Visualising computational intelligence through converting data into formal concepts

ANDREWS, S. <http://orcid.org/0000-0003-2094-7456>, ORPHANIDES, C. and POLOVINA, S. <http://orcid.org/0000-0003-2961-6207>

Published version

ANDREWS, S., ORPHANIDES, C. and POLOVINA, S. (2010). Visualising computational intelligence through converting data into formal concepts. In: XHAFA, F., BAROLLI, L., NISHINO, H. and ALEKSY, M., (eds.) Proceedings of the 2010 international conference on P2P, parallel, grid, cloud and internet computing (3GPCIC). IEEE Computer Society, 302-307.

Visualising Computational Intelligence through converting Data into Formal Concepts

Simon Andrews*, Constantinos Orphanides† and Simon Polovina*
Conceptual Structures Research Group, Communication and Computing Research Centre
Faculty of Arts, Computing, Engineering and Sciences, Sheffield Hallam University, Sheffield, UK
{s.andrews, s.polovina}@shu.ac.uk †corphani@my.shu.ac.uk

***Abstract*—Formal Concept Analysis (FCA) is an emerging data technology that complements collective intelligence such as that identified in the Semantic Web by visualising the hidden meaning in disparate and distributed data. The paper demonstrates the discovery of these novel semantics through a set of FCA open source software tools *FcaBedrock* and *In-Close* that were developed by the authors. These tools add computational intelligence by converting data into a Boolean form called a Formal Context, prepare this data for analysis by creating focused and noise-free sub-Contexts and then analyse the prepared data using a visualisation called a Concept Lattice. The Formal Concepts thus visualised highlight how data itself contains meaning, and how FCA tools thereby extract data's inherent semantics. The paper describes how this will be further developed in a project called CUBIST, to provide in-data-warehouse visual analytics for RDF-based triple stores.**

***Keywords*-Formal Concept Analysis, FCA, Formal Context, Formal Concept, visualisation, Concept Lattice, data warehousing, in-warehouse analytics, attributes, objects, Galois connection, Semantic Web, RDF, distributed data, disparate data**

I. Introduction

As its core, the Semantic Web comprises of design principles, collaborative working groups and a variety of enabling technologies [16]. It includes formal specifications such as the *Resource Description Framework (RDF)*, *Web Ontology Language (OWL)* and a variety of data interchange formats, such as *RDF/XML* and *N-Triples* [14]. These technologies provide a formal description of concepts, terms and relationships that capture and integrate meaning with distributed data within a given domain. New data technologies are emerging that can analyse, annotate and visualise such data and promote collective computational intelligence. In this vein, a data analysis method that has been rapidly developed during the past two decades and focuses on knowledge presentation, information management and identifying conceptual structures among semantic data is *Formal Concept Analysis (FCA)*.

II. Formal Concept Analysis

Formal Concept Analysis is a term that was introduced by Rudolf Wille in 1984 and builds on "applied lattice and order theory that was developed by Birkhoff and others in the 1930's" [18] and was initially developed as a subsection of Applied Mathematics, based on the mathematisation of concepts and concepts hierarchy.

A *Formal Concept* is constituted by its *extension*, comprising of all objects which belong to the Concept, and its *intension*, comprising of all attributes (properties, meanings) which apply to all objects in the extension [18]. The set of all objects and attributes together with their relation to each other form a *Formal Context*, which can be represented by a cross-table [12].

Airlines	Latin America	Europe	Canada	Asia Pacific	Middle east	Africa	Mexico	Caribbean	USA
Air Canada	×	×	×	×	×		×	×	×
Air New Zealand		×		×					×
Nippon Airways		×		×					×
Ansett Australia				×					
Austrian Airlines		×	×	×	×	×			×

The cross-table above is a Formal Context representing the destinations of five airlines, where the elements on the left are formal objects (airlines) and the elements at the top are formal attributes (destinations). If an object has a specific attribute, it is indicated by placing a cross in the corresponding cell of the table. An empty cell indicates that the corresponding object does not have the corresponding attribute. In the Airlines Context, Air Canada flies to Latin America but does not fly to Africa.

A central notion of FCA is a duality called a '*Galois connection*'. This connection is often observed between two types of items that relate to each other. A Galois connection implies that "if one makes the set of one type larger, they will be related to a smaller set of the other type, and vice versa" [12]. In the airlines example, the combination of destinations, Asia Pacific, Europe and USA, are flown to by four airlines. If Middle East is added to the list of destinations, the number of airlines reduces to two.

The definition of a Formal Concept is extended by the idea of closure: the extension contains all objects that have the attributes in the intension, and the intension contains

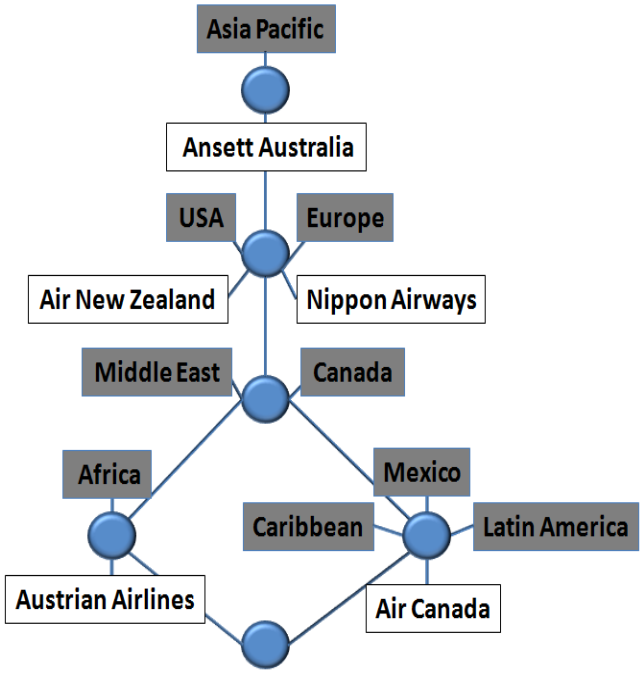

Figure 1. A Lattice corresponding to the Airlines Context

all attributes shared by the objects in the extension. In the example of the two airlines that fly to Asia Pacific, Europe, USA and the Middle East, it can be seen from the table that Canada is also flown to by the same two airlines. Adding Canada to the list of destinations completes (closes) that particular Formal Concept.

A strength of FCA is that the Galois connections between the Formal Concepts can be visualised in a *Concept Lattice* (Figure 1), which is an intuitive way of discovering hitherto undiscovered information in data and portraying the natural hierarchy of Concepts that exist in a Formal Context.

Each node in the lattice is a Formal Concept. The objects (airliners) are labeled below the nodes, while the attributes (destinations) are labeled above the nodes. Extracting information from a lattice is straightforward and easy to understand. In order to see which attributes are featured by an object, one begins from the node where the object is located and starts heading upwards in the lattice. Any attributes one meets along the way are the attributes featured by that object. For example, if one heads upwards in the lattice from Air New Zealand (object), one will collect the attributes USA, Europe and Asia Pacific. This can be interpreted as 'Air New Zealand flies to USA, Europe and Asia Pacific'. Similarly, in order to see which objects have a specific attribute, one heads downwards in the lattice. Any objects one meets along the way are objects which feature that particular attribute. For example, heading downwards from Canada (attribute), one will collect the objects Austrian Airlines and Air Canada. This can be interpreted as 'Canada is flown to by Austrian Airlines and Air Canada'. Although the Airline Context is only a small example of FCA, it is evident that the Concept Lattice provides information that is not evident from at looking the table alone.

III. Motivation

Going beyond the simple example above, it has been shown that FCA can be usefully applied to large sets of data and that FCA has applications in data mining [10]. Rather than X-sized data it can handle XX-sized data. Programs, such as In-Close[1] [1], exist which are capable of handling and processing large Formal Contexts. However, issues arise when trying to visualise the Concept Lattice. Formal Contexts that have tens of attributes and thousands of objects can easily contain tens, if not hundreds of thousands of Formal Concepts [9]. The *Mushroom* data set [4], for example, has 23 attributes (properties of mushrooms) and 8124 objects (mushrooms). The Formal Context that results from the data set contains over 220,000 Formal Concepts. Lattice visualisation software does not exist that can compute lattices with such large numbers of nodes. Even if such tools existed, the results would be highly complex and unreadable, unless a sophisticated means of managing the lattice was employed.

Another issue is the fact that disparate and distributed data do not, by definition, exist in a unified form. They need to be converted into Formal Contexts first, in order for FCA to be carried out. This is done by *discretising* and *Booleanising* data; taking each many-valued attribute in a data set and converting it into as many Boolean attributes as it has values and by scaling continuous values using ranges [3].

The task of converting data into Formal Contexts can be time consuming, is open to interpretation and usually requires a programming element. FcaBedrock[2], a tool that has been developed to facilitate this process, is described later on.

IV. Interpreting Data for FCA

A. Data Discretisation and Booleanisation

Data Discretisation is defined as "a process of converting continuous data attribute values into a finite set of intervals with minimal loss of information" [8]. Data mining tasks often involve dealing with continuous attributes, which can result in a decrease of performance [8]. This can be resolved by producing discretised versions of continuous attributes, making them easier to handle and increasing the performance of mining tasks.

On the other hand, *Data Booleanisation* involves taking a many-valued attribute in a data set and converting it into as many Boolean attributes as it has values [7]. This is also the approach used to convert categorical attributes in FCA, as

[1]https://sourceforge.net/projects/inclose
[2]https://sourceforge.net/projects/fcabedrock

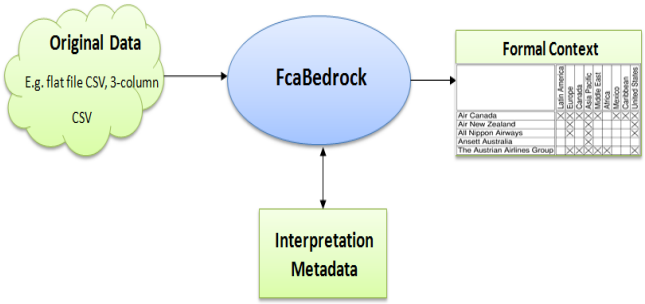

Figure 2. FcaBedrock Process.

they are converted by creating a Formal Attribute for each of the attribute categories [2].

B. FcaBedrock

FcaBedrock is a tool for creating Formal Contexts for FCA [3] by converting many-valued data into Boolean data (many-valued attributes into Formal Attribute). By using a process of guided automation, the tool obtains the metadata required for conversion, such as attribute names and attribute types. The metadata are stored, can be edited, used for subsequent conversions and act as a record of how data was interpreted (Figure 2). FcaBedrock currently takes many-column CSV and 3-column CSV files as input, but a version is being developed that also takes RDF-S and OWL formats [11]. The tool can convert categorical (nominal), Boolean and continuous attribute types into Formal Attributes. As opposed to classical FCA, FcaBedrock converts continuous attributes by using disjoint ranges, rather than progressive scaling [19], since ranges such as *0-9*, *10-19*, *20-29* lead to a less dense Context than the corresponding scales *<10*, *<20*, *<30* and make the size of Concepts in a data set more manageable. Large data sets are easily converted into two popular FCA formats, Burmeister (.cxt) [13] and FIMI[3] (.dat).

Creating Formal Contexts using FcaBedrock is straightforward, versatile and reproducible. The tool has a variety of features, which give the user a high degree of control over the conversion process, such as the ability to repeat metadata for similar attributes and the ability to auto-detect the metadata, directly from the input file, if desired. FcaBedrock also provides features for data analysis (or data preparation); it allows the exclusion of attributes from a conversion if they are not of particular interest in, or appropriate for, the analysis. Furthermore, the conversion can be restricted to only those objects with user-specified values.

V. Concept and Lattice Generation

Several tools exist for visualizing lattices, such as the *ToscanaJ* kit [5], a complete suite of tools for creating

[3]http://fimi.cs.helsinki.fi/

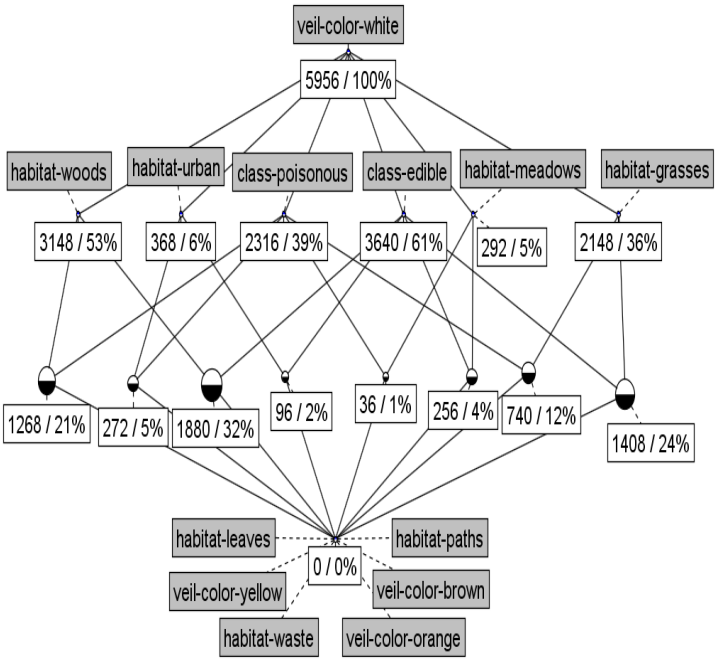

Figure 3. Visualising a *Mushroom* sub-Context in ConExp.

and using Conceptual Information Systems and *ConExp*[4], a tool for analysing Formal Contexts, exploring dependencies between attributes, counting Formal Concepts and building Concept lattices using Contexts in popular FCA formats as input.

However, as mentioned earlier, the actual usefulness of the lattices produced by lattice visualization software heavily relies on the size of the Formal Contexts given as input. Formal Contexts with a large number of Formal Concepts can result in performance issues and the production of unmanageable, unreadable lattices. FcaBedrock's data preparation features address these issues, by allowing the creation of sub-Contexts, based on exclusion and restriction criteria set by the user. This means of focusing the analysis can result in significantly smaller and easier to visualise lattices. Figure 3 shows an example of applying the attribute exclusion and restriction features of FcaBedrock to the *Mushroom* data set [4]. The data set originally comprises of 8124 objects, 23 attributes and 125 Formal Attributes[5]. 20 attributes were excluded from the conversion, except from the *class* (poisonous and edible), *veil-color* and *habitat* attributes. Furthermore, the habitat attribute was restricted to *woods*, *urban*, *meadows* and *grasses*. The sub-Context returned 5956 objects and 13 Formal Attributes.

The lattice produced some interesting information. For example, in the sample, mushrooms that live in woods,

[4]http://sourceforge.net/projects/conexp/

[5]The number of Formal Attributes varies according to user interpretation of the original data.

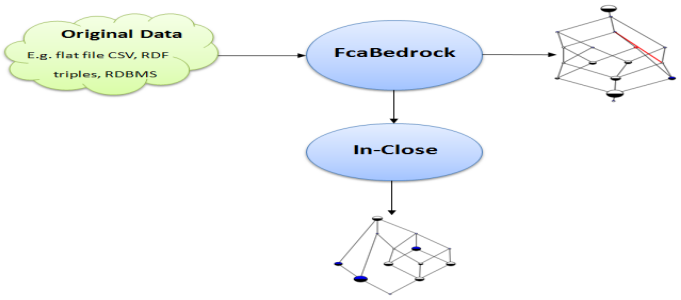

Figure 4. Visualising Formal Contexts using FcaBedrock and In-Close.

meadows, grass and urban areas all have white veils. The proportions of edible to poisonous mushrooms in the various habitats would suggest that woods and meadows are the best places for mushroom pickers to go.

VI. Dealing with Noise

The above example has shown how the production of smaller and easier to visualise lattices is possible by using FcaBedrock. However, this has been achieved by significantly reducing the size of the Formal Context by restricting the data conversion. An alternative approach, that involves all of the data, is to focus the analysis on large Concepts. Small Concepts (noise) can be filtered out of a Context using the In-Close program (Figure 4). In-Close accepts Formal Contexts in the Burmeister (.cxt) format as input, and computes its Concepts [1]. In-Close allows the user to exclude from the computation Concepts with fewer than user-specified numbers of attributes and objects (so-called *minimum support*). After the computation of Concepts, In-Close outputs the same Burmeister file, but with only those Concepts that have the minimum support set by the user. By setting the minimum support high enough so that a relatively small number of Concepts are produced, 'noise-free' Contexts can be produced which are easily managed and readable in lattice visualization software.

Figures 5 and 6 show the results in ConExp of minimum support applied to the *Mushroom* data. To compare features of edible and poisonous mushrooms in the sample, two sub-Contexts were created using FcaBedrock; one containing all the edible mushrooms and the other containing all the poisonous ones. Each sub-Context was processed by In-Close to produce a manageable number of Concepts, by setting appropriately large values for minimum support. In each case, the minimum number of attributes in a Concept was set to ten. For the edible mushrooms, the minimum number of objects in a Concept was set to 1900, resulting in a Context containing 17 Concepts. For the poisonous mushrooms, the minimum number of objects in a Concept was set to 885, resulting in a Context containing 14 Concepts. These sizes, although somewhat arbitrary, were set to provide lattices with a similar, readable, number of nodes.

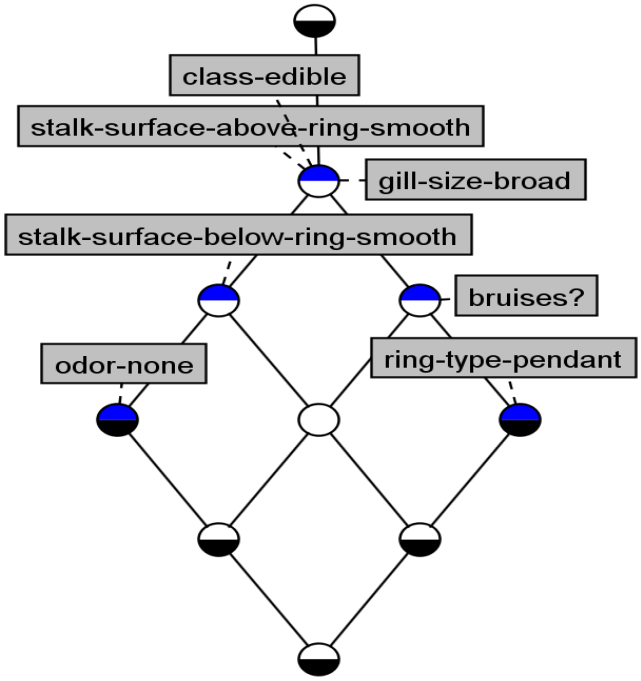

Figure 5. Edible Mushroom Concept Lattice.

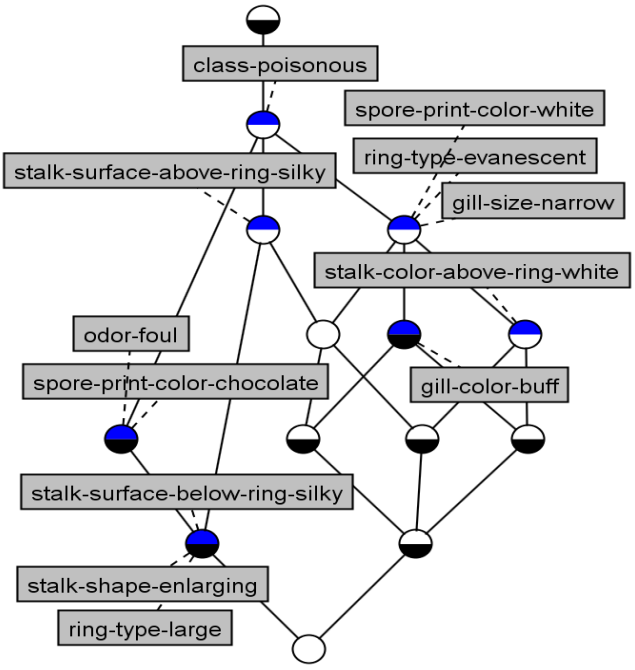

Figure 6. Poisonous Mushroom Concept Lattice.

In each case, the resulting Concepts involved most of the mushrooms in the corresponding 'noisy' sub-Context (3334 out of 3916 poisonous mushrooms and 2848 out of 4208 edible mushrooms). The lattices in ConExp were further simplified by hiding attributes that were commonly

supported in both (the purpose here was to highlight the difference between the sub-Contexts, not the similarities). For example, *veil-type-partial* and *ring-number-one* were present in both edible and poisonous Concepts and were removed. This process led to the edible Concept lattice being reduced to ten Concepts. Differences between the two sub-Contexts were now clear. For example, a smooth stalk would seem to indicate that a mushroom is good to eat, whereas those with silky stalks should be avoided. Those with a combination of white spores, an evanescent ring and narrow gills are probably dangerous; safer to try mushrooms with broader gills. Less surprising is the fact that foul smelling mushrooms should probably be left alone; those with no smell are safer. Noting that the number of objects (mushrooms) involved in a Concept decreases as one navigates downwards in a lattice, having a pendant ring is probably only a corroboratory factor in deciding on the wholesomeness of a mushroom. The significance of a mushroom having bruises is interesting, perhaps indicating that edible mushrooms are more likely to show damage from foraging animals.

VII. CUBIST

The further development of this work will form a core part of CUBIST ("Combining and Uniting Business Intelligence with Semantic Technologies"), a research project awarded under the European Unions 7th Framework Programme, 5th ICT call, topic 4.3: Intelligent Information Management; STREP Project No.: FP7 257403. CUBIST aims to develop an approach for Business Intelligence that augments Semantic Technologies with BI capabilities and provides conceptually relevant and user-friendly FCA-based visual analytics. CUBIST will find applications within the Semantic Web and specifically the use of RDF. CUBIST aims to deliver high performance in-warehouse interactive visual analytics for information warehouses and triple stores.

A. Semantic Web and RDF

In the development of the Semantic Web, the use of the RDF schema and triples is proposed, rather than using traditional XML [6]. In XML, the same information can be represented using various structures that all have the same meaning to a person reading them [6], [11], [14]. However, each document can produce different XML trees when parsed by a machine [6]. These are problems which the RDF schema tries to resolve, as RDF gives some standard ways of writing statements so that however it occurs in a document, the same effects can be produced in RDF terms.

FcaBedrock is being developed to accept RDF/XML files as input. RDF uses the *subject-predicate-object* logic, which is the same logic used for the 3-column CSV format currently supported by FcaBedrock. Functionality is to be added for deriving data encoded in RDF vocabularies such as *Friend of a Friend*[6] *(FOAF)* and in authoring ontologies languages such as *OWL*. By using FcaBedrock as a semantic data preparation tool, FCA can find further applications in the Semantic Web and make knowledge representation, information management and visualizing conceptual structures among semantic data possible.

B. Triple Stores

Triple Stores are column-oriented data warehouses for storing and retrieving RDF metadata using query languages for RDF, such as the *SPARQL Protocol and RDF Query Language*[7]. There is a growth of interest in industry because, using sophisticated indexing techniques, high performance data analysis is possible over billions of triples [15], [17]. As part of CUBIST, FcaBedrock and In-Close will be developed to capture and process disparate and distributed data and be integrated into such triple stores, with the objective of providing ways to visualise and explore hitherto undiscovered BI using FCA (Figure 7).

VIII. CONCLUSION

Using freely available software tools it is possible to visualise hidden meaning in data. FCA is shown to be applicable to large-scale data. As well as the analysis of the Mushroom data set presented here, useful analysis (not presented here) has been carried out on the *Adult* and *Internet Advertisement* data sets from UCI [4] and an internal set of student-related data at Sheffield Hallam University. FcaBedrock is to be presented at the 19th International Conference on Conceptual Structures [3], ICCS 2010, and work to further demonstrate the applicability of the techniques described here is being prepared for other venues, such as the International Conference on Concept Lattices and Their Applications. Further work is required to integrate the processes described here, and to provide intuitive and responsive interfaces to them. Interest at a European level has been demonstrated by the funding of CUBIST, in which this further work can be carried out. CUBIST is bringing together European data warehousing companies, universities with expertise in FCA and commercial use-case partners to develop powerful, insightful and intuitive RDF-based FCA Visual Analytics for BI. Disparate data will come from a range of structured and unstructured sources, providing rich and complex challenges for CUBIST to provide collective intelligence through the use of FCA as an emerging data technology.

REFERENCES

[1] Andrews, S. (2009). *In-Close, A Fast Algorithm for Computing Formal Concepts*. Available: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-483/paper1.pdf. Last accessed 06 May 2010.

[6] http://www.foaf-project.org/

[7] http://www.w3.org/TR/rdf-sparql-query/

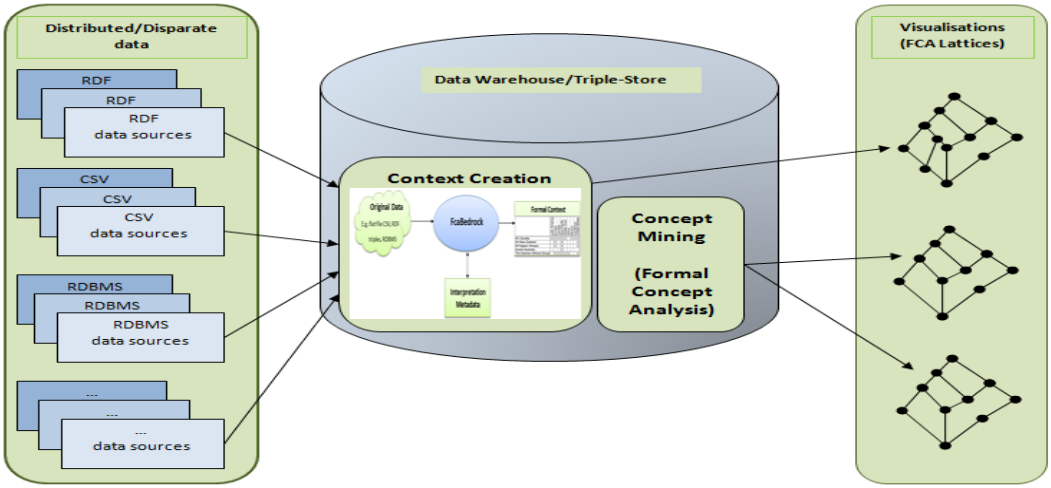

Figure 7. Incorporation of FcaBedrock and In-Close into BI-enabled Triple Stores.

[2] Andrews, S. (2009). *Data conversion and interoperability for FCA*. In: *Conceptual Structures Tools Interoperability Workshop*, ICCS 2009: Moscow.

[3] Andrews, S. and Orphanides, C. (2010). *FcaBedrock, a Formal Context Creator*. Submitted to: Croitoru, M., Ferre, S. and Lukose, D. (eds.) *ICCS 2010*, [www.iccs.info] (accepted paper).

[4] Asuncion, A. and Newman, D.J. (2007). *UCI Machine Learning Repository* [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.

[5] Becker, P. and Correia, J.H. (2005). *The ToscanaJ Suite for Implementing Conceptual Information Systems*. In *Formal Concept Analysis*, LNCS, Vol. 3626, pp. 324-348, Springer Berlin / Heidelberg.

[6] Berners-Lee, T. (1998). *Why RDF model is different from the XML model*. Available: http://www.w3.org/DesignIssues/RDF-XML. Last accessed 10 May 2010.

[7] Imberman, S. and Domanski, B. (1999). *Finding Association Rules from Quantitative Data using Data Booleanization*. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.14.4447&rep=rep1&type=pdf. Last accessed 11 May 2010.

[8] Jin, R. Breitbart, Y. and Muoh, C. (2009). *Data discretization unification*. In: *Knowledge and Information Systems*. 19(1), pp. 1-29.

[9] Krajca, P., Outrata, J., Vychodil, V. (2008) *Parallel Recursive Algorithm for FCA*. In: Belohlavek, R., Kuznetsov, S.O. (eds.), *Proceeding of the Sixth International Conference on Concept Lattices and their Applications*, pp. 71-82, Palacky University, Olomouc.

[10] Kaytoue-Uberall, M., Duplesssis S. and Napoli, A. (2008). *Using Formal Concept Analysis for the Extraction of Groups of Co-expressed Genes*. In: Le Thi, H.A., Bouvry, P., Pham Dinh, T. (eds.) *MCO 2008*. CCIS vol. 14, pp. 439-449. Springer-Verlag, Berlin/Heidelberg.

[11] Passin, T. B. (2004). *Explorers Guide to the Semantic Web*. Manning, Greenwich: CT.

[12] Priss, U. (2008). *Formal Concept Analysis in Information Science*. In: Cronin, B. (ed.). *Annual Review of Information Science and Technology*, ASIST, vol. 40.

[13] Priss, U. (2008) *FcaStone - FCA File Format and Interoperability Software.* In: Croitoru, M., Jaschkë, R., Rudolph, S. (eds.), Conceptual Structures and the Web, *Proceedings of the Third Conceptual Structures and Tool Interoperability Workshop*, pp. 33-43.

[14] Semantic Web. (2010). *The Semantic Web*. Available: http://semanticweb.org/wiki/Main_Page. Last accessed 28 Apr 2010.

[15] Slezak, D., Wroblewski, J., Eastwood, V. and Synak, P. (2008). *Brighthouse: an analytic data warehouse for ad-hoc queries*. In: *Proceedings of the VLDB Endowment*. vol. 1(2), pp. 1337-1345. ACM Digital Library.

[16] World Wide Web Consortium. (2010). *Design Issues*. Available: http://www.w3.org/DesignIssues/. Last accessed 08 May 2010.

[17] White, P.W. and French, C.D. (1998). *Database system with methodology for storing a database table by vertically partitioning all columns of the table*. US Patent 5,794,229, August 11, 1998.

[18] Wille, R. (2005). *Formal Concept Analysis as Mathematical Theory of Concepts*. In: Ganter, B., Stumme, G. and Wille, R. (eds.) *Formal Concept Analysis: Foundations and Applications*. Berlin: Springer. pp. 1-6.

[19] Wolff, K.E. (1993). *A First Course in Formal Concept Analysis*. Available: http://www.fbmn.h-da.de/home/wolff/Publikationen/A_First_Course_in_Formal_Concept_Analysis.pdf. Last accessed 03 May 2010.