cubist

Your Business Intelligence

The 2nd CUBIST Workshop

CUBIST-WS-12

Simon Andrews, Frithjof Dau (Eds.)

Simon Andrews, Frithjof Dau (Eds.):

The 2nd CUBIST Workshop

CUBIST-WS-12

Preface

This volume contains the papers accepted to the second CUBIST workshop.

CUBIST (Combining and Uniting Business Intelligence with Semantic Technologies) is a research project funded by the European Commission under the 7th Framework Programme of ICT, topic 4.3: Intelligent Information Management. The project started in October 2010. CUBIST follows a best-of-breed approach that combines essential features of Semantic Technologies, Business Intelligence and Visual Analytics. CUBIST aims to

- persist federated data in a semantic Data Warehouse; a hybrid approach based on a BI enabled triple store,
- and to provide novel ways of applying visual analytics in which meaningful diagrammatic representations based on Formal Concept Analysis will be used for depicting the data, navigating through the data and for visually querying the data.

As one can see from this description, CUBIST requires expertise from a variety of research fields, which cannot be provided by a single research organization. For this reason, the second CUBIST workshop which addresses the main topics of CUBIST has been set up. The workshop aims at providing a forum for both research and practice for CUBIST-related topics and technologies in order to facilitate interdisciplinary discussions. The first CUBIST workshop was held in conjunction with the 19th International Conference on Conceptual Structures (ICCS 2011), which was held on 25 - 29 July 2011 at the University of Derby, United Kingdom, and attracted submissions from outside the CUBIST consortium. The second workshop was held in conjunction with the 10th International Conference on Formal Concept Analysis (ICFCA 2012) which was held on 6 – 10 May 2012 at the University of Leuven, Belgium. We are proud that again the workshop received and accepted submissions from outside the CUBIST consortium, which indicates that the project and the workshop are addressing contemporary topics of interest to researchers in the fields. In total we had six submissions, three with some CUBIST members as authors, and five of the submissions were accepted.

We, the chairs, want to express our appreciation to all authors of submitted papers and to the program committee members for their work and valuable comments.

April 2012, Simon Andrews and Frithjof Dau

CUBIST-WS-12 Organization

Chairs

Simon Andrews (Sheffield Hallam University, UK)

Frithjof Dau (SAP AG, Germany)

Program Committee

Cassio Melo (Centrale Recherche S.A. (CRSA) - Laboratoire MAS, France)

Constantinos Orphanides (Sheffield Hallam University, UK)

Emre Sevinc (Space Applications Services NV, Belgium)

Kenneth McLeod (Heriot-Watt University, UK)

Marie-Aude Aufaure (Centrale Recherche S.A. (CRSA) - Laboratoire MAS, France)

Simon Polovina (Sheffield Hallam University, UK)

Yuri Kudryavcev (PMSquare, Australia)

Ivan Launders (BT Innovation & Design, UK)

Martin Watmough (CIBER, UK)

Table of Contents

Network Reduction Based on Structural Equivalence of Nodes

Alexey Lakhno and Andrey Chepovskiy

Higher School of Economics,
Data Analysis and Artificial Intelligence Department,
Pokrovskiy boulevard 11, 109028 Moscow, Russia
alakhno@gmail.com, achepovskiy@hse.ru

Abstract. Network reduction is considered in application to visual analytics of relational data. We construct a reduced network by merging structurally equivalent nodes. This network specifies the initial structure of relations in a more compact way without any loss of information. The analysis of such reduction is presented in application to the communication network from Stanford Large Network Dataset Collection. It is shown how the reduction based on structural equivalence can help in visualization of large networks.

Keywords: network analysis, network reduction, structural equivalence, relations visualization

1 Introduction

Networks provide a natural representation of information about relations between objects. We consider networks as attributed graphs. The nodes of these graphs correspond to the objects of a knowledge domain, while the edges can be treated as the relations between them. Social, communication and transaction networks are the examples of networks to deal with in real analytical problems. However large size of these networks can be an obstacle for their analysis. Network reduction addresses this problem.

Structural equivalence as a concept was introduced by Lorrain and White [1] who proposed that individuals in a social network are role equivalent if they have the same neighborhood. As it is mentioned in [2] there is no loss of structural information by combining structurally equivalent nodes into a single subset and representing them as a single structural entity. This approach can be considered as a clusterization technique. Let's call it *se*-reduction as an abbreviation from structural equivalence reduction. It is straightforward to construct the network of any size that does not contain structurally equivalent nodes and thus is absolutely irreducible using the above technique (Sect. 2). However the experiments with the real networks representing social and communication relations showed the significant level of underlying graph reduction. This fact allows to use the *se*-reduction in visual analytics as a clutter reduction technique for visualization of semantic networks. In this paper we present the algorithm that can be used for *se*-reduction and discuss the results of experiments with EU email communication network from Stanford Large Network Dataset Collection [3].

Visual representation is extremely useful for exploring and analyzing networks. A solid survey on graph drawing can be found in [4, 5]. Some layout approaches such as force-directed or layered methods are general purpose while others are intended for visualization of specific subgraphs. Multistripe layout suitable for visualization of relations incident to the selected set of nodes provides an example of the second approach [6]. Anyway, large amount of nodes and links result in poorly readable drawings because of link crossings and common visual overload. Using the *se*-reduction as a preprocessing step before layout construction we can get significant clutter reduction preserving full information about the initial relational structure. There is a number of clutter reduction techniques addressing the problem of large datasets in information visualization [7]. Paper [8] considers drawing graphs using modular decomposition. Merging the nodes of a module has better reduction potential in comparison to *se*-reduction. However it does not preserve the information about edges inside the module. The other clutter reduction technique used in graph drawing is the edge bundling [9, 10]. It can be applied further to *se*-reduction in order to improve the readability of drawing.

The rest of the paper is organized in the following way. Section 2 provides formal definitions necessary for further understanding. Section 3 presents an algorithm used for *se*-reduction. Section 4 discusses the result of *se*-reduction applied to EU email communication network. It provides the statistics revealing the structure of *se*-reduced graph and answering the question "what fragments are actually reduced". Section 5 illustrates the visualization of network fragments using the *se*-reduction technique. Finally, we summarize and conclude our work in Sect. 6.

2 Formal definitions

Let $G = \langle V, E \rangle$ be an underlying graph of a network. Here we consider an undirected graph without selfloops and multiple edges. Such a graph can be obtained from a general network by merging multiple edges, removing selfloops and ignoring edge orientation. Here V is a set of nodes and E is a set of edges. If a node $v \in V$ then $N(v) = \{u \in V | (v, u) \in E\}$ denotes a *neighborhood* of v.

2.1 *se*-reduced graph

Definition 1. *Two nodes u, v of a graph* $G = \langle V, E \rangle$ *are called structurally equivalent* $u \sim_{se} v$ *if and only if they have the same neighborhood* $N(u) = N(v)$*, that is* $\forall w \in V$ *holds* $(v, w) \in E \Leftrightarrow (u, w) \in E$.

Clearly $\sim_{se} \subseteq V \times V$ is an equivalence relation and specifies the partition of set V into equivalence classes. An equivalence class of $v \in V$ is defined as $[v] = \{u \in V \mid u \sim_{se} v\}$ and the set of all possible equivalence classes is given by the quotient set $V/\sim_{se} = \{[v] \mid v \in V\}$. An equivalence class is called *trivial* if it contains only one node.

Proposition 1. *In a graph $G = \langle V, E\rangle$ that does not contain selfloops there can't be any edges connecting the nodes from one equivalence class, that is $\forall u, v \in V$ holds $u \sim_{se} v \Rightarrow (u, v) \notin E$.*

Proof. Suppose $\exists u, v$ such that $u \sim_{se} v$ and $(u, v) \in E$. This means $u \in N(v)$. But $(u, u) \notin E$ as there is no selfloops in G. So $u \notin N(u)$ and $N(u) \neq N(v)$. Finally we came to a contradiction with the fact $u \sim_{se} v$. □

Proposition 2. *If there is an edge $(u, v) \in E$ in a graph $G = \langle V, E\rangle$ then every node u' from $[u]$ is connected to every node v' from $[v]$, that is $\forall u' \in [u], v' \in [v]$ holds $(u', v') \in E$.*

Proof. Suppose $\exists u' \in [u]$ and $v' \in [v]$ such that $(u', v') \notin E$. The fact $u' \in [u]$ means $N(u') = N(u)$. Similarly $v' \in [v] \Rightarrow N(v') = N(v)$. It is known that $u \in N(v)$ as $(u, v) \in E$. Thus $u \in N(v')$ and $(u, v') \in E$. This means $v' \in N(u)$ and $v' \in N(u')$. So we came to a contradiction with the fact $(u', v') \notin E$. □

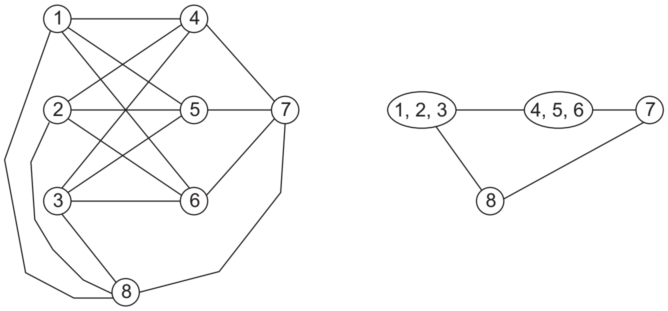


Fig. 1. Initial graph G and corresponding *se*-reduced graph $R(G)$: G contains 8 nodes and 16 edges while $R(G)$ contains 4 nodes and 4 edges. The edges of $R(G)$ mean following four facts: each node from the set $\{1, 2, 3\}$ is connected to each node from the set $\{4, 5, 6\}$, each node from the set $\{4, 5, 6\}$ is connected to node 7, each node from the set $\{1, 2, 3\}$ is connected to node 8, nodes 7 and 8 are connected.

Definition 2. *For a given graph $G = \langle V, E\rangle$ an se-reduced graph is defined as $R(G) = \langle R_V(G), R_E(G)\rangle$ where*

$$R_V(G) = V/\sim_{se} = \{[v] \mid v \in V\},$$

$$R_E(G) = \{([u], [v]) \mid (u, v) \in E\}.$$

Thanks to Proposition 2 $(u, v) \in E \Rightarrow \{(u', v')|u' \in [u], v' \in [v]\} \subseteq E$. So every edge $([u], [v]) \in R_E(G)$ corresponds to the set $\{(u', v')|u' \in [u], v' \in [v]\}$ of edges in the initial graph G.

Fig. 1 shows an example of *se*-reduction. Every node from $R_V(G)$ corresponds to a group of structurally equivalent nodes of G while every edge $([u],[v]) \in R_E(G)$ means that every node in G from $[u]$ equivalence class is connected to every node from $[v]$ equivalence class. Obviously $|R_V(G)| \leq |V|$ and $|R_E(G)| \leq |E|$. We use two basic characteristics for evaluation of graph reduction efficiency: node reduction ratio $r_V(G) = 1 - |R_V(G)|/|V|$ and edge reduction ratio $r_E(G) = 1 - |R_E(G)|/|E|$. Reduction ratios can take on values in range $[0, 1)$. The higher reduction ratios are the more efficient graph reduction is. Zero values of reduction ratios mean no reduction.

Example 1. Graph $G = \langle V, E \rangle$ where $V = \{1, \ldots, n\}$, $E = \{(i, i+1) | i = 1..n\}$ gives an example of absolutely irreducible graph as it does not contain structurally equivalent nodes. As $|R_V(G)| = |V|$ and $|R_E(G)| = |E|$ it follows that $r_V(G) = 0$ and $r_E(G) = 0$.

Example 2. For any graph G the *se*-reduced graph $R(G)$ is absolutely irreducible. By construction $R(G)$ does not contain any *se*-equivalent nodes. So the *se*-reduction does not make sense being applied iteratively.

2.2 Neighborhood subgraph

Assume we have a selected set of nodes in a network. Neighborhood subgraph concept gives a natural way of extracting subnetwork concerned with this set. If $u, v \in V$ then $d(u, v)$ denotes the number of edges in the shortest path between u and v. In case of no path between u and v assume $d(u, v) = \infty$.

Definition 3. *For a given set of nodes $V' \subset V$ neighborhood subgraph of radius r is such a graph $O(G, V', r) = \langle O_V(G, V', r), O_E(G, V', r) \rangle$ that*

$$O_V(G, V', r) = \{v \in V | \exists u \in V', d(u, v) \leq r\},$$

$$O_E(G, V', r) = \{(u, v) \in E | u, v \in O_V(G, V', r)\}.$$

3 Reduction algorithm

Assume the initial graph $G = \langle V, E \rangle$ is specified with a list of edges and its nodes are numbered from 1 to $|V|$. The following algorithm is used in order to construct the *se*-reduced graph $R(G)$:

1. Form adjacency lists for the nodes of G. Nodes in the lists should be ordered by their identifiers.
2. For every node $v \in V$ calculate a hash function of its adjacency list. The calculated values will be used for a preliminary comparison of node neighborhoods $N(v)$.
3. For every node $v \in V$ define whether it should be distributed to a new equivalence class or added to one of the equivalence classes of processed nodes. In order to make a decision we look up all the classes with the same value of hash function.

4. When the equivalence classes are formed we look through the edges of the initial graph G in order to construct the edges of $R(G)$ according to its definition $R_E(G) = \{([u], [v]) \mid (u, v) \in E\}$.

Stage 1 can be performed with $O(|E| \log |V|)$ time complexity. Stages 2 and 4 require $O(|E|)$ time as every edge of G can be processed in $O(1)$. Stage 3 time complexity significantly depends on the hash function distribution. However if the number of equivalence classes with coincident values of hash function is limited with a constant, we get $O(\sum_{[v] \in R_V(G)} |[v]|^2)$ time complexity. The following inequality gives an upper estimate which is more straightforward for the interpretation: $\sum_{[v] \in R_V(G)} |[v]|^2 \leq (\sum_{[v] \in R_V(G)} |[v]|)^2 = |V|^2$. Thus total time complexity of the algorithm is $O(|E| log |V| + |V|^2)$. Actually this value is overrated and the algorithm can be used to process the networks containing several million nodes and edges in about several seconds.

4 Reduction experiment

Current section presents the result of *se*-reduction applied to EU email communication network [3]. The network was generated using email data from a large European research institution. The data was collected and anonymized for a period of 18 months and covers a dataset of around three million emails. Nodes of the underlying graph correspond to email addresses and edges mean the fact of communication, that is one or more email letters.

After removing selfloops and merging parallel edges of opposite direction we got a graph $G = \langle V, E \rangle$ containing 265214 vertices and 364481 undirected edges. Corresponding *se*-reduced graph $R(G) = \langle R_V(G), R_E(G) \rangle$ contains 56347 nodes and 138763 edges. This means around 79% node reduction ratio and 62% edge reduction ratio. There occurred 6787 equivalence classes containing two or more nodes. The biggest equivalence class contains 7633 nodes. Figure 2 shows the distribution of the equivalence class sizes. Figure 3 shows the nodes distribution among the classes according to their sizes. It turns out there are 22 equivalence classes of size 1000 or more that contain in sum 54061 nodes (about 20% of all nodes). Figures 4 and 5 contain dot charts revealing typical structure of the reduced fragments. The first chart shows that there are no edges in the reduced graph connecting equivalence classes of size more than two as there are no dots (x, y) for which $x > 2$ and $y > 2$ simultaneously. However the second chart proves the existence of nontrivial equivalence classes connected with up to 285 single nodes. Besides there occurred an equivalence class containing 22 nodes that are connected with 7 other nodes each. The fragment is specified with 8 nodes and 7 edges in the reduced graph $R(G)$ instead of 29 nodes and 154 edges in the original graph G.

5 *se*-based network visualization

Figure 6 provides an example of *se*-based network fragment visualization. The fragment was extracted from EU communication network as a neighborhood sub-

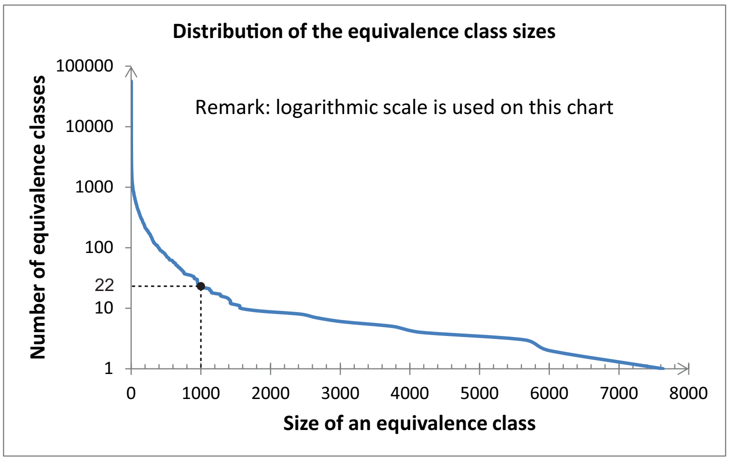


Fig. 2. The curve shows the number of equivalence classes containing more than or equal to a specific number of nodes. For example there are 22 equivalence classes of size 1000 or more.

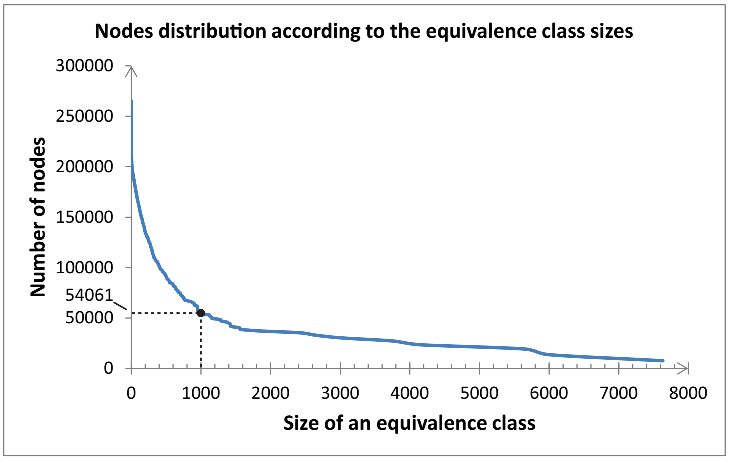


Fig. 3. The curve shows the number of nodes in equivalence classes containing more than or equal to a specific number of nodes. For example there are totally 54061 nodes in the equivalence classes of size 1000 or more.

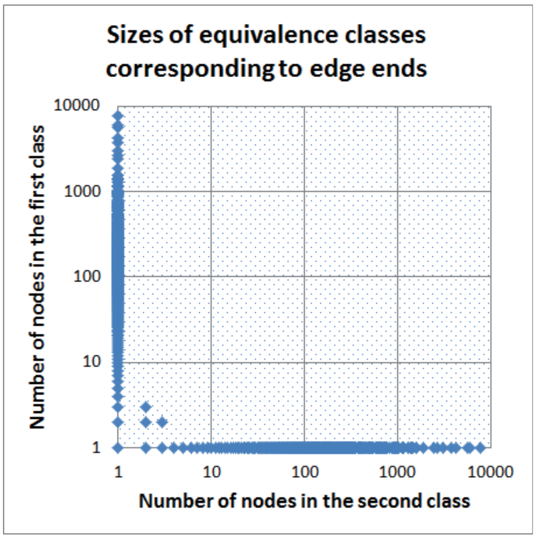


Fig. 4. Each dot corresponds to an edge of the *se*-reduced graph $R(G)$: x and y coordinates mean the sizes of equivalence classes corresponding to the ends of the edge. The chart is symmetric across the line $y = x$ as the edges have no direction.

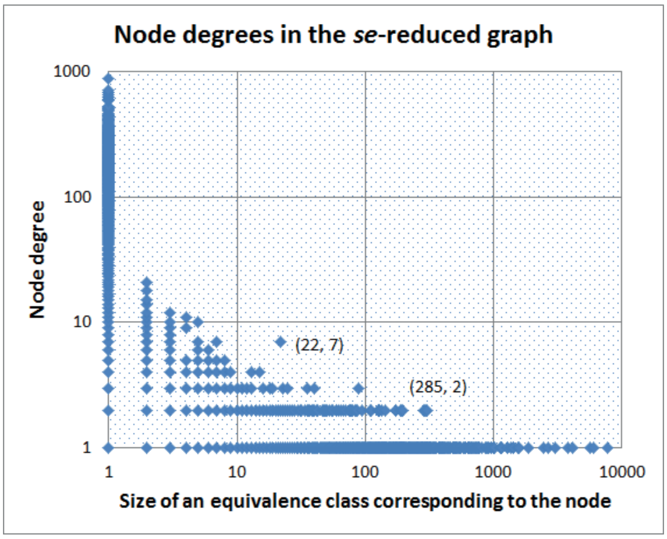


Fig. 5. Each dot corresponds to a node of the *se*-reduced graph $R(G)$: x-coordinate means the size of an equivalent class corresponding to the node while y-coordinate means the number of incident edges in the *se*-reduced graph. For instance, dot $(22, 7)$ corresponds to the class containing 22 equivalent nodes with 7 incident edges each.

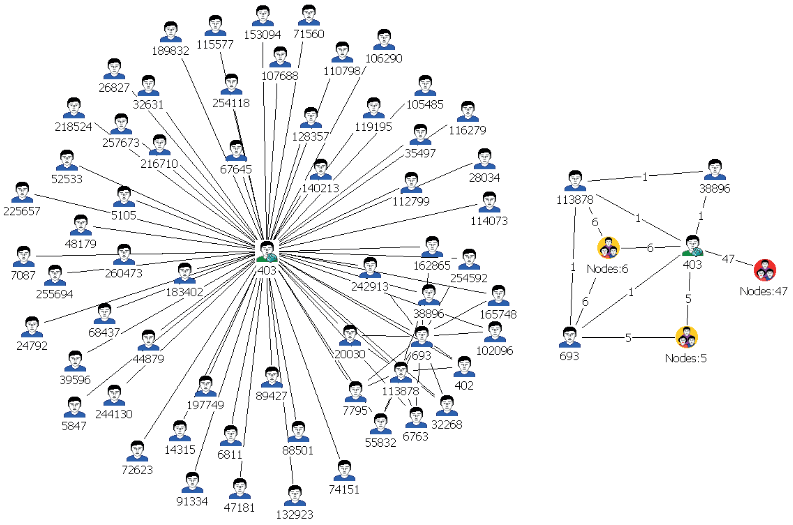


Fig. 6. Neighborhood subnetwork $G' = O(G, \{403\}, 1)$ extracted from EU communication network and the result of its se-reduction. Numbers on the edges of $R(G)$ mean the numbers of merged edges of the initial subnetwork.

graph $G' = O(G, \{403\}, 1)$. It contains 62 nodes and 80 edges. Corresponding se-reduced graph contains 7 nodes and 11 edges. There occurred three nontrivial equivalence classes in $R(G')$ containing 5, 6 and 47 nodes respectively. The drawing of $R(G)$ provides the visual representation for G that avoids visual overload merging repeated fragments of relational structure.

6 Conclusion

Complex structure and large size of real semantic networks make difficulties for construction of clear visual representations. Network reduction based on structural equivalence of nodes is considered as a technique addressing this problem. We presented the algorithm of se-reduction with $O(|E|log|V| + |V|^2)$ time complexity and provided the analysis of se-reduction applied to EU communication network that proves the reduction efficiency and reveals the structure of reduced fragments. Finally the approach to visualization of networks was illustrated.

References

1. Lorrain, F., White, H.C.: Structural equivalence of individuals in social networks. Journal of Mathematical Sociology, Vol. 1, pp. 49–80, 1971.
2. Wasserman, S., Faust, K., Social Network Analysis. Cambridge: Cambridge University Press, 1994.

3. EU email communication network dataset from Stanford Large Network Dataset Colection, `http://snap.stanford.edu/data/email-EuAll.html`
4. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall, 1999.
5. Kaufmann, M., Wagner, D. (eds.): Drawing Graphs: Methods and Models. Springer-Verlag, 2001.
6. Lakhno, A.P., Chepovskiy A.M.: Visualization of Semantic Network Fragments Using Multistripe Layout. Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, Lecture Notes in Artificial Intelligence, Vol. 6743, Heidelberg: Springer, 2011.
7. Ellis, G., Dix, A.: A Taxonomy of Clutter Reduction for Information Visualisation. Visualization and Computer Graphics, IEEE Transactions on, Vol. 13, No. 6, pp. 1216–1223, 2007.
8. Papadopoulos, C., Voglis, C.: Drawing Graphs Using Modular Decomposition. Journal of Graph Algorithms and Applications, Vol. 11, No. 2, pp. 481–511, 2007.
9. Ganser, E., Koren, Y.: Improved Circular Layouts. In: Kaufmann, M., Wagner, D. (eds) GD 2006. LNCS, Vol. 4372, pp. 386–398, Heidelberg: Springer, 2007.
10. Pupyrev, S., Nachmanson, L., Kaufmann, M.: Improving Layered Graph Layouts with Edge Bundling. In: Brandes, U., Cornelsen, S. (eds.) GD 2010. LNCS, Vol. 6502, pp. 329–340, Hidelberg: Springer, 2011.

Semantic Web Spatial Representation of Biomedical Images

D.N.F Awang Iskandar[1,3], Kenneth M^{c}Leod[1], and Albert Burger[1,2]

[1] Dept. of Computer Science, Heriot-Watt University, Edinburgh, UK
[2] MRC Human Genetics Unit, Edinburgh, UK
[3] FCSIT Universiti Malaysia Sarawak, Malaysia
dnfaiz@fit.unimas.my
kenneth.mcleod@hw.ac.uk
ab@macs.hw.ac.uk
http://www.macs.hw.ac.uk/bisel

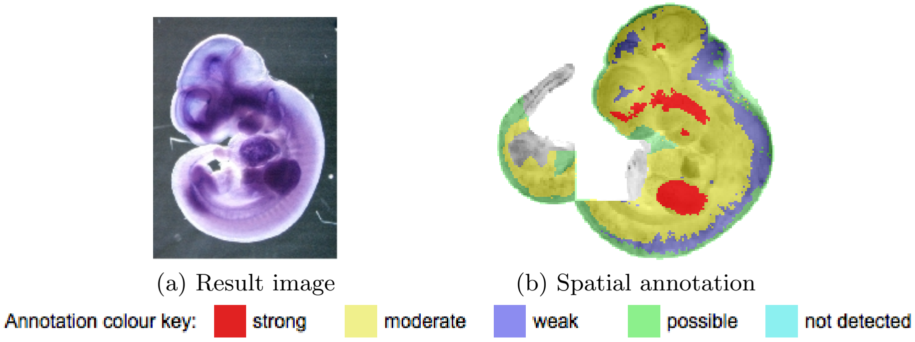
Abstract. Biomedical images contain vast amounts of information. Regrettably, much of this information is only accessible by domain experts. This paper looks at one particular use case in which this scenario occurs. Motivation for the use of a semantic representation of images is given. Subsequently a benchmark representation is developed, and an exploration of an existing geospatial technology undertaken. The benchmark is lossless - and thus too expensive. The geospatial technique makes assumptions that are not valid. Accordingly, the need for a new *biospatial* representation is demonstrated.

Keywords: biomedical, geospatial, image processing, semantic web

1 Introduction

Semantic Web technologies, applications and tools have facilitated great steps forward in the life science and health care data exchange. However, developing appropriate semantic representations, including designing spatio-temporal ontologies, remains difficult and challenging [1]. Activities described in this paper centre on the semantic description of biomedical images. This paper describes a biological use case in order to motivate this research before documenting the initial undertakings.

Gene expression information allows biologists to discover relationships between genes, in particular when genes are active in the same location. This co-expression information provides insights into the ways in which relationships between genes affect the development of a tissue. A gene is a unit of instructions that provides directions for one essential task, i.e., the creation of a protein. Gene expression information describes whether or not a gene is expressed (active) in a location. Broadly speaking there are two types of gene expression information: those that focus on where the gene is expressed, and those whose primary concern is the strength of expression. This work concentrates on the former category, and in particular a technology called *in situ* hybridisation gene expression.



(a) Result image

(b) Spatial annotation

Fig. 1: A sample image of the mouse embryo at Theiler Stage 18 from the EMAGE database

Although this work is applicable to a wide range of biomedical applications and domains, only one biological use case has been considered to date — *in situ* gene expression for the developmental mouse, which is available from the Edinburgh Mouse Atlas of Gene Expression (EMAGE) [2] database.

Information on gene expression is often given in relation to a tissue in a particular model organism. Here the model organism is the mouse. This organism is studied from conception until adulthood. The time window is split into 28 Theiler Stages (TS). Stages 1 to 26 form the unborn, or developmental, mouse. Each stage has its own anatomy, and corresponding anatomy ontology called EMA [3]. Moreover, EMA contains a number of 3D models representing different stages of the developmental mouse.

The result of an *in situ* experiment is documented as an image displaying an area of a mouse (from a particular TS) in which some subsections of the mouse are highly coloured as depicted in Fig. 1(a). Areas of colour indicate that the gene is expressed in that location. Furthermore, the image provides some indication of the level (strength) of expression: the more intense the colour, the stronger the expression. Results are analysed manually under a microscope. A human expert determines in which tissues the gene is expressed, and at what level of expression. As volume information is not the main focus of the experiment, its description uses vague natural language terms such as strong, moderate, weak or present. For example, the gene *bmp4* is strongly expressed in the future brain from TS15. This is a so-called textual annotation. Additionally, result images can be mapped onto 3D models of the mouse creating so-called spatial annotations depicted in Fig. 1(b). The spatial annotations are normally generated by EMAGE, whilst the textual annotations are produced by the researchers who performed the experiment.

The next section utilises the use case to explain the problem focused upon in more depth. Subsequently, Section 3 contemplates the research vision before Section 4 examines related work. Section 5 presents the initial study taken to create a spatial representation for the use case images. A discussion on semantic web spatial representation for biomedical images in Section 6 concludes the paper.

2 Problem and Motivation

EMAGE contains a large volume of information within its gene expression images (both the original experimental result, and the spatial annotations). Unless the viewer is an expert anatomist, they are unable to interpret and understand this information. Theoretically, a non-expert could consult the textual annotations; however, these annotations have far less granularity and are often incomplete. For example, whilst the images display exactly where the gene is expressed, the textual annotations often abstract the location to the overarching structure or region. Thus, rather than saying the gene is expressed in the venous valve of the right atrium, the annotation might say the gene is expressed in the heart. Furthermore, if an experimenter is only interested in the heart, (s)he will only provide textual annotations for the heart, ignoring all the potential annotations that could be created for other regions.

Whilst human experts are able to deal with the above issues, non-experts and computers are not. Using the raw images makes more knowledge available, and provides more precision. Thus all work based on these images becomes more reliable and more relevant. Likewise, if the full content of the images is available it becomes possible to integrate that information more readily with information from other domains. For example, it becomes possible to start organising, comparing, and querying *all* the gene expression information in EMAGE by gene family, or by pathway. Thus enabling the discovery of new knowledge and relationships. To illustrate, Andrews [4] demonstrates the potential worth of applying one data mining technique (formal concept analysis) to EMAGE's textual annotations. Yet, these annotations represent a tiny percentage of the information available within EMAGE. If all information were available, data mining techniques would be both essential for navigating the vast dataset, and crucial for knowledge discovery within it.

Even though this paper focuses on a biological use case it should be clear that is equally applicable to the medical domain where countless MRI, CT and PET scans are taken each day. These images are analysed by experts who produce textual summaries that are then stored alongside the original images. Much like the biological images, these medical images contain large volumes of potentially useful information that is currently unavailable to non-expert viewers unless they have extensive medical training.

Rapid improvements in technology means that the cost of experimentation is steadily decreasing, whilst the quality of the resultant images is increasing. Accordingly, the number of images available, and the size of those images, is growing. This expansion amplifies the need to make this information more readily accessible.

A researcher wants to integrate as much information from as many images as possible. Yet, these images are often distributed across the globe. For example,

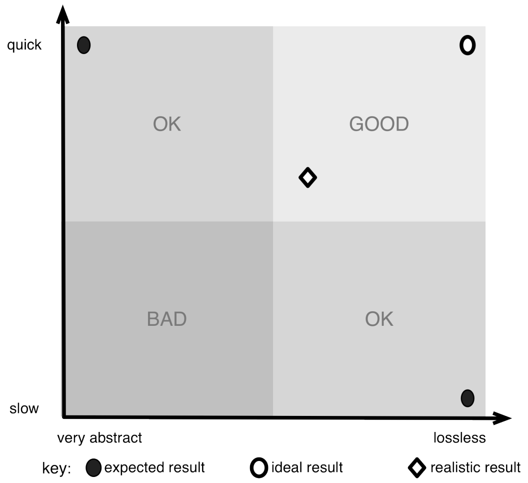


Fig. 2: Visual summary of the possible outcomes of this research. Clearly, the ideal result is unlikely and thus care must be chosen to select the best balance.

EMAGE is based in Edinburgh, whilst a complementary resource called the Allen Brain Atlas [5] is located in Seattle. Both of these resources contain hundreds of thousands of images equating to tens of TB of data. Aggregating these images for study and analysis is therefore something of a challenge. A side benefit of creating semantic descriptions of images is that these descriptions are likely to be smaller than the images themselves, and thus cheaper to share internationally with other researchers.

The above discussion states the case for making the information contained within biomedical images more easily accessible. Yet, this is only a partial solution to the surrounding problem. It is necessary to not only make the information available, but to facilitate the efficient use of that information. Biology is increasingly an information centric science [6], as such the information must be readily available for computational access and analysis.

3 Research Vision

The long term vision of this work is to find an appropriate semantic description of the spatial information recorded within biological images held as part of the EMAGE resource.

Ideally it will be possible to create a representation of the images that contains every single piece of information the image does, and yet does not require significant processing thereby incurring significant computational cost. Intuitively this seems unrealistic. If the representation is lossless, the physical size (in MB) of the representation will be more than for an abstract representation.

Whilst individually these differences may be insignificant, when considering a library of hundreds of thousands of images, the cumulative effect is likely to have a significant impact on performance. Accordingly, as Fig. 2 summarises, there is a need to chose an appropriate trade-off between performance and the level of detail captured and represented.

Medicine is a safety critical domain, mistakes may result in patients dying. Within our chosen use case, this is not true. This quick comparison demonstrates that the requirements for precision and accuracy are likely to differ from one application to another. Accordingly, the trade-off needs to be specific to each use case. Initially, this work shall focus exclusively on the chosen use case.

The notion of "semantic" is technology neutral and one of the popular technology platforms is the Resource Description Framework (RDF). The remainder of this paper will consider the first steps in the creation of a RDF-based spatial representation of biomedical image data.

Before starting it is necessary to first consider related work, in particular with a view to adapting existing technology rather than 'reinventing the wheel'.

4 Related Work

Spatio-temporal ontology representations have been studied to represent data in research related to semantic image retrieval, Geographic Information Systems (GIS) and medical atlases. Hudelot *et al.* (2008) [7] used a fuzzy spatial relation ontology to represent the knowledge-based recognition of brain structures in 3D magnetic resonance images. On the other hand, Mechouche *et al.* (2008) [8] used specific spatial relations to describe orientations between the different patches and sulcus segments in MRI images. To retrieve specific images based on the spatial location of objects in an image, Awang Iskandar (2009) [9], [10] calculated the distance of queried objects using SPARQL.

In GIS, the spatio-temporal representation is known as geospatial — defined as "of or relating to the relative position of things on the earth's surface" [11]. The geospatial semantics research domain emerges from the combination of GIS and semantic interoperability that is popularised by the Semantic Web technologies [12]. The W3C Geospatial Incubator group (GeoXG) have developed the W3C Geospatial Vocabulary [13] to enable Web representation of physical location and geography. Recently, a Geospatial Resource Description Framework (GRDF) was proposed to provide general, semantics-aware and expressive language for the domain [14]. To facilitate the use of semantic web technologies for GIS, OWLIM-SE [15] and TopBraid Composer (along with AllegroGraph) [16] integrated geospatial extensions into their semantic repositories. However, both platforms only support two-dimensional geospatial data.

For the current problem, working with the W3C Geospatial Vocabulary seems to be a sensible initial approach. Therefore, in this work we study, explore and adapt the W3C Geospatial Vocabulary for our use case.

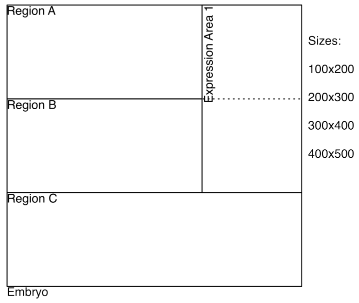


Fig. 3: A simplified gene expression image

5 Simulating the Biomedical Image Representations

From Section 4 it is apparent that there is a wealth of existing work that may be applicable in our use case. Section 5.2 considers one such technology. It seems likely that a number of other technologies and ideas shall be equally promising. Thus in order to facilitate a fair and meaningful comparison, first a suitable benchmark approach is discussed in Section 5.1.

Rather than dealing with the complexity of a real biomedical image, the first set of experiments discussed in this paper use a simplified image depicted in Fig. 3, which is an embryo that is broken down into three distinct regions (A, B, and C). These regions represent different tissues or biological structures. Fig. 3 depicts the common scenario in which an area of expression overlaps multiple regions (in this case 'Expression Area 1' covers A, and B).

5.1 The Benchmark

Ideally the semantic representation of each image will be lossless as that ensures no crucial information is neglected. Intuitively this approach seems too expensive to be feasible. Accordingly, a brief experiment was conducted to test the practicality of a lossless solution. Additionally, this representation can be used as a benchmark for comparing more abstract representations designed in the future.

The RDF triples for the image are modelled using the representation depicted in Fig. 4. EMBRYO_IMAGE is the box which contains each REGION. Both of these concepts have an ID (or name that functions as an ID). A REGION is comprised of multiple pixels. Every pixel in the image is recorded in this representation. Associated with each PIXEL is:

- `id` that is an identifier for the pixel;
- `x-coord` value of x pixel's co-ordinate;
- `y-coord` value of y pixel's co-ordinate;
- `colour` of the pixel;

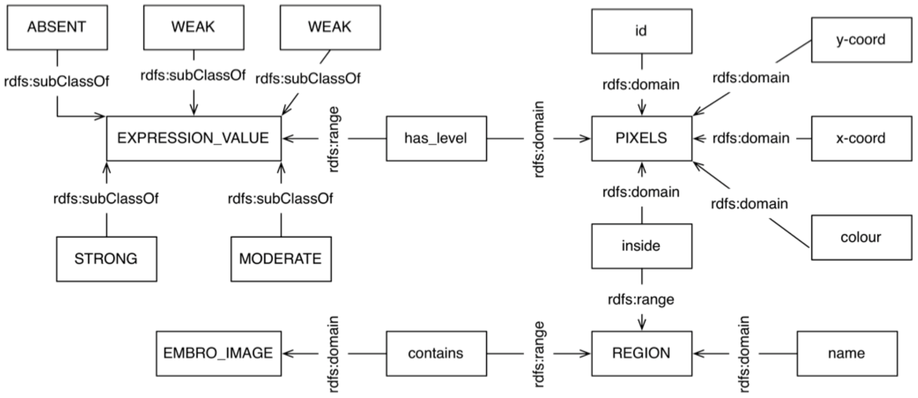


Fig. 4: Visualisation of RDF pixel-based representation

- which region (A, B, or C) the pixel is `inside`; and
- `has_level` indicates which level of expression (if any) the pixel represents.

EMAGE contains over 400,000 images[4] of varying sizes. The smallest images are 100 pixels by 160 pixels. This means, at a minimum, EMAGE contains $6,400,000,000$ pixels[5]. If each pixel has 6 items of information (and thus triples) associated with it, that means EMAGE requires at least $38,400,000,000$ triples to be stored. In reality most images are bigger than 100x160, indeed some are as large as 1920x1200, thus the true number of pixels/triples will be considerably larger.

In order to simply the exercise, Fig. 3 was taken to be 100x200 pixels in size. Then a number of different RDF/XML files were created and loaded into a triplestore. In some files the size of Fig. 3 was stretched to make it 200x300, 300x400 or 500x600. Likewise, the number of images in each RDF file was changed (1, 10, or 100 images). Consequently, the content of the RDF files ranged from 1 image at size 100x200 all the way to 100 images at size 500x600.

Immediately noticeable was the size of the larger files — the 100 images of size 500x600 consumed roughly 15GB of disk space. This caused resource issues with the machine running the triplestore, and so an n3 representation was generated (roughly 7.5GB). Loading this file into the repository took over an hour, and required over 30GB of disk space[6]. Whilst a triplestore expert may reduce both measurements, it is unlikely they would be reduced to the extent that the full EMAGE data set could be represented in this manner with the kind of hardware available in the current economic climate.

As such, whilst the lossless pixel-based solution encodes every piece of information contained within an image, it is unsuitable for real world application in

[4] Not including the 3D models.

[5] 400,000 * 100 * 160

[6] RDFS inferencing was enabled.

the current use case. Nevertheless, it functions well as a baseline to which other abstractions can be compared.

5.2 Using Geospatial Vocabulary

The discussion now moves to the trial application of the WGS84 Geo Positioning RDF Vocabulary [13], in particular the implementation offered by OWLIM-SE. Ideally, this solution would work perfectly thus mitigating the need for further research. Yet, as this discussion makes clear, whilst certain aspects are appropriate for the biological use case others simply do not work.

Again, the image from Fig. 3, particularly the RDF file generated in Section 5.1 containing a single 100x200 image, was used to test this geospatial SPARQL capabilities. Because the technology is now geospatial, it is necessary to amend the previously discussed RDF by replacing the triples documenting the x and y co-ordinates with triples containing latitude and longitude. As the latter are based on the globe, there is no direct mapping to the former. Yet we assumed one. Thus the x value became the longitude and the y value the latitude. For example, the triple `<x-coord rdf:datatype="http://www.w3.org/2001/XMLSchema#integer>68</x-coord>` became `<wgs84_pos:long>68</wgs84_pos:long>`.

Using SPARQL and OWLIM geospatial extensions, we queried the RDF triples to:

- find pixels within a circle;
- find pixels within a rectangle;
- find pixels within a polygon; and
- compute distance between two pixels.

Being able to find pixels within a circle, rectangle and polygon would enable the discovery of genes expressed within that region. This is the starting point from which almost all analysis in the use case builds. Finding the distance between entities of interest is likewise useful as we can discover how close specifics gene expression patterns are to a point of interest.

In general most of the functionality worked, to a degree, for the use case. Three particular problems were encountered. Firstly, the scale was completely wrong. The unborn mouse is a very small object, yet the globe is very big. So whilst EMAGE uses pixels[7] or μm as the unit of distance, the geospatial community prefers km and miles. Secondly, whilst the pixels within a polygon/circle/rectangle were accurately returned, the distance was not. The globe is a sphere, and this fact is used when calculating distance. The unborn mouse changes shape over the course of its development; starting as an imperfect sphere, graduating to a “blob” and eventually becoming the recognisable shape of a mouse. Accordingly, the geometry used to calculate distance is not applicable, and thus supplied erroneous results. Thirdly, the geospatial SPARQL has a distance query limitation. The latitude is limited to the range -90 (South) to +90 (North) and longitude is limited to the range -180 (West) to +180 (East). In

[7] 3D pixels, called *voxels* can also be used.

the our use case, we do not have a negative pixel locations nor a limit on the number of pixels in an image.

It is no surprise that a solution developed for the geospatial world is not directly applicable to our domain. Yet, by undertaking this exercise we have learnt that appropriately modified implementations of certain spatial operations, i.e. a relevant extension to the SPARQL query language, could benefit the processing of abstractions of information contained in biomedical images

6 Conclusion

CUBIST [17] is an EU FP7 project exploring next generation BI, through the exploitation of Semantic Technologies and Visual Analytics.

Our work started with the goal of making the EMAGE image data available to the CUBIST partners, thus enabling a BI centric exploration of the data. As such, the overarching goal of this activity is to generate a semantic representation of the data contained within biomedical images. The techniques developed should likewise allow medical images to be used.

This paper demonstrates that a lossless pixel-based rendition can be constructed with relative ease. Yet, the volume of data makes this impractical. The second thread of activity described within these pages explores the application of an existing geospatial technology.

Whilst initially promising, it is clear that the geospatial technology is very much *geo* specific. It makes a number of assumptions and decisions based on the Earth and its known properties. Whilst this approach is entirely valid, regrettably, it ensures the technology cannot be applied to the current use case. This suggests the need for a *bio*spatial representation and associated machinery.

Although the exact form and nature of this biospatial solution requires further research, some clear requirements have been gathered from this paper:

- the unit of measurement is in pixels/voxels or μm;
- space should be described using the standard x and y Cartesian system, with this reflected in the queries and query output;
- there should be no limit on the size (resolution) of the image or the objects contained within.

There is still the possibility that the existing geospatial solutions can be changed to better reflect the biospatial reality, this shall be explored in our future work.

Acknowledgments. This work is part of the CUBIST project (Combining and Uniting Business Intelligence with Semantic Technologies), funded by the European Commission's 7th Framework Programme of ICT under topic 4.3: 'Intelligent Information Management'.

References

1. Kuhn, W., Raubal, M., Gärdenfors, P.: Cognitive Semantics and Spatio-Temporal Ontologies. Introduction to a Special Issue of Spatial Cognition and Computation. 7(1), 3–11 (2007)
2. The Edinburgh Mouse Atlas of Gene Expression (EMAGE), `http://www.emousemouseatlas.org/emage/home.php`
3. Baldock, R., Davidson, D.: Anatomy ontologies for bioinformatics: principles and practise, chap. The Edinburgh Mouse Atlas, pp. 249–265. Springer Verlag (2008)
4. Andrews, S., M^{c}Leod, K.: Gene co-expression in mouse embryo tissues. In proceedings of the 1st CUBIST Workshop. CEUR Workshop Proceedings, ISSN 161–0073 (2011)
5. Allen Brain Atlas, `http://developingmouse.brain-map.org/`
6. Antezana, E., Kuiper, M., Mironov, S.: Biological knowledge management: the emerging role of the semantic web technologies. Brief. Bioinform, 10(4):392–407 (2009)
7. Hudelot, C., Atif, J., Bloch, I.: Fuzzy Spatial Relation Ontology for Image Interpretation. Fuzzy Sets System. 159(15), 1929–1951 (2008)
8. Mechouche, A., Morandi, X., Golbreich, C., Gibaud, B.: A Hybrid System for the Semantic Annotation of Sulco-gyral Anatomy in MRI Images. In: 11th International Conference on Medical Image Computing and Computer-Assisted Intervention - Part I, pp. 807–814, Springer-Verlag (2008)
9. Awang Iskandar, D.N.F: Visual Ontology Query Language. In: 1st Conference on Networked Digital Technologies, pp. 65 – 70, (2009)
10. Awang Iskandar, D.N.F: A Visual Ontology Query Language to Bridge the Semantic Gap in Content-based Image Retrieval. IJWA. 1(4), 183–193 (2010)
11. “geospatial.” Collins English Dictionary - Complete & Unabridged 10th Edition. HarperCollins Publishers.
12. Kuhn, W.: Geospatial Semantics: Why, of What, and How? Journal on Data Semantics (Special Issue on Semantic-based Geographical Information Systems, Spring 2005, LNCS 3534): 1-24 (2005)
13. WGS84 Geo Positioning: an RDF Vocabulary, `http://www.w3.org/2003/01/geo/wgs84_pos`
14. Alam, A. Khan, L., Thuraisingham, B.: Geospatial Resource Description Framework (GRDF) and Security Constructs. J. Computer Standards and Interfaces (Special Issue: Secure Semantic Web) 33(1), 35 – 41 (2011)
15. geospatial indexing in OWLIM, `http://www.ontotext.com/owlim/geospatial`
16. AllegroGraph 4.4, `http://www.franz.com/agraph/allegrograph/`
17. CUBIST project, `http://cubist-project.eu/`

Towards Scalingless Generation of Formal Contexts from an Ontology in a Triple Store

Frithjof Dau, SAP Research

Abstract. The EU-funded research project CUBIST investigates how Formal Concept Analysis can be applied as a Visual Analytics tool on top of information stored in a Triple Store (TS). This paper provides first steps for utilizing SPARQL in order to generate formal contexts out of the data in the TS, where the emphasis is put on using object-properties between individuals. Thus is complements FcaBedrock, which will be used in CUBIST as well and focuses on the scaling of datatype-properties between individuals and literals. It is discussed how the approaches of this paper and FcaBedrock can be combined.

1 Introduction

The EU funded research project CUBIST[1] targets new approaches to Business Intelligence (BI) by combining essential features of Semantic Technologies, Business Intelligence and Visual Analytics based on FCA (Formal Concept Analysis).

The Visual Analytics part of CUBIST is complementing traditional BI-means by utilizing FCA for analyzing the data in the triple store. FCA is a well-known theory of data analysis which allows to conceptually clustering objects with respect to a given set of attributes and then visualize the (lattice-ordered) set of clusters, e.g. by means of Hasse-diagrams. The starting point of FCA is a *formal context (O,A,I)* consisting of a set O of formal objects, a set A of formal attributes, and an incidence-relation $M \subseteq O \times A$ between the formal objects and attributes. There exists a variety of FCA-tools[2], but nearly all of them take a formal context as input. Real data to be analyzed, however, often comes in different forms:

- conceptually, often attributes are not binary, but have values like numbers, strings, or dates (e.g. we have *many-valued attributes*)
- technically, data can come in form of csv-files, databases, triple stores, etc

For dealing with many-valued attributes, the most-used method is *conceptual scaling* [1]. Essentially, for a given-many valued attribute, a conceptual scale is a specific context with the values of the many-valued attribute as objects. The choice of the attributes of the scale is a question of the design of the scale: The attributes are meaningful attributes to describe the values; they might be different entities or they might even be the values of the property again. Using a conceptual scale, a dataset with a many-valued attribute can be "translated" into a formal context, where the objects are the objects of the dataset and the attributes are the attributes of the conceptual scale.

[1] www.cubist-project.eu

[2] See http://www.upriss.org.uk/fca/fcasoftware.html for a maintained list of FCA-software

To the author's knowledge, there are essentially two tools which allow for scaling real datasets:

- ToscanaJ [2] is a suite of tools which allows to creating conceptual scales out of data from a relational database and then interactively visualizing and exploring the generated concept lattices
- FcaBedrock [3,4] is a tool which converts csv-files into formal contexts. It is "taking each many-valued attribute and converting it into as many Boolean attributes as it has values and converting continuous values using ranges." [4]

The approach of ToscanaJ is a two-step approach: First, in the design phase of a conceptual information system, the conceptual scales are created by an FCA-experienced designer. In the run-time phase, these scales are used by the user to explore the data. The downside of this approach is that the scales are predefined in the design-phase, which in turn implies that the lattice-structure of the scales is fixed, thus ToscanaJ does not really allow new *structural* insights into the data to be obtained. In the beginning of CUBIST, a modified version of ToscanaJ, called ToscanaJTS ("TS" for "Triple Store") has been developed which acts on a triple store instead of a database [5,6]. ToscanaJTS shows the applicability of FCA on top of a triple store, but it inherits the above discussed downsides of ToscanaJ as well.

The approach of FcaBedrock slightly differs from ToscanaJ. Similar to ToscanaJ, there is information needed on how to convert real data into a formal context. As stated in [4] "FcaBedrock solves these problems by documenting data conversions in re-usable, editable, meta-data-files called bedrock files." The difference to ToscanaJ is that the formal attributes of the generated formal context are not manually defined during the design phase, but they are created on the fly from the real data and the meta-information in the bedrock files. In this respect, FcaBedrock better serves the purpose to generate formal contexts out of the real data on the fly. For CUBIST, though, there are the following disadvantages: First, FcaBedrock puts a focus on many-valued attributes, but –as it will be discussed in the next sections- there are indeed possibilities to obtain binary relations directly (without scaling) out of the information in a triple store. Secondly, the existing version of FcaBedrock needs csv-files as input. This disadvantage, though, is targeted: A new version of FcaBedrock which acts directly on a triple store is currently developed within CUBIST.

Data in a triple store modeled with RDFS is essentially structured as follows:

- First of all, we have *individuals* in a triple store. Individuals are instances of RDFS-types, which are hierarchically ordered classes
- There are binary relationships called *object properties* between individuals. Similarly to types, these properties can be hierarchically ordered.
- Finally, we can assign values (strings, numbers, dates, etc) to individuals by means of *datatype properties.*

Having said this, a triplestore-enabled version of FcaBedrock can essentially deal with the conversion of datatype properties into formal contexts. On the other hand, as object properties are binary relations between indivuals, they give naturally rise to formal contexts, where the domain of an object property can serve as the set of formal objects and the range as the set of formal attributes. This first idea is too narrow. This paper discusses first steps on how to utilize object properties in different ways in order to generate formal contexts between resources in a triple store. As this paper deals with utilizing object properties for FCA, and as FcaBedrock deals with utilizing datatype properties, the ideas presented in the paper are complementing the ideas which underly the creation of context in FcaBedrock. It is planned to combine the approach of FcaBedrock and the ideas in this paper to provide in CUBIST a full-fledged approach to generate formal contexts out of the data (individuals, object properties, datatype properties) in a triple store.

2 Prerequisites

We will exemplify most of the ideas with the data from the HWU use case [7] in CUBIST. In order to understand the examples, the use case and the underlying ontology [8,9] shall be briefly introduced.

The HWU use case deals with data about gene expressions in mouse embryos. The core information are triples of the form (gene, tissue, level of expression), where:

- A *gene* is a unit of instruction providing directions for tasks in the development of a mouse, e.g. the creation of a protein.
- A *tissue* is a anatomical part of a mouse embryo. Tissues are ordered via a part-of-relation. Moreover, each tissue is uniquely assigned to a *Theiler Stage*, being a “time-slot” in the development of a mouse.
- The *level of expression* or *strength* states whether a gene is expressed in a tissue, or whether it is known that it is not expressed (this can even be more fine-grained described as weakly, moderate or strongly expressed), or whether it is unknown whether the gene is expressed or not (this can even be more fine-grained described as “not examined” or “possible).

Such a triple is called *textual annotation* and concluded from some *experiments*. Each experiment consists of one or more textual annotations.

The data of the HWU use case has been converted to an RDFS-ontology and stored in a triple store (OWLIM[3]). The schema of the HWU-ontology is provided in Fig. 1.

In order to create formal contexts out of the HWU-data in the triple store, we have developed a small tool which takes a SPARQL-query as input and converts the result of the SPARQL-query into a context. Essentially, the tool works as follows: The result of a SPARQL-query is a table, where the columns correspond to the query variables. It is possible that cells in the table are not filled (this can happen if the OPTIONAL-clause of SPARQL is used). The names of the query variables determine whether the variable is used to generate a formal object or a formal attribute: Variables starting with “o” generate objects, variables starting with “a” generate attributes, and all other variables have no impact on the generated context. If more than one variable starts with “o”, then the result for these variables are simply concatened (with a divider ‘--‘) to generate an object name. The case of more than one variable starting with “a” is handled in a similar manner. Finally, if we have a row in the SPARQL-result which generates both an object and an attribute, a cross in the corres-

[3] http://www.ontotext.com/owlim

ponding incidence relation is set. An (artificial) example for this algorithm is provided in Fig. 2.

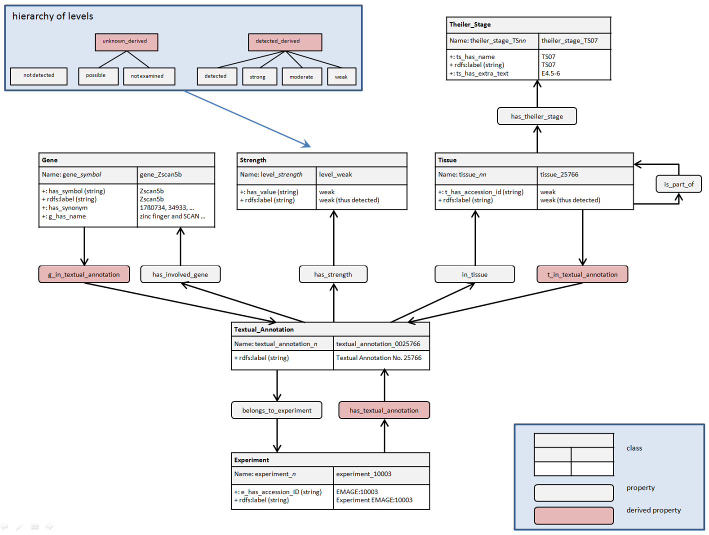


Fig. 1. The ontology for the HWU use case in CUBIST

The result of a SPARQL-query

Obj1	Obj2	Att1	Att2
O1			
O1	O2		
	O3		
		A1	
		A1	A2
			A2
O4		A3	
O5	O6	A4	
O5	O7	A5	A6
O5	O7	A4	

The generated formal context

	A1	A1—A2	A2	A3	A4	A5—A6
O1						
O1—O2						
O3						
O4				X		
O5—O6					X	
O5—O7					X	X

Fig. 2. From SPARL-query-results to formal contexts

Before we come to the next section, let us finally state two general assumptions:

1. URIs are the unique identifiers for resources, but they might be too clumsy to be used as names for the resources. We will use labels instead, thus we assume that each resource in the triple store is appropriately labeled using `rdfs:label`, and we more over assume that different entities have different labels.
2. Secondly, as SPARQL is agnostic to inferencing, we naturally assume that the information in the triple store is closed under RDFS-entailment.

Both assumptions hold in the given HWU-dataset.

3 SPARQRL-queries

3.1 Simple Pattern: Linking entities with a chain of properties

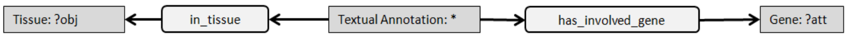
The very essence of a formal context is the incidence relation, being a binary relation between objects and attributes. RDF-properties in turn are binary relations as well (between RDF-resources). Thus any RDF-property `linkingproperty` already gives rise to a formal context, via the following SPARQL-query:

Most basic SPARQL-query for generating a formal context

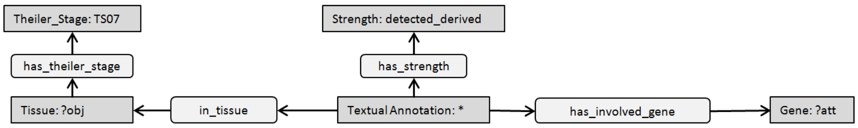
```
SELECT DISTINCT ?obj ?att WHERE {
?objRessource rdfs:label ?obj .
?attRessource rdfs:label ?att .
?objRessource :linkingproperty ?attRessource . }
```

This pattern is anyhow too simple as a general pattern for generating contexts, and we have to extend it. For the following considerations, let us assume we want to investigate the tissues of Theiler stage 07 and which genes are detected in those tissues.

1. First of all, as RDF is graph-based, we can have in RDF chains of properties between resources. This particularly applies when we have reified relations. For our example investigation, note that there is no direct property linking tissues and genes: Instead, we have textual annotations linking them. So we have to consider the following chain of properties:[4]



2. In the following, resources which are not directly queried, like the (unknown) textual annotations, will be called "intermediate resources".
3. The property `in_tissue` in the example above moreover shows that in the chain of properties, some properties might be traversed in their opposite direction.
4. It is sensible to assume that all queried resources as well as intermediate resources are retrieved from some RDF-type. In our example, we have Tissue and Gene for the queried resources and Textual_Annotation for the intermediate resource.
5. Finally, we might further restrict the set of objects, or the set of attributes, by some constraints. In our example, we are interested in investigating a specific Theiler stage and thus instead of taking all tissues into account, we restrict ourselves only to those tissues from that Theiler stage. Similarly, we might impose constraints on the intermediate resources as well. In our example, we are only interested into combinations of tissues and genes where the gene is detected (maybe weakly, moderate or strongly) in the tissue, but we are not interested into combinations where the gene is not detected. Thus we have the following additional constraints:



We are now prepared write a SPARQL-query which generates the wished context.

[4] We use the notion of conceptual graphs [10] and query graphs [11] to diagrammatically depict the queries.

SPARQL Query 1	Explanation
`SELECT DISTINCT ?obj ?att WHERE {`	Select Clause for objects and attributes
`?x1 rdf:type :Tissue ; rdfs:label ?obj .`	Type of objects (see 3))
`?x1 :has_theiler_stage :theiler_stage_TS07 .`	Additional Constraint for objects (see 4))
`?x3 rdf:type :Gene ; rdfs:label ?att .`	Type of attributes (see 3))
`?x2 rdf:type :Textual_Annotation.`	Type of intermediate resource (see 3))
`?x2 :in_tissue ?x1 .`	1st and 2ndproperty in the chain of properties (see 1),
`?x2 :has_involved_gene ?x3 .`	first prop. is traversed in opposite direction. (see 2))
`?x2 :has_strength :level_detected_derived .}`	Add. constraint for intermediate ressources (see 4))

It should be noted that this SPARQL-query does not retrieve **all** tissues of Theiler Stage 07: Instead, only those tissues are retrieved where a gene is detected. Let us call such a query “object-restricted”. Vice versa, not all genes are retrieved, but only those who are detected in some tissue of Theiler stage 07. Let us call those queries “attribute-restricted”. In other words: In the formal context, we have by definition neither empty rows (due to the query being object-restricted) nor empty columns (query being attribute-restricted). One might want to change this to object-unrestricted queries, i.e. retrieving all tissues of Theiler stage 07, allowing empty rows, and/or attribute-unrestricted queries, i.e. retrieving all genes, allowing empty columns. So we have four variants of query 1 to consider.

As there are nearly 7000 genes, a query which retrieves all of them and adds them as formal attributes in the formal context does not seem sensitive. But as there are only 16 tissues in Theiler stage 07, retrieving all of them and adding them as formal objects to the formal contexts is reasonable. So let us consider the object-unrestricted variant of query 1. There are two ways to obtain this in a SPARQL-query: Either via utilizing the OPTIONAL-clause of SPARQL, or using the UNION-operator. Both queries are given below.

SPARQL Query 1a, utilizing OPTIONAL	SPARQL Query 1b, utilizing UNION

SELECT DISTINCT ?obj ?att WHERE {
?x1 rdf:type :Tissue ; rdfs:label ?obj .
?x1 :has_theiler_stage :theiler_stage_TS07 .
OPTIONAL
{ ?x3 rdf:type :Gene ; rdfs:label ?att .
 ?x2 rdf:type :Textual_Annotation.
 ?x2 :in_tissue ?x1 .
 ?x2 :has_involved_gene ?x3 .
 ?x2 :has_strength :level_detected_derived .}
}
ORDER BY ?obj ?att
	SELECT DISTINCT ?obj ?att WHERE {
	{ ?x1 rdf:type :Tissue ; rdfs:label ?obj .
	 ?x1 :has_theiler_stage :theiler_stage_TS07 . }
	UNION
	{ ?x1 rdf:type :Tissue ; rdfs:label ?obj .
	 ?x1 :has_theiler_stage :theiler_stage_TS07 .
	 ?x3 rdf:type :Gene ; rdfs:label ?att .
	 ?x2 rdf:type :Textual_Annotation.
	 ?x2 :in_tissue ?x1 .
	 ?x2 :has_involved_gene ?x3 .
	 ?x2 :has_strength :level_detected_derived . }
	}
	ORDER BY ?obj ?att

Beginning of resultset	**Beginning of resultset**
obj att ------------------------------------- EMAP:25772 EMAP:42 Etv5 EMAP:42 Smad2 …	obj att --- EMAP:25772 EMAP:42 EMAP:42 Etv5 EMAP:42 Smad2 …

Table 1. SPARQL-query for gene-tissue combinations of TS 07, all TS07 tissues retrieved

From an RDF-point of view, these queries are semantically (slightly) different: In query 1a, we have a row in the resultset with a tissue *t* and without a gene if and only if the gene belongs to Theiler stage 07 and no gene is detected in that tissue, whereas in query 1b we have a row in the resultset with a tissue *t* for any *t* belonging to Theiler stage 07. An example where the resultsets differ is tissue EMAP:42, as it can be seen in Table 1. So the resultset of query 1b is a superset of the resultset of query 1a. Anyhow, the formal contexts generated with queries 1a and 1b are indeed the same, thus from an FCA-perspective, the queries are equivalent.

Please note moreover that in query 1b, the clause querying the tissues is repeated in the UNION clause, whereas a repetition of the clause is not. This renders query 1b (slightly) more complicated.

Having these two differences in mind, one can conclude that query 1a has to be preferred over query 1b.

The patterns of both queries can easily be transferred to the case of attribute-unrestricted queries. For queries which are both object- and attribute-restricted, only the UNION-variant can be easily extended.

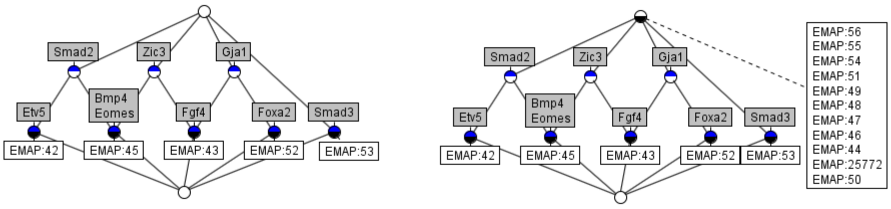


Fig. 3. Concept lattices retrieved from query 1 (left) , and query 1a or query 1b (right)

3.2 Combining different variables to objects or attributes

In the previous section, we have investigated for Theiler stage 07 tissue-gene combinations such that the gene is detected in the respective tissue. There are anyhow different levels of being detected: weak, moderate, and strong. Moreover, in some experiments a gene is detected without information on how strong the expression of the gene in that tissue is. So we have four kinds of being detected, called “weak”, “moderate”, “strong” and “detected”. All of them are subsumed by an artificial strength called “detected derived”. See Fig. 1, where these levels are depicted. The information on the level of expression is not provided by the SPARQL-queries of the last section, but it can easily be added by slightly altering the queries. We adopt query 1b as following, highlighting the changes in the query:

SPARQL query 1b with level of expression added

```
SELECT DISTINCT ?obj ?att1 ?att2  WHERE {
?x1 rdf:type :Tissue ; rdfs:label ?obj .
?x1 :has_theiler_stage :theiler_stage_TS07 .
OPTIONAL
{     ?x3 rdf:type :Gene ; rdfs:label ?att1 .
      ?x2 rdf:type :Textual_Annotation.
      ?x2 :in_tissue ?x1 .
      ?x2 :has_involved_gene ?x3 .
      ?x2 :has_strength :level_detected_derived .
      ?x2 :has_strength ?x4 .
      ?x4 rdf:type :Strength ; rdfs:label ?att2 . }
}
ORDER BY ?obj ?att1 ?att2
```

In the next figure, the resulting concept lattice is provided.

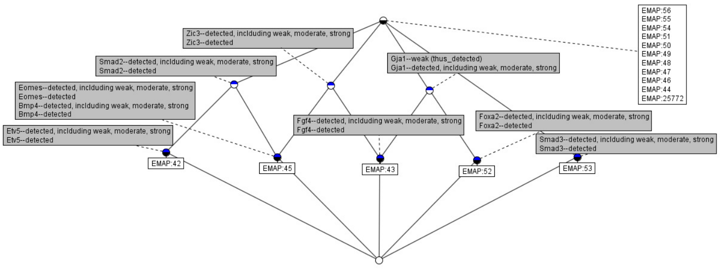

EMAP:56
EMAP:55
EMAP:54
EMAP:51
EMAP:50
EMAP:49
EMAP:48
EMAP:47
EMAP:46
EMAP:44
EMAP:25772
Zic3--detected, inclduding weak, moderate, strong
Zic3--detected
Smad2--detected, inclduding weak, moderate, strong
Smad2--detected
Gja1--weak (thus_detected)
Gja1--detected, inclduding weak, moderate, strong
Eomes--detected, inclduding weak, moderate, strong
Eomes--detected
Bmp4--detected, inclduding weak, moderate, strong
Bmp4--detected
Fgf4--detected, inclduding weak, moderate, strong
Fgf4--detected
Foxa2--detected, inclduding weak, moderate, strong
Foxa2--detected
Etv5--detected, inclduding weak, moderate, strong
Etv5--detected
Smad3--detected, inclduding weak, moderate, strong
Smad3--detected
EMAP:42
EMAP:45
EMAP:43
EMAP:52
EMAP:53

Fig. 4. Concept lattice of altered query 1b, now providing information about strengths

The lattices of Fig.3 and Fig. 4 have the same structure: Essentially, only the information in the attribute labels are more fine-grained. In other cases, combining more result variables to attributes might yield in valuable, new structural insights. To provide an HWU-example for this effect, we consider the following query to retrieve contradicting textual annotations, which was possible in the previous version of the tool:

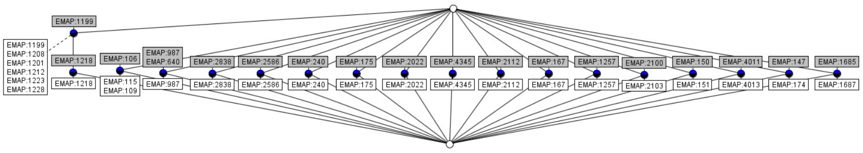
SPARQL query 2 for finding contradictions

```
select distinct ?o0 ?a0   where {
  ?x0 rdf:type :Tissue ; rdfs:label ?o0 .
  ?x1 rdf:type :Tissue ; rdfs:label ?a0 .
  ?x2 rdf:type :Gene ;   rdfs:label ?o1 .
  ?ta1 :in_tissue ?x0; :has_involved_gene ?x2; :has_strength :level_detected_derived .
  ?ta2 :in_tissue ?x1; :has_involved_gene ?x2 ; :has_strength :level_not_detected .
  {
  { ?x0 :is_part_of ?x1 .  Filter(!sameTerm(?x1,?x0)) }
  UNION
  {  Filter(sameTerm(?x0,?x1)  )  }   }}
```

This query retrieves pairs of tissues t_1 and t_2, where

- t1 is_part_of or the same tissue as t2 (that is, we are using propagation of tissues), and
- there exists a gene g which is (possibly weakly, moderate, or strong) detected in t_1 and not detected in t_2

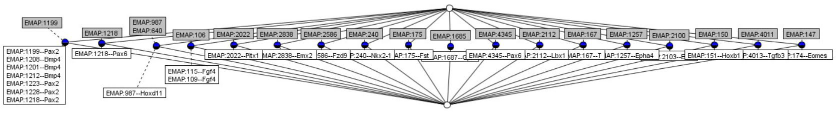
Of course, if a gene *g* is expressed in a t_1, then one can conclude that it is expressed in t_2 as well. That is, the query finds out pairs of tissues (t_1, t_2) where different experiments concerning the gene *g* come to contradicting results. As we will discuss the example further, we fix the following notation describing the roles of the tissues: The tissue t_1 will be called lower or detected tissue, and t_2 will be called upper or undetected tissue. Next, the concept lattice generated by query 2 is provided.



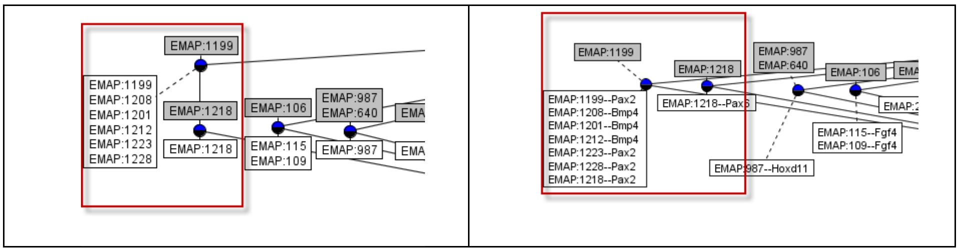
Most of the lattice does not provide, apart from the tissue-names, any structural information. On the left hand side, however, this lattice does reveal some insights:

- There are two contradicting tissue pairs with EMAP:106 as upper (undetected) tissue, as we have the two lower tissues EMAP:109 and EMAP:115. We don't know anyhow how many pairs of textual annotations cause these contradictions: it must be at least two pairs, but for example, we might have a contradiction for many genes, detected in EMAP:109 and not detected in EMAP:106. The same concern applies to any node in the lattice: We never know how many contradicting pairs of textual annotations we have for one node.
- Similarly, we have two contradicting tissue pairs with EMAP:987 as lower (detected) tissue.
- The left hand side is most interesting: There is a number of contradicting tissue pairs with EMAP:1199 as upper tissue, namely 7 (6+1). Maybe the experiment(s) investigating EMAP:1199 deserve a closer look? We have moreover a dependency: Whenever a lower (detected) tissue contradicts with EMAP:1218 as upper (nondetected) tissue, it contradicts with EMAP:1199 as well (but not vice versa).

Now, for this query, we lose the information which genes cause the contradictions. Knowing about the involved genes would allow to partly cope with the questions raised above. We do only a slight change in the query by adding the variable `?o1` to the list of variables in the select-clause, i.e. we reuse the last query and change it to query 2b by starting with "`select distinct ?o0 ?a0 ?o1 where` ". Below, the corresponding lattice is provided.



In the last discussion, we speculated that "we might have a contradiction for many genes, detected in EMAP:109 and not detected in EMAP:106." But now we see this contradiction is only causedwe by one gene (Fgf4), and same holds true for all nodes: all contradicting tissue pairs are caused by **exactly one** gene. Moreover, the lattices are not isomorphic: The difference is highlighted in the next screenshots:



Note that the contradiction between EMAP:1218 and EMAP:1119 as upper tissues and EMAP:1218 as lower tissue are caused by *different*, thus the attribute dependency between EMAP:1218 and EMAP:1119 on the left hand side is lost on the right hand side. It seems even more that EMAP:1119 deserves a closer observation.

3.3 Attributes of different types

So far, in the queries we have provided we have as objects or attributes either entities of **one** RDF-type, or combinations (via string-concatenation) of different types into one object or attribute. In some cases, though, it can be desirable to have objects or attributes of different types. This shall be exemplified with query 2 where we have analyzed contradiction pairs of tissues. These tissues in turn are assigned to Theiler stages. In the last query, we used

```
?x1 rdf:type :Tissue ; rdfs:label ?a0 .
```

to query the tissues. If we replace this line by

```
?x1 rdf:type :Tissue         .
?x1 :has_theiler_stage ?ts1 .
?ts1 rdfs:label ?a0          .
```

We obtain the Theiler stages instead. Now, utilizing the SPARQL-“UNION”-operator, it is possible to “combine” these slightly different queries, resulting in a formal context where the attributes are either tissues or Theiler stages.

The result lattice extends the lattice for query 2 by adding Theiler stages. It looks as follows:

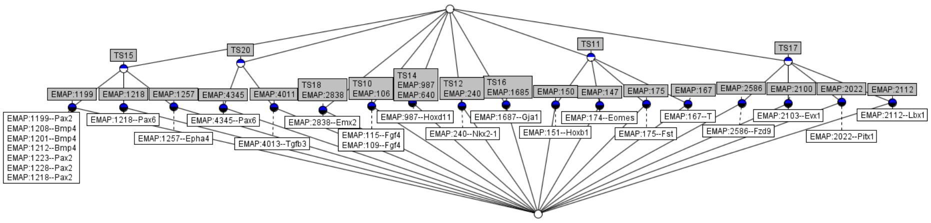


Fig. 5. Concept lattice for contradicting pairs of tissues

The nice lattice structure is unsurprising: It is caused by the fact that each tissue is assigned exactly one Theiler stage (and the part_of-relation only applies to tissues in the same stage).

The approach hereby exemplified it particularly helpful if attributes (here: tissues) are in turn classified by other attributes (here: Theiler stages). For this reason, it can easily be transferred to RDFS-instances and their corresponding types, thus utilizing the type-hierarchy in an RDFS-ontology. We have not exemplified this approach in this paper as the underlying ontology does not provide interesting type hierarchies.

4 Summary and next steps

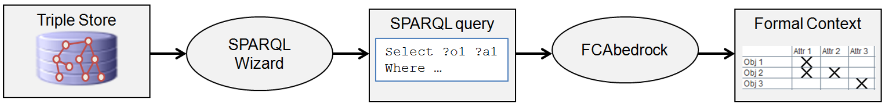
In the previous section, we have discussed how object properties between individuals in a triple store can be utilized for generating formal contexts. We have seen that only utilizing plain object properties between individuals is not sufficient: Instead, one should consider chains of object properties with constraints for intermediate nodes, and one should consider different means for adding formal attributes generated from *different* RDFS-types to the formal context. This is anyhow only a first step for generating formal contexts in CUBIST. There are essentially two tasks to be conducted:

1. First, it cannot be expected from a CUBIST user to write SPARQL-queries on her own. For this reason, we have to find common patterns for the generation of formal contexts of SPARQL-queries. Such patterns can be used in a wizard which guides the user in the creation of formal contexts without showing any SPARQL-queries.
2. Second, as already mentioned, the ideas presented in this paper complement the approach of FcaBedrock, thus it has to be investigated how these approaches can be combined.

A closer look at our approach reveals that there is no need that the SPARQL-queries, as presented in the paper, only return (the labels of) RDFS-individuals: Of course, one can extend the approach to queries where some of the variables return literals, e.g. strings or numbers. In our approach, for a given query-variable, each literal would be taken as it is for generating a formal attribute, which would be in most cases not desirable. Instead, for such variables, the process of conceptual scaling –as carried out by FcaBedrock- should apply. Indeed, as any SPARQL-query returns a table, it is straight-forward to feed such a table into FcaBedrock. Having said this, a possible workflow for the generation of contexts is as follows:

1. A user selects the type of individuals she wants to investigate.
2. A wizard guides the user in the creation of a SPARQL-query. For example, the wizard could provide properties or chains of properties (starting with the given RDFS-type), and the user can select which chain(s) should be used in the generation of the context. Moreover, it could be possible that in this step, the user adds additional constraints on intermediate nodes in the chain.
3. For each (chain of) properties selected by the user, the range consists either of individuals or of literals of a given type. In the former case, the labels of the individuals are used for the generation of attributes, whereas in the latter case, the values are transformed into formal contexts with the help of FcaBedrock.

That is, generally speaking, the meta-information which generated a formal context out of the triple store consists of a SPARQL-query and, for each query variable, instructions on how the results of the variable are used to generate objects and attributes. The whole process can be depicted as follows:



```
Select ?ol ?al
Where …
```

	Attr 1	Attr 2	Attr 3
Obj 1	X		
Obj 2	X	X	
Obj 3			X

In the further course of CUBIST, the consortium will investigate and implement a unified approach for the generation of formal context which takes individuals, object properties and datatype properties into account, leading to a fully-fledged generation of context out of the triple store on the fly.

Acknowledgement

This work is part of the CUBIST project ("Combining and Uniting Business Intelligence with Semantic Technologies"), funded by the European Commission's 7th Framework Programme of ICT, under topic 4.3:Intelligent Information Management.

The herein used data orginates data from the Edinburgh Mouse Atlas of Gene Expression (EMAGE) and is released under a creative commons license.

5 References

1. Ganter, B., Wille, R.: Conceptual Scaling. In: Roberts, F. (ed.) Applications of Combinatorics and Graph Theory to the Biological and Social Sciences. IMA, vol. 17, pp. 139–168. Springer, Heidelberg (1989)
2. Becker, P., Correia, J.H.: The ToscanaJ Suite for Implementing Conceptual Information Systems. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 324–348. Springer, Heidelberg (2005)
3. Andrews, S.: Data Conversion and Interoperability for FCA. In: CS-TIW 2009, pp.42–49 (2009), http://www.kde.cs.uni-kassel.de/ws/cs-tiw2009/proceedings_final_15July.pdf
4. Andrews, S., Orphanides, C.: FcaBedrock, a Formal Context Creator. In: Croitoru,M., Ferre, S. and Lukose, D. (eds.): Proceedings of ICCS 2010, Kuching, Malaysia. LNAI 6208, Springer-Verlag (2010)
5. F. Dau and B. Sertkaya. Formal Concept Analysis for Qualitative Data Analysis over Triple Stores. In: Olga De Troyer, Claudia Bauzer Medeiros, Roland Billen, Pierre Hallot, Alkis Simitsis and Hans Van Mingroot (eds): Advances in Conceptual Modeling. Recent Developments and New Directions - ER 2011 Workshops FP-UML, MoRE-BI, Onto-CoM, SeCoGIS, Variability@ER, WISM. Springer, LNCS, vol 6999, 2011.
6. F. Dau and B. Sertkaya. An Extension of ToscanaJ for FCA-based Data Analysis over Triple Stores. In F. Dau (ed):
7. Proceedings of the 1st CUBIST (Combining and Uniting Business Intelligence with Semantic Technologies) Workshop 2011, CEUR proceedings, vol 753, 2011.
   Simon Andrews, Keneth McLeod: Gene Co-Expression in Mouse Embryo Tissues. In F. Dau (ed): Proceedings of the 1st CUBIST (Combining and Uniting Business Intelligence with Semantic Technologies) Workshop 2011, CEUR proceedings, vol 753, 2011.
8. M^{c}Leod, K., Ferguson, G., Burger, A.: Argudas: arguing with gene expression information. In: Paschke, A., Burger, A., Splendiani, A., Marshall, M.S., Romano, P. (eds.) Proceedings of the 3rd International Workshop on Semantic Web Applications and Tools for the Life Sciences (December 2010)
9. Richardson, L., Venkataraman, S., Stevenson, P., Yang, Y., Burton, N., Rao, J., Fisher, M., Baldock, R.A., Davidson, D.R., Christiansen, J.H.: EMAGE mouse embryo spatial gene expression database: 2010 update. Nucleic Acids Research 38, Database issue, D703–D709 (2010)
10. J. F. Sowa: Conceptual Structures: Information Processing in Mind and Machine. Addison Wesley Publishing Company Reading, 1984.
11. F. Dau: Query Graphs with Cuts: Mathematical Foundations. In A. Blackwell, K. Marriott, A. Shimojima (Eds): Diagrammatic Representation and Inference. LNAI 2980, Springer Verlag, Berlin–New York 2004, 32-50.

ConSeQueL - SQL Preprocessor Using Formal Concept Analysis with Measures

Jan Kanovsky and Juraj Macko

Dept. Computer Science
Palacky University, Olomouc
17. listopadu 12, CZ-77146 Olomouc
Czech Republic
email: {kanovskyjan}@centrum.cz, {juraj.macko}@upol.cz

Abstract. In this paper the SQL preprocessor ConSeQueL is introduced. ConSeQuel is based on a Formal Concept Analysis (FCA) with the measures. The input dataset is provided as collection of the text files, which consist of the formal context file and measure files. This collection is preprocessed and the result is inserted into the predefined tables of the relational database. For preprocessing there are used two methods. The first method produces a collection of some formal concepts with aggregated measures. The second method to the attribute implications with aggregated measures. Both outputs provide interesting information for the financial analysis. ConSeQueL provides the information very similar to OLAP cube.

1 Introduction

The formal concept analysis (FCA) is well known method for data analysis. This method was enhanced by adding measures to the objects and/or attributes and such measures are aggregated. The input data for FCA is the formal context, in this case enhanced by measures. The result is a collection of the formal concepts, enhanced by aggregated measures. Other possible output of FCA is a collection of an attribute implications, which is enhanced by measures as well. Based on enhanced FCA theory the novel SQL preprocessor ConSeQueL was developed. Paper is structured as follows: In Preliminaries the basic theory is introduced. In Theoretical Background of the Preprocessor the enhancement of FCA is proposed. In SQL Preprocessor, ConSeQueL is introduced from various points of view, namely a functionality, an architecture, used algorithms etc. Conclusion and Future Research consists of some notes and the future perspectives.

2 Preliminaries

Just the fundamentals and basic definitions are shown. In this paper we use classical measure only. For more details about classical measures we refer to [3]. The following aggregation functions are used: $min()$ for minimum, $max()$ for maximum and $avg()$ for arithmetic mean. The aggregation operator are denoted by Θ. More details about aggregation operators can be found in [4].

2.1 Formal Concept Analysis [1]

The input dataset for FCA is the formal context, which is a relation between the set of objects X and the set of attributes Y, is denoted by $\langle X, Y, I \rangle$ where $I \subseteq X \times Y$. The formal concept of the formal context $\langle X, Y, I \rangle$ is denoted by $\langle A, B \rangle$, where $A \subseteq X$ and $B \subseteq Y$. $\langle A, B \rangle$ is a formal concept iff $A^{\uparrow} = B$ and $B^{\downarrow} = A$. The concept forming operators $()^{\uparrow}$ and $()^{\downarrow}$ are defined as $A^{\uparrow} = \{y \in Y \,|$ for each $x \in X \,:\, \langle x, y \rangle \in I\}$ and $B^{\downarrow} = \{x \in X \,|\,$for each $y \in Y \,:\, \langle x, y \rangle \in I\}$. The set A is called an extent and the set B an intent. The set all formal concepts of $\langle X, Y, I \rangle$ is denoted by $\mathcal{B}(X, Y, I)$ and equipped with a partial order $\leq$ forms a concept lattice of $\langle X, Y, I \rangle$.

2.2 Attribute Implication [1] and Minimal Generators [6]

An attribute implication $E \Rightarrow F$ over a set of the attributes Y is valid in a set $M \subseteq Y$ iff $E \subseteq M$ implies $F \subseteq M$. Formally $\|E \Rightarrow F\|_M = 1$ if $E \Rightarrow F$ is true in M. Let $\mathcal{M} \subseteq 2^Y$. We say that the attribute implication $E \Rightarrow F$ is valid in $\mathcal{M}$ if $E \Rightarrow F$ is valid in each $M \in \mathcal{M}$. For the formal context $\langle X, Y, I \rangle$ we define $\mathcal{M} = \{\{x\}^{\uparrow} | x \in X\}$ and we write $\|E \Rightarrow F\|_{\langle X,Y,I \rangle} = 1$ if $E \Rightarrow F$ is true in $\langle X, Y, I \rangle$. For an intent B of the formal concept $\langle A, B \rangle \in \mathcal{B}(X, Y, I)$ we define *generators* and *minimal generators* of B as $\mathrm{gen}(B) = \{D \subseteq Y \,|\, D^{\downarrow\uparrow} = B\}$ and $\mathrm{mgen}(B) = \{D \in \mathrm{gen}(B) \,|\, E^{\downarrow\uparrow} \subset B$ for every $E \subset D\}$. The closure operator $\uparrow\downarrow$ induces an equivalent relation on 2^Y. All generators $gen(B)$ belong to the same equivalent class. It means that all elements of the equivalence class has the same closure. The elements of each equivalent class equipped by order relation $\leq$ defined by $\subseteq$, forms a structure with one greatest element B and at least one minimal element. Each minimal element of the equivalence class is a minimal generator $mgen(B)$. The intent B can be seen as a maximal generator of B.

3 Theoretical Background of the Preprocessor

The theoretical background of the preprocessor consists of the FCA enhancement. The formal context is enhanced by measures. One or more measures are assigned to the objects and/or attributes. The formal concepts are enhanced by aggregated measures. The same approach is used by the attribute implications, moreover some interesting ratios are considered.

3.1 Formal Concept Analysis with Measures [2]

Definition 1 (Measure of Object and Attribute). *A Measure of the object is mapping $\Phi : X \to \mathbb{R}^+$ and a Measure of the attribute is mapping $\Psi : Y \to \mathbb{R}^+$.*

Definition 2 (Value of Extent and Intent). *The Value of extent is mapping $v : A_{\mathcal{B}(X,Y,I)} \to \mathbb{R}^+$defined as $v\,(A) = \underset{x \in A}{\odot}\, \Phi\,(x)$, where $\odot$ is either the symbol* *for the sum Σ (the "sum" operation) or the symbol for cardinality $|A|$ or the arbitrary aggregation function Θ. A is an extent of the formal concept $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$. Similarly, the value of the intent is mapping $w : B_{\mathcal{B}(X,Y,I)} \to \mathbb{R}^{+}$ defined as $w(B) = \underset{y\in B}{\odot} \Psi(y)$, where B is an intent of the formal concept $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$*

Let $X = \{Car1, \ldots, Car20\}$ be a set of sold cars and Y be a set of components. $Y = \{AC, AB, ABS, TMP, EG, AT\}$ consists of Air Conditioning (AC), Airbag (AB), Antilock Braking System (ABS), Tempomat (TMP), Extra Guarantee (EG) and Automatic Transmission (AT). Table 1 represents the formal context $I \subseteq X \times Y$, which means that the "car has additional component". In the example there are used two different measures: measure $\Phi(X)$ which represents the *Price of Car* and $\Psi(Y)$, which represents the *Price of Component*. The formal context from Table 1 results to $\mathcal{B}(X, Y, I)$, which consists of 23 concepts (in Table 2) e.g. the concept $\langle A_{20}, B_{20}\rangle = \langle\{Car12, Car14\}, \{AC, AB, TMP, EG, AT\}\rangle$.

	1. *AC*	2. *AB*	3. *ABS*	4. *TMP*	5. *EG*	6. *AT*	$\Phi(X)$ = Price in EUR
Car1	×	×					16 000
Car2		×	×	×			12 000
Car3		×	×	×	×		14 000
Car4	×			×	×		16 000
Car5	×				×		12 000
Car6	×	×	×				12 000
Car7		×	×	×			12 000
Car8			×				14 000
Car9							16 000
Car10		×					12 000
Car11		×		×			12 000
Car12	×	×	×	×	×	×	14 000
Car13		×		×			16 000
Car14	×	×		×	×	×	16 000
Car15			×		×		14 000
Car16	×	×					12 000
Car17	×	×					12 000
Car18		×	×	×			16 000
Car19			×				16 000
Car20	×	×	×		×		14 000
$\Psi(Y)$ = Price in EUR	1 000	500	800	600	250	100	

Table 1. The formal context of the cars, the additional components, the price of the car and the price of the component

Let $\Phi(X)$ be the *Price of the Car*, namely $\Phi(Car12) = 14000$, $\Phi(Car14) = 16000$. Putting $\odot = \Sigma$, then $v(A_{20}) = \underset{x\in A_{20}}{\Sigma} \Phi(x) = \Phi(Car12) + \Phi(Car14) = 14000 + 16000 = 30000$, which is the extent value of the concept A_{20}. It can

	Extent	Intent	Extent value
	Cars	Components	Total Price
1	X - all cars	$\emptyset$	278 000
2	$\{2,3,4,7,11,12,13,14,18\}$	$\{TMP\}$	128 000
3	$\{3,4,5,12,14,15,20\}$	$\{EG\}$	100 000
4	$\{1,4,5,6,12,14,16,17,20\}$	$\{AC\}$	124 000
5	$\{1,2,3,6,7,10,11,12,13,14,16,17,18,20\}$	$\{AB\}$	190 000
6	$\{2,3,6,7,8,12,15,18,19,20\}$	$\{ABS\}$	138 000
7	$\{3,4,12,14\}$	$\{EG,TMP\}$	60 000
8	$\{4,5,12,14,20\}$	$\{EG,AC\}$	72 000
9	$\{2,3,7,11,12,13,14,18\}$	$\{AB,TMP\}$	112 000
10	$\{3,12,14,20\}$	$\{AB,EG\}$	58 000
11	$\{3,12,15,20\}$	$\{ABS,EG\}$	56 000
12	$\{1,6,12,14,16,17,20\}$	$\{AC,AB\}$	96 000
13	$\{2,3,6,7,12,18,20\}$	$\{AB,ABS\}$	94 000
14	$\{4,12,14\}$	$\{AC,TMP,EG\}$	46 000
15	$\{3,12,14\}$	$\{TMP,EG,AB\}$	44 000
16	$\{12,14,20\}$	$\{EG,AB,AC\}$	44 000
17	$\{2,3,7,12,18\}$	$\{AB,ABS,TMP\}$	68 000
18	$\{3,12,20\}$	$\{ABS,EG,AB\}$	42 000
19	$\{6,12,20\}$	$\{ABS,AC,AB\}$	40 000
20	$\{12,14\}$	$\{AB,EG,AC,TMP,AT\}$	30 000
21	$\{3,12\}$	$\{AB,EG,TMP,ABS\}$	28 000
22	$\{12,20\}$	$\{AB,AC,EG,ABS\}$	28 000
23	$\{12\}$	Y - all components	14 000

Table 2. The formal concepts with the extent value

be interpreted as the "Total price of all cars with additional components AC, AB,TMP, EG and AT is 30 000 EUR". Let $\Psi()$ be the $Price\ of\ the\ Component$. Putting $\odot = \Sigma$, we have intent value of the formal concept A_{20} calculated as $w(B_{20}) = 1000+500+600+250+100 = 2450$. More mappings $\Phi(x)$ can be used. In our example $\Phi(x)$ was defined as $Price$, but $\Phi_1(x)$ can be defined as $Price$, $\Phi_2(x)$ as $Costs$ and $\Phi_3(x)$ as $Benefit$ - with different measures assigned to an object. In the example Σ was used, but any aggregation operator Θ can be utilized, to achieve various $v_i(A)$ - values for extent e.g. $Average\ sales$, $Minimal\ costs$ or $Maximal\ benefit$. Similarly more attribute measures and corresponding values of intent can be obtained.

3.2 Attribute Implications with Measures

Consider the attribute implication $\{AC, ABS\} \Rightarrow \{AB\}$ which is true in the formal context from the previous example. It means that all cars with the components Air Conditioning and ABS from the context table will also have the component Airbag. As we know from the previous example the each attribute (car component) has at least one measure assigned (in our example $\Psi(Y)$ was the $Price$). For each set of the attribute implication $E \Rightarrow F$, hence for the antecedent E and consequent F can be defined value similarly as it was done by the intent.

Definition 3 (Value of Antecedent and Consequent). *The Value of an antecedent is mapping* $p : E_{\mathcal{B}(X,Y,I)} \rightarrow \mathbb{R}^{+}$ *defined as* $p\,(A) = \underset{y \in E}{\Sigma} \Psi\,(y)$*. Similarly, the value of a consequent is mapping* $q : F_{\mathcal{B}(X,Y,I)} \rightarrow \mathbb{R}^{+}$ *defined as* $q\,(B) = \underset{y \in F}{\Sigma} \Psi\,(y)$*, where* E *is an antecedent and* F *is a consequent of the attribute implication* $E \Rightarrow F$*.*

The valid implication $E \Rightarrow F$ in the formal context means: When the attributes from E are present in a row, also the attributes from F are present there. One can say that the attributes from E always "brings" the additional attributes from F. Using the antecedent and the consequent values $p(E)$ and $q(F)$, it can be denoted by $p(E)_E \rhd q(F)_F$,which means "The value of E will bring the additional value of F".Taking the attribute implication $\{AC, ABS\} \Rightarrow \{AB\}$ and the above defined attribute measures we get $(Price(AC)+Price(ABS))_{\{AC,ABS\}} \rhd Price(AB)_{\{AB\}}$ which is $1800_{\{AC,ABS\}} \rhd 500_{\{AB\}}$. When the customer buys the car with the *Air Conditioning* and the *Antilock Braking System*, such car will also have the *Air Bag*. When customer have decided to invest 1800 EUR to the *Air Conditioning* and the *Antilock Braking System*, the customer will also invest additional 500 EUR to the *Air Bag*.

Even it is interesting information two problems appears. The first problem is generally the very huge set of the attribute implications. There is a couple of methods, how to select the reasonably big subset of the attribute implications based on several criteria. One criterion say, that the subset must be minimal and all other attribute implication can be derived from this subset. Well-known is Duquenne-Guigues Base. Another reasonable subset of the attribute implications is based on the minimal generators. It's easy to see, that in the formal context $mgen(B) \Rightarrow B$ or $mgen(B) \Rightarrow (B \backslash mgen(B))$ always holds. The advantage of such subset is that each attribute implication deals with the particular intent. The reason is, that the sets of $mgen(B)$ are organized in the equivalence classes induced by $\downarrow\uparrow$. The same it can be done for all attribute implications based on the minimal generators. In this case is not important that such base of attribute implications is not minimal. By using an algorithm for calculating the minimal generators a support (the amount of objects in the corresponding extent) is usually considered. The equivalent class of the trivial intent Y usually consists of a huge amount of minimal generators. Such class is ignored, because a support of the corresponding intent is usually 0. Hence the amount of the frequent minimal generators (with given minimal support ¿ 0) can be reasonable. Another trivial example is $mgen(B) = B$, when we have $mgen(B) \Rightarrow \emptyset$. Such implications are not considered in this paper. The second problem is that not all attribute implications with the values can be significant for the users. For example the attribute implication with value $1800_{\{AC,ABS\}} \rhd 500_{\{AB\}}$ can be interesting by one particular car but amount of such cars in the formal concept can be small. As a solution can be considered the support mentioned above. But it is usually not enough. Financial managers work not only with the natural measures (e.g. the quantity) but also with the financial measures (the price, the sales volume). Even if we have the huge amount of the cheap cars, it can be insignificant comparing to smaller amount of the more expensive cars. Hence we need to use the extent value of the concept, which corresponds to the given attribute implication. The attribute implication $\{AC, ABS\} \Rightarrow \{AB\}$ with the values $1800_{\{AC,ABS\}} \triangleright 500_{\{AB\}}$ has the corresponding formal concept $\langle\{6, 12, 20\}, \{ABS, AC, AB\}\rangle$ with the extent value $v(\{Car6, Car12, Car20\}) = \Sigma\{Price(Car6), Price(Car12), Price(Car20)\} =$ 12 000 + 14 000 + 14 000 = 40 000 EUR (where $\odot = \Sigma$). We can read it as "1800 EUR can bring additional 500 EUR and it corresponds to 3 cars with the total price 40 000 EUR". Hence the financial manager will be interested in such implications only which give the minimal extent value of the corresponding formal concept. Moreover some additional ratios can be defined as, **Consequent Antecedent Ratio: CAR** $= \frac{q(F)}{p(E)}$, **Consequent Antecedent Support Ratio: CASR** $= \frac{q(F)}{p(E)} * |A|$, **Consequent Antecedent Value Ratio: CAVR** $= \frac{q(F)}{p(E)} * v(A)$, where A is an extent of the formal concept. Such ratios have a very clear meaning to the user and can be used for constraining the set of all attribute implications by the one of the ratios mentioned above.

$intent$	$mgen(B) \Rightarrow B\backslash mgen(B)$								
B	$E \Rightarrow F$	$p(E)$	$q(F)$	$w(B)$	$\lvert A\rvert$	$v(A)$	CAR	$CASR$	$CAVR$
$\{1, 2, 3, 4, 5, 6\}$	$\{3, 6\} \Rightarrow \{1, 2, 4, 5\}$	900	2350	3250	1	14000	2, 61	2, 61	36556
$\{1, 2, 3, 4, 5, 6\}$	$\{1, 3, 4\} \Rightarrow \{2, 5, 6\}$	2400	850	3250	1	14000	0, 35	0, 35	4958
$\{1, 2, 3, 5\}$	$\{1, 3, 5\} \Rightarrow \{2\}$	2050	500	2550	2	28000	0, 24	0, 49	6829
$\{1, 2, 3\}$	$\{1, 3\} \Rightarrow \{2\}$	1800	500	2300	3	40000	0, 28	0, 83	11111
$\{1, 2, 4, 5, 6\}$	$\{6\} \Rightarrow \{1, 2, 4, 5\}$	100	2350	2450	2	30000	23, 50	47, 00	705000
$\{1, 2, 4, 5, 6\}$	$\{1, 2, 4\} \Rightarrow \{5, 6\}$	2100	350	2450	2	30000	0, 17	0, 33	5000
$\{1, 4, 5\}$	$\{1, 4\} \Rightarrow \{5\}$	1600	250	1850	3	46000	0, 16	0, 47	7188
$\{2, 3, 4, 5\}$	$\{3, 4, 5\} \Rightarrow \{2\}$	3150	500	3650	2	28000	0, 16	0, 32	4444
$\{2, 3, 4\}$	$\{3, 4\} \Rightarrow \{2\}$	1400	500	1900	5	68000	0, 36	1, 79	24286

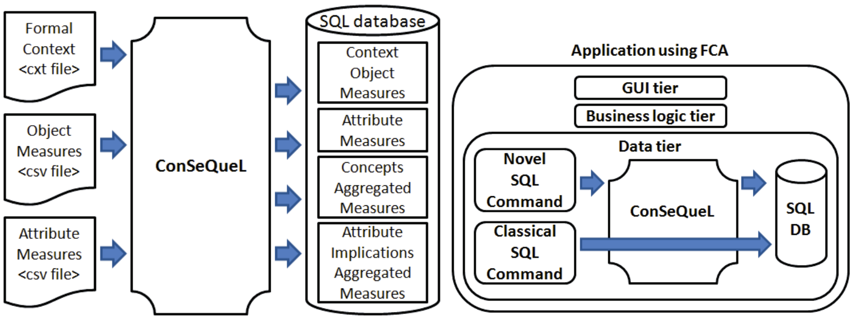
Table 3. Minimal generators, attribute implications, values, ratios

4 SQL Preprocessor

ConSeQueL is the SQL preprocessor based on the theory introduced above. The term "SQL preprocessor" means, that as an input we use novel SQL commands using data files with the formal context (CXT file) and the values (CSV file). After the preprocessing we get the data in relational database, which we can process later by classical SQL commands. The output of such preprocessor can be either collection of INSERT queries without processing them in SQL database (so called SQL dump) or the real data in a database.

4.1 Architecture and Technology

ConSeQuel is designed in C using PostgreSQL ver. 9.0.4 as a relational database. ConSeQueL operates on a command line (shell or cmd). ConSeQueL can be used as a part of an arbitrary application using a relational database and FCA functionality. The architecture is depicted in the Figure 1(i). In the 3-tier software architecture, the ConSeQueL can be inserted in between the logic and the data tier (see Figure 1(ii)). The logic tier is connected to ConSeQueL and uses either classical SQL commands or novel SQL commands (described below). Classical SQL commands are directly send to the SQL database. The novel SQL commands are preprocessed by ConSeQueL and then the results are sent to the SQL database.



(i) Functinality and architecture of ConSeQueL

(ii) ConSeQueL as a part of 3-tier architecture

Fig. 1. ConSeQueL - functionality and use in 3-tier architecture

4.2 Functionality and Novel SQL Commands

ConSeQueL preprocessor uses the following novel SQL commands:

- **CREATE CONTEXT** *contextname*
  FROM *contname.cxt* [*−oval objmeasures.csv*] [*−aval attrmeasures.csv*]
  It process the formal context from inputname.cxt into the database table with a given *contextname*.When the object and/or the attribute measures are assigned to them (up to 5 measures to the object and up to 5 measures to the attribute) they are processed as well.

- **CREATE CONCEPT** *conceptname* **FROM** *contextname*
  It will process the given formal context *contextname* into the collection of the formal concepts with theirs supports (number of objects and/or attributes in each formal concept). This collection called *conceptname* is inserted into the predefined table. When the measures are assigned, they will be processed as well. There are four basic aggregations used namely $SUM()$, $AVG()$, $MIN()$ and $MAX()$ for each assigned measure.

- **CREATE IMPLIC** *implicname* **FROM** *contextname*
  It process the given formal context *contextname* into the collection of the attribute implications. The antecedent (the minimal generator) and the consequent of each attribute implication are calculated as well. When the measures are defined, they will also be processed. Aggregation $SUM()$ is used for each assigned measure. Moreover the ratios CAR, $CASR$ and $CAVR$ are calculated for each assigned measure.

The novel SQL commands result to the data in the predefined tables. The queries on this data can be done via classical SQL commands. The example of the ConSeQuel results can be seen in Table 2 and Table 3.

4.3 Algorithms and Implementation

There are couple of algorithms used for computing the formal concepts. In ConSeQueL our own implementation of FCbO algorithm is used (for more details see [8]). This algorithm is enhanced on output, where the aggregations of the assigned measures are calculated for each computed formal concept. The computation of the minimal generators ensures the implementation of the Prince algorithm [7] (see also Acknowledgements). This implementation was used as it is, with the similar modification as above.

4.4 Database Structure

The above presented SQL commands result to an inserting of the data into the predefined tables. The ConSeQueL consists of 7 tables: 1. **desc-contexts** - mescription of the formal contexts' collections, 2. **context** - collection of the formal contexts and measures assigned to the objects, 3. **values-of-attributes** - measures assigned to the attributes, 4. **desc-concepts** - description of the concepts' collections, 5. **concept** - collection of the concepts including aggregations, 6. **desc-implic** - description of attribute implications' collections and 7. **implic** - Collections of the attribute implications including antecedent (minimal generator), consequent, aggregated values and ratios. The tables "context", "desc-contexts" and "values-of-attributes" are used as an input tables only. The data structure is not designed for efficient querying the objects, attributes, assigned measures and description from these three tables. The main purpose of the ConSeQueL is to compute the concepts and the attribute implications with the aggregated values which are stored in tables "concept" or "implic" respectively. These two tables are designed as data warehouse tables and can be queried very simply (e.g. no joins are necessary). The remaining tables "desc-concepts" and "desc-implic" are descriptive only and using such tables is voluntary. In the Figure 2 the whole database structure is depicted. The selected details can be found in the Table 4.

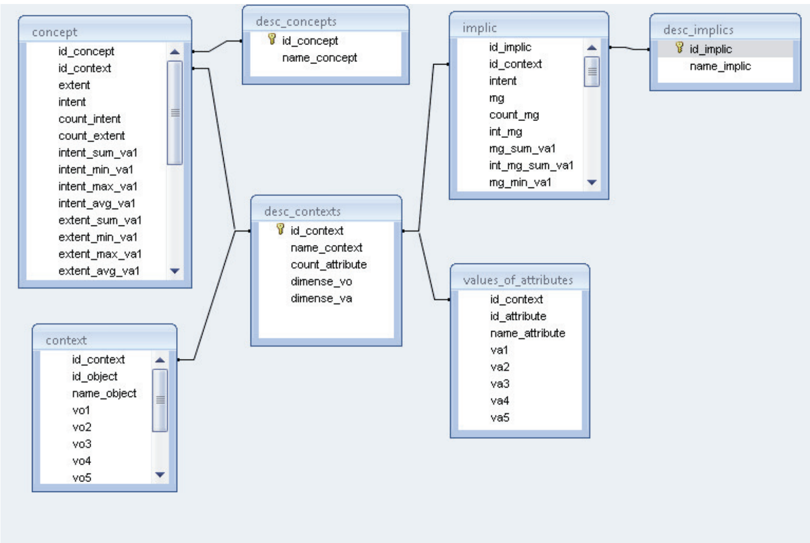


Fig. 2. Database structure of ConSeQueL

table	column	description	data type
concept	extent	extent	text
concept	intent	intent	text
concept	count_intent	amount of the attributes	number
concept	count_extent	amount of the objects	number
concept	intent_sum_va1	sum of measure 1 of all attributes in intent	number
concept	intent_min_va1	minimum of measure 1 of all attributes in intent	number
concept	intent_max_va1	maximum of measure 1 of all attributes in intent	number
concept	intent_avg_va1	average of measure 1 of all attributes in intent	number
concept	extent_sum_va1	sum of measure 1 of all objects in extent	number
concept	extent_min_va1	minimum of measure 1 of all objects in extent	number
concept	extent_max_va1	maximum of measure 1 of all objects in extent	number
concept	extent_avg_va1	average of measure 1 of all objects in extent	number
implic	intent	intent	text
implic	mg	minimal generator, antecedent	text
implic	count_mg	amount of attributes in minimal generator	number
implic	int_mg	consequent	text
implic	mg_sum_va1	sum of measure 1 of all attributes in antecedent	number
implic	int_mg_sum_va1	sum of measure 1 of all attributes in consequent	number
implic	CAR_va1	CAR ratio for measure 1	number
implic	CASR_va1	CASR ratio for measure 1	number
implic	CAVR_va1	CAVR ratio for measure 1	number

Table 4. Selected columns in database structure of ConSeQueL

5 Conclusion and Future Research

The application ConSeQueL is a significant part of the diploma work of Jan Kanovsky and the theory behind will be a part of the doctoral thesis of Juraj Macko. The novelty of ConSeQueL can be seen from the theoretical practical point of view. FCA with measures is generalization for OLAP technology. The attributes in OLAP are organized in dimensions. FCA with measures can use independent attributes [2]. The practical contribution of ConSeQueL can be seen as an application of the FCA with measures. ConSeQueL presented in this paper is the introductory attempt in this field and a couple of open question will be part of the future research (the efficient data structure for attribute values, efficient computing the aggregations etc.) Some experiments are done and authors leave the results for the extended version of this paper. The future perspective is to enhance the existing preprocessor to fuzzy settings and also to measure based constraints [2].

Acknowledgements: The authors would like to thank Dr. Jan Outrata for consulting the efficient implementation of FCbO algorithm. Authors would like to thank also Sadok Ben Yahia and Tarek Hamrouni for Prince implementation [7] available on $http://www.fst.rnu.tn/sbenyahia/software\_release.htm$.

References

1. Ganter B., Wille R.: *Formal Concept Analysis. Mathematical Foundations.* Springer, Berlin, 1999.
2. Macko J.: *Formal Concept Analysis as a Framework for Business Intelligence Technologies.* F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 195–210. Springer, Heidelberg (2012)
3. Wang Z., Klir G.: *Generalized measure theory*, Springer, New York, 2009
4. Calvo T., Kolesárová A., Komorníková M., Mesiar R. *Aggregation operators: Properties, classes and construction methods* Aggregation Operators: New Trend and Applications, p. 3-106 , Eds: Calvo T., Mayor G., Mesiar R., Physica Verlag, (Heidelberg 2002)
5. Maier D.: *The theory of relational databases*, Computer Science Press, Rockville, 1983
6. Belohlavek R., Macko J.: *Selecting Important Concepts Using Weights.* In: P. Valtchev, R. Jäschke (Eds.): ICFCA 2011, LNAI 6628, pp. 65–80, Springer Heidelberg, (2011)
7. T. Hamrouni, S. Ben Yahia, and Y. Slimani: Prince: An algorithm for generating rule bases without closure computations. In Tjoa, A.M., Trujillo, J., eds.: Proceedings of the 7th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2005), Springer-Verlag, LNCS 3589, Copenhagen, Denmark. (2005) 346-355
8. Outrata J., Vychodil V.: Fast Algorithm for Computing Fixpoints of Galois Connections Induced by Object-Attribute Relational Data. Information Sciences 185(1)(2012), pp. 114-127

The Transaction Concept in Enterprise Systems

Simon Polovina

Conceptual Structures Research Group
Communication and Computing Research Centre (CCRC)
Sheffield Hallam University, UK S1 2NU
S.Polovina@shu.ac.uk

Abstract. Many enterprises risk business transactions based on information systems that are incomplete or misleading, given that 80-85% of all corporate information remains outside of the processing scope of such systems. Computer technology nonetheless continues to become more and more predominant, illustrated by SAP A.G. recognising that 65-70% of the world's transactions are run on their technology thus "have to do a good job". Using SAP as the illustrative case study, the benefits of Service-Oriented Architecture (SOA) and associated technologies are explored. Together with Business Process Management / Modelling (BPM), social media, Business Intelligence (BI), semantic, multi-agent systems (MAS) and other technologies, enterprise architecture frameworks, principles such as Moore's core-context, and conceptual structures, a practical roadmap is identified to TOA (Transaction-Oriented Architecture (TOA). TOA picks up POA (Process-Oriented Architecture) along the way, and is predicated on the Transaction Concept (TC). The TC builds upon work in Resources, Events, Agents (REA) and the Transaction Graph (and Lattice). The TC is the essence of enterprise systems that allow SAP, their competitors, customers, suppliers and partners to do an ever better job with the world's transactions.

Keywords: Service-Oriented Architecture (SOA), SAP A.G., Business Process Management / Modelling (BPM), Social Media, Combining and Unifying Business Intelligence with Semantic Technologies (CUBIST), POA (Process-Oriented Architecture), TOA (Transaction-Oriented Architecture (TOA), Resources, Events, Agents (REA), Conceptual Structures, Transaction Graph, Transaction Lattice

1 Introduction

Many enterprises risk business transactions based on information systems that are incomplete or misleading, augmenting the claim that 80-85% of all corporate information remains outside of the processing scope of such systems [1] [2]. As these enterprise systems become more and more predominant the issue becomes increasingly acute; indeed SAP as the significant enterprise systems' vendor have noted that 65-70% of the world's transactions are run on SAP thus "have to do a good job" [3]. Enterprise systems are being expected to align more and more with the essence of the

adfa, p. 1, 2011.

enterprise and through the productivity of computers lever this knowledge about itself and become more successful.

2 Approaches

Accordingly there has been a substantial push to Service-Oriented Architecture (SOA) and an eco-system that in SAP's case is epitomised by the Enterprise Services Workplace (ESW, http://esworkplace.sap.com). Allied to these approaches has been the integration of Business Intelligence (BI) that, continuing with SAP as the exemplar, has been the emergence of the High-Performance Analytic Appliance (SAP HANA, http://sap.com/hana) architecture. A continuation of BI is to apply semantic technologies that structure unstructured data. These information extraction technologies take knowledge management a stage further by discovering knowledge hitherto hidden in that data, thereby capturing much more of that elusive 80-85% of corporate information. The Combining and Unifying BI with Semantic Technologies project (CUBIST, www.cubist-project.eu) is an exemplar of extracting meaning from structured and unstructured data to discover knowledge.

2.1 Service-Oriented Architecture (SOA)

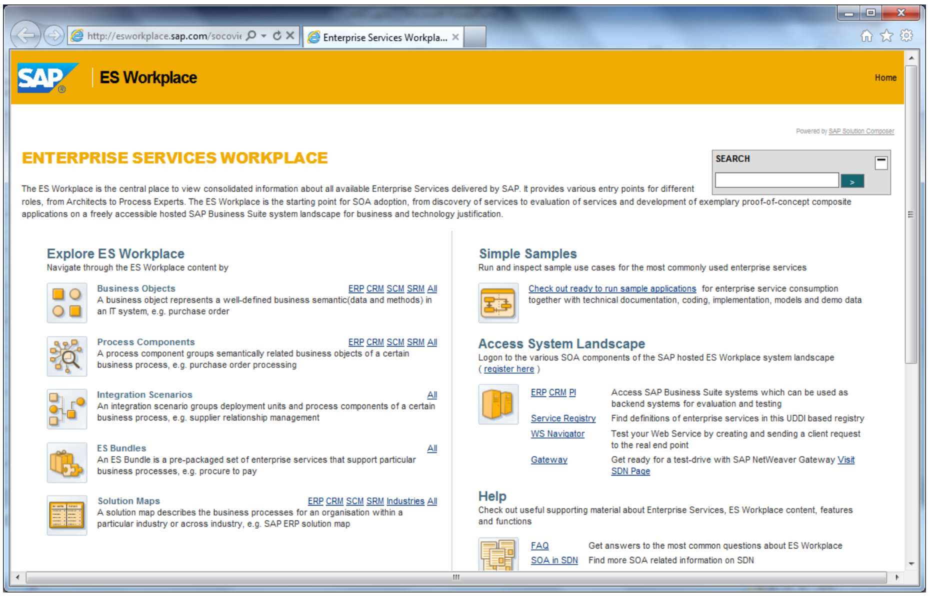
SOA recognises the limitations of existing enterprise applications that have been built along the lines of large functional silos such as Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Supplier Relationship Management (SRM), Financials, or even for specific markets (the so-called 'verticals' applications) such as Oil and Gas, Healthcare, Banking, Telecommunications or Public Sector.

Whilst all these applications reflect actual applications than a technology seeking an application (e.g. Database Systems), the semantics (or 'meaning') of business activity are at a much lower granular level than those applications imply. Rather, like object-oriented approaches, business activity is made up of a number of service components, namely 'business objects' that can be orchestrated into business processes according to the business requirements. With intermediate levels of orchestration of these objects into process components that in turn can become part of deployment units, enterprise applications are individually configured in a way that better aligns with the given business need. Standardisation is achieved at the component rather than the application level, thus taking advantage of reusability. Agility is achieved by re-orchestrating or enabling new components in direct response to changing business needs. To allow flexible orchestration and re-orchestration, the service interfaces of each business object are also defined according to a standardised governance process. Even the data of each business object are built according to Global Data Types (GDTs).

All in all, SOA provides an operational architecture that makes component based software development become realistic for enterprise systems. SOA development distinguishes itself from object-orientation to the extent that each component is centred on providing a *service*; in our case a composite element of business semantics that *adds value* to the enterprise application. The technical nature of each component is encapsulated by its business meaning, thus can be directly applied by Business Process Management (BPM) and its associated Business Process Modelling (whose acronym is also BPM) to orchestrate business processes.

2.2 Enterprise Services Workplace (ESW)

To operationalise SOA (i.e. to make it possible) many vendors and their partners and customers have recognised that the definition and service interfaces of the many resulting business objects cannot be conceptualised, developed and implemented by the vendors alone. Rather it requires a collaborative eco-system that in SAP's case is epitomised by the Enterprise Services Workplace (ESW, http://esworkplace.sap.com), Figure 1. As such, vendors, partners, customers and anyone essentially can contribute to the construction of the SOA. Many components can draw on the vendors established expertise, as a 'de-assembly' of their existing enterprise applications or the creation of 'enterprise services'. Some components may only consist of their interfaces defined in WSDL, at least describing to this extent the business semantics of that component using the Web Services recommendations that SOA is conventionally based upon.


http://esworkplace.sap.com/socovie
Enterprise Services Workpla...
SAP
ES Workplace
Home
Powered by SAP Solution Composer
SEARCH
>

ENTERPRISE SERVICES WORKPLACE

The ES Workplace is the central place to view consolidated information about all available Enterprise Services delivered by SAP. It provides various entry points for different roles, from Architects to Process Experts. The ES Workplace is the starting point for SOA adoption, from discovery of services to evaluation of services and development of exemplary proof-of-concept composite applications on a freely accessible hosted SAP Business Suite system landscape for business and technology justification.

Explore ES Workplace
Navigate through the ES Workplace content by

Business Objects — ERP CRM SCM SRM All
A business object represents a well-defined business semantic(data and methods) in an IT system, e.g. purchase order

Process Components — ERP CRM SCM SRM All
A process component groups semantically related business objects of a certain business process, e.g. purchase order processing

Integration Scenarios — All
An integration scenario groups deployment units and process components of a certain business process, e.g. supplier relationship management

ES Bundles — All
An ES Bundle is a pre-packaged set of enterprise services that support particular business processes, e.g. procure to pay

Solution Maps — ERP CRM SCM SRM Industries All
A solution map describes the business processes for an organisation within a particular industry or across industry, e.g. SAP ERP solution map

Simple Samples
Run and inspect sample use cases for the most commonly used enterprise services

Check out ready to run sample applications for enterprise service consumption together with technical documentation, coding, implementation, models and demo data

Access System Landscape
Logon to the various SOA components of the SAP hosted ES Workplace system landscape (register here)

ERP CRM PI	Access SAP Business Suite systems which can be used as backend systems for evaluation and testing
Service Registry	Find definitions of enterprise services in this UDDI based registry
WS Navigator	Test your Web Service by creating and sending a client request to the real end point
Gateway	Get ready for a test-drive with SAP NetWeaver Gateway Visit SDN Page

Help
Check out useful supporting material about Enterprise Services, ES Workplace content, features and functions

FAQ	Get answers to the most common questions about ES Workplace
SOA in SDN	Find more SOA related information on SDN

Fig. 1. Enterprise Services Workplace (ESW)

2.3 Business Intelligence

In SOA, business objects are conceptualised through the collaboration of the participants as SAP describes in an 'ecosystem' (www.sap.com/partners/ecosystem). This ‘top down’ process might be integrated with ‘bottom up’ knowledge discovery from the data itself. The patterns of the behaviour of the data (e.g. sales figures for a given set of customers in a product-market) is interpreted through a query of that data i.e. Business Intelligence (BI). Latterly there has been a bringing closer together of SOA and BI, with the latter’s in-memory analytics. The former (SOA) as enterprise applications has traditionally been structured according to online transaction processing (OLTP) whereas the latter (BI) relies on online analytical processing (OLAP), resulting in an overhead of de-normalising data from the former and creating data cubes to permit the latter to take place. By enabling this to take place in computer memory rather than disk access, information can be requested and responded to instantaneously as SAP states in “real real time” through its High-Performance Analytic Appliance (HANA, http://sap.com/hana) architecture. HANA will go beyond the simple placing of OLTP and OLAP together in-memory into a much more integrated architecture that reflects SAP’s now ongoing commitment to offer “innovation without disruption”.

2.4 Semantic Technologies

To assist BI, semantic technologies such as those being investigated in CUBIST can extract the information from both structured and unstructured data. It enables more informed querying to take place as well as discover hitherto hidden meaning from the data. In CUBIST this is envisaged through an integration of Semantic Web technology through OWLIM (www.ontotext.com/owlim) with Formal Concept Analysis (FCA, www.fcahome.org.uk), and tested on data use cases in i) bioinformatics, ii) space telemetry and iii) matching career opportunities to candidates. FCA is an automated technique that identifies the conceptual structures among data sets. FCA is a formal method as it mathematically discovers the concepts from the patterns in the data according to the objects and attributes that make up that data. Moreover, these formal concepts are related to other formal concepts in a lattice structure (known as Galois connections). Through these interrelated formal concepts, FCA thus has the potential to complement the cognitive conceptualisation of business objects and service interfaces with those that are machine-generated from data.

3 Automation in BPM

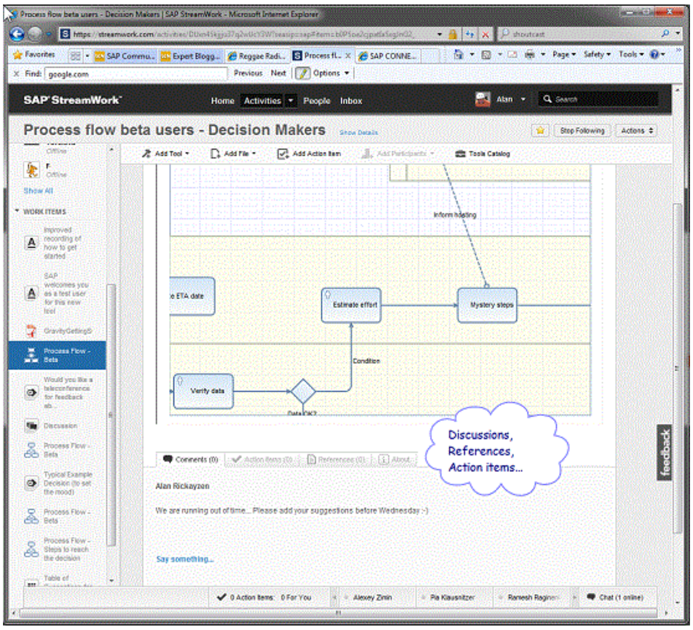
From the above approaches, we can begin to appreciate how enterprise systems will more expressively align with the enterprises they are meant to support and enhance. As these approaches become more and more established, more and more enterprises will have systems that at last can record the business transactions that embody the purpose of the enterprise. As the approaches become established technologies, business will begin to take them for granted. This may be the height of enterprise systems, but it is possible to foresee that it will in turn offer new opportunities for enterprises. One evident possibility is introducing computer automation into the hitherto human-centric businesses process orchestration itself though software agents as described shortly. This potential development could be built upon the existing developments in the use of social media to facilitate collaborative BPM. In SAP, that existing development is their StreamWork^{TM} technology.

3.1 Social Media and BPM

SAP's StreamWork (Figure 2) provides a representative use case how social media can be applied to BPM. Working with partners well-known in this field such as Google, Novell, Evernote, Scribd, and Box, it takes the general collaborative features found in social media technologies and augments them with BPM tools such as Business Process Modelling Notation (BPMN). It thus as described earlier brings SOA into the suite of tools available in social media. Let us therefore begin to outline the extent of automation in the future stages of these tools' development.

3.2 Adding Software Agents, Conceptual Structures

The presence of BPM tools in social media has thereby enabled collaborative computer-mediated BPM to take place, thus opening the way for more technologies to be integrated in this environment. One pertinent route is the incorporation of software agents as partners in the collaborative process. These agents would not only bring the productivity of computers as counterpart to the creativity of the human experts in the BPM process, these software agents can search and appropriate the many resources of the Web, Intranets and Internet and distil their findings to the benefit of the human collaborators.



Alan Rickayzen

We are running out of time... Please add your suggestions before Wednesday :-)

Fig. 2. SAP StreamWork^{TM}

Conceptual Structures (CS) might also be brought into BPM. Notably, CS is about technologies that "harmonises the creativity of humans with the productivity of computers. CS recognise that organisations work with concepts; machines like structures". CS "advances the theory and practice in connecting the user's conceptual approach to problem solving with the formal structures that computer applications need to bring their productivity to bear in solving these problems" (http://extra.shu.ac.uk/iccs2007). CS enables "Knowledge Architectures [that] give rise to smart applications that allow enterprises to share meaning across their interconnected computing resources and to realize transactions that would otherwise remain as lost business opportunities." [4].

4 Conceptual and Architectural Frameworks

Whilst there are expressive CS tools such as Conceptual Graphs (CGs) and Formal Concept Analysis (FCA), ISO Common Logic, as well as Semantic Web technologies based on such as Description Logics (DL) and others such as FLogic (e.g. by Ontoprise, www.ontoprise.com) or Datalog, the agents would need a frame of reference in order to fulfil their role as knowledgeable providers.

For this purpose the human participants may refer to Enterprise Architecture Frameworks such as TOGAF (in SAP's case enhanced by the SAP EAF) or Zachman, or Principles like Moore's Core-Context [5]. Working with concepts, the human experts (e.g. Enterprise or Solution Architects, or Business Process Experts) would appreciate that "all models are wrong, but some are useful" [5, p. 424]. Put simply, they would see them as frameworks that offer solutions that recognise "It is better to be vaguely right than exactly wrong" [6, p. 272]. There thus remains an element of human intuition that, as evidenced in experiences such as those from Artificial Intelligence, cannot easily be computer programmed thus also remain outside of enterprise systems. The software agents therefore have to overcome to some useful degree their computer-based limitations to be effective participants

Multi-Agent Systems (MAS) offer one promising route forward. One avenue of research has how they might be deployed in enterprise systems through Resources, Events, Agents (REA) [7]. Other work has taken on this approach and, using CGs, provided an early requirements capture specification, known as *Tr*ansaction *A*gent *M*odelling (TrAM) [8]. A variant of this work has been the Transaction Graph, using CGs to apply TrAM in Enterprise Architecture projects [10]. This work augments Enterprise Architecture Frameworks so that semantic enterprise applications can be built. Some of this work is illustrated by a Transaction Graph (in CGs) and Transaction Lattice (in FCA) for a University case study is shown by Figure 3 [9]. Other work describes a Financial Trading case study as well as for health, mobile services, manufacturing and in learning [2]. Work has begun towards rather an ambitious project using ISO Common Logic for Open Semantic Enterprise Architecture (OpenSEA, www.open-sea.org).

5 The Transaction Concept

The term 'transaction' appears in a number of the above works. They demonstrate 'Transaction' as a high-level declarative statement that identifies the enterprise itself rather than a number of lower-level transactions that support its business processes (or, in SAP's terms the transactions that make up its ERP and other systems). Rather, the Transaction is a concept that restates the enterprise's mission statement, but presenting it in a balanced way that shows what an enterprise is willing to sacrifice ('pay') to satisfy the desires in its mission statement. It captures the fact that enterprises do not always seek to maximise their profit in purely monetary ways. Even many outwardly profit-oriented enterprises present their mission statements in qualitative ways (e.g. quality of service, duty to all stakeholders, society, and reputation to name a few).

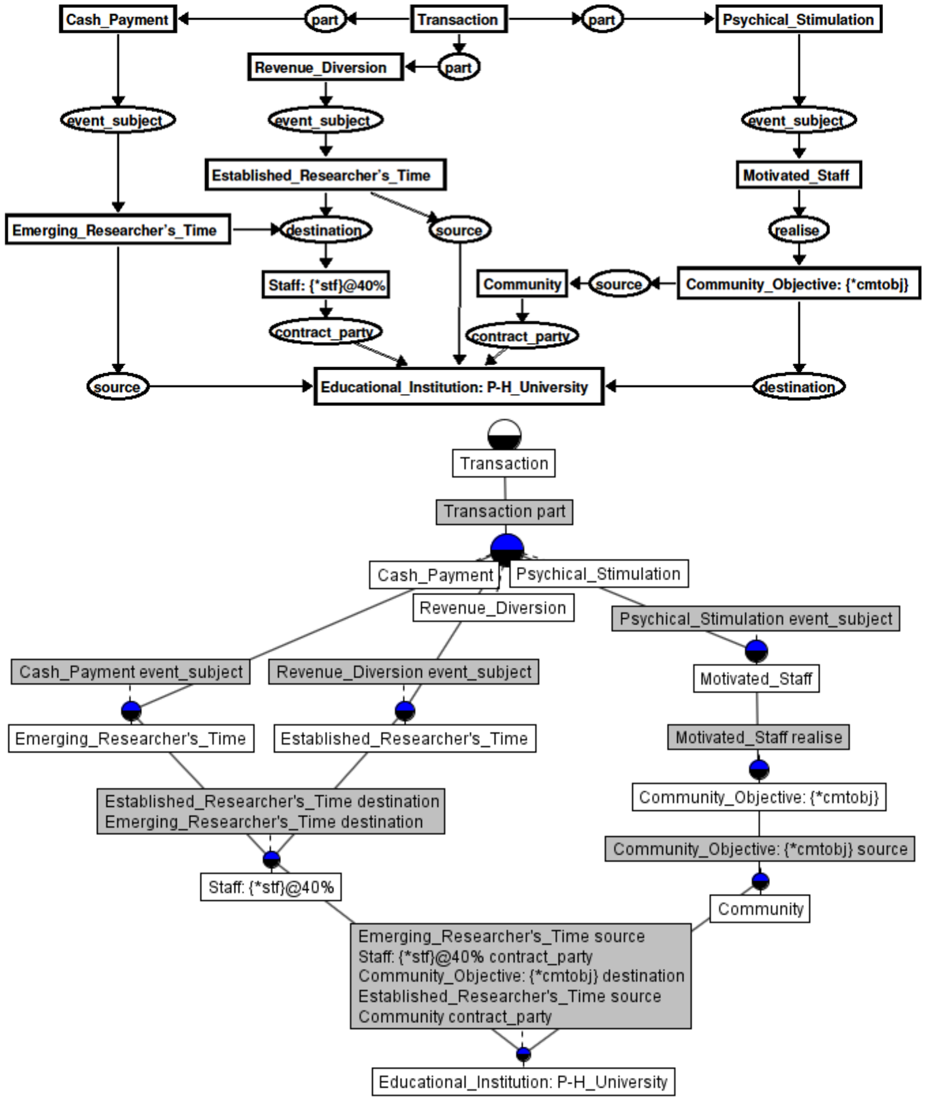

Cash_Payment
part
Transaction
part
Psychical_Stimulation
part
Revenue_Diversion
event_subject
event_subject
event_subject
Established_Researcher's_Time
Motivated_Staff
Emerging_Researcher's_Time
destination
source
realise
Staff: {*stf}@40%
Community
source
Community_Objective: {*cmtobj}
contract_party
contract_party
source
Educational_Institution: P-H_University
destination
Transaction
Transaction part
Cash_Payment
Psychical_Stimulation
Revenue_Diversion
Psychical_Stimulation event_subject
Cash_Payment event_subject
Revenue_Diversion event_subject
Motivated_Staff
Emerging_Researcher's_Time
Established_Researcher's_Time
Motivated_Staff realise
Community_Objective: {*cmtobj}
Established_Researcher's_Time destination
Emerging_Researcher's_Time destination
Community_Objective: {*cmtobj} source
Staff: {*stf}@40%
Community
Emerging_Researcher's_Time source
Staff: {*stf}@40% contract_party
Community_Objective: {*cmtobj} destination
Established_Researcher's_Time source
Community contract_party
Educational_Institution: P-H_University

Fig. 3. Transaction Graph (top diagram), Lattice (bottom diagram)

Whilst possibly confusing when we think of transactions, this is consistent with the concept of a transaction given by the dictionary definitions of this term (e.g. www.merriam-webster.com/dictionary/transaction). We may therefore make a distinction by using Transaction with an initial uppercase T as opposed to transaction beginning with a lowercase 't' (i.e. Transaction vs. transaction; the overarching strategic Transaction(s) that epitomises the very enterprise itself as opposed to the many day-to-day system level transactions). It may be viewed as roughly analogous to 'cloud' or 'kite' (or business-level) use cases vs. 'sea-level'/'fish-level' system use cases (http://www.youtube.com/watch?v=FNkuGEr1gB4). To explicate the Transaction (note uppercase initial T), a Transaction Use Case Diagram (TUCD) exists [10].

Given that even the use of uppercase T vs. lowercase t as a device may still provide an inadequate level of distinction, and as supported by the foregoing discussion, the term Transaction Concept (TC) is given to mean *T*ransaction.

5.1 The Semiotic Ladder

The dimensions of the TC can further be illustrated by the Semiotic Ladder, Figure 4.

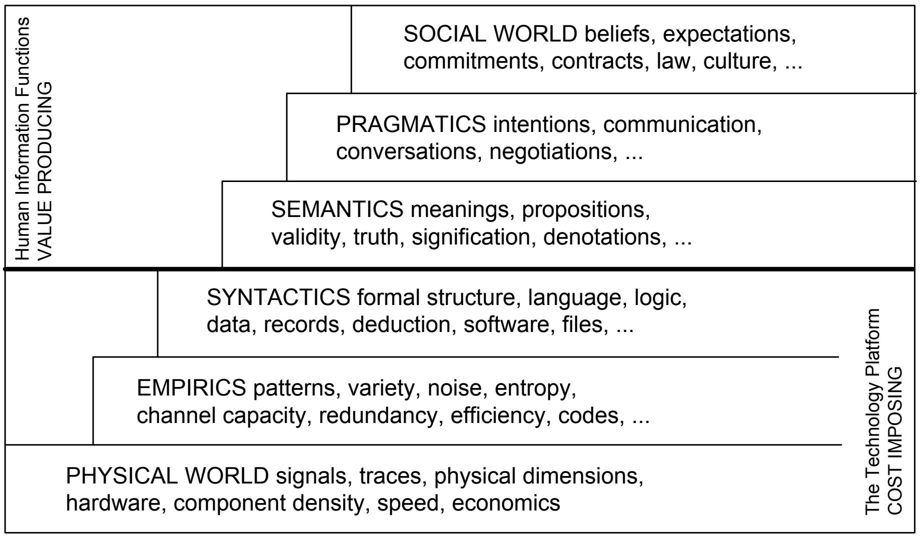

Human Information Functions
VALUE PRODUCING
SOCIAL WORLD beliefs, expectations, commitments, contracts, law, culture, ...
PRAGMATICS intentions, communication, conversations, negotiations, ...
SEMANTICS meanings, propositions, validity, truth, signification, denotations, ...
SYNTACTICS formal structure, language, logic, data, records, deduction, software, files, ...
EMPIRICS patterns, variety, noise, entropy, channel capacity, redundancy, efficiency, codes, ...
PHYSICAL WORLD signals, traces, physical dimensions, hardware, component density, speed, economics
The Technology Platform
COST IMPOSING

Fig. 4. The Semiotic Ladder

Essentially, the 'cost imposing' layers in this ladder structure illustrate the areas in which the productivity of computers ('the technology platform') benefit information (enterprise) systems much better than manual systems. It is where computers are much better than humans, hence the success of technologies such as data processing that nowadays we cannot imagine being without. The 'value producing' layers are where humans are better than computers. Classical experiences from Artificial Intelligence have shown how poorly computers perform in these areas, but more recent technologies such as knowledge management systems, and SOA/BPM with MAS as described can play a pertinent role.

Enterprise Architecture Frameworks are intended to capture the Enterprise holistically, in line with an Architect's remark when asked what Architects do i.e. "from a blank sheet of paper to the position of last nail in the wall". Essentially, these frameworks cover the whole range of the ladder. Transactions too are wide-ranging, from those supporting ACID (Atomicity, Consistency, Isolation, Durability) at the database level and the transactions in SAP systems, through to business transactions denoting an agreement between a buyer and a seller to exchange an asset for payment through to the TC itself. Accordingly, transactions (from little t to Big T) transcend the steps of the ladder, thereby Enterprise Architecture Frameworks.

Transactions describe why the enterprise exists. It explicates what the enterprise offers, what it desires in return, and the assessed risks in achieving these rewards. It gives the enterprise its sense of direction; consequently it can give enterprise systems the same direction, in line with the expectations of the semiotic ladder.

5.2 Transaction-Oriented Architecture (TOA)

A Transaction-Oriented Architecture (TOA) provides the framework by which an enterprise's business processes are orchestrated according to the TC. The TOA brings purpose and direction to SOA, further assisted by Process-Oriented Architecture (POA), which offers a reference architecture by which SOA can be orchestrated according to business processes, for example the way that Business Objects in SAP are orchestrated into Process Components. TOA culminates SOA and enterprise applications' productivity including SAP's to the height of the real, transactional world that enterprises operate.

Figure 5 illustrates TOA as the capstone of a pyramid. It shows how the ESW underpins SOA, which is additionally supported through the discovery of new Business Objects from BI knowledge discovery projects like CUBIST. SOA and POA are supported by social media and BPM, illustrated by SAP's StreamWork technology. MAS can help automate the orchestration of SOA and POA. These software agents may act as software partners to the human participants in simulating drafts of processes and interactively feed back the extent to which they exemplify Moore's Core-Context Principles in a given POA project. Thus, like the Semantic Web, technology meaningfully enters the domain of the hitherto alien territory of human information functions in the semiotic ladder. The TOA uses the Transaction Graph and the Transaction Lattice (here for a Financial Trading case study [10]). Using REA and TrAM, MAS is in TOA too. Technology's usefulness is driven up the semiotic ladder.

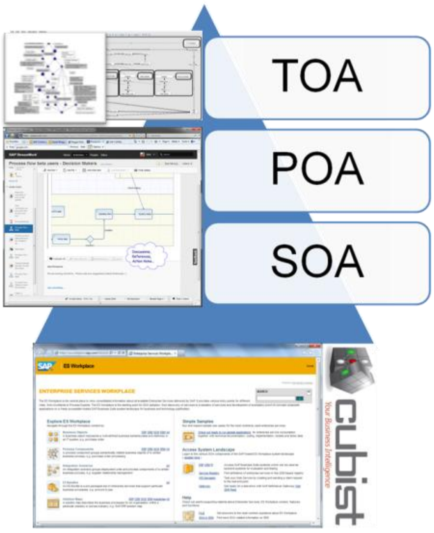


Fig. 5. The TOA

6 Concluding Remarks

TOA offers a practical roadmap for the future development of SOA, BPM, BI, Social Media, MAS, Semantic Technologies and Conceptual Structures. It supports the development of Architectures for Enterprise Applications that ameliorate the 80-85% of corporate information that remains outside of the processing scope of existing enterprise systems. By supporting the expressivity of Enterprise Architecture Frameworks with the described technologies, technology – the productivity of computers – enters the human information functions in the semiotic ladder. Computer productivity is merged with human creativity, reflecting Moore's Core-Context principles that enterprises are distinguished from each other by the creativity of the human participants. SOA has become a reality, exemplified by SAP StreamWork and the ESW. With associated developments in POA and the emerging Semantic and MAS Technologies, we can begin to envisage Conceptual Structures that better incorporate that missing 80-85%. And with the TC as its heart, allow SAP, its competitors, customers, suppliers and partners to do an ever better job with the world's transactions.

7 References

[1] D. &. R. J. Seidman, "Preface," *IBM Systems Journal,* vol. 43, no. 3, p. 449, 2004.

[2] S. Polovina and S. Andrews, "A Transaction-Oriented Architecture for Structuring Unstructured Information in Enterprise Applications," in *Intelligent, Adaptive and Reasoning Technologies: New Developments and Applications*, Hershey, IGI-Global, 2011.

[3] Forbes, "It Doesn't Take Two Years to Create a Good Strategy," Forbes, 10 07 2011. [Online]. Available: http://www.forbes.com/sites/sap/2011/10/07/saps-bill-mcdermott-it-doesnt-take-two-years-to-create-a-good-strategy/. [Accessed 01 03 2012].

[4] U. Priss, S. Polovina and R. Hill, "Preface," in *Conceptual Structures: Knowledge Architectures for Smart Applications, LNAI 4604*, Heidelberg, Springer, 2007, p. i.

[5] G. Moore, Living on the Fault Line, New York: Harper Business, 2002.

[6] G. Box and N. Draper, Empirical Model-Building and Response Surfaces, Chichester, UK: Wiley Series in Probability and Statistics, 1987.

[7] C. Read, Logic Deductive and Inductive, London: Grant Richards, 1898.

[8] D. Vymětal and C. V. Scheller, "MAREA: Multi-Agent REA-Based Business Process Simulation Framework," in *ICT for Competitiveness 2012*, Karviná, Czech republic, 2012.

[9] R. Hill, S. Polovina and D. Shadija, "Transaction Agent Modelling: From Experts to Concepts to Multi-Agent Systems," in *Conceptual Structures: Inspiration and Application, LNAI 4068*, Heidelberg, Springer , 2006, pp. 247-259.

[10] I. Launders, The Transaction Graph: Requirements Capture in Semantic Enterprise Architectures, Saarbrücken, Germany: Lambert Academic Publishing, 2012.

[11] S. Andrews and S. Polovina, "A Mapping from Conceptual Graphs to Formal Concept Analysis," in *Conceptual Structures for Discovering Knowledge, LNAI 6828*, Heidelberg, Springer, 2011, pp. 63-76.