Combining and Uniting Business Intelligence with Semantic Technologies
Acronym: CUBIST Project No: 257403
Small or Medium-scale Focused Research Project FP7-ICT-2009-5 Duration: 2010/10/01-2013/09/30

FCA Integration in the Triple Store, Version #2

This document describes the FCA Service component, which is part of the 2nd version of the CUBIST integrated prototype.

Type	Prototype
Document ID:	CUBIST D3.3.2
Work package:	WP3
Leading partner:	SHU
Author(s):	Constantinos Orphanides (SHU)
Dissemination level:	PU
Status:	Final
Date:	18. Mar. 2013
Version:	1.0

Versioning and contribution history

Version	Description	Contributors
0.1	Draft	Constantinos Orphanides (SHU)
0.2	Feedback/Review	Emre Sevinç (SAS)
0.3	Incorporation of feedback from 0.2	Constantinos Orphanides (SHU)
0.4	Feedback/Review	Frithjof Dau (SAP)
0.5	Incorporation of feedback from 0.4	Constantinos Orphanides (SHU)
0.6	Final draft	Constantinos Orphanides (SHU)
1.0	Final version	Constantinos Orphanides (SHU)

Reviewers

Name	Affiliation
Emre Sevinç	SAS
Frithjof Dau	SAP

Table of Contents

1 Introduction

This document provides an overview of the FCA Service component in CUBIST. This component is used to create formal contexts out of a triple store. The NowaSearch front-end component issues a request to the FCA Service to create a formal context, to be then visualised as a concept lattice by the CUBIX Visual Analytics component. The overall CUBIST architecture is seen in Figure 1.

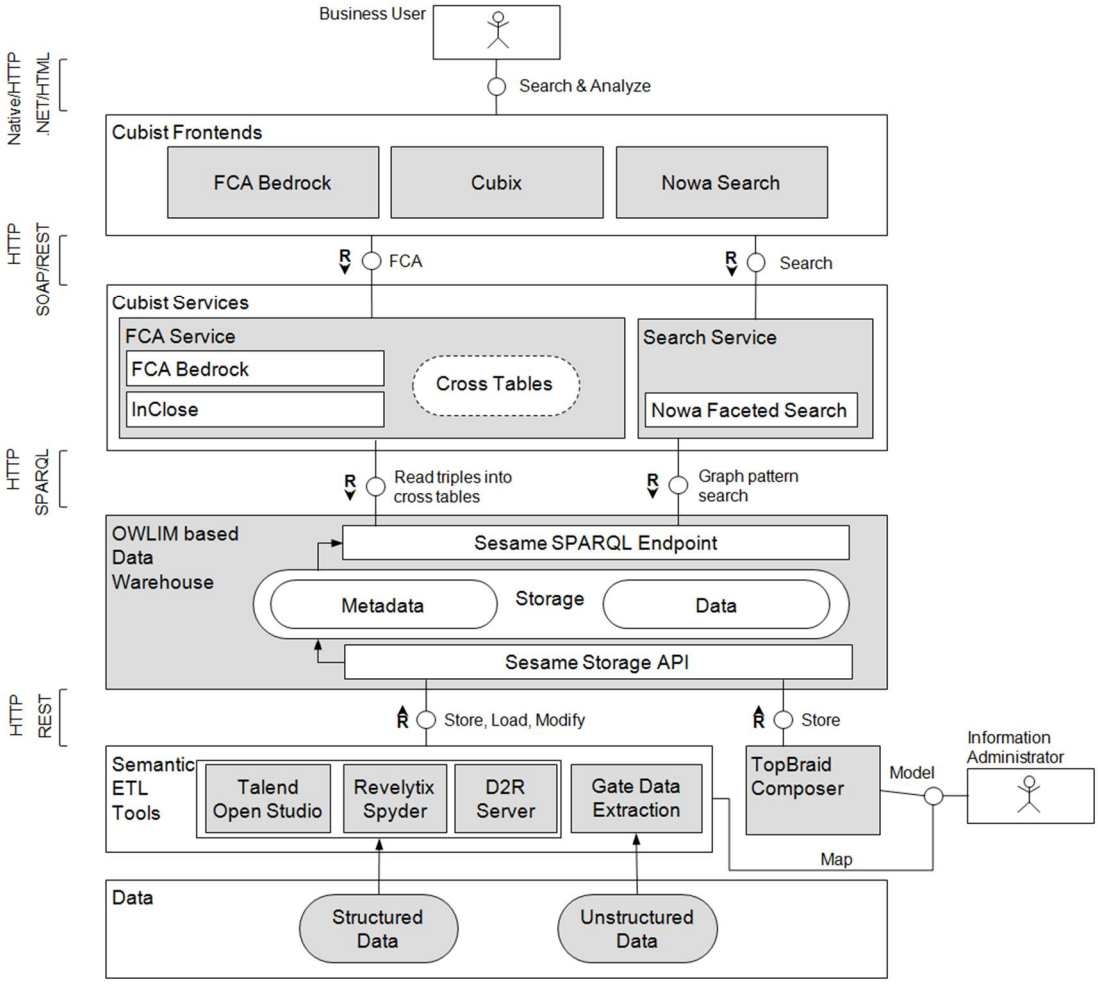

Figure 1. CUBIST Architecture

2 Overview

2.1 Installation and Binaries

For instructions on how to install the FCA Service and to gain access to the binary file please refer to D1.3.1/D1.3.2.

2.2 Architecture

The FCA Service is built using C# on the Microsoft .NET 4 Framework, using the REpresentational State Transfer[1] (REST) architecture.

2.3 Web Methods

The FCA service exposes three web-methods which are used to create a formal context out of a triple store. The two web-methods accept input and return output using the JavaScript Object Notation[2] (JSON) format, while the third one returns output in XML. A help-page of the FCAService can be accessed here: http://cubist.hallam.shu.ac.uk/FcaBedrock.svc/help

The web-methods are explained below:

Resource	URL	Description
POST newformalcontext	http://cubist.hallam.shu.ac.uk/FcaBedrock.svc/newformalcontext	This method is an HTTP post method which accepts the following parameters: 1) string repositoryConnection: the information needed to connect to a triple store. 2) string repositoryId: the ID of the repository in the triple store. 3) string sparqlQuery: the SPARQL query that the FCA service will execute on the triple store to fetch data. 4) List<Variable> allVariables: a list of a

[1] http://en.wikipedia.org/wiki/Representational_state_transfer

[2] http://en.wikipedia.org/wiki/JSON

		custom class called Variable, which holds all the necessary metadata needed to convert each variable of the SPARQL query to a formal attribute. The Variable class consists of the following properties: • bool AddAttributeName: Indicates whether to include the attribute name when creating formal attribute names. • string AttributeName: The name of the attribute. • string AttributeType: The type of the attribute. Possible values are Categorical, Boolean, Continuous, Ordinal and Date • string BinningType: How to bin the attribute (only applies to Continuous, Ordinal and Date attributes). Possible values are Equal width binning, Equal frequency binning and Standard deviation based binning. • bool IsObject: Indicates whether a variable represents objects. • String ManualBins: A string containing bins set by the user (only applies to Continuous, Ordinal and Date attributes) • int NumberOfBins: The number of bins to be created (possible values range from 2 to 10)

		• string ScalingType: What type of scaling to apply (only applies to Continuous, Ordinal and Date attributes). Possible values are Discrete (e.g. 0-10, 10-20, 20-30, …) and Progressive (e.g. 0-10, 0-20, 0-30, …) • string VariableName: the name of the variable as it appears in the SPARQL query. 5) int minSupportObjs: The minimum-support for objects that the high performance concept miner (InClose) component should apply to the formal context created by the FCAService. 6) int minSupportAtts: The minimum-support for attributes that the high performance concept miner (InClose) component should apply to the formal context created by the FCAService. 7) double faultToleranceLevel: the level of fault tolerance that should be applied to the formal context. The web-method creates a formal context based on parameters 1-7 and returns a unique formal context ID (which represents the formal context created) to the consumer of the service. The formal context ID can be then used to retrieve the actual formal context by calling the web-method below.
GET formalcontext?id={id}&format=json	http://cubist.hallam.shu.ac.uk/FcaBedrock.svc/formalcontext?id={id}&format=json	This method retrieves the formal context having the formal context ID issued with the request, or null if the particular ID does not exist. The results are returned in JSON.

GET formalcontext?id={id}&format=xml	http://cubist.hallam.shu.ac.uk/FcaBedrock.svc/formalcontext?id={id}&format=xml	This method retrieves the formal context having the formal context ID issued with the request, or null if the particular ID does not exist. The results are returned in XML.

3 An Example

Following is an example demonstrating how the FCA Service can be used to create formal contexts out of a triple store.

Let us say that we are querying the data from the HWU use-case and we are interested in finding the tissues in which Gene Bmp4 is strongly detected, moderately detected, or detected. By making the appropriate selections in the "Analyze" button of NowaSearch (Figure 2), we then proceed to the "Scaling" button of NowaSearch (Figure 3), where several scaling options are available to the user, depending on the nature of the properties selected. The "Scaling" panel of NowaSearch reflects the parameters of the "POST newformalcontext" web-method of the FCAService (first web-method defined in section 2.3): The first three parameters have been generated by NowaSearch, whereas parameters 4-7 are to be set in this panel.

In this example, both of the attributes are to be scaled as a "Categorical" attribute (i.e. a one-to-one mapping to create one formal attribute for each of the values of the two variables).

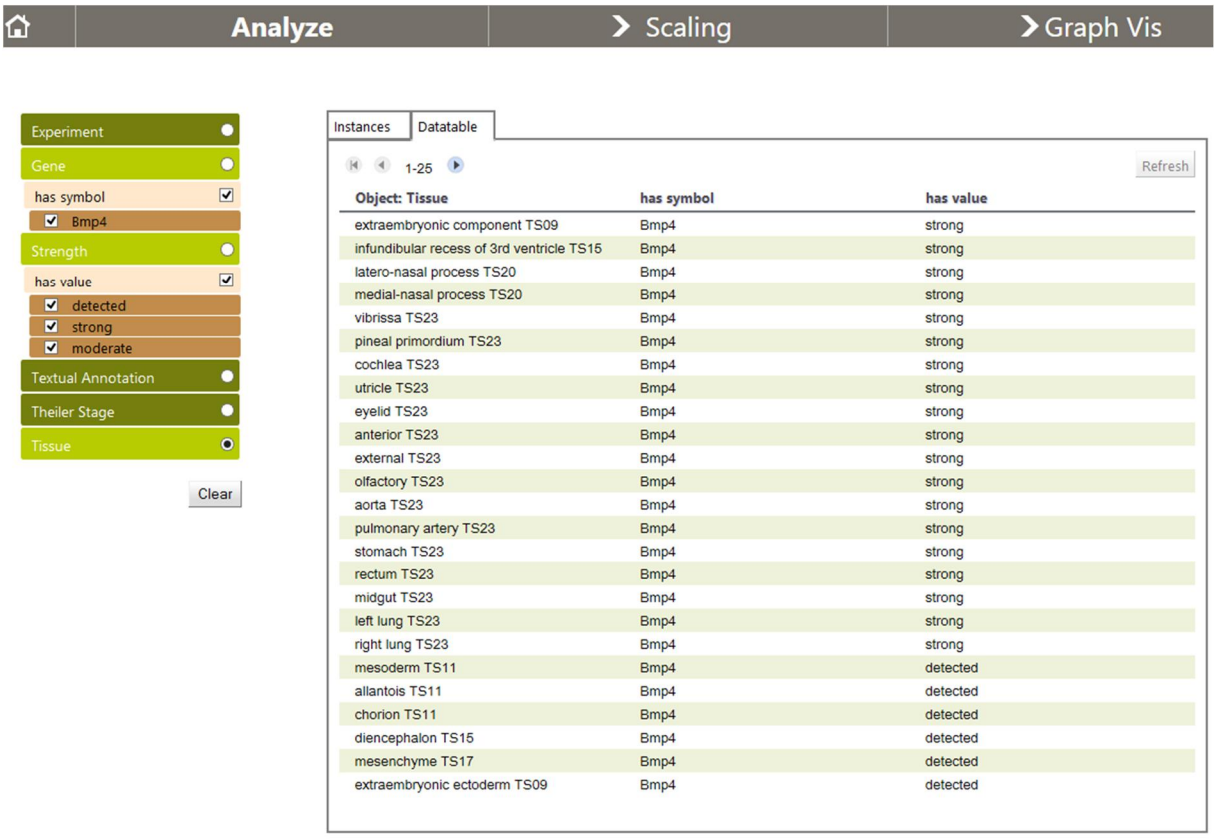

Figure 2. "Analyze" section of NowaSearch.

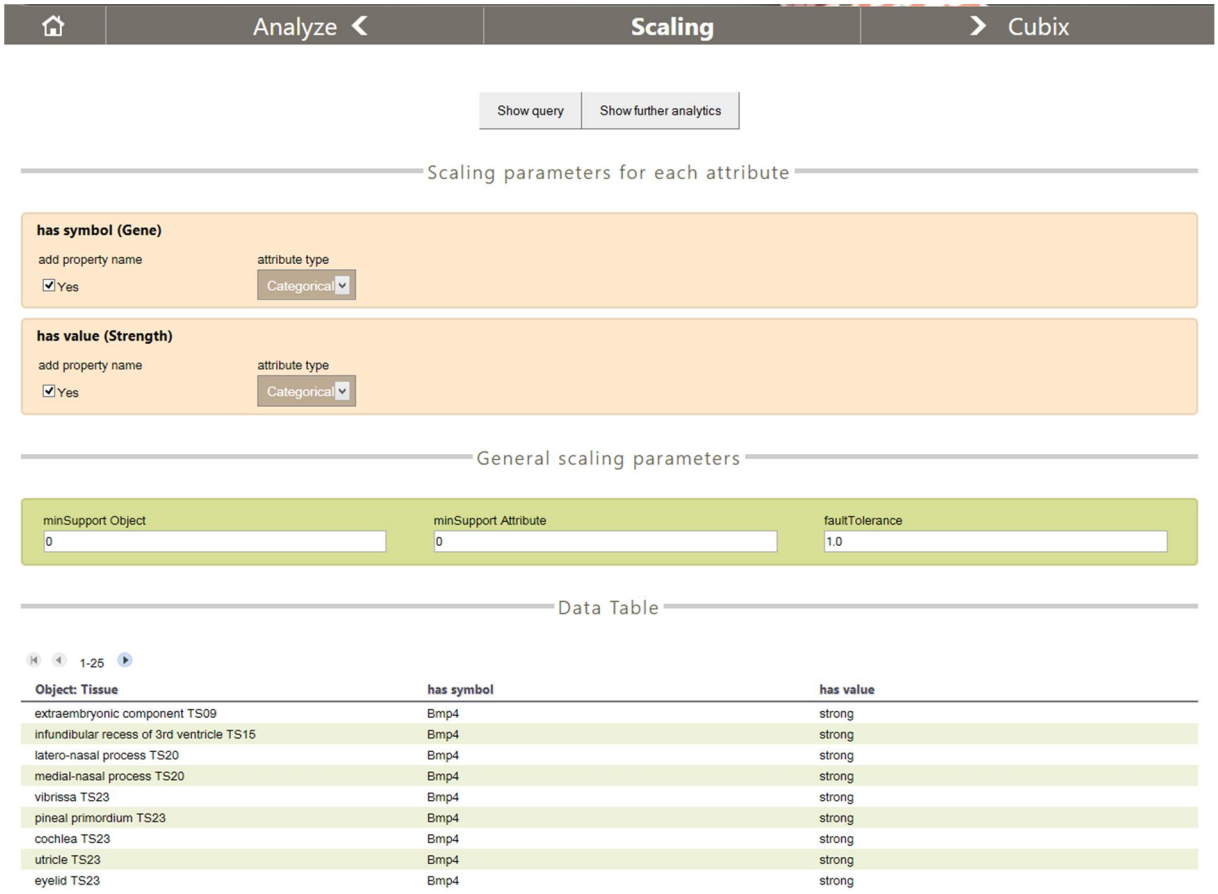

Figure 3. “Scaling” section of NowaSearch.

Once we have defined the above, pressing the “CUBIX” button will result in the following actions:

1) NowaSearch will issue a POST request (first web-method defined in section 2.3) to the FCA Service to create a formal context based on the data and scaling parameters requested by the user.
2) The FCA Service will create the formal context and returns a context ID to NowaSearch.
3) NowaSearch passes the context ID to CUBIX.
4) CUBIX issues a GET request (second web-method defined in section 2.3) to the FCA Service, to request the formal context with the context ID received from NowaSearch and visualisation takes place (Figure 4).

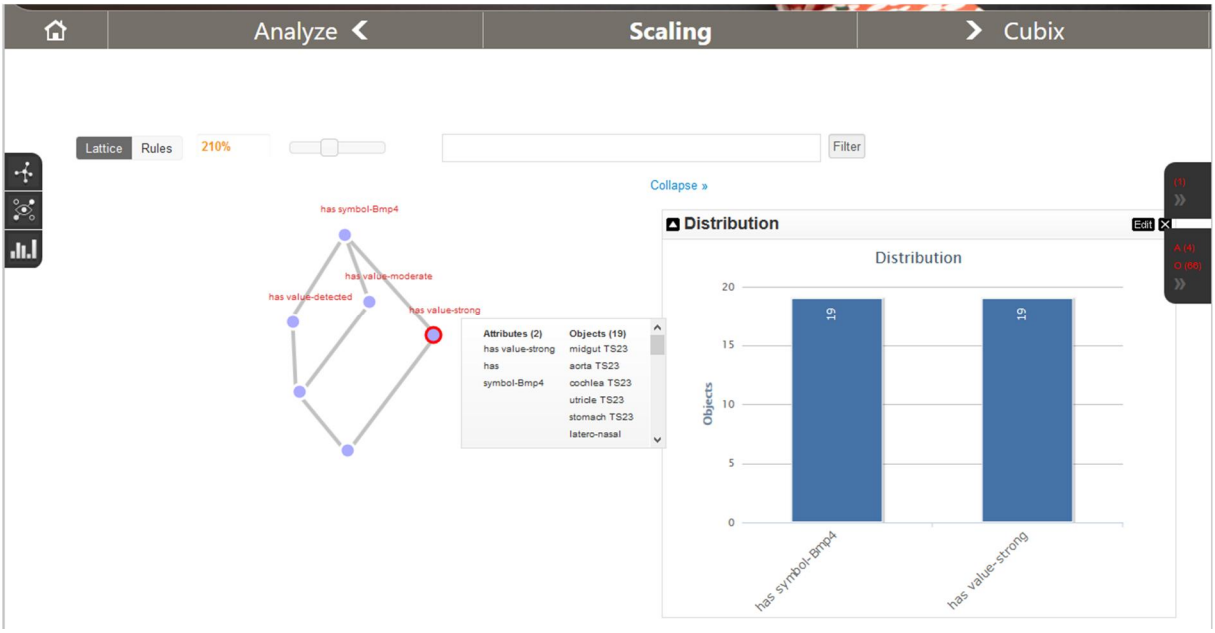

Figure 4. Visualizing the resulting formal context in CUBIX.

The formal contexts generated by the FCA Service can be accessed directly by issuing an HTTP GET (second web-method in section 2.3) on http://cubist.hallam.shu.ac.uk/FcaBedrock.svc/formalcontext?id={id}&format=json (by replacing {id} with the actual formal context ID) to retrieve the formal context. An example of what the JSON output the FCA Service produces looks like is shown in Figure 5 below; this is the same formal context that was used in the example above.


```
[
  "B",
  "",
  "66",
  "4",
  "",
  "extraembryonic component TS09",
  "infundibular recess of 3rd ventricle TS15",
  "latero-nasal process TS20",
  "medial-nasal process TS20",
  "vibrissa TS23",
  "pineal primordium TS23",
  "cochlea TS23",
  "utricle TS23",
  "eyelid TS23",
  "anterior TS23",
  "external TS23",
  "olfactory TS23",
  "aorta TS23",
  "pulmonary artery TS23",
  "stomach TS23",
  "rectum TS23",
  "midgut TS23",
  "left lung TS23",
  "right lung TS23",
  "mesoderm TS11",
  "allantois TS11",
  "chorion TS11",
  "diencephalon TS15",
  "mesenchyme TS17",
  "extraembryonic ectoderm TS09",
  "allantois mesoderm TS11",
  "amnion mesoderm TS11",
  "chorion ectoderm TS11",
  "chorion mesoderm TS11",
  "yolk sac mesoderm TS11",
  "eye TS18",
  "inner ear TS19",
  "otocyst TS17",
  "extraembryonic component TS07",
  "extraembryonic ectoder
```

```
m TS08",                                "fore-paw TS20",                        "X.X.",
   "inner cell mass TS04",              "hind-paw TS20",                        "X.X.",
   "polar trophectoderm TS              "telencephalon TS15",                   "X.X.",
04",                                    "mesenchyme TS23",                      "X.X.",
   "infundibular recess of 3            "meninges TS23",                        "X.X.",
rd ventricle TS17",                     "has symbol-Bmp4",                      "X.X.",
   "telencephalon TS20",                "has value-strong",                     "X.X.",
   "1st branchial arch TS15             "has value-detected",                   "X.X.",
",                                      "has value-moderate",                   "X.X.",
   "Rathke's pouch TS15",               "XX..",                                 "X.X.",
   "future brain TS16",                 "XX..",                                 "X.X.",
   "1st branchial arch man              "XX..",                                 "X.X.",
dibular component TS17",                "XX..",                                 "X.X.",
   "1st branchial arch maxi             "XX..",                                 "X.X.",
llary component TS17",                  "XX..",                                 "X.X.",
   "apical ectodermal ridge             "XX..",                                 "X.X.",
 TS17",                                 "XX..",                                 "X.X.",
   "1st branchial arch man              "XX..",                                 "X.X.",
dibular component TS18",                "XX..",                                 "X.X.",
   "epithelium TS16",                   "XX..",                                 "X.X.",
   "epithelium TS17",                   "XX..",                                 "X.X.",
   "nasal epithelium TS17",             "XX..",                                 "X.X.",
   "otocyst TS18",                      "XX..",                                 "X.X.",
   "branchial arch TS17",               "XX..",                                 "X.X.",
   "limb TS17",                         "XX..",                                 "X.X.",
   "ear TS17",                          "XX..",                                 "X.X.",
   "latero-                             "XX..",                                 "X.X.",
nasal process TS17",                    "XX..",                                 "X.X.",
   "medial-                             "X.X.",                                 "X.X.",
nasal process TS17",                    "X.X.",                                 "X.X.",
   "epithelium TS20",                   "X.X.",                                 "X.X.",
   "mandibular component                "X.XX",                                 "X.X.",
ectoderm TS17",                         "X.X.",                                 "X.X.",
   "apical ectodermal ridge             "X.X.",                                 "X.X.",
 TS19",                                 "X.X.",                                 "X..X",
   "mesenchyme TS19",                   "X.X.",                                 "X..X",
   "embryo TS17",                       "X.X.",                                 "X..X"
   "embryo TS19",                       "X.X.",                              ]
```

Figure 5. The formal context used by CUBIX to generate the lattice visualization in Figure 4.