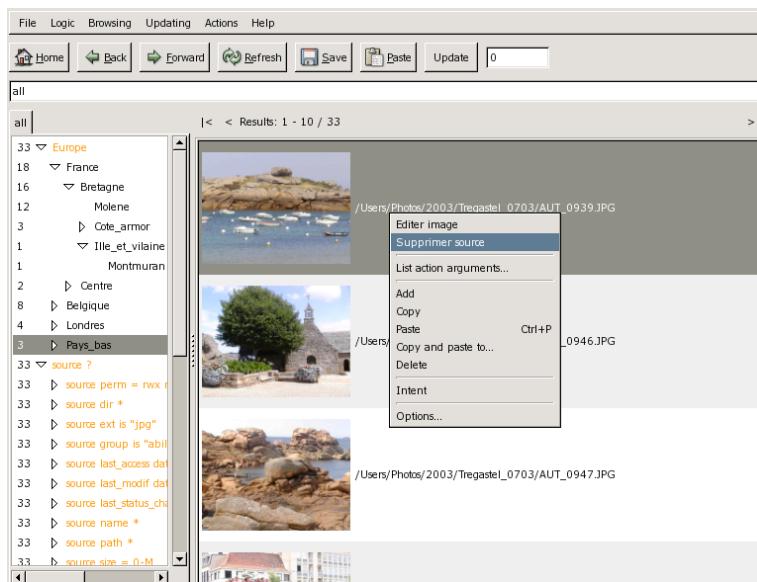


FIG. 40 – Appliquer une action

9.4.3 Supprimer un objet

Notre fichier source étant maintenant détruit, il n'est plus utile de garder l'objet correspondant. Nous allons le supprimer en le sélectionnant puis appliquant l'action **Delete** du menu contextuel de la fenêtre des objets (cf. figure 41).

9.4.4 Exporter des objets

Une fois, notre recherche aboutie, nous avons envie d'exporter la liste des objets pour effectuer un diaporama avec le logiciel GQview. Pour cela (cf. figure 42), nous ne sélectionnons aucun objet dans la fenêtre des objets pour que l'ensemble des objets soient pris en compte, puis via le menu **File > Export > Export file...** nous générerons le fichier "Monument de Bretagne.gqv" mémorisant la liste des fichiers sources au format gqv.

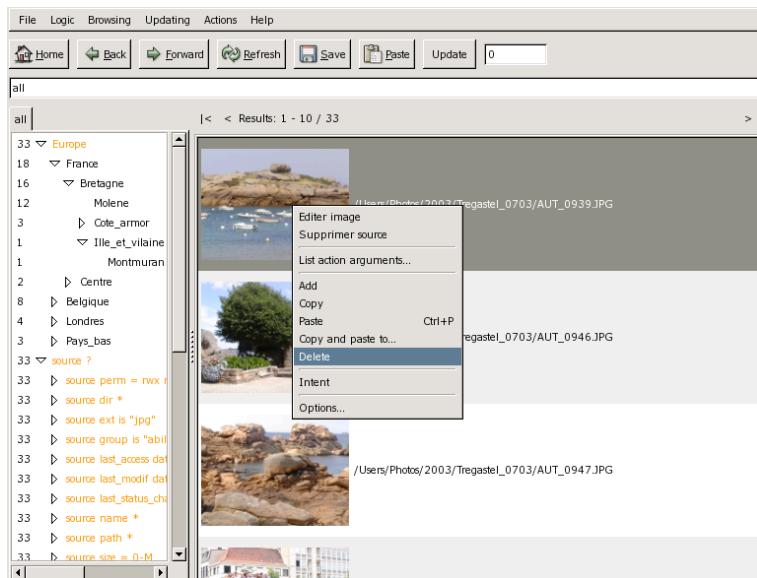


FIG. 41 – Supprimer un objet

Table des figures

1	Interface graphique de l'application Camelis	4
2	Ordre de mise à jour des différentes fenêtres de l'interface	5
3	Fenêtre de navigation	6
4	Construction d'une requête	7
5	Modification de la requête à partir de la fenêtre de navigation	8
6	Classification des valeurs d'une propriété correspondant à une date	12
7	Menu File	16
8	Sous-menu File > Import	17
9	Fenêtre de dialogue pour l'importation/exportation de contexte	17
10	Fenêtre de dialogue pour l'importation de fichiers	18
11	Fenêtre de gestion des sources importées	19
12	Sous-menu File > Export	20
13	Fenêtre de dialogue pour l'exportation de fichiers	21
14	Fenêtre de gestion des exports	21
15	Statistiques du contexte courant	22
16	Menu Logic	23
17	Fenêtre de gestion des règles de mise à jour	25
18	Menu Browsing	25

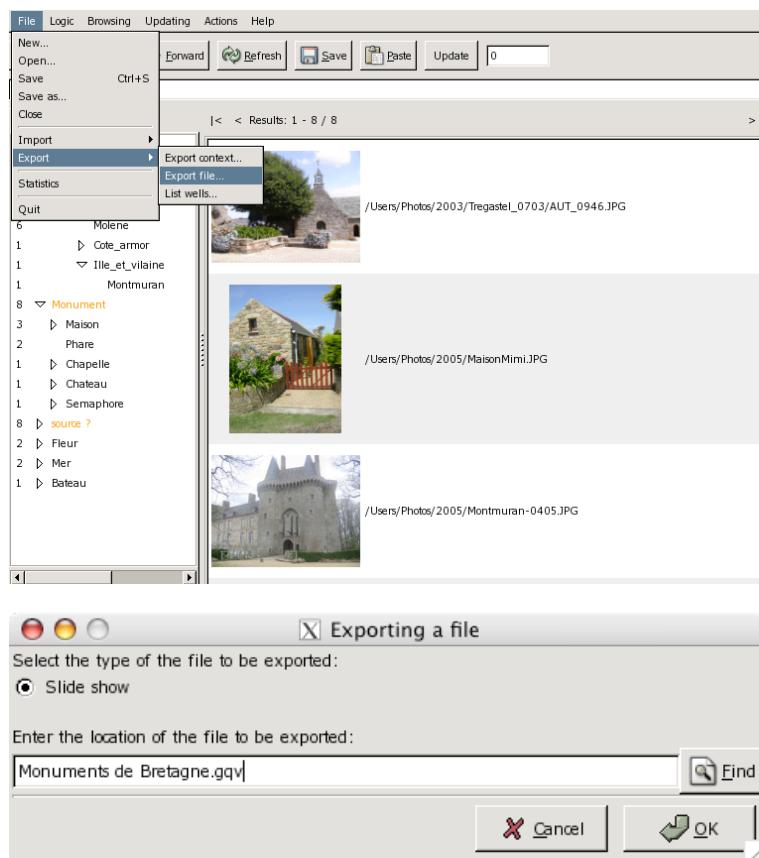


FIG. 42 – Exporter des objets

19	Fenêtre de gestion de l'affichage de la fenêtre de navigation	27
20	Fenêtre de gestion de l'affichage de la fenêtre des objets	28
21	Menu Update	29
22	Menu Actions	29
23	Fenêtre de définition d'une action	30
24	Fenêtre de définition d'une action applicable sur un ensemble d'objets	31
25	Fenêtre de gestion des actions	32
26	Menu contextuel de la fenêtre de navigation	33
27	Exemple d'onglets dans la fenêtre de navigation	34
28	Menu contextuel de la fenêtre des objets sélectionnés	35
29	Fenêtre donnant les paramètres d'un objet utilisables dans une action	37
30	Menu contextuel de la zone des requêtes	38
31	Créer des objets par importation de fichiers sources	40
32	Créer une propriété initialement associée à des objets	41
33	Associer un objet à une propriété déjà créée	43
34	Dissocier un objet d'une propriété	44
35	Créer un axiome	45
36	Rechercher des objets à partir de la fenêtre de navigation	48
37	Rechercher des objets à partir d'une requête - Cacher des propriétés	49
38	Rechercher des objets à partir de la fenêtre des objets	50
39	Définir une action	51
40	Appliquer une action	52
41	Supprimer un objet	53
42	Exporter des objets	54

Table des matières

1	Introduction	3
2	Démarrer l'application	3
3	Présentation générale	4
4	La navigation	6
5	Les propriétés	9
5.1	Chaîne de caractères	9
5.2	Entier	10
5.3	Date	10
5.4	Heure	11
5.5	Droits sur un fichier	11
5.6	Classification des valeurs	12
6	Les éléments du contexte	12
6.1	Les sources	13
6.1.1	Répertoires	13
6.1.2	Fichiers et URL	13
6.1.3	Fichiers BibTEX	13
6.1.4	Fichiers mp3	13
6.1.5	Fichiers jpg	13
6.1.6	Signets Mozilla	13
6.1.7	Répertoires Mozilla	14
6.1.8	Fichiers CSV	14
6.1.9	Fichiers interface ocaml	14
6.1.10	Enregistrements DBLP	14
6.1.11	Résultats de recherche DBLP	14
6.2	Les données extrinsèques	14
6.3	Les axiomes	14
6.4	Les tags	15
6.5	Les exports	15
6.6	Les règles de mise à jour	15
6.7	Les actions	15
7	Les différents menus	15
7.1	Le menu « File »	15
7.1.1	Chargement/Sauvegarde de contextes	16
7.1.2	Importation/Exportations d'objets	16
7.1.3	Statistiques et quitter l'application	22
7.2	Le menu « Logic »	23

7.2.1	Gestion des propriétés	23
7.2.2	Gestion des règles	24
7.3	Le menu « Browsing »	24
7.3.1	Navigation dans l'historique des requêtes	26
7.3.2	Mise à jour de la zone des requêtes	26
7.3.3	Gestion de l'affichage des fenêtre	27
7.4	Le menu « Updating »	28
7.5	Le menu « Actions »	29
7.5.1	Add action	30
7.5.2	List actions...	31
8	Les menus contextuels	32
8.1	Fenêtre de navigation	32
8.1.1	Mise à jour de la zone de requête	33
8.1.2	Gestion de l'affichage de la fenêtre	34
8.1.3	Gestion des propriétés	35
8.2	Fenêtre des objets sélectionnés	35
8.2.1	Appliquer des actions	36
8.2.2	Manipuler les objets	36
8.2.3	gestion de l'affichage de la fenêtre	37
8.3	Zone de requête	37
9	Exemples de manipulations courantes	38
9.1	Créer des objets	38
9.1.1	Objets liés au système de fichiers	38
9.1.2	Objets purement Camelis	39
9.2	Créer des propriétés, les associer aux objets et les classifier	39
9.2.1	Créer une propriété initialement associée à un ou plusieurs objets	39
9.2.2	Créer une propriété non initialement liée à un objet	39
9.2.3	Créer des associations objet-propriété	42
9.2.4	Supprimer des associations objet-propriété	42
9.2.5	Hiérarchiser les propriétés en créant des axiomes	42
9.3	Rechercher des objets	42
9.3.1	Rechercher des objets à partir de la fenêtre de navigation	46
9.3.2	Rechercher des objets à partir de la zone de saisie des requêtes	47
9.3.3	Rechercher des objets à partir de la fenêtre des objets	47
9.4	Agir sur les objets ou les supprimer	47
9.4.1	Définir une action pour un type d'objets donnés	51
9.4.2	Appliquer une action sur un objet	52
9.4.3	Supprimer un objet	52
9.4.4	Exporter des objets	52