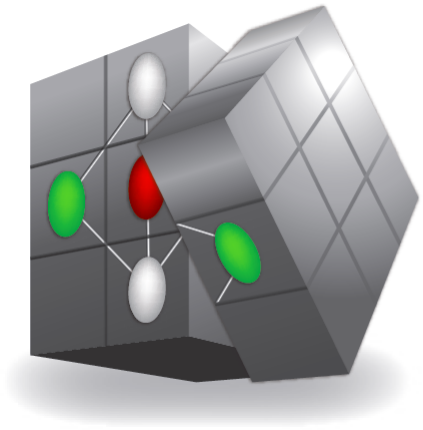
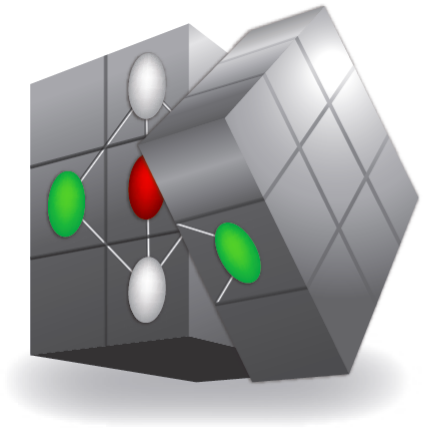


cubist

Your Business Intelligence

Project Overview



cubist

Your Business Intelligence

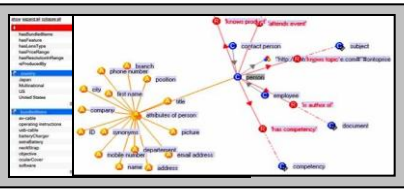
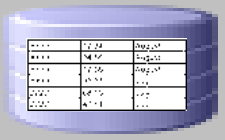
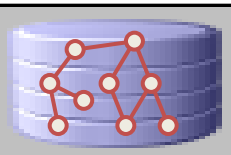
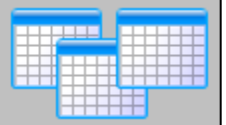
CUBIST in a nutshell: Developing an approach for semantic and user-friendly Business Intelligence by

- **augmenting Semantic Technologies with BI capabilities, and**
- **providing conceptually relevant and user friendly visual analytics.**

Motivation

- Increased proportion of unstructured data (>80%)
  - Not accessible for classical BI solutions
  - Can be better leveraged by means of Semantic Technologies (ST)
- Insufficient user interfaces for Business Intelligence (BI)
  - Improved visual analytics, based on Formal Concept Analysis (FCA), for qualitative Data Analysis
  - Complementing to existing approaches for quantitative Data Analysis

CUBIST Main Idea
From classical to semantic BI

Classical Business Intelligence		Semantic Business Intelligence
	**Output**	 
restricted queries / analytics	**User Interaction**	flexible and visual queries / analytics
Data Warehouse 	**Store**	**Triple Store** 
ETL	**Gathering Information**	Semantic ETL
 databases	**Data sources**	 databases  Forums, blogs  Office docs

CUBIST Main Idea
From classical to semantic BI

CUBIST: Developing an approach for semantic and user-friendly BI

Providing conceptually relevant and user friendly visual analytics.

- Formal Concept Analysis / Galois Lattices
- Faceted navigation
- Graph-based navigation

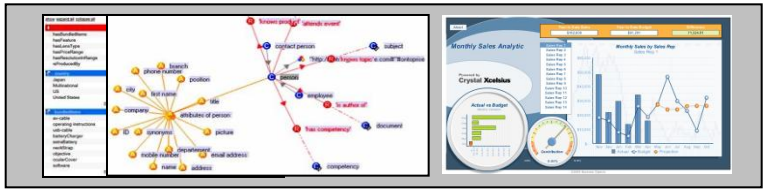
Augmenting Semantic Technologies with BI capabilities

- Triple store as persistency layer
- Flexible Data Warehouse design
- Extending SPARQL with OLAP functionalities
- Reasoning / Deriving implicit facts

Federating data from both unstructured and structured sources

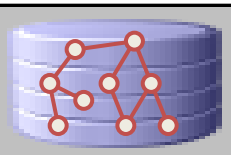
- Enhanced ETL
- Text Mining
- Information Extraction

Semantic Business Intelligence



flexible and visual queries / analytics

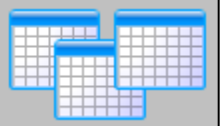
Triple Store



Semantic ETL


databases


Forums, blogs


Office docs

Comparing classical BI with ST

Feature	Classical BI	Semantic Technologies
Federating data from unstructured and structured sources	bad	good
Explicit meaning of data / explicit relations	bad	good
Querying/accessing huge amounts of data	good	average
OLAP functionalities	good	bad
Agile DW schema	bad	good
Inferring implicit facts	bad	good
ETL flexibility	bad	average
Data reuse	bad	good
Use of external datasources / data location independence	bad/average	good

Research Challenges

UI challenges for FCA-based VA

- Look like keyword but semantic search front-ends
- Faceted navigation
- Graphical UI to build graph pattern queries

Technical challenges for FCA-based VA

- Parallel processing algorithms
- Algorithms for many-core architectures
- Making use of Single Instruction-Multiple Data (SIMD) architecture

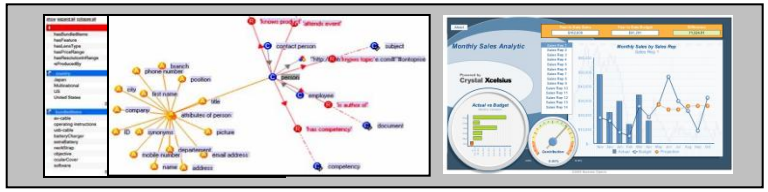
BI enabled Triple Store

- No existing query language: extending SPARQL with OLAP functionalities
- Performance and scalability of triple store

Semantic ETL

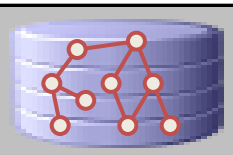
- Quality of source data: particularly “noisy” unstructured data
- Quality of the extraction process: uncertainty of data has to be addressed
- Identity resolution
- Federating unstructured and structured sources

Semantic Business Intelligence



flexible and visual queries / analytics

Triple Store



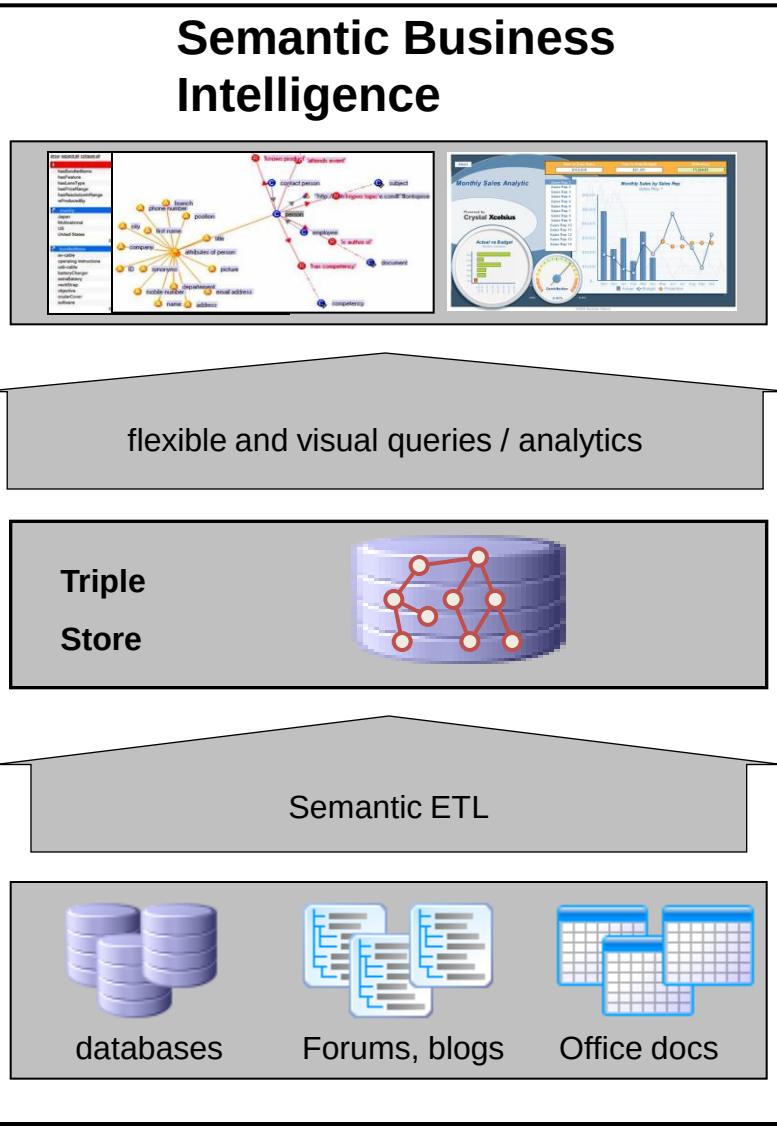
Semantic ETL

Expected Results

FCA-based VA
- Performance formal concept miner
- Large scale FCA capability
- FCA visualization of BI

BI enabled Triple Store
- OLAP extensions for SPARQL
- Discovery mechanism for finding implicit information
- Triple store based information warehouse
- Integration of triple store with FCA

Semantic ETL
- Semantically enriched lineage information
- Error detection within extracted data and CUBIST information warehouse
- Semantic ETL component that provides SPARQL endpoints for various data sources



CUBIST – Project Details

Instrument

- Instrument: STREP
- Theme: ICT-2009-4.3
- Call: FP7 Call 5
- Lead: SAP Research
- Duration: 36 Months
- Start: 2010/10

Consortium

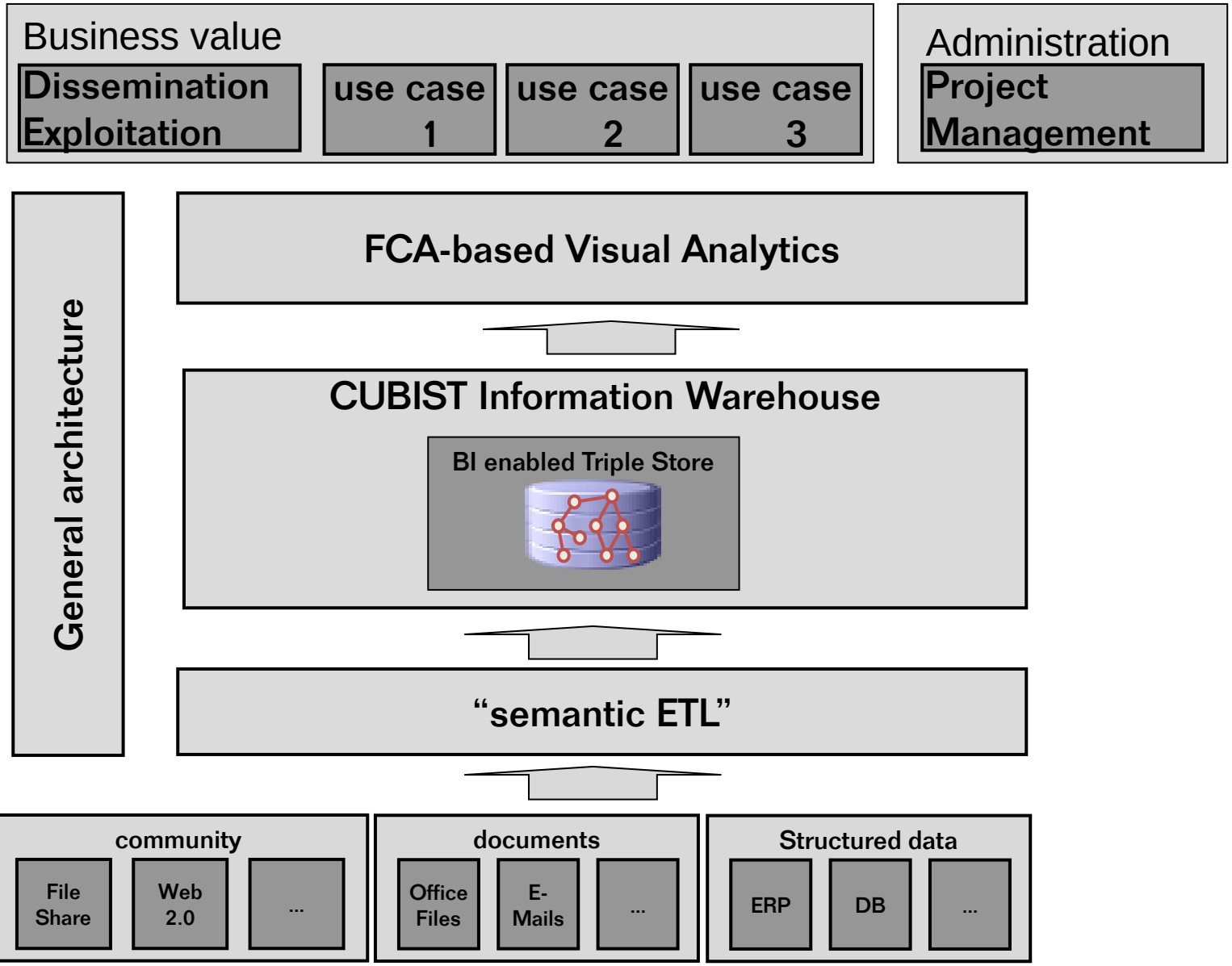
Technological Partners

- SAP (Germany)
  - Coordinator and technological partner
- Ontotext (Bulgaria)
  - Expertise in Semantic Technologies
- Sheffield Hallam University (UK)
  - Expertise in FCA
- Centrale Rechereche S.A. (France)
  - Expertise in FCA and Visual Analytics

Use Case Partners

- Heriot-Watt University (UK)
- Space Applications Services (Belgium)
- Innovantage (UK)

CUBIST Highlevel Architecture



Data Preparation

Data Preparation

Semantic ETL

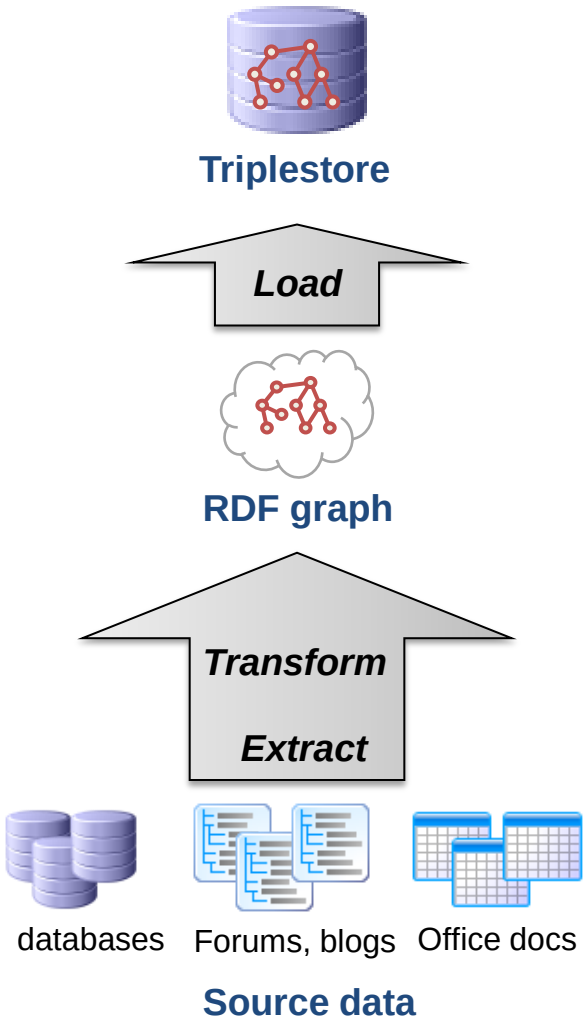
- refers to Extract, Transform, Load (ETL) of “classical BI”
- Extract: from structured & unstructured data sources
- Transform: to triple/ graph data
- Load: into triple store

Semantic ETL from structured data sources

- Mostly transformation from databases to triple data
- Relational DB to RDF Mapping Language (R2RML)
  - Db2triples from antidot
- Data Integration tools
  - Talend Open Studio for Data Integration

Semantic ETL from unstructured data sources

- Text Mining, Semantic annotations



Formal Concept Analysis

What is Formal Concept Analysis?

Formal Concept Analysis is:

- a mathematization of the philosophical understanding of concepts
- a human-centered method for conceptually clustering and structuring data
- a method to visualize data and its inherent structures, implications and dependencies

FCA in three Minutes (i)

How can we describe the concept "BI products from SAP"?

- Extensionally by enumerating all ***objects***:
  - BO Xcelsius, BO Crystral Reports, …
- Intensionally through ***attributes***:
  - "is an SAP product", "is a BI tool", …

Generally, a ***concept*** is divided into two mutually dependent parts:

- Its ***extension*** are all objects that share all the attributes of the concept,
- Its ***intension*** are the attributes which precisely describe the objects of the concept.

The concepts form a hierarchy: A concept C1 is a ***subconcept*** of C2, iff

- the extension of C1 is a subset of the extension of C2
- the intension of C2 is a subset of the extension of C1

} equivalent

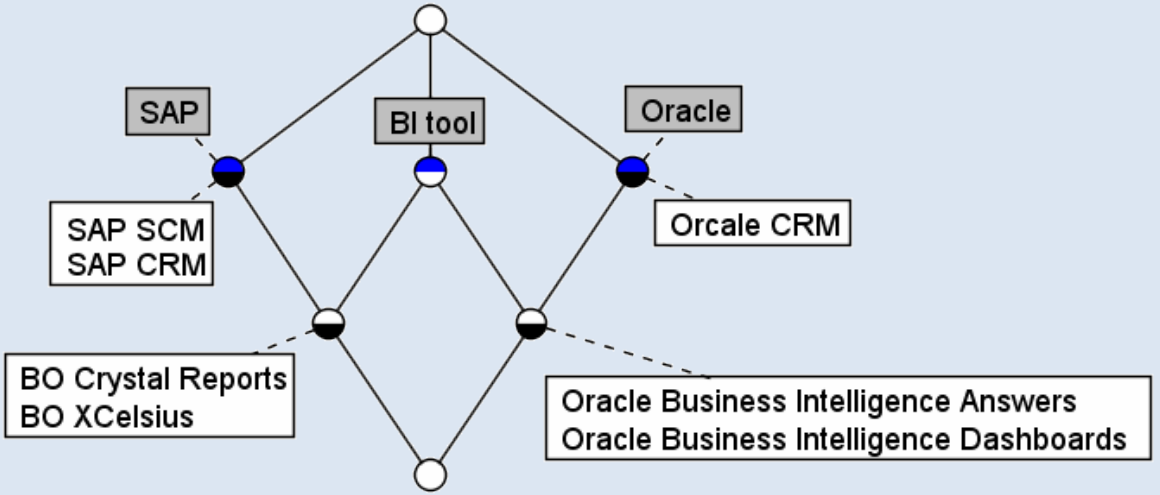
Theorem: For a given universe of discourse, the concept hierarchy is a complete lattice

FCA in three Minutes (ii)

A toy **formal context**

	SAP	Oracle	BI tool
BO XCelsius	X		X
BO Crystal Reports	X		X
SAP CRM	X		
SAP SCM	X		
Oracle Business Intelligence Dashboards		X	X
Oracle Business Intelligence Answers		X	X
Orcale CRM		X	
Oracle Fusion GRC		X	

Its derived **concept lattice**



Small, Real Example Context: Feature Comparison Matrix

The table below is to be visualized as a concept lattice.

Table 1: SAP® Crystal Reports® Software Comparison Matrix

High-Productivity Report Creation Features	2008 and 2011	Version for Visual Studio 2010	Version for Eclipse	XI	10	9	8.5
Wizards and experts for report creation	D	D	D	D,P,S	A,D,P,S	A,D,P,S	D,P,S
Database expert for graphical table linking	D	D	D	D,P,S	A,D,P,S	A,D,P,S	D,P,S
Field explorer to manage report fields	D	D	D	D,P,S	A,D,P,S	A,D,P,S	D,P,S
Drill down in runtime	D	D	D	D,P,S	A,D,P,S	A,D,P,S	D,P,S
Autosave	D			D,P,S	A,D,P,S	A,D,P,S	D,P,S
Editable preview window	D		D	D,P,S	A,D,P,S	A,D,P,S	D,P
Browse field data	D	D	D	D,P,S	A,D,P,S	A,D,P,S	D,P,S
Move, resize, and multiselect objects	D	D	D	D,P,S	A,D,P,S	A,D,P,S	D,P,S
Custom templates	D			D,P,S	A,D,P,S	A,D,P,S	
Repository for component reuse	D			D,P	A,D,P	A,D,P	
Workbench tool for managing projects	D			D,P,S			
Automatic patch notification and installation	D	D	D	D,P,S			
Start page to stay connected with latest information	D	D	D	D,P,S			
HTML preview	D	D	D	D,P			
Extension points for custom wizards, toolbar buttons, and designer events			D				
Extension points for toolbar buttons	D						
Multilingual report design and view	D						
Dual monitor support	D	D	D				

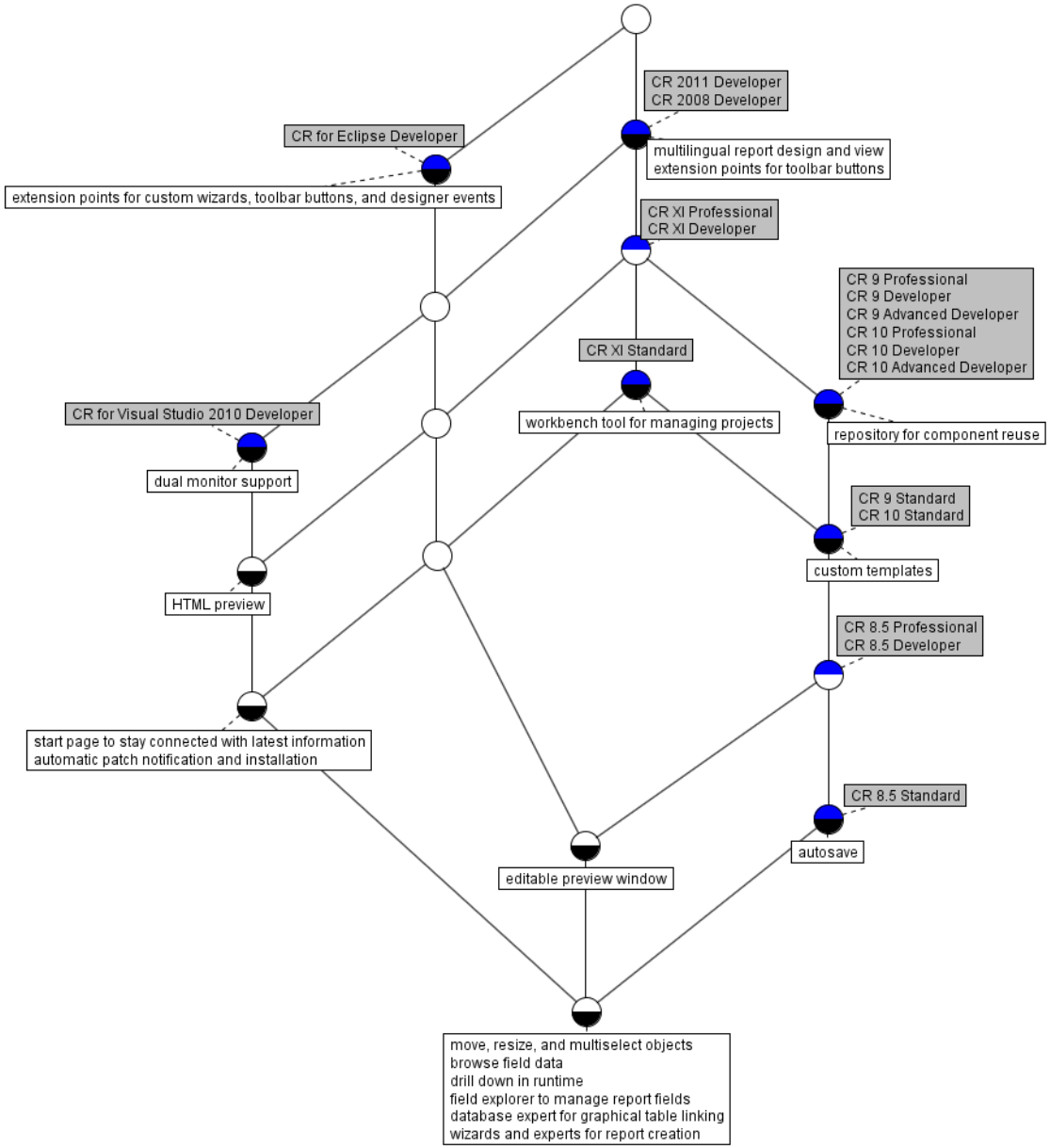
SAP® Crystal Reports® Versions

Key Editions:
Advanced Developer, Developer, Professional, Standard

Version 8.5		D	P	S
Version 9	A	D	P	S
Version 10	A	D	P	S
Version XI		D	P	S
Version 2008 and 2011		D		
Version 2011 only		D 2011		
Version 2008 only		D 2008		

Source: **Comparison of features by version** for SAP Crystal Reports and SAP Crystal Server Software. Pdf-brochure, www.sap.com

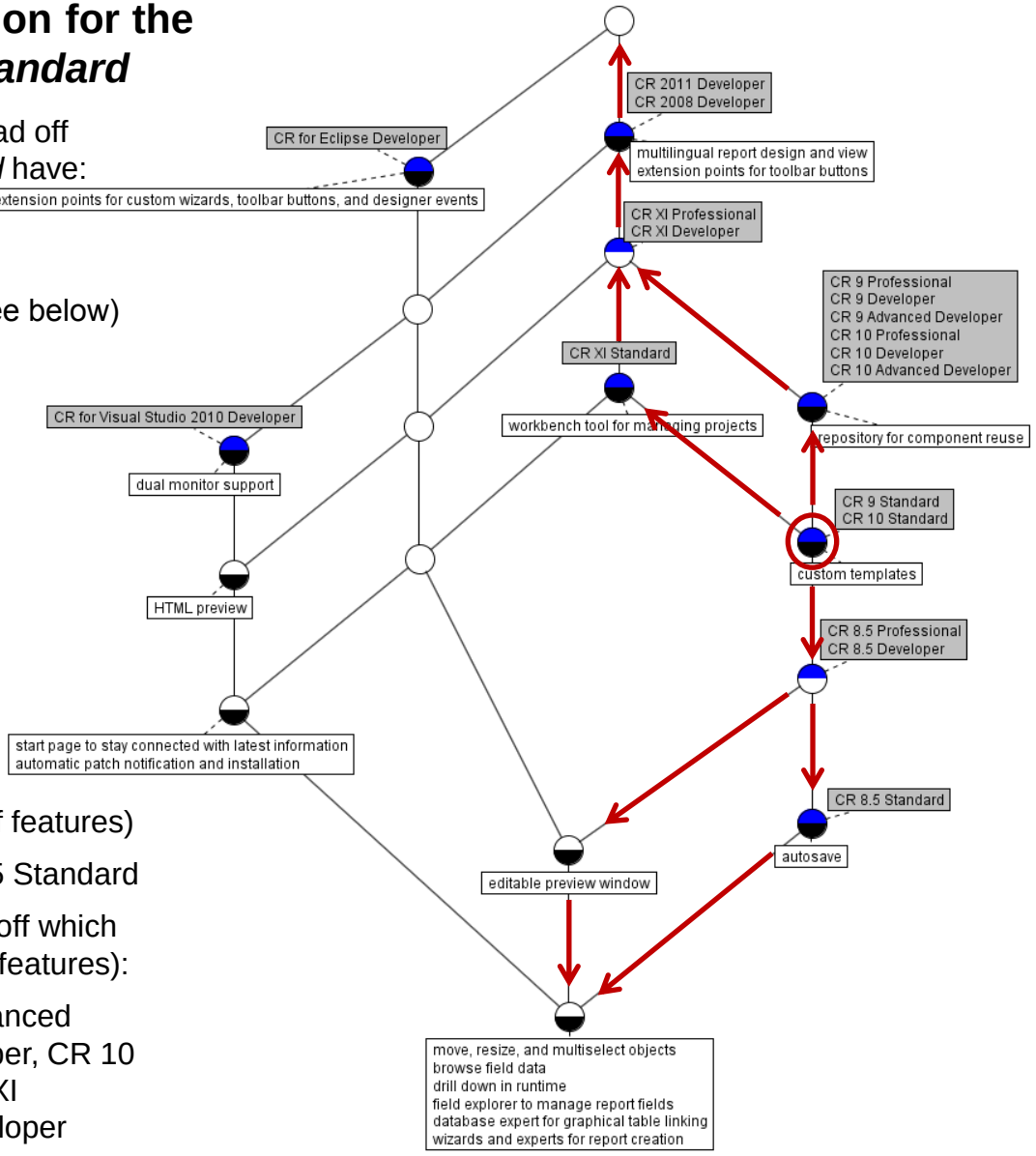
Feature Comparison Matrix: Concept Lattice



Feature Comparison Matrix: Reading the Concept Lattice

Here is how you read off the information for the versions *CR 9 Standard* and *CR 10 Standard*

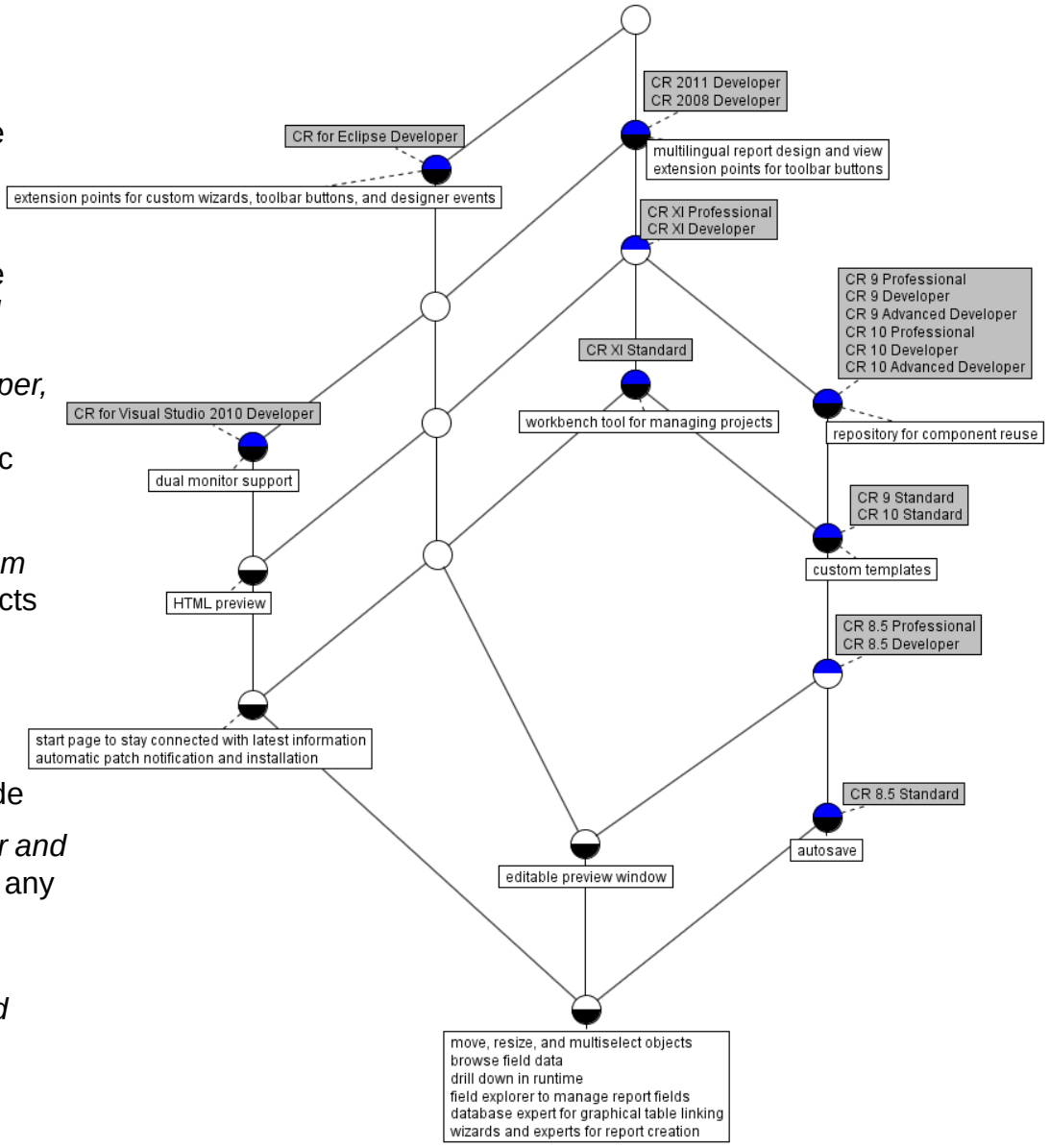
- Following all possible paths downwards, we can read off which features *CR 9 Standard* and *CR 10 Standard* have:
  - custom templates
    - indeed the distinguishing feature of these versions, ompared to “weaker” versions (see below)
  - Editable preview window
  - Autosave
  - Move, resize, and multiselect objects;
  - Browse field data
  - Drill down in runtime
  - Field explorer to manage report fields
  - Database expert for graphical table linking
  - Wizards and experts for report creation
- Following all possible paths downwards, we can read off versions are weaker (i.e., have a subset of features)
  - CR 8.5 Professional, CR 8.5 Developer, CR 8.5 Standard
- Following all possible paths upwards, we can read off which versions are stronger (i.e., they have a superset of features):
  - CR 9 Professional, CR 9 Developer, CR 9 Advanced Developer, CR 10 Professional, CR 10 Developer, CR 10 Advanced Developer, CR XI Professional, CR XI Developer, CR 2008 Developer, CR 2011 Developer



Feature Comparison Matrix: Reading the Concept Lattice

Some more things one can read off

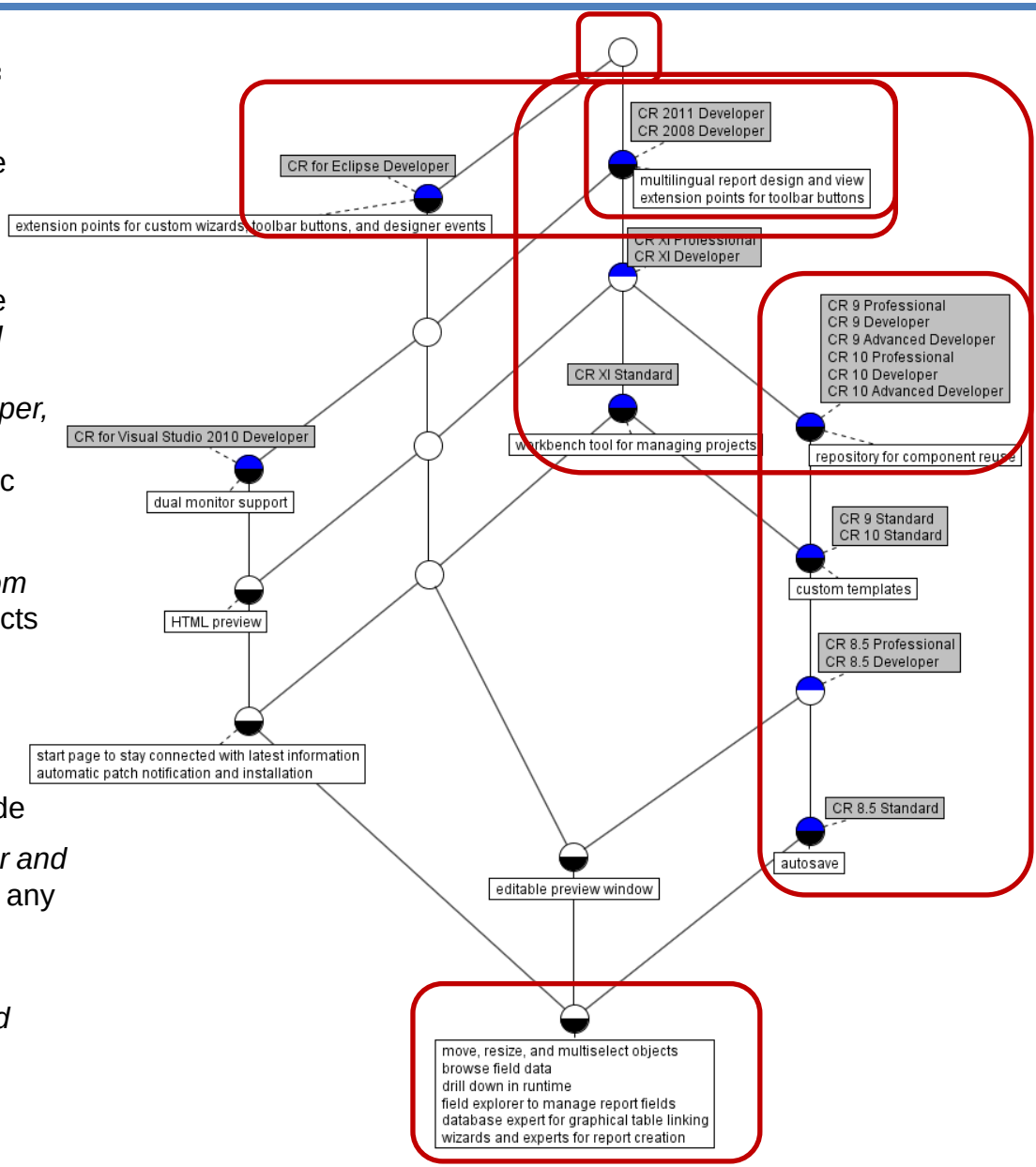
- *CR 2011 Developer* and *CR 2008 Developer* have exactly the same features
  - Because they are on the same node
- *CR 2011 Developer* and *CR 2008 Developer* have more features than *CR XI Professional* and *CR XI Developer*, which in turn have more features than *CR XI Standard, CR 9 Professional, CR 9 Developer, CR 9 Advanced Developer, CR 10 Professional, CR 10 Developer, CR 10 Advanced Developer*, etc
  - Reading the lattice downwardly
- *Autosave* is featured in more products than *Custom templates*, which in turn is featured in more products *than repository for component reuse*, etc
  - Reading the lattice upwardly
- There is no product having all features
  - As there is no product name on the top node
- But *CR for Eclipse Developer, CR 2011 Developer and CR 2008 Developer* are the best products (i.e. for any of those, there is no product with a superset of features)
- *Move, resize, and multiselect objects, browse field data*, etc are featured in all products



Feature Comparison Matrix: Reading the Concept Lattice

Some more things one can read off

- *CR 2011 Developer* and *CR 2008 Developer* have exactly the same features
  - Because they are on the same node
- *CR 2011 Developer* and *CR 2008 Developer* have more features than *CR XI Professional* and *CR XI Developer*, which in turn have more features than *CR XI Standard, CR 9 Professional, CR 9 Developer, CR 9 Advanced Developer, CR 10 Professional, CR 10 Developer, CR 10 Advanced Developer*, etc
  - Reading the lattice downwardly
- *Autosave* is featured in more products than *Custom templates*, which in turn is featured in more products *than repository for component reuse*, etc
  - Reading the lattice upwardly
- There is no product having all features
  - As there is no product name on the top node
- But *CR for Eclipse Developer, CR 2011 Developer and CR 2008 Developer* are the best products (i.e. for any of those, there is no product with a superset of features)
- *Move, resize, and multiselect objects, browse field data*, etc are featured in all products



Conceptual Scaling

From many-valued to single-valued contexts

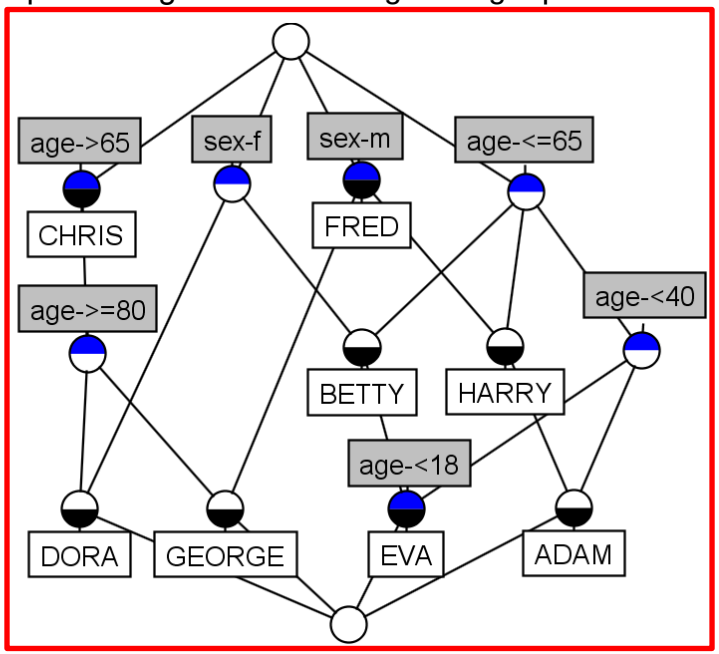
In terms of data, formal contexts are boolean data. While data predominantly exist in non-boolean forms, FCA introduces non-boolean attributes via many-valued contexts, using *conceptual scaling* to transform them into single-valued contexts. Each non-boolean attribute is given a *scale*, each scale being a context in itself.

Below is an example of scaling the sex and age of eight persons (first table). In order to convert this many-valued context into a single-valued formal context, a scale was created for each of the many-valued attributes; one for sex and one for age.

- The first scale contains the two possible values of sex, ‘m’ for male and ‘f’ for female.
- Age, however, has potentially many values as it is a continuous attribute, so rather than using individual values, a scale was created using boundary values: ‘<18’, ‘<40’, ‘≤65’, ‘>65’ and ‘≥80’. ‘ADAM’ is 21 years old, so has the formal attributes ‘<40’ and ‘≤65’. ‘GEORGE’ is 90 years old, so he has the formal attributes ‘>65’ and ‘≥80’.
- The two scales are then merged to form the complete formal context representing ‘the sex and age of eight persons’ (second table).
- The resulting lattice can be seen on the right.

	sex	age
ADAM	m	21
BETTY	f	50
CHRIS	?	66
DORA	f	88
EVA	f	17
FRED	m	?
GEORGE	m	90
HARRY	m	60

	sex		age				
	m	f	*<18*	*<40*	*<=65*	*>65*	*>=80*
ADAM	X			X	X		
BETTY		X			X		
CHRIS						X	
DORA		X				X	X
EVA		X	X	X	X		
FRED	X						
GEORGE	X					X	X
HARRY	X				X		



Conceptual Scaling in CUBIST

In CUBIST, we deal with numerous attribute types including dates, numbers and strings. Each attribute type requires it's own approach to convert it into a formal context and analyze it. The attribute types encountered in CUBIST, along with a brief description of how each type is scaled, are mentioned below.

Attribute Types

- **Categorical**
  - The predominant attribute type is categorical. This is the typical many-valued attribute, e.g. 'Color' can have multiple values such as red, green, blue, black, yellow, etc.
  - Categorical attributes are converted by creating one formal attribute for each of the attribute categories, e.g. Color-red, Color-green, Color-blue, etc.
  - In the conversion process, each value read in from the data is compared with the category values listed for the corresponding attribute. A match is recorded as a true value in the formal context.
- **Boolean**
  - A Boolean attribute can be interpreted as a single formal attribute. Typically, a Boolean attribute in a dataset is one that has two categories that represent *true* and *false*.
  - During the conversion process, the corresponding value read in from the data file is compared with the true value. If they match, this is recorded as a *true* value in the formal context.
  - Treating a Boolean attribute as categorical can also allow for both the *true* and *false* values to appear in the formal context.
- **Continuous**
  - Continuous attributes are dealt with by discretising the data using user-defined ranges (e.g., 0–*10*, *10*–*20*...), or by hierarchical scaling (e.g., > *0*, > *10*, > *20*...).
  - During the conversion process, a continuous value read in from the data file is compared with the appropriate boundary values, assigning a true value to the corresponding formal attribute in the formal context.

Conceptual Scaling in CUBIST

- **Attribute Types (continued)**
  - **Ordinal**
    - Ordinal attributes are categorical attributes where the ordering of the attribute values is important.
    - Taking as an example an attribute Education containing higher education degrees, it is important that a foundation course comes before a Bachelors degree, a Doctorate comes after a Masters degree and so on.
    - Ordinality also allows the creation of sensible ranges to summarise values for non-continuous attributes. Using the same example, creating ranges for the Education attribute can be used to group education levels to undergraduate and postgraduate: Foundation-Bachelors, Masters-Doctorate.
    - Capturing the order of non-numerical data is a feature that allows analyses that are not easy using traditional means of data analysis.
    - When an ordinal attribute has been declared, a numerical value is assigned to each of the attribute values. During the conversion process, an attribute value is read, it's numerical counterpart is found and compared with the appropriate boundary values, assigning a true value to the corresponding formal attribute in the formal context.
  - **Date**
    - The most common date formats are supported. As an example, creating ranges for a dates attribute can be used to group months into seasons: 01/12/2010 to 28/02/2011, 01/03/2011 to 31/05/2011, 01/06/2011 to 31/08/2011 and 01/09/2011 to 30/11/2011.
    - Declaring ranges for date attributes is possible using the same process used for declaring ranges for continuous attributes. Adding hours, minutes and seconds to a date is also possible – 23/09/2011 01:55:40 AM is valid as a date value.
    - During the conversion process, a date value is read and compared with the appropriate boundary values, assigning a True value to the corresponding formal attribute in the formal context.

Concise Comparison of Different Visual Analytics Means

What the next slides are about …

The next slides provide a few thoughts on different kinds of analyzing some data, in order to compare the following **Visual Analytics** means:

1. Traditional BI Visual means (here: a bar chart)
2. A graph-based visualization (here: force-based layout)
3. A visualization based on Formal Concept Analysis (here: concept lattices)

Toy Example Data Set

Skill	Persons with that Skill
IE	Anja, Ben, Ernst, Fred, Ken
ETL	Chris, Fred, Mark
BI	Ben, Chris, Fred, Lemmy, Mark, Naomi
ST	Anja, Diana, Ernst, Fred, Gerald, Harriet, Ken, Owen
FCA	Anja, Diana, Gerald, Harriet, Ian, John, Ken, Owen
VIZ	Anja, Diana, Ian

Possible Information Needs:

1. Show me the count of people for a given skill
2. Show me the skills and how many people share some skills, in order to get an idea on how strongly skills are related
3. Show me the skills and people such that I get an idea of the distribution of skills among people and dependencies between skills

Converting the Data (Analytic Model)

Raw Data

Skill	Persons with that Skill
IE	Anja, Ben, Ernst, Fred, Ken
ETL	Chris, Fred, Mark
BI	Ben, Chris, Fred, Lemmy, Mark, Naomi
ST	Anja, Diana, Ernst, Fred, Gerald, Harriet, Ken, Owen
FCA	Anja, Diana, Gerald, Harriet, Ian, John, Ken, Owen
VIZ	Anja, Diana, Ian

Bar Chart Data

Skill	#People
IE	5
ETL	3
BI	6
ST	8
FCA	8
VIZ	3

Counting the number of people per skill

Graph Data

	IE	ETL	BI	ST	FCA	VIZ
IE						
ETL	1					
BI	1	3				
ST	4	1	1			
FCA	2	0	0	5		
VIZ	1	0	0	2	3	

Counting the number of people who share two skills

FCA Data (Formal Context)

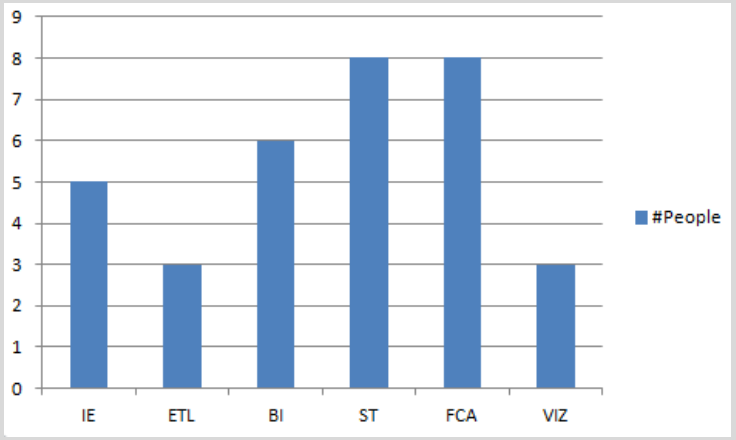
	IE	ETL	BI	ST	FCA	VIZ
Anja	X			X	X	X
Ben	X		X			
Chris		X	X			
Diana				X	X	X
Ernst	X			X		
Fred	X	X	X	X		
Gerald				X	X	
Harriet				X	X	
Ian					X	X
John					X	
Ken	X			X	X	
Lemmy			X			
Mark	X	X	X			
Naomi	X		X			
Owen				X	X	

Visualizing the Data

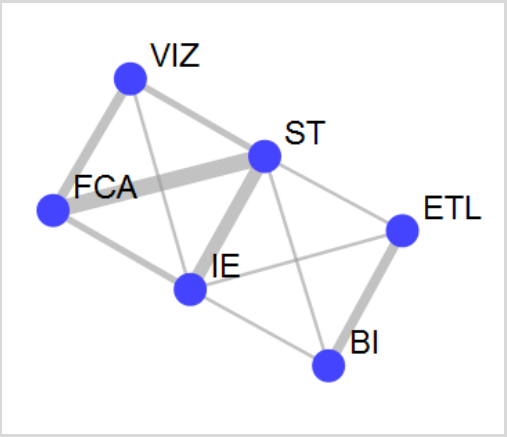
Raw Data

Skill	Persons with that Skill
IE	Anja, Ben, Ernst, Fred, Ken
ETL	Chris, Fred, Mark
BI	Ben, Chris, Fred, Lemmy, Mark, Naomi
ST	Anja, Diana, Ernst, Fred, Gerald, Harriet, Ken, Owen
FCA	Anja, Diana, Gerald, Harriet, Ian, John, Ken, Owen
VIZ	Anja, Diana, Ian

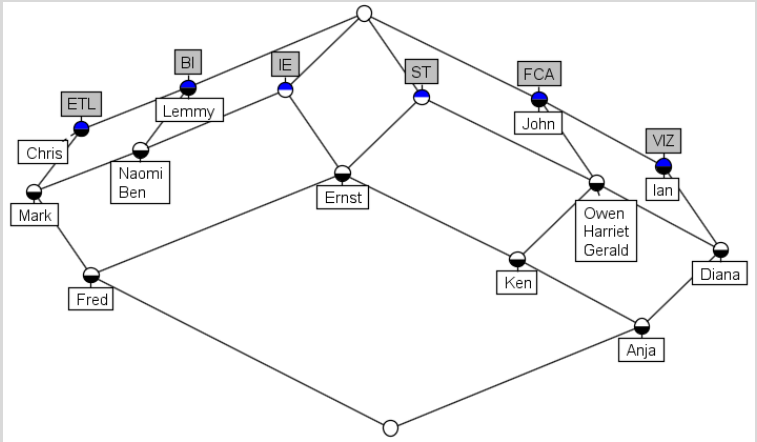
Bar Chart



Graph

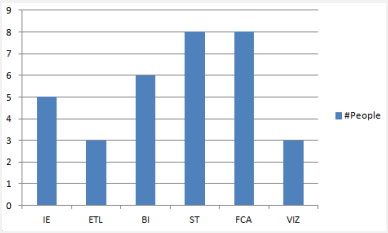


FCA Concept Lattice



Comparison

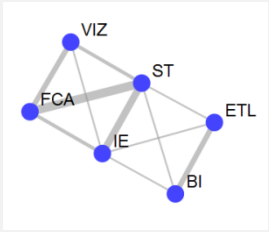
Bar Chart



- Many well-known visualizations
- Good (readable and comprehensible) layouts
- Good for analyzing numbers

- Loss of information (what people)
- Misleading for overlapping attributes (counting people manyfold)
- Not utilizing relationships between entities

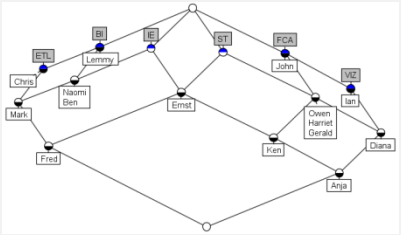
Graph



- Attractive visualizations
- (Relatively) easy to understand
- Utilizing and showing links between entities (skills)

- Loss of information (what people)
- Bad for analyzing numbers

FCA lattice



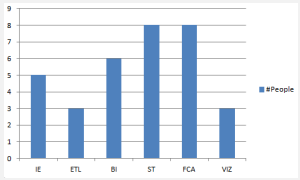
- No loss of information
- Meaningful clusters in one node
- Showing dependencies between entities (both people and skills)

- Number of nodes might explode
- Finding good layout is unsolved (nice layout in example is accidential and has been manually created)
- Unfamiliar means for analytics
- Scalability
- Bad for analyzing numbers

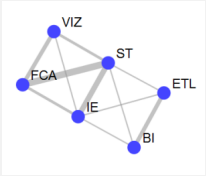
Conclusion

Remember the information needs from the beginning

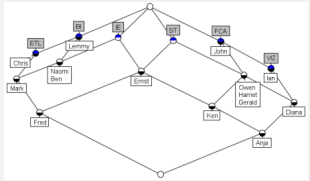
Show me the count of people for a given skill



Show me the skills and how many people share some skills, in order to get an idea on how strongly skills are related



Show me the skills and people such that I get an idea of the distribution of skills among people and dependencies between skills



Conclusion

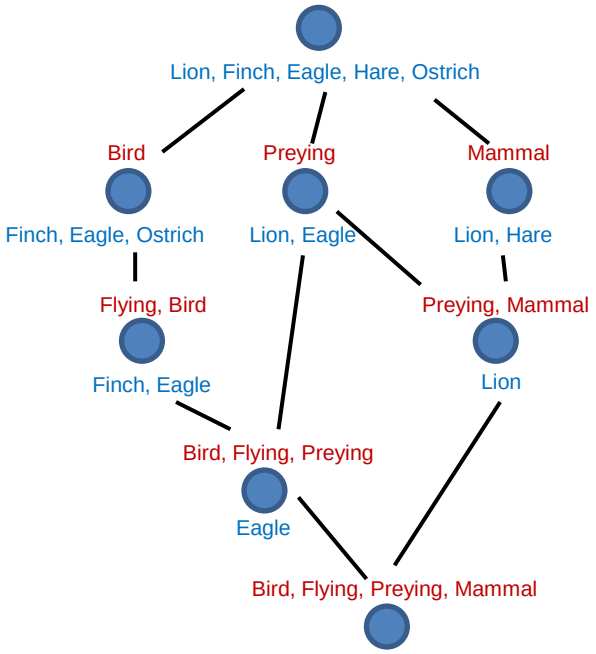
- Each visualization has ist own strengths and weaknesses
- Each type of visualization is suited for a specific type of information needs
- Thus the visualizations are complementing
- Thus future BI tools should provide all types of visualizations
  - For example, side by side with linking-and-brushing

Interactive Visual Analytics for Formal Concept Analysis

Association Rules

Displays patterns of co-occurrence between data under the form:
Premise => Conclusion

Concept lattice



Association rules

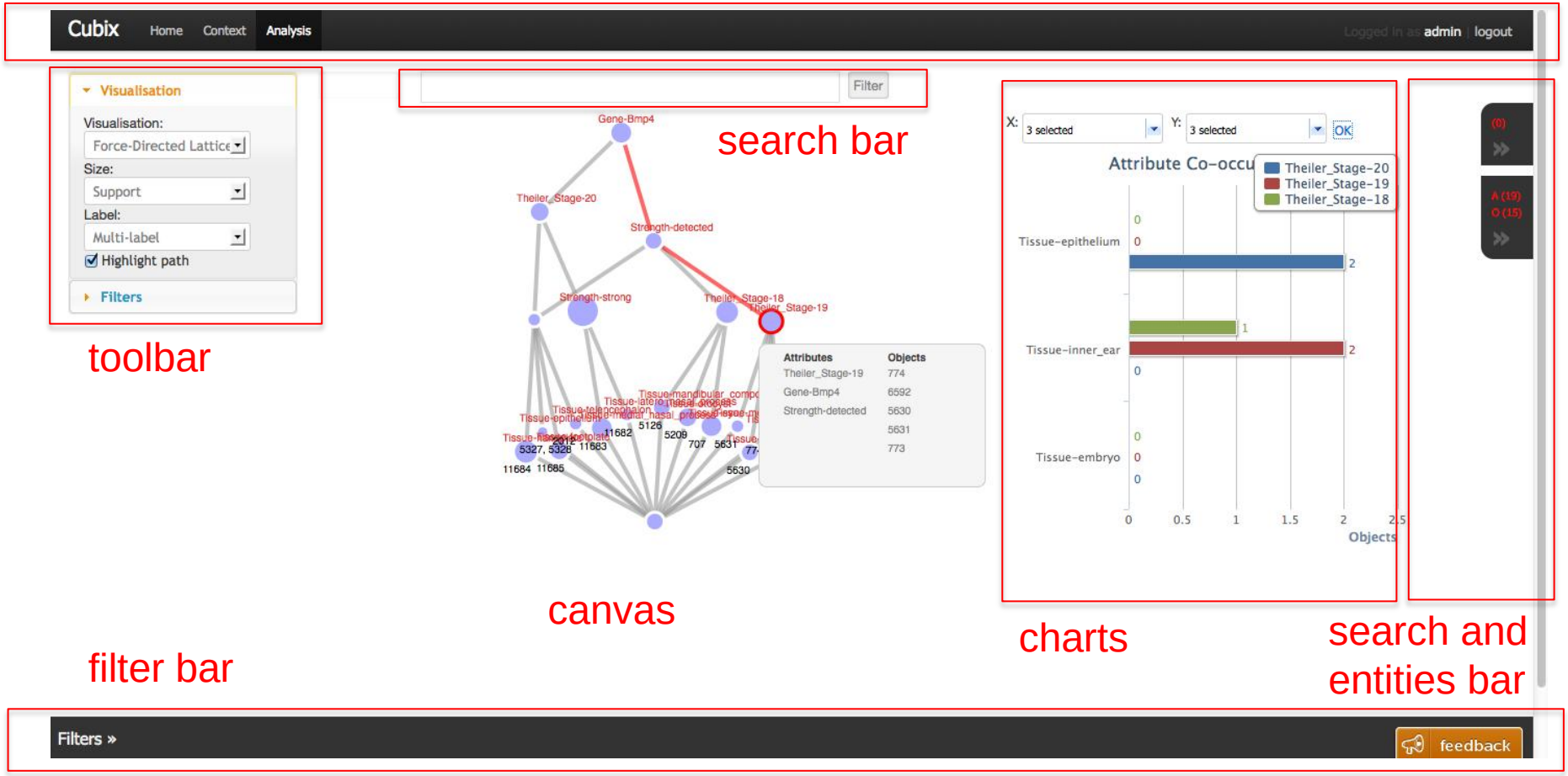
Conf.	#	Attributes		Attributes	#
100%	2	Flying	=>	Bird	3
50%	2	Preying	=>	Flying, Bird	1
50%	2	Preying	=>	Mammal	1

Visual Analytics

- Visual analytics focuses on **massive** and **dynamic** volumes of information
- Supports **human judgment**
- by means of **visual representations and interaction techniques** in the analysis process [Keim *et al. 2001*]

- Visual Analytics for FCA combines:
  - Traditional BI (charts)
  - Graph-based visualization (graphs)
  - Concept visualization (concept lattice)

Cubix User Interface



Filtering / drilling-down

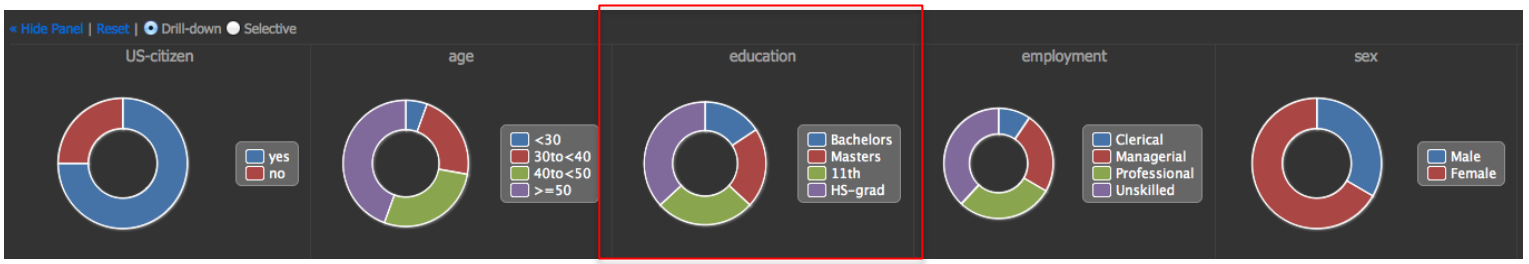
Filtering takes effect when clicking on pie slices

Linking and brushing [Kein 2002]

Selective
Progressive

Display charts according to attribute type

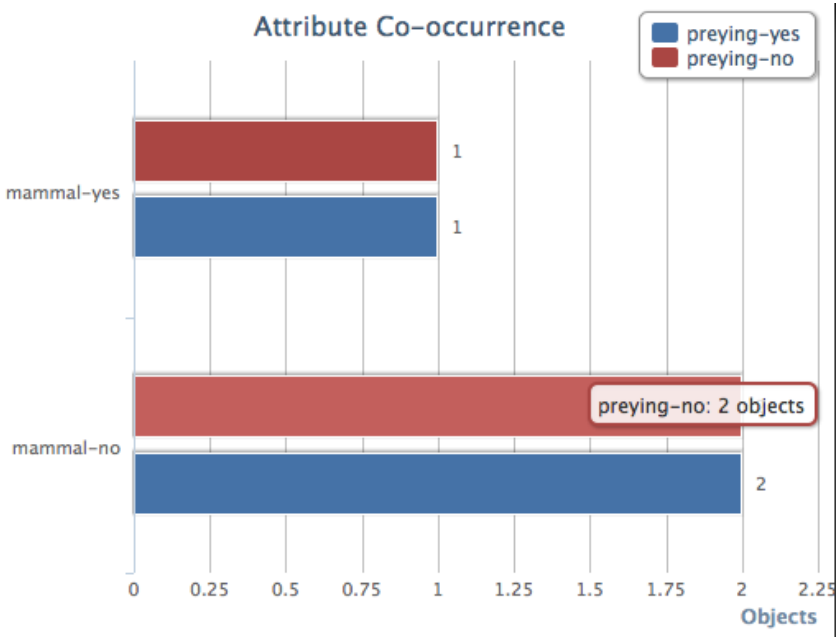
e.g. pie, time slider, map



Dashboard

Graphs to display different perspectives of conceptual data

E.g. co-occurrence of attributes



Customizable dimensions
Attr x objs x concepts

E.g. timeline for a monitored value

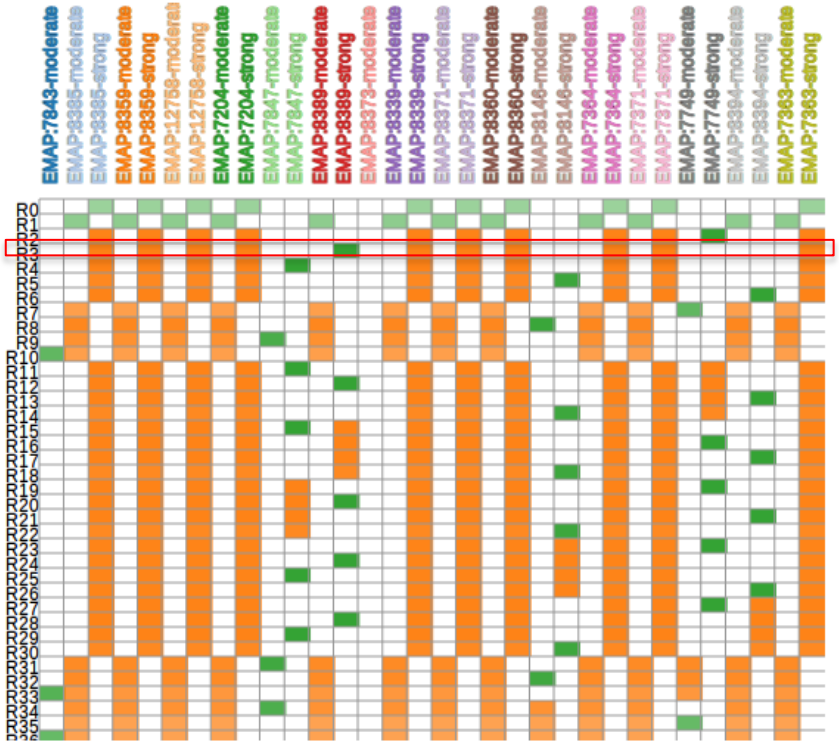
Visualization of Association Rules

New visual metaphors for association rules

```
5 < 1 > age-30to<40 education-HS-grad sex-Male US-citizen =[100%]=> < 1 > employment-Unskilled;
6 < 1 > age-30to<40 employment-Clerical =[100%]=> < 1 > education-Bachelors sex-Male US-citizen;
7 < 1 > age-30to<40 employment-Managerial =[100%]=> < 1 > education-Masters sex-Female;
8 < 1 > age-30to<40 employment-Unskilled sex-Male US-citizen =[100%]=> < 1 > education-HS-grad;
9 < 1 > age-30to<40 sex-Female =[100%]=> < 1 > education-Masters employment-Managerial;
10 < 2 > age-30to<40 US-citizen =[100%]=> < 2 > sex-Male;
11 < 1 > age-40to<50 =[100%]=> < 1 > employment-Clerical sex-Female;
12 < 1 > age->=50 education-Bachelors US-citizen =[100%]=> < 1 > employment-Managerial sex-Female;
13 < 1 > age->=50 education-HS-grad sex-Male US-citizen =[100%]=> < 1 > employment-Managerial;
14 < 1 > age->=50 employment-Managerial sex-Male US-citizen =[100%]=> < 1 > education-HS-g
15 < 1 > age->=50 employment-Unskilled sex-Male US-citizen =[100%]=> < 1 > education-11th;
16 < 3 > age->=50 =[100%]=> < 3 > US-citizen;
17 < 1 > education-Bachelors sex-Male US-citizen =[100%]=> < 1 > age-30to<40 employment-Cle
18 < 3 > education-Bachelors =[100%]=> < 3 > US-citizen;
19 < 1 > education-Masters =[100%]=> < 1 > age-30to<40 employment-Managerial sex-Female;
20 < 1 > education-11th =[100%]=> < 1 > age->=50 employment-Unskilled sex-Male US-citizen;
21 < 1 > education-HS-grad employment-Unskilled sex-Male US-citizen =[100%]=> < 1 > age-30tc
22 < 2 > education-HS-grad =[100%]=> < 2 > sex-Male US-citizen;
```

List of rules - Conexp

Matrix view - Cubix



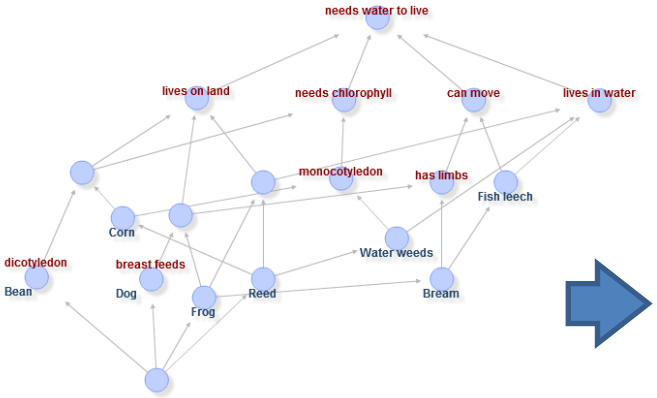
Visual Analytics Tree Extraction from Concept Lattices

Tree Extraction and Lattice Visualisation

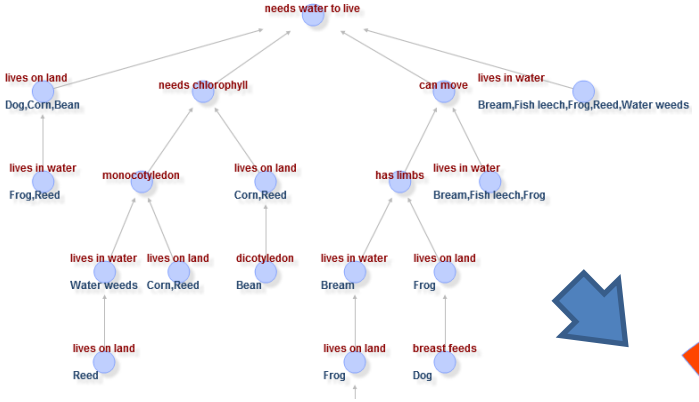
- Goals
  1. Combine tree visualizations for lattice representation
     - Reduces visual clutter (no edge crossing)
     - Simplifies navigation
     - Allows tree visual metaphors (*sunburst, tree maps*, etc)
  2. Enhance lattice readability through visual variables
     - Display conceptual metrics
     - Allows comparison between concepts
- This approach combines
  - Lattices reduction (stability, support, confidence)
  - Representation (colors, size, shape)
  - Layout (sunburst and trees)

Trees as Lattice Alternatives

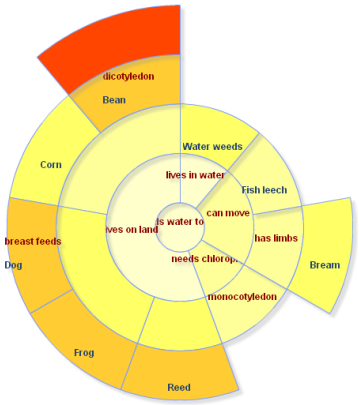
- **Idea**: using known lattices measures to select best parent for each node
  - stability, support, confidence, similarity and topological features



Original lattice



tree from the lattice



Tree visualizations

Tree-extraction process

1. Select an order of criteria
2. A bottom-up approach select only one parent from each node
3. Options for displaying applied metrics are enabled

Parent selection criterion

Consider A for a *set of Attributes* and B for *a set of Objects*

Stability

$$\sigma(A,B) = \frac{Card(\{C \subseteq A \mid C' = B\})}{2^{Card(A)}}$$

Measures how a concept remains the same even when some of their attr/objs are removed

Support

$$\varphi(B) = \frac{Card(B')}{Card(G)}$$

Measures the frequence of concept objects set

Shared attributes

$$\phi(E) = \frac{Card(B \cap D)}{Card(M)}$$

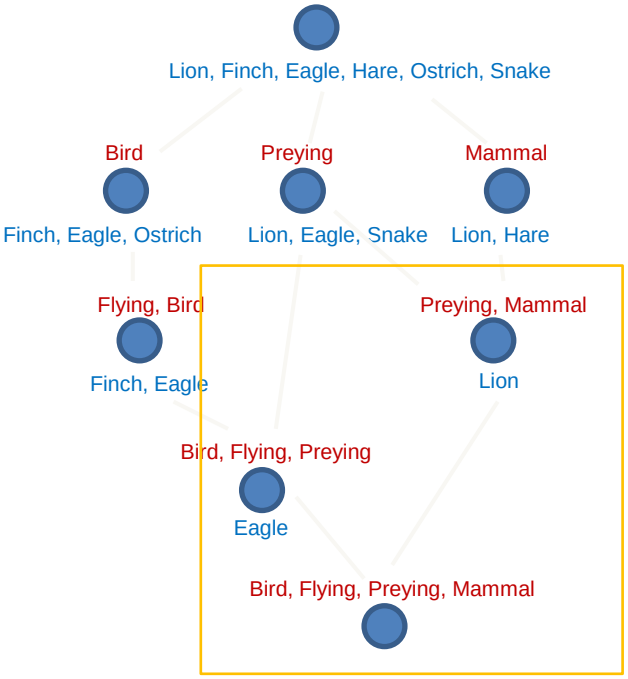
It represents the degree of similarity between parent and child nodes.

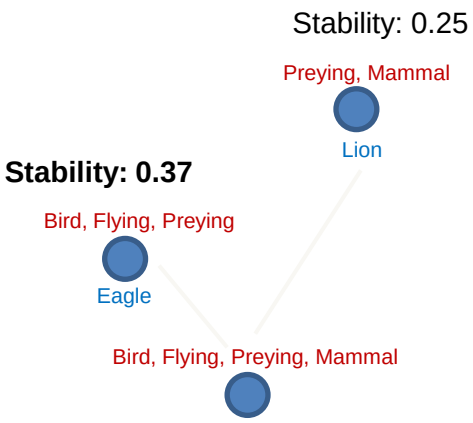
Confidence

$$\delta(E) = \frac{Card(C)}{Card(A)}$$

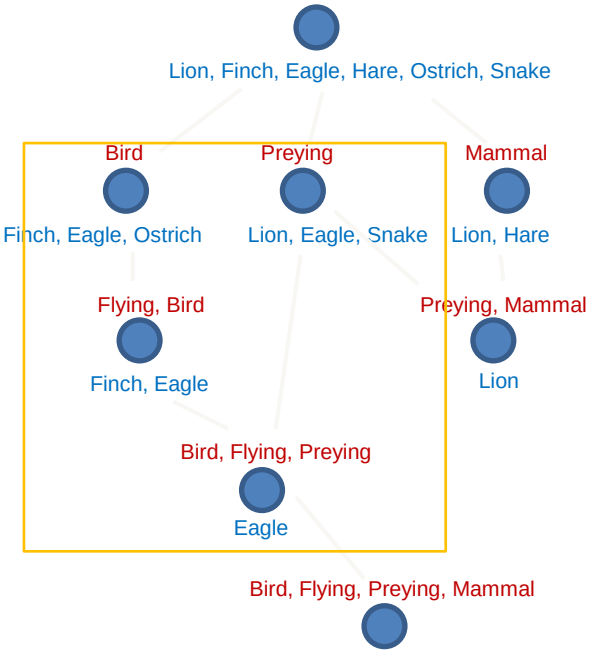
It measures how strong the implication is between a parent concept in a child concept.

Example: Stability

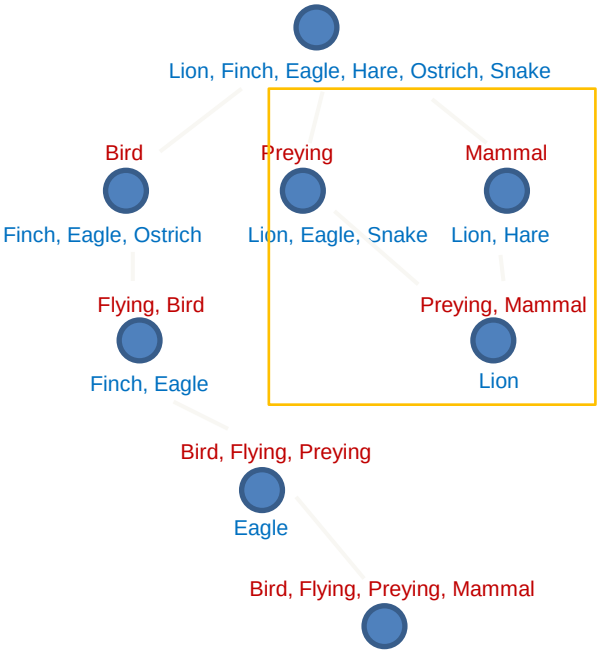




Example: Stability



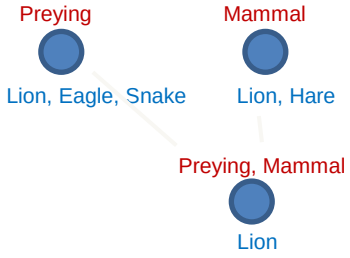
Example: Stability



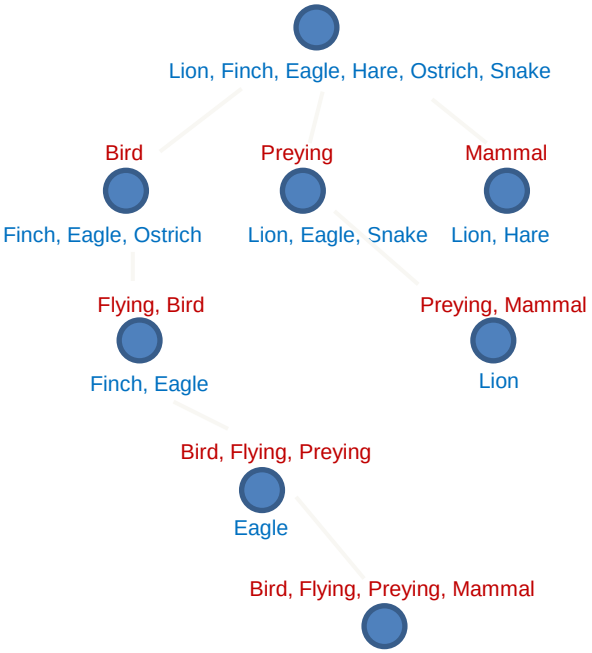
Tie: go to next criterion

Confidence: 0.3
Stability: 0.5

Confidence: 0.5
Stability: 0.5



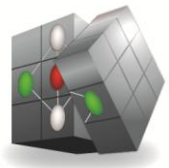
Example: Stability



Final tree

When to apply each measure?

- Observing real-world analogies
  - E.g. Class of customers that represents "*married men who often buy beers on Fridays*"
  - ➢ Stability
- Frequent pattern analysis
  - E.g. *Which genes appear most frequently together?*
  - ➢ Support
- Similarity analysis
  - E.g. *Which classes of animals are similar?*
  - ➢ Shared attributes
- Confidence analysis
  - E.g. *Which kinds of product are often bought together in the supermarket?*
  - ➢ Confidence



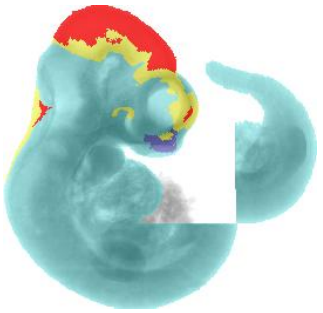
Conclusion

- Use of trees in place of lattices
  - Easier to read, interpret and navigate
  - Have a variety of compact visualizations
  - Loss of information can be compensated by preserving some characteristic
- Use of visual variables to indicate computed metrics used in reduction

Biomedical Atlases
(Heriot-Watt University)

HWU Use Case

- In situ gene expression for the developmental mouse
  - E.g., the gene Oxt2 contributes to the creation of the brain
  - Developmental mouse = conceived to birth
  - http://www.emouseatlas.org
- 2 structured sources; 400, 000 images
  - EMAP = anatomy (ontology)
  - EMAGE = database of developmental mouse in situ gene expression results
  - Images, e.g.,



- Outcome: relationships between genes providing potential info on phenomenon contributing to healthy development of tissues

Semantic Business Intelligence for Space Control Centres

(Space Applications Services)

SAS Use Case - Overview

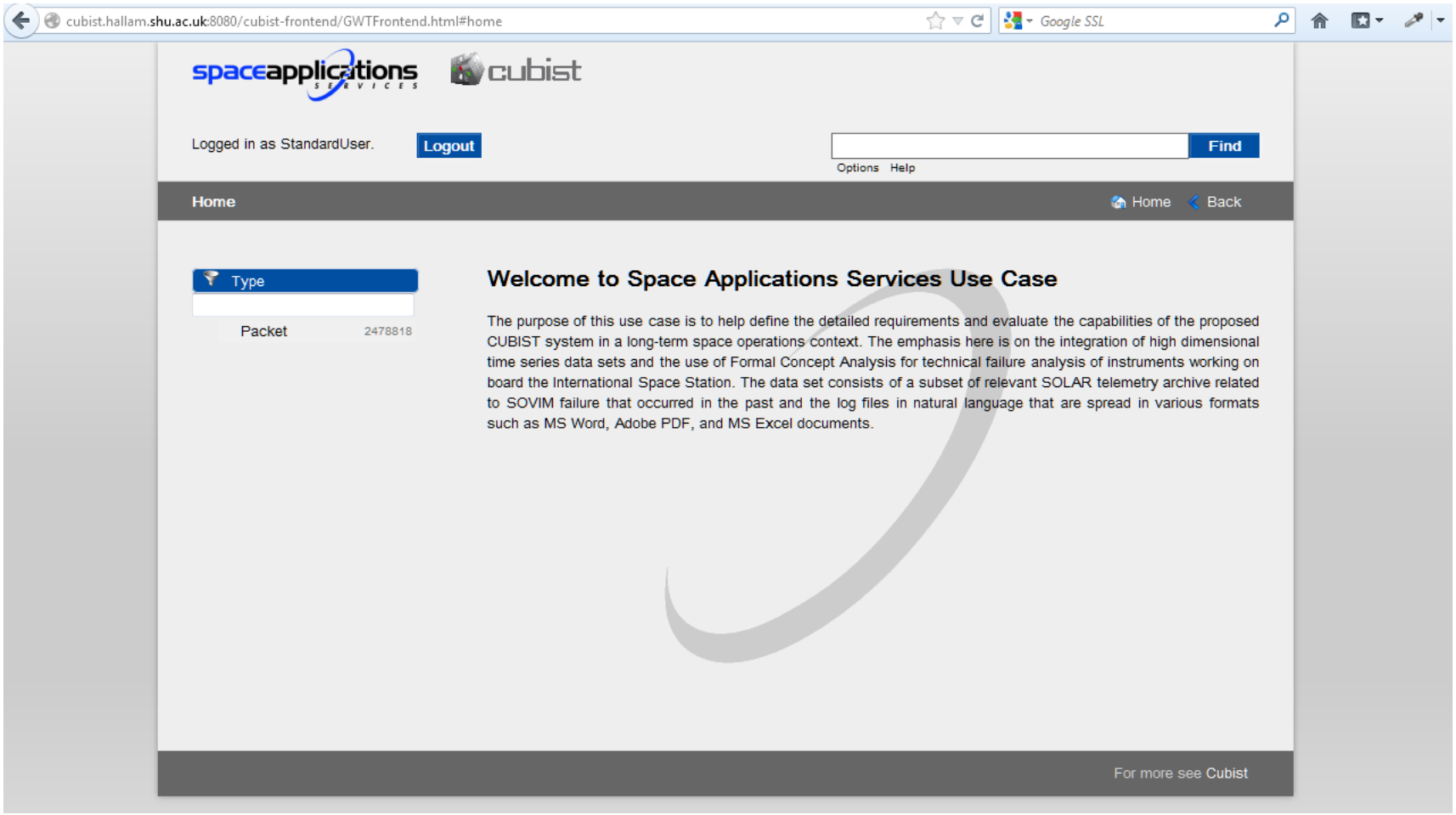
- This Use Case is set within the domain of space industry.
- Today, in space control center operations, much time is spent on transferring, discussing, reviewing, and copying information between the operations partners. The search and replay of operational data for the correct analysis of the on-board situations is also time-consuming. This problem is especially emphasized by the fact that many of the data stores are distributed and have different user interfaces.
- The current operations show an increasing need for a system providing the operator with a unified interface for fast access to information and analytics, especially during anomalous situations where time constraints are stronger. In this Use Case, we envision CUBIST to combine Business Intelligence and Semantic technologies to aggregate various data sources -structured, as well as unstructured- available to the operators, providing support in analysis and decision making on console. The CUBIST concept will be applied on the data resources of the SOLAR payload, a long-term ESA mission on the International Space Station. CUBIST allows to quickly analyze complex data patterns and to extract essential and relevant information, which is crucial in real-time space operations.

- SOLAR is an external payload installed on Columbus in February 2008:
- It is an integrated platform accommodating three instruments: SOVIM, SOLSPEC and SolACES.
- Its scientific objective is the measurement of the solar spectral irradiance throughout a large part of the electromagnetic spectrum
- The Belgian User Support and Operations Center (B.USOC) ensure 24/7 operations support for SOLAR

- **Structured data sources**
- Payload Telemetry:
- House Keeping data
  - Processed Parameters
    - 1 Telemetry packet sent every second
    - 1 Packet: 343 parameters/sec
- **Unstructured data sources**
- On Board Short Term Plan Unstructured data sources:
- Columbus Support Tools
- System Problem Reports
- Payload Operations Data File
- Daily Operations Report
- SOLAR Predictor Tool
- Local Bugs database
- Documentation

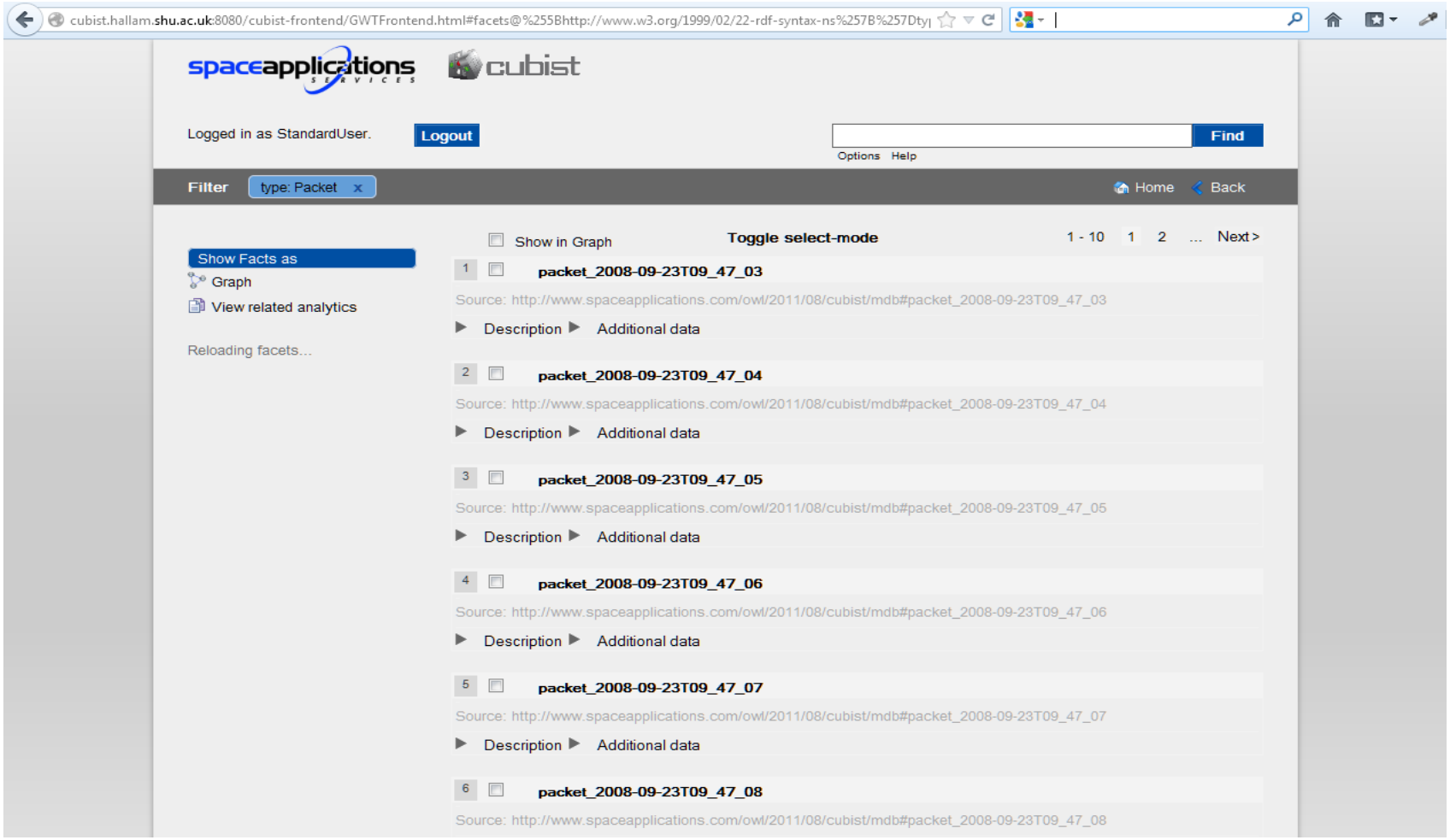
SAS Use Case – Prototype v1

- The first version of the SAS Use Case prototype has been built as customized version of the general integrated prototype.
- The Use Case prototype v1 has been configured precisely with the official SAS colour scheme and design:



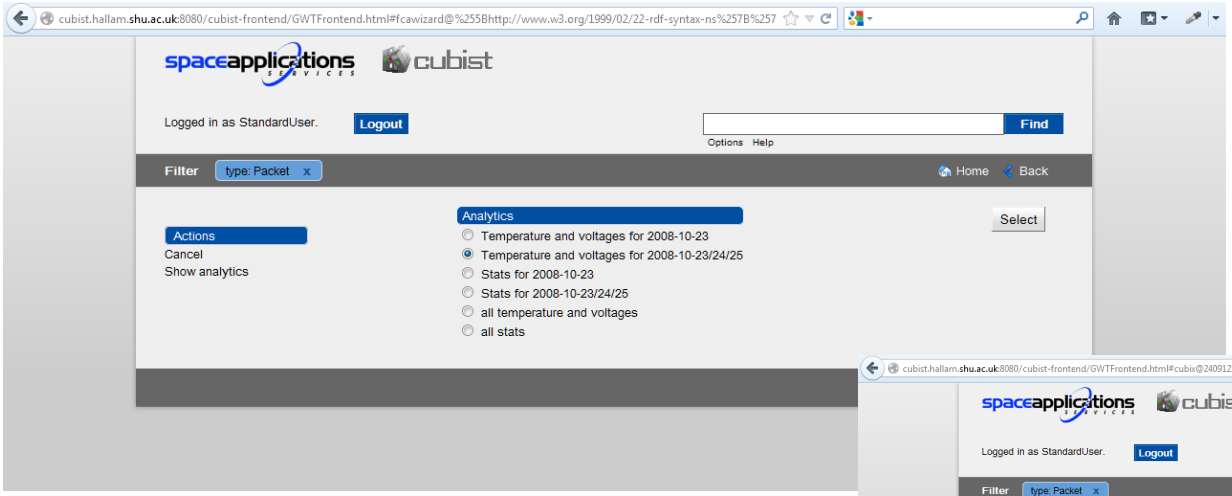
SAS Use Case – Prototype v1 – Triplified Telemetry Data

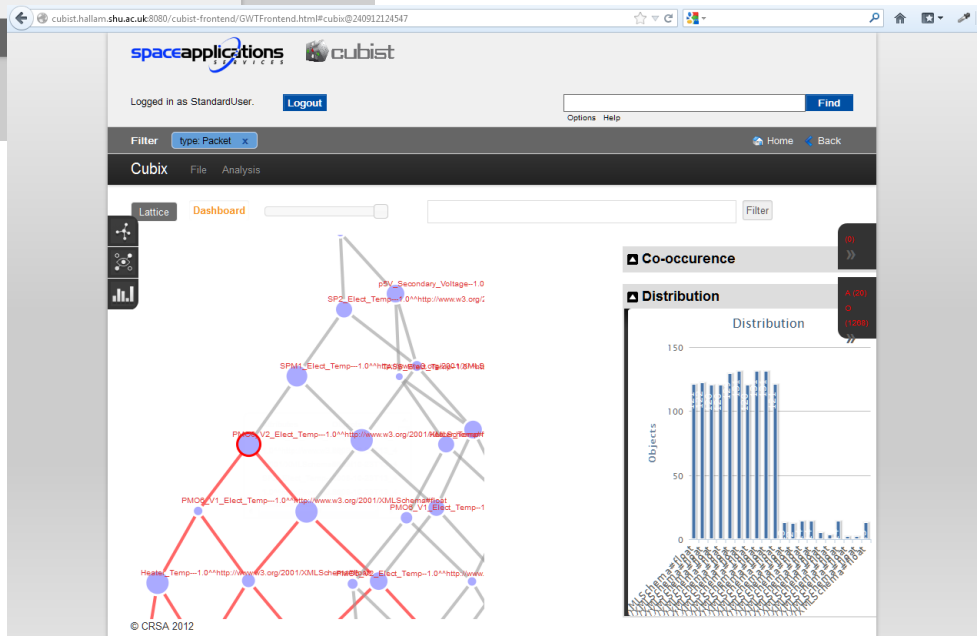
- The data backend of the prototype is connected to the triplified version of the structured telemetry data that resides in the OWLIM semantic triple store.
- The original telemetry data have been triplified using Talend in a batch process and using a predefined ontology. This infrastructure is deemed useful for future data conversion tasks, especially if new periods, that is, subsections, of telemetry data is to be analysed in the future:



SAS Use Case – Prototype v1 – Preliminary Analytics

- Some preliminary analytics have been prepared.
- The analytics are first described by the expert operators working with the SOLAR system at B.USOC.
- They are then manually converted to proper SPARQL-based analytics and shown in the user interface with human-readable labels.





Thank You!

- **Dr. Frithjof Dau**
  Researcher, Business Intelligence Practice I SAP Research Dresden
- **SAP AG** I Chemnitzer Str. 48 I 01187 Dresden I Germany
-
- T +49 351 4811-6152 I F +49 6227 78-51425 I mailto: frithjof.dau@sap.com
- www.sap.com

© 2011 SAP AG. All rights reserved