

01/08/2008



SoCQ GUI : Manuel d'utilisation v3.0

Description des données / Fonctionnement de l'application



Julien Deflaux

Sommaire

Introduction.....	2
I. Description des données	3
A. Le Schema.....	3
B. Les BindingPattern.....	4
C. La Table	4
D. Les Attribute.....	5
II. Fonctionnement de l'application	6
A. Ajout d'une Base De Données	7
B. Ajout d'une table.....	7
C. Editeur de requêtes	7
D. Visualisation des Tables dans l'éditeur.....	8
E. Afficher/Masquer/Réduire les Tables	8
F. Masquer/Réduire une Table	9
G. La sélection des éléments	9
H. Modification de la structure d'une Table.....	9
I. Modifier les propriétés des éléments	9
J. Ajouter des tuples.....	9
K. Sauvegarde / Chargement	10

Introduction

Ce document décrit le fonctionnement de l'interface graphique (GUI = Graphic User Interface) permettant de gérer le moteur de requêtes continues **SoCQ**.

L'interface a été réalisée en **java** sous forme de plugin Eclipse RCP. Le jdk utilisé est le 1.6.

Actuellement l'interface permet de créer des modèles de données et de visualiser leur représentation et de gérer les tuples associés à ces modèles.

On ne s'attachera pas ici à la sémantique des objets manipulés, mais juste à leur cycle de vie (création, modification, destruction).

I. Description des données

En tout premier lieu, il est nécessaire de définir comment vont être représentées les modèles de données dans le GUI.

Les différents **types de données** à modéliser sont les suivants : Table, Schema, BindingPattern, Attribute.

Une **Table** contient un Schema et un ou plusieurs BindingPattern. Le **Schema** contient une liste d'Attribute. Une **BindingPattern** possède un Attribute correspondant à son Service et contient une liste d'Attribute en entrée et une liste en sortie.



Une **Table** et donc son Schema peut être une **Relation** ou un **Stream** (flux de données).


La représentation de ces types de données a donc été calquée sur les représentations UML des tables

A. Le Schema

Le Schema correspond à une Table qui peut soit être une Relation, soit être un flux de données (Stream).

Cette distinction est visible sur la représentation graphique du Schema via son icône :

- Si  précède le nom du Schema, la Table est une **Relation**.
- Si  précède le nom du Schema, la Table est un **Stream**.

Chacun des **Attribute** du Schema est précédé de l'icône . Pour plus de détails sur les Attribute, se référer au paragraphe *I-D. Les Attribute*.

Ainsi, la représentation d'un Schema peut avoir une des deux allures suivantes :







Figure 1 : Représentation d'une Relation



Figure 2 : Représentation d'un flux

B. Les BindingPattern

La représentation des BindingPattern est similaire à celle du Schema. A savoir :

- Le **nom** de la BindingPattern est précédé de l'icône 
- L'Attribute correspondant à son **Service** est précédé de l'icône 
- Les Attribute correspondant aux paramètres **d'entrée** sont précédés de l'icône 
- Les Attribute correspondant aux paramètres **de sortie** sont précédés de l'icône 

Pour plus de détails sur les Attribute, se référer au paragraphe *I-D. Les Attribute*.

Ainsi, la représentation d'une BindingPattern à l'allure suivante :



Figure 3 : Représentation d'une BindingPattern

C. La Table

La Table ne faisant que regrouper un Schema et, soit aucune, soit une ou plusieurs BindingPattern qui utilisent ce Schema, on représente ces différentes entités qui composent la Table dans un cadre de couleur bleu clair :

Dans le coin supérieur gauche de la Table est Affiché son nom en police Times taille 25. La taille de cette police est volontairement grande par rapport au contenu de la Table pour permettre une meilleure visibilité lorsqu'il y a une multitude de Table à afficher.

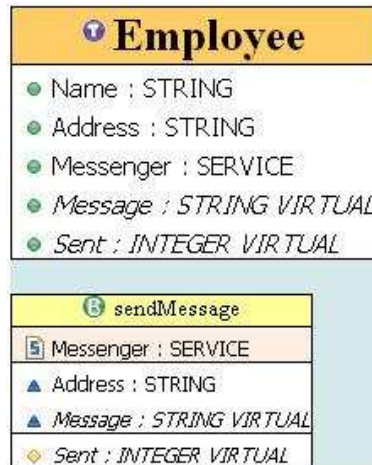


Figure 4 : Représentation d'une Table

D. Les Attribute

Les Attributes possèdent **trois caractéristiques** principales :

- Leur **nom** <name> qui est une chaîne de caractère
- Leur **type** <type> qui est un des types suivants :
 - Real
 - Binary
 - Integer
 - Service
 - String

La sémantique de chacun de ces types n'est pas précisée dans ce document.

- Un booléen qui précise s'ils sont **Virtual** ou non.

Les Attribute sont donc représentés par une chaîne de caractère ayant la syntaxe suivante :

<name> : <type> [VIRTUAL]

De plus, des icones peuvent précéder leur affichage. Ainsi ils sont précédés par l'icône :

- ● par défaut
- 📄 s'il s'agit de l'Attribute correspondant au Service d'une BindingPattern.
- 🗨 s'il s'agit d'un Attribute d'entrée d'une BindingPattern.
- ⬇ s'il s'agit d'un Attribute de sortie d'une BindingPattern.

II. Fonctionnement de l'application

Lors du lancement de l'application, aucune Table n'est encore créée. L'application se présente sous cette forme :

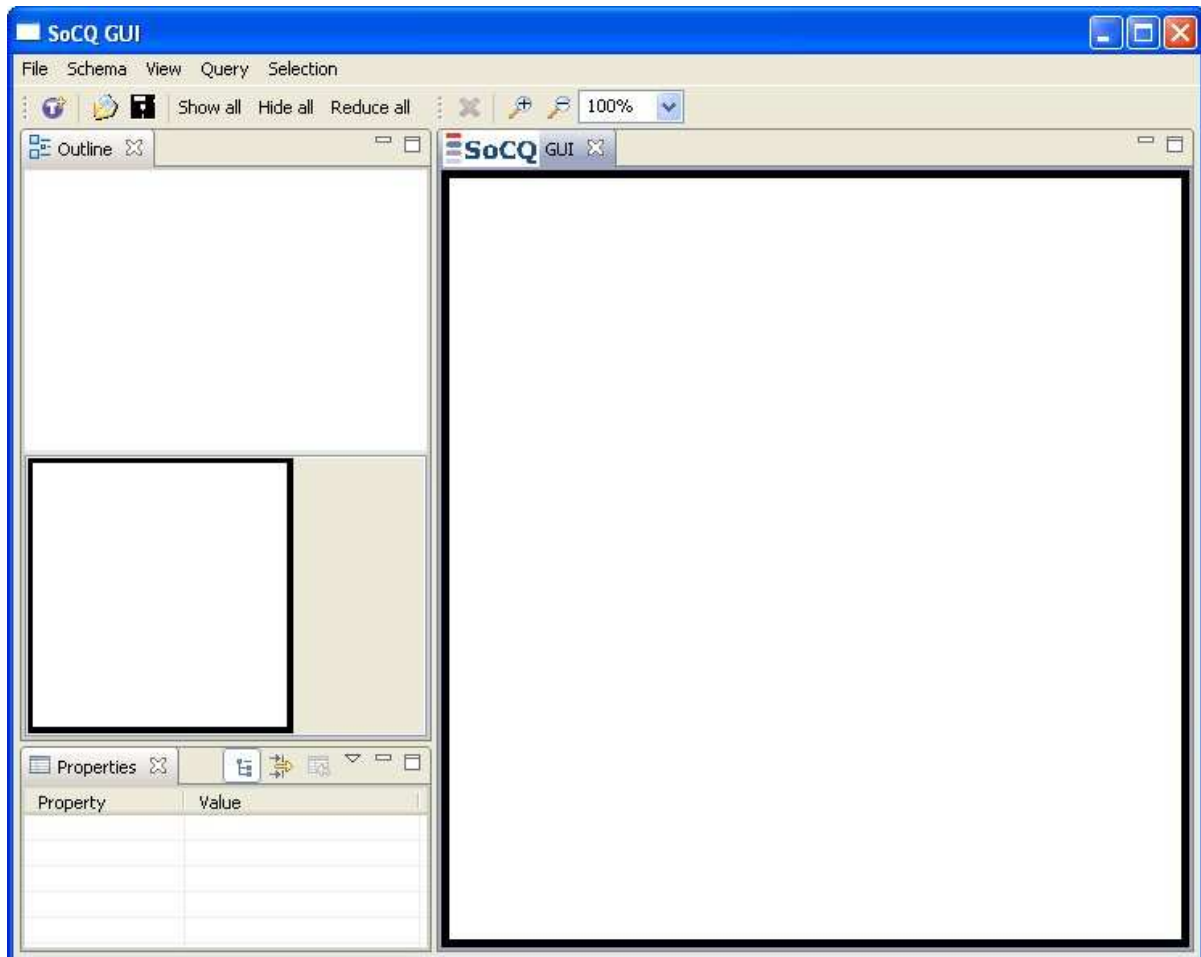


Figure 5 : GUI au lancement

Le GUI est composé de plusieurs éléments :

- Un Editeur de Tables, dans l'onglet SoCQ GUI.
- Une vue sous forme d'arbre dans l'onglet Outline.
- Une miniature de la visualisation des Tables en bas de l'onglet Outline.
- Une Feuille de propriétés Properties.
- Une barre d'outils/menu.


Un approfondissement de chacun de ces éléments est traité dans la suite du document en fonction des actions proposées à l'utilisateur

A. Ajout d'une Base De Données

Il est possible d'ajouter un nouveau Schema de Base de Données, via le Menu File-> New DataBase.

Un nouvel onglet apparaîtra dans la zone de l'éditeur, ce qui permet de disposer d'une nouvelle base de donnée indépendante des autres.

B. Ajout d'une table

Il faut ouvrir l'éditeur de requêtes grâce au bouton  de la barre d'outils, ou grâce au menu *Query* de la barre de menu.

La syntaxe des requêtes est détaillée dans la documentation *Grammaire & Parseur*.

C. Editeur de requêtes

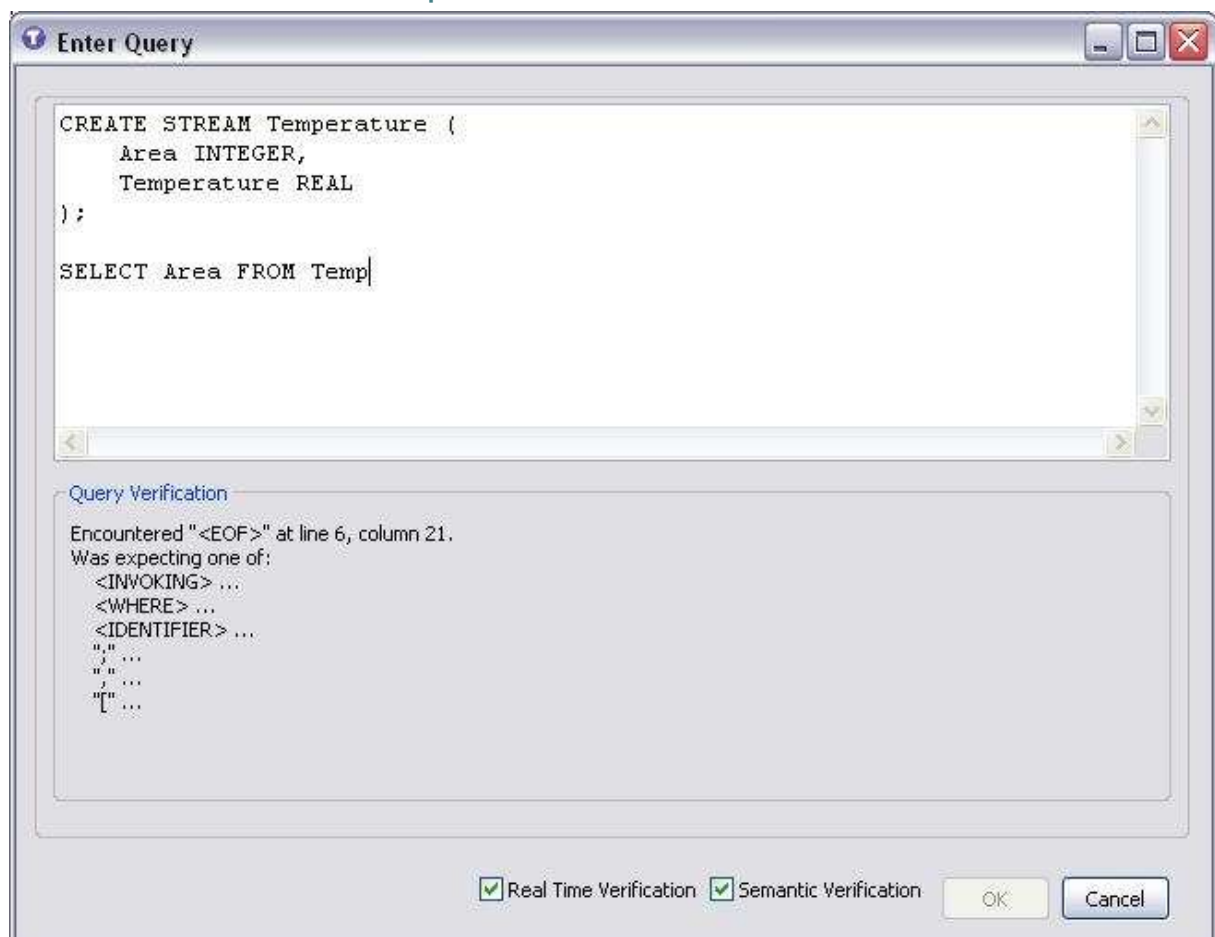


Figure 6 : Fenêtre de saisie de Requête

Un cadre permet d'entrer la requête.

Le cadre *QueryVerification* permet de vérifier en temps réel le résultat du parseur. Le parseur parcourt la requête et fait une vérification syntaxique et sémantique.

La vérification sémantique tient compte de l'ensemble des tables présentes dans l'éditeur.

Le bouton OK permet de devenir actif si la requête est sans erreur. L'objet correspondant est alors créé dans l'éditeur.

D. Visualisation des Tables dans l'éditeur

La table est représentée comme décrite dans la partie I. *Description des données*

Le zoom de la barre de menu peut entraîner des difficultés de visibilité globales, palliées par la vue miniature des tables.

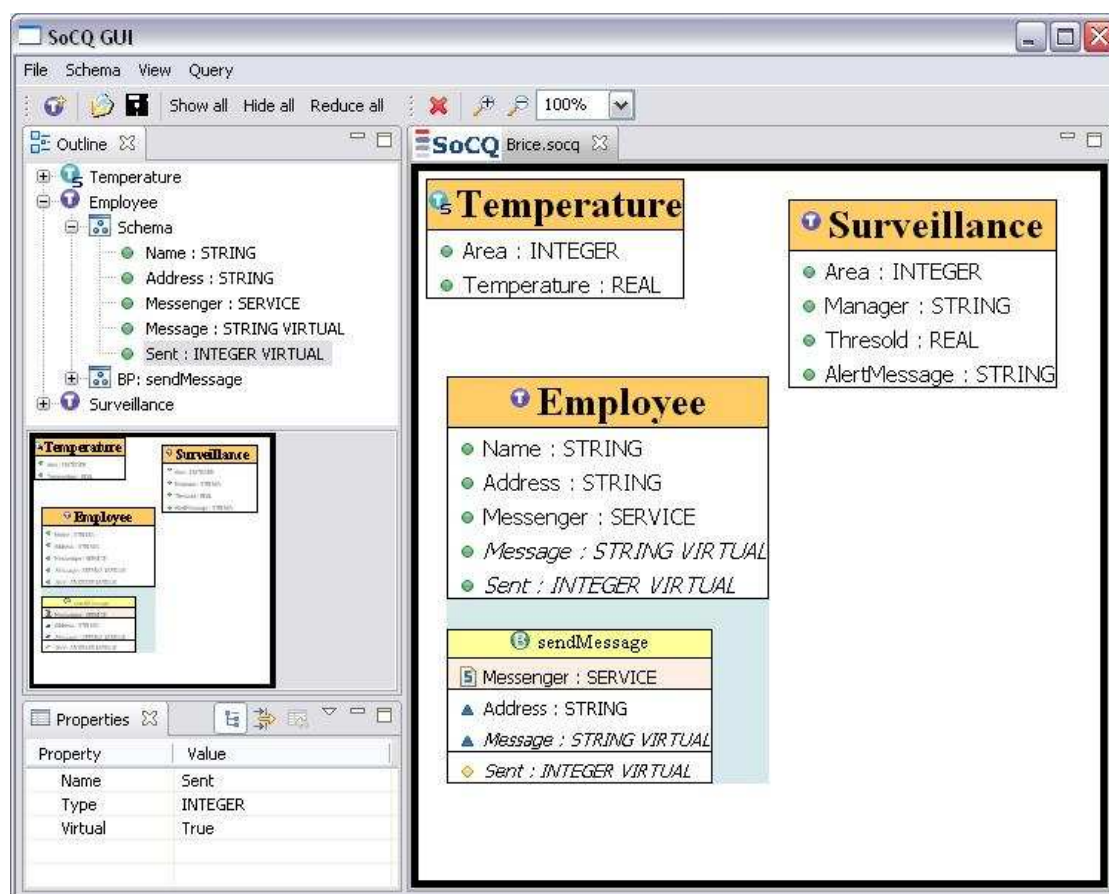


Figure 7 Visualisation d'une Table

E. Afficher/Masquer/Réduire les Tables

Il est possible d'afficher, masquer ou réduire l'ensemble des tables à travers un des boutons correspondant dans la barre de menu.

Ces fonctions sont aussi accessibles dans le menu *View*.

F. Masquer/Réduire une Table

Il est possible d'effectuer une de ces actions via le menu contextuel droit.

Pour effectuer une de ces actions via le menu déroulant du haut, il faut qu'une table soit sélectionnée.

G. La sélection des éléments

Pour sélectionner une table, il faut soit cliquer sur son cadre bleu, soit sur le cadre orange qui contient son nom.

Pour sélectionner un attribut, il faut lui cliquer dessus.

H. Modification de la structure d'une Table

Pour pouvoir modifier la structure d'une Table, il faut double cliquer sur celle-ci.

La fenêtre d'édition de requêtes va alors s'ouvrir avec la requête correspondante à la Table.

Remarque : Il ne faut pas que la table contienne des tuples, sinon il n'est plus possible de modifier sa structure.

I. Modifier les propriétés des éléments

Il faut sélectionner un élément, et dans la fenêtre de propriétés, en bas à gauche apparaît la liste des propriétés modifiables de cet élément.

Principalement on peut changer le nom des éléments, ou leur visibilité.

Remarque : pour modifier le nom d'un attribut il faut qu'il ai été sélectionné dans la Table et non dans la BindingPattern.

J. Ajouter des tuples

Pour ajouter des tuples, il faut cliquer sur une Table et faire apparaître le menu déroulant droit -> add Tuple.

La fenêtre d'édition de tuples permet d'ajouter des tuples dans TOUS les champs, sans vérifier si l'attribut est virtuel ou non. Ce qui peut poser un problème lors de l'export des tuple dans un fichier.

K. Sauvegarde / Chargement

L'utilisateur peut sauvegarder son travail à l'aide du bouton de sauvegarde de la barre d'outils.

L'ensemble des tables sera sauvegardé dans un fichier texte où chacune des tables est représentée par sa requête.

Si cette table contient des tuples, les requêtes d'insertion de tuples seront ajoutées dans ce fichier.

Ainsi les fichiers sont facilement éditables en dehors du GUI, à l'aide d'un éditeur de texte quelconque.

On peut donc charger via le bouton de chargement, n'importe quel fichier texte ayant comme contenu une requête de description de données.

Si la requête n'est pas correcte, une boîte de dialogue s'affiche pour indiquer l'erreur, et les anciennes tables présentes dans le visualisateur ne sont pas effacées.

Si la requête est correcte, toutes les anciennes tables du visualisateur sont effacées et remplacées par celles correspondantes au contenu du fichier.