Combining and Uniting Business Intelligence with Semantic Technologies

Acronym: CUBIST
Project No: 257403

Small or Medium-scale Focused Research Project
FP7-ICT-2009-5
Duration: 2010/10/01-2013/09/30

Requirements analysis (RDF triple stores for BI)

Abstract: This deliverable will provide concrete requirements for an RDF triple store in order to be applicable within a BI context, such as requirements for data volumes and query language support.

Type	Report
Document ID:	CUBIST D3.1.1.
Workpackage:	WP3
Leading partner:	SAP
Author(s):	Frithjof Dau (SAP)
Dissemination level:	PU
Status:	Final
Date:	13 April 2011
Version:	1.0

Versioning and contribution history

Version	1. Description	Contributors
0.1	Draft	Frithjof Dau (SAP)
0.2	Draft, extended chapter 1,4, added chapter 5 (to be removed later) to discuss queries from use case partners	Frithjof Dau (SAP)
0.4	Extended sections concerning datasets and queries from Innovantage	Hazzaz Imtiaz (Inno)
0.5	Reaction and comments to usecase requirements	Ontotext
1.0	Addressed comments of reviews	Frithjof Dau (SAP)

Reviewers

Name	Affiliation
Constantinos Orphanides	SHU
Cassio Melo	CRSA

Table of contents

1 Introduction

Instead of using a "classical" database, a core idea of CUBIST is to use an RDF triple store as persistency layer in the context of Business Intelligence. The objectives of WP3 is thus to develop a methodology, as well as a supporting implementation that renders an RDF triple store (a "Semantic Information Warehouse", so-to-speak) suitable for BI applications.

However, compared to data warehouses which are tailored to suit BI needs, triple stores lack dedicated BI functionalities:

- SPARQL 1.0, being the query language for triple stores which is the official W3C-recommendation, currently provides little or no means for OLAP cube-type queries used in a typical BI context. Examples of BI-related functionality missing from SPARQL 1.0 include: extended aggregate functions, sub-queries, projected expressions and common statistical functions, standardised full text search and complex graph path expressions.
- Moreover, common indexing and materialisation strategies for triple stores are not tailored for the specific needs of BI, particularly for querying the data in the BI context.

A major goal of CUBIST's WP3 is to cope with these issues.

In task T3.1, OLAP extensions to SPARQL are first investigated and then implemented. It has to be noted that the SPARQL Working Group at W3C has already planned to include some of this functionality in SPARQL 1.1. For the point in time when CUBIST was submitted to the EC, the working group was hardly active, but meanwhile, a working draft for the next version of SPARQL, namely SPARQL 1.1, has been published in October 2010 (see http://www.w3.org/TR/sparql11-query/).

New planned features of SPARQL include:

- Aggregates
- Subqueries
- Negation
- Complex expressions in the SELECT clause

For the time being, it is not clear to what extent these extensions will satisfy the needs of the CUBIST use cases one the one hand, and whether they will really make it into the next SPARQL recommendation. Within CUBIST, the requirements of the use cases with respect to the abilities of SPARQL are the main driving force for extending SPARQL. For these reasons, task T3.1 task will include four main activities:

1. gathering requirements from the use cases for OLAP related query functionalities missing from SPARQL 1.0,
2. evaluating the planned extensions within SPARQL 1.1,
3. providing specific proposals for additional extensions and recommendations for a scalable implementation supporting such extensions, and
4. gathering requirements from the use cases regarding the data volume (billions of triples) and query performance (thousands of queries per hour)

Task 3.2 in turn focuses on advanced indexing and materialisation strategies that will support the upcoming SPARQL extensions as well as provide improved scalability and performance of the triple store with respect to data storage and querying in a typical business analytics scenario. As this deliverable belongs to task 3.1, we will not further detail indexing and materialisation strategies.

This deliverable will provide concrete requirements for an RDF triple store in order to be applicable within a BI context, such as requirements for data volumes, query language support and query performance. Its outcome will be the basis for deliverable D3.1.2 “OLAP extensions to SPARQL”, in which concrete recommendations for OLAP extensions to the SPARQL language, together with recommendations for efficient implementation support of these extensions, will be provided.The most important basis for this requirements analysis (i.e., w.r.t. extending SPARQL with BI functionalities) are non-technical queries CUBIST is expected to be able to process. These queries have been gathered from the use case partners as initial input for this deliverable and are provided in chapter 4.

Naturally, the queries act on the information available in the CUBIST information warehouse, i.e. on the information harvested from the data sources provided by the use case partners. Thus it is essential to have an understanding of the data sources. For this reason, a list of all datasources is provided in chapter 3. The type of the datasource (e.g., whether it is unstructured or structured) affects to some extend the kind of queries which can be run on them. In chapter 2, a first attempt to classify the datasources is provided, including some initial discussion how information from these sources is (technically) harvested and persisted in the information warehouse. At this point it seems important to mention that a detailed classification of the datasources and the technical means to harvest them is not subject of work package 3, thus not subject of this deliverable: These investigation will be thoroughly conducted in work package 2 “Semantic ETL and Data Integration” and its result be documented in the deliverables D2.1.x “Semantic ETL from unstructured data sources” and D2.2.x “Semantic ETL from structured data sources” (with x=1,2: each of the deliverables comes in two versions v1 and v2 which are due in M12 and M24, respectively).

Based on the initial queries from the use case partners and the federated datasources, this deliverable investigates in what respect SPARQL 1.0 has to be extended, and provides concrete requirements for query language support. In order to do so, the non-technical queries in chapter 4 are enumerated and query by query investigated w.r.t. the needed SPARQL query support. This is done in chapter 5.

2 Types of Data Sources and Respective Semantic ETL Technologies

This chapter provides an attempt to classify the different kinds of data sources which are to be digested by the CUBIST system. As already said in the introduction, a detailed classification of the datasources and the technical means to harvest them is not subject of this deliverable, but will be targeted in deliverables of WP2. Instead, we focus on those aspects of the datasources which are important for the querying abilities of the CUBIST system.

To this end, a first important distinction is to differentiate between metadata and annotations on the one hand and facts on the other hand:

1) When structured sources like databases are digested, the content of these sources can be persisted as facts (in form of RDF triples) in the information warehouse. These facts are *normally* taken to be granted to be 100% accurate. For example, in the Innovantage use case, a database with information about companies will be federated, and the information in this database is considered to be true. Anyhow, even information in databases might be imprecise or even faulty. For example, in the Hariot-Watts use case, the databases contain results of experiments, and a given result might be these results might be inaccurate, or different results might be conflicting (indeed, finding such conflicts is a goal of the use case).

2) When unstructured sources like text documents are digested, we will mainly digest not facts *inside* the document, but *about* the document, i.e. metadata. Some of the metadata assigned to a document will be accurate (a simple example is the crawling date), whereas other metadata will be more or less imprecise (e.g., a text analysis might yield that a document is about some concept, and the text analysis engine assigns some precision, e.g. 87%, to this assignment). The main instrument for assigning metadata to the documents are technologies from the field of information extraction, like using dictionaries and name catalogs, rules, etc.

   Depending on some predefined inherent form/structure of the text documents, it might anyhow be possible to derive some facts out of them. For example, in the Innovantage use case, job ads on the internet are crawled, and if all job ads from a given website have a specific from (e.g. offering date, offering company, job title, job description), then it is possible to digest some facts out of them (e.g. "company *X* offers job *Y*" or "job Y hasSalary salary *Z*").

So far we have argued that the kind of datasources (most importantly, the distinction between unstructured and structured datasources) affects the kind of information which can be federated into the semantic information warehouse and thus affects the kind of queries which can be asked to the information warehouse.[1] Moreover, the quality of the process which extracts the information of the datasources and stores it into the information warehouse, i.e. the quality of the semantic ETL, is important as well.

[1] Though not mentioned yet, it is plain that the *quality* of the datasources is important as well.

2.1 Sources from WWW

Sources from the internet cover web pages, social streams (e.g. from twitter or LinkedIn), blogs and forums, etc. There are essentially two ways to federate data from these sources into the CUBIST information warehouse:

1) Some websites (e.g. Twitter) provides APIs in order to access the data
2) Other sources need to be digested with crawlers

WWW sources are unstructured sources, so essentially the considerations for unstructured sources from the beginning of this chapter apply. Anyhow, the backends of those WWW sources which are relevant for CUBIST (e.g. websites with job advertisements) are in most cases databases. If the respective content provider offers an API where facts from the backend databases can be retrieved, these sources can be somehow considered structured sources.

2.2 Unstructured textdocuments (word,txt,pdf)

In most cases, unstructured text documents (e.g. manuals or reports written by humans, like the console logs in the Space Application Services use case) are files on a fileshare. In this case, the fileshare has to be crawled and the crawled documents have to be processed with IE technologies. The considerations for unstructured sources from the beginning of this chapter apply.

2.3 Structured textdocuments (word, txt, pdf, CSV)

Text documents are not necessarily unstructured documents: Some text documents have a strict format and internal structure which allows for considering them being structured sources and retrieving facts out of them. An example of documents of this type are telemetry documents in the Space Application Services use case: Here telemetry data is stored in a structured textual format. A dedicated script running on the crawled texts thus can transform the data into facts stored as triples in the information warehouse. Such a script needs to include a mapping from the data in the documents to appropriate concepts or predicates in the ontology.

2.4 Excel-Files

Spreadsheets vary to a large extent in terms of complexity, and the higher the complexity of a spreadsheet is, the more complicated it is to federate the data of the spreadsheet into a triple store. For less complex spreadsheets which are essentially tables, a standard approach to import the contained data is to model the spreadsheet rows as instances and the columns as attributes of the instances. Technically, this can be done by exporting the spreadsheet to a csv-file, which is then imported similarly to structured textdocuments, or by excel-macros which directly import the data in the spreadsheet into the triple store.

2.5 XML

XML is a textual, yet semi-structured data format which serves the purpose to encode documents in machine-readable form. Due to the structured form of XML, it is essentially possible to transform all information in an XML document into RDF triples and thus making it searchable. For XML-documents, there exists the well-known distinction between being well-formed and being valid, the latter w.r.t. to a given schema document. Being well-formed means that the XML-document is syntactically correct, whereas as valid document moreover conforms to a given schema document (a DTD or XSD file), thus so-to-speak being semantically well-formed (is the schema is understood to

provide some semantic information of the data in the XML-file). Obviously, if a schema definition for an XML-document exists, it is easier to import the data in the xml-file into the triple store in a correct manner.

2.6 Databases

Importing data from relational databases into triple stores is a well investigated field, and there exists already a variety of tools for supporting this task, e.g. *Triplify*[2] or *D2R Server*[3]. Databases contain both schema information (structures of tables) as well as instance data (the entries in the tables). Schema information of the database can be mapped to schema information in the ontology (e.g., tables can be mapped to classes or properties, table columns can be mapped to attributes or properties), and entities can then be imported and accordingly assigned to classes or properties. The essence of importing a database is thus the mapping from the schema of the database to the ontological schema. Depending on the quality of this mapping, the entities in the database can be queried from the triple store. Moreover, some of the tools allow on-the-fly data translation of the data, i.e. there is no need to replicate data in triple stores, instead queries to the triple store are real-time translated to SQL-queries to the relational database.

2.7 Existing Ontologies

Reusing ontologies is an important aspect of ST. For a triple store, existing ontologies can either be loaded directly to the store, or - assumed that the triple store supports it- imported by using the `owl:import` command.

[2] http://triplify.org

[3] http://www4.wiwiss.fu-berlin.de/bizer/d2rq/index.htm

3 Data Sets from Use Case Partners

3.1 HWU

This use case does not contain any unstructured data. All data to be used in this use case will come from a structured data repository. The only unstructured data items that may prove pertinent to this use case, are the journal publications that often precede an experiment being published in EMAGE. Such publications regularly contain more information than the resource. Yet, this information is hidden inside electronic articles, only some of which are public domain. As this data is not central to the use case, it shall not be discussed further.

The biological use case initially centres on two databases provided by the EMAP[4] project, namely EMAP and EMAGE.

The data in EMAGE uses the two methods of describing the mouse anatomy found in EMAP - for more information see Deliverable D7.1.1, *Requirements Analysis*. Due to the nature of biology, EMAGE is both inconsistent and incomplete.

Name:	**EMAP**
Type:	Database
Ownership:	Public access after registration
Technical access:	SQL endpoint
Content:	**EMAP** contains a series of anatomy ontologies that describe the development of the mouse from conception until just before birth, and a series of 3D spatial models that perform the same task;

Name:	**EMAGE**
Type:	Database
Ownership:	Public access after registration
Technical access:	SQL endpoint
Content:	**EMAGE** includes results from, and manual annotations of, in situ gene expression experiments undertaken on the developmental mouse.

[4] www.emouseatlas.org

3.2 SAS

This section lists the datasources from Space Application Services, divided into unstructured and structured sources.

As already described in the CUBIST DoW, the data of space centres has varying availability restrictions.

- R3: Requires USOC's permission only. Support of B.USOC has been obtained for granting access to this category of data: see letter of support of B.USOC in the appendix of the DoW. This data source is sufficient for reaching the very large data volumes required for CUBIST (telemetry datasets).
- R2: Property of scientists or ESA. Extra permission may be granted by data owners. The letter of support of B.USOC in the DoW appendix ensures B.USOC's support and mediation for accessing such additional data sources.
- R1: Restricted to ISS (International Space Station) operations environment Extra permission has to be asked from ESA, via the B.USOC. The letter of support of B.USOC in the DoW appendix ensures B.USOC's support and mediation for accessing such additional data sources.
- Public: Some data used in B.USOC comes from public sources, e.g. ISS TLE and attitude data that is freely published by NASA.

3.2.1 Structured Sources

Name:	**OSTP (Onboard Short Term Planner)**
Type:	Structured text
Ownership:	R1: Restricted to ISS (International Space Station) operations environment
Content:	OSTP (Onboard Short Term Planner) is a web application that displays timeline information about ground and onboard procedures, schedules, and activities. Data can be received from a remote site in a textual, structured format. However, the primary interface that OSTP presents to the user is horizontally scrollable web page displaying with a timeline over several days and various events scattered on it. B.USOC operators view OSTP in read-only mode.

Name:	**IPV (International Procedure Viewer)**
Type:	XML
Ownership:	R1: Restricted to ISS (International Space Station) operations environment
Content:	IPV is a web application for managing and displaying the database of onboard and ground procedures. The procedures are stored in a well-structured XML format that describes each procedure step by step, sets preconditions and control checks.

Name:	Telemetry
Type:	Structured text
Ownership:	R3: Requires USOC's permission only
Content:	Telemetry is data organized into packets that is sent by the payload to the control centre. A representative telemetry contains temperature measurements, voltage and current readings, various operational states and reports, such as rotation axes of moving parts.

Name:	Science data
Type:	Structured text
Ownership:	R2: Property of scientists or ESA
Content:	Although technically part of telemetry, science data belongs to the scientific community represented by the PI (Primary Investigator) and its use more restrictive than that of the telemetry.

Name:	MDB
Type:	Relational database
Ownership:	R2: Property of scientists or ESA
Content:	The mission database contains a machine-readable description of the telemetry, including the size of various parameters sent in telemetry packets and their interpretation from binary to engineering values.

Name:	Telecommands
Type:	Structured text
Ownership:	R2: Property of scientists or ESA
Content:	Telecommands are structured data sent to payload during the operations. They may contain control structures for shutting up or starting various modules, as well as uploads of data and scripts. A complete history of telecommands over the operating live of the payload is saved and is made available to the operational environment and to the scientific partners.

Name:	Auxiliary data
Type:	Structured text
Ownership:	Public
Content:	Most of auxiliary data comes from public sources. For instance, current B.USOC operations related to the SOLAR payload heavily depend on TLE (two-line elements) to predict the position of ISS and on the ISS attitude timeline (ATL) to predict the orientation of ISS towards the Sun. These two external data sources are combined in order to create e.g. an optimal shiftplan for operator's presence.

Name:	System and software logfiles
Type:	Structured text
Ownership:	R3: Requires USOC's permission only
Content:	Some of the software products used throughout the operations produce logfiles that record operator actions and other supplementary information in computer-processable form. Typical examples of such log files are apache logs and windows event logs.

3.2.2 Unstructured data sources

Name:	CEFN/ICN (Columbus Flight Notes/Intra-Console Notes)
Type:	Unstructured text
Ownership:	R1: Restricted to ISS (International Space Station) operations environment
Content:	Columbus Flight Notes is an issue tracking web application for the Columbus environment, using by USOC operators and the personnel of the Col-CC Columbus control centre. It is coded in PHP and does not seem to provide any functionality to export the contents of the notes in machine-readable format. ICNs or Intra-Console Notes are a special datatype within the same application that is intended to be used within a single location, e.g. B.USOC and thus — invisible to the rest of the Columbus environment.

Name:	Console Logs
Type:	Unstructured text
Ownership:	
Content:	Console logs are short timestamped messages that the operators update every time them make an operation. Examples of such messages are: "File renaming complete.", "OC requests to provide an anomaly report for the missing data", "Handing over to John", etc. Console logs are primarily used for passing over information inbetween shifts and can also be used for forensics analysis.

Name:	DOR (Daily Operations)
Type:	Excel
Ownership:	R3: Requires USOC's permission only
Content:	Daily operations reports are Excel documents that are filled semi-automatically from a large and complex Excel spreadsheet every 24 hours. They include the summary of the operations by ISS path, as well as reports on discrepancies and anomalies.

Name:	SOLAR Mission Tool
Type:	Excel
Ownership:	??????
Content:	This is a large Excel file that serves to generate DORs, implement configuration control, and provides a multitude of functionality in order to simplify the manual work of B.USOC operators. The file is constantly updated with payload-related information by the operators during their shifts.

Name:	PODF (Payload Operations Data File)
Type:	XML
Ownership:	R1: Restricted to ISS (International Space Station) operations environment
Content:	PODF or Payload Operations Data File is a set of documents governing the operations of the payload in the pre- defined situations, such as power loss and particular space weather conditions. These documents are available in IPV and can be retrieved as machine- and human-readable XML files.

Name:	JOIP (Joint Operations Interface Procedures), FR (Flight Rules), PL Reg (Payload Regulations)
Type:	Unstructured text (pdf-files created in Word)
Ownership:	R1: Restricted to ISS (International Space Station) operations environment
Content:	Just like PODF, JOIP governs the operations of the payload, JOIP governs the communications between various parties involved in the Columbus environment: NASA, ESA, Roskosmos, flight directors, columbus and and payload operators. FR (Flight Rules) and PL Reg (Payload Reguations) are similar to JOIP and cover standard situations and operations tasks for Columbus. Most of these documents are created in Microsoft Word and distributed as PDF files.

Name:	SPRdb (System Problem Report database), AR (Anomay Database), IOT-TS (Industrial Operator Testing Ticketing System)
Type:	XML
Ownership:	R1: Restricted to ISS (International Space Station) operations environment
Content:	SPRdb (System Problem Report database) is an online issue tracking system for ground and onboard issues about Columbus, its payloads, and control centres. It is used mainly to report problems to ground segment and payload developers while CEFN is used for operations communication. It has several sections that list SPRs, ARs (Anomaly Reports) and IOT-TS (Industrial Operator Testing Ticketing System) tickets. The contents of the SPRdb database can be extracted into XML files.

Name:	eRoom
Type:	Unstructured text (pdf-files or Word-files)
Ownership:	R1: Restricted to ISS (International Space Station) operations environment
Content:	eRoom is a Documentum™ installation that serves as a repository of interface control documents, specifications, technical notes, procedures, protocols, minutes, presentations, reports, manuals, operational products, databases, and emails. Most files are kept in PDF or Office formats.

Name:	B.USOC Wiki
Type:	Wiki
Ownership:	R3: Requires USOC's permission only
Content:	B.USOC Wiki is a local installation of the eGroupware™ software that serves as a handbook and is updated by operators and other ground staff when needed.

Name:	Local bugs database
Type:	Relational database (bugzilla)
Ownership:	R3: Requires USOC's permission only
Content:	The local bug database is running on Bugzilla and is used by B.USOC operators to enter bug reports related to the software running inside B.USOC.

3.3 Innovantage

The Innovantage use case contains both structured and unstructured data. The structured data is contained in MySQL RDBMS and contains UK vacancy data, geo-spatial information, company directory and application. The unstructured data consists of vacancy description, crawled company webpages, public domain news sources, company forums, social streams such as twitter etc.

3.3.1 Structured data sources

Name:	**Jobboard and Company Vacancy database**
Type:	MySQL Database
Ownership:	Innovantage
Content:	This includes various tables containing information about jobboard vacancies, company vacancies, jobboard advertisers, contacts, mapping advertisers to 3rd party company directory etc.

Name:	**Geo-spatial database**
Type:	MySQL Database
Ownership:	public
Content:	Comprehenvise geographic locations data of UK, containing all cities, counties, regions etc.

Name:	**3rd Party company database**
Type:	MySQL Database
Ownership:	DNB or Market Location
Content:	Comprehensive UK Company directory with name, address, phone, URL, Sector hierarchy, Sales, Profit, Employee Information

Name:	**Statistical database**
Type:	MySQL Database
Ownership:	Innovantage
Content:	Job Category hierarchy, Mapping from Job titles to categories

Name:	**User database**
Type:	MySQL Database
Ownership:	Innovantage
Content:	Application level data such as users, user alerts, users saved searches, access control, API clients and monitoring, crawl statistic, job root for company websites, lead management for clients etc.

3.3.2 Unstructured data sources

Name:	Company web site crawled by the focussed crawler
Type:	WWW
Ownership:	public
Content:	Contains aboutus, contactus, partners pages which can be used to depict company profile and activity

Name:	Job Vacancy Description
Type:	Text/ XML
Ownership:	Innovantage
Content:	Vacancy description can contain job description, company profile, job location, salary, required skills, experience, education etc.

Name:	social streams from Twitter/Facebook
Type:	WWW
Ownership:	Public
Content:	Profiles, textual information extracted from company pages

Name:	news networks, financials, blogs and forums
Type:	WWW
Ownership:	Public
Content:	Textual information about companies in news items, public opinions, recruitment activities

Name:	External RSS feeds, alerts
Type:	WWW
Ownership:	Public
Content:	Textual information about companies, recruitment activities

4 Queries from use Case Partners

In the following, queries which are either already performed by the use case partners or which the use case partners want to be answered by the CUBIST system are listed. These queries serve as input for the extension of SPARQL 1.0 as needed in CUBIST.

4.1 HWU

4.1.1 Queries which can already been answered without CUBIST

Currently, EMAGE makes the following queries available to end users via its web interface (italicised terms are variables to be instantiated):

- Where is *gene G* (not) detected?
- Where is *gene G* (not) detected in developmental stages *S1* to *S2*?
- What genes are (not) detected in *anatomy term T*?
- What genes are (not) detected in *anatomy term T* in developmental stages *S1* to *S2*?
- What genes are (not) detected in *3D spatial region R*?
- Where are the genes involved in *biological process*[5] *P* (not) detected?

4.1.2 Queries which can only be answered with CUBIST

Future queries that may be possible through the application of technology developed during CUBIST:

- Detection of possible inconsistencies and errors in the gene expression information;
- How does the level of expression *for gene G* change from developmental stage S1 to stage S2?
- How does the pattern of expression[6] for *gene G* change from *developmental stage S1* to *stage S2*?
- Which genes have similar expression patterns in *stages S1* to *S2*?
- What do the genes involved in *biological process P* have in common?
- What pathways[7] does *expression pattern cluster C* correspond to?
- Based on *expression pattern EP* from *experiment E*, which stage in the pathway was *E* performed at?

[5] Part of the Gene Ontology (www.geneontology.org)

[6] An expression pattern is a spatial representation of where the gene is expressed.

[7] Pathway data is not provided in EMAGE, but is freely available from a number of resources.

4.2 SAS

For Space Application Services, it is more important to distinguish between queries in normal situations, and queries which have to be answered when some anomalies occur. CUBIST is relevant for Space Application Services to analyze anomalous situations, thus the second kind of queries correspond to those queries which have to be answered by the CUBIST system.

4.2.1 Typical search activities

During Nominal activities, as part of the routine operations

- Console Logs: go through all console logs since previous shifts to check for open issues, actions to follow up, general awareness, ...
- TM archive: Check for completeness of the archive
- TM Displays: during periods of 'data connection' monitoring of the on-going activities by telemetry monitoring of the housekeeping data of the instruments and payload (temperatures, modes, command schedule tracking, ...)
- ISS ATL: check for possible reboosts of the station which requires a special SOLAR configuration
- SOLAR Mission Tool: check for upcoming activities:
    - Files needed are indeed available onboard
    - Feasibility schedule with respect to possible ATL changes, SAA passes and other constraints
    - Timeline coherence
- CEFN : assess open CEFNs, some examples and required actions:
    - Timeline review: check OSTPV and SOLAR Mission Tool, provide input through CEFN where needed
    - Requests for Powerdown: Check for payload developer's MEMO for minimal required SOLAR power consumption + provide into CEFN
    - IPV review: Check whether PODF related to B.USOC Payloads are correctly uplinked into new IPV+ provide approval (or approval with modification)
    - SOLAR Commanding (CEFN with B.USOC as author): check the SOLAR commanding plan for upcoming 24 hours, should be compliant with the SOLAR Mission Tool and using the correct PODF
    - Open ARs : check AR that may affect the payload, if needed report to COL FCT
    - Ground Segment operations, Check for CEFNs regarding ground segment maintenance or so, check impact on the SOLAR operations and inform Ground Controller
    - Older CEFNs: check previous inputs
- MDB: check command stack for upcoming activities
- IPV/PODF: retrieve correct PODF for execution activities

- IOT-AR/IOT-SPR: check open actions on B.USOC side
- SOLAR Documentation: used as background information to successfully operate SOLAR
- OSTPV: check SAA passes (instrument constraint), KU-S band (TDRS→ TM/TC constraint), night-day cycle (SOLAR constraint), SOLAR resources, correct planning. ....
- SOLAR Predictor Tool (tool developed by JMW)/ SOLAR Beta angle: check for Sun Visbility Windows
- BUSOC Mail: check for PIs input, files that have been provided, requests from other operations

4.2.2 Search activities in anomalous situations

During an anomalous situation and during near real time period afterwards (hours/days), some of the following searches could also be performed on console as part of the analysis

- Displays: check which parameters were off nominal
- Archive:
  - Retrieve specific parameters of the occurrence/period to insert in an Xcell file
  - Replay the occurrence using the displays
- MDB/Command History:
  - Check which commands have been sent
- IPV/PODF: check which procedure has been used
- IOT-AR: check whether the anomaly has occurred before and the analysis or CARTs disposition, if so
  - CEFN: If Anomaly has occurred before check recovery procedure used back then
  - IOT-SPRs: check the offline analysis if the anomaly has occurred before
  - Check Console logs of previous occurrence
- Console Log:
  - Check the logs of all involved parties during the occurrence
- Voice (VoCS): requests replay of the loops coordination (this has never happened for B.USOC)
- COL DMS: retrieve engineering documentation to support the analysis or recovery actions (Safety DP, User Manuals, ...)
- SOLAR Documentation: check reference documentation for analysis
- SOLAR operations Documentation: check reference on the agreed operations concept
- SOLAR shiftplanner Tool (JMW tool) : check which ground controller is on call
- SOLAR POC list: check numbers of PI/Payload Developer/.. to contact them for support
- SOLAR Mission Tool:
  - retrieve current planning to update it

- retrieve filenames to be uploaded

- SOLAR dropbox: retrieve files to be uploaded
- Flight Rules/ PL Reg: retrieve which Flight Rule or PL Reg is applicable (if so)

4.3 Innovantage

4.3.1 Queries which can already been answered without CUBIST

Currently Insight end users can make following queries

- Job Vacancy search by title, description, advertiser organization name or type (agency or direct employer), location hierarchy, category hierarchy, date posted, business sector, salary, jobtype, source(job boards or company website)
- Quick Vacancy search through keywords using Lucene text search facility
- Statistics from search results such as % of jobs from various sources, locations, job categories, industry sector etc.
- Company lookup for the advertisers from 3rd party directories such as Market Location or DNB, jobs associated and various recruitment statistic about an advertiser
- Yahoo news articles relevant to a particular advertiser by keyword match

4.3.2 Queries which can only be answered with CUBIST

- Radius job search around a town, county or postcode or part of a postcode by number of miles using the geo-spatial extension of sparql
- Skill, qualifications, company information within job description
- Job categorisation based on description, title and company sector
- Facts and sentiments about companies from external unstructured sources
- Mining hidden concepts and relations within the data

5 Queries from use Case Partners with comments

HWU

Requirement ID	Requirement Content
HWU01	Where is *gene G* (not) detected?
HWU02	Where is *gene G* (not) detected in developmental stages *S1* to *S2*?
HWU03	What genes are (not) detected in *anatomy term T*?
HWU04	What genes are (not) detected in *anatomy term T* in developmental stages *S1* to *S2*?
HWU05	What genes are (not) detected in *3D spatial region R* ?
HWU06	Where are the genes involved in *biological process P* (not) detected?
HWU07	Detection of possible inconsistencies and errors in the gene expression information
HWU08	How does the level of expression *for gene G* change from developmental stage S1 to stage S2?
HWU09	How does the pattern of expression for gene G change from developmental stage S1 to stage S2?
HWU10	Which genes have similar expression patterns in *stages S1* to *S2*?
HWU11	What do the genes involved in *biological process P* have in common?
HWU12	What pathways does expression pattern cluster C correspond to?
HWU13	Based on *expression pattern EP* from *experiment E*, which stage in the pathway was *E* performed at?

ID	Covered by SPARQL 1.0	Covered by SPARQL 1.1	Existing extensions	CUBIST specific extension	Comments
HWU01	x	x			Negative queries implemented by FILTER NOT EXISTS
HWU02	x	x			Negative queries implemented by FILTER NOT EXISTS
HWU03	x	x			Negative queries implemented by FILTER NOT EXISTS
HWU04	x	x			Negative queries implemented by FILTER NOT EXISTS
HWU05				x	3D spatial support is needed
HWU06	x	x			having molecular events in the store required

HWU07	x	x			Not a query Needs custom analysis & pre-processing. Only single control queries are possible
HWU08	x	x			Only single control queries are possible
HWU09	x	x			Checks for existing/missing gene expression are possible
HWU10	-	-	-	-	Not a query Needs custom analysis & pre-processing
HWU11	-	-	-	-	Not a query Needs custom analysis & pre-processing
HWU12	x				
HWU13	-	-	-	-	Not a query Needs custom analysis & pre-processing

SAS

Requirement ID	Requirement Content
SAS01	Console Logs: go through all console logs since previous shifts to check for open issues, actions to follow up, general awareness, ...
SAS02	TM archive: Check for completeness of the archive
SAS03	TM Displays: during periods of 'data connection' monitoring of the on-going activities by telemetry monitoring of the housekeeping data of the instruments and payload (temperatures, modes, command schedule tracking, ...)
SAS04	ISS ATL: check for possible reboosts of the station which requires a special SOLAR configuration
SAS05	SOLAR Mission Tool: check for upcoming activities: Files needed are indeed available onboard
SAS06	SOLAR Mission Tool: check for upcoming activities: Feasibility schedule with respect to possible ATL changes, SAA passes and other constraint
SAS07	SOLAR Mission Tool: check for upcoming activities: Timeline coherence
SAS08	CEFN : Timeline review: check OSTPV and SOLAR Mission Tool, provide input through CEFN where needed
SAS09	CEFN : Requests for Powerdown: Check for payload developer's MEMO for minimal required SOLAR power consumption + provide into CEFN
SAS10	CEFN : IPV review: Check whether PODF related to B.USOC Payloads are correctly uplinked into new IPV+ provide approval (or approval with modification)
SAS11	CEFN : SOLAR Commanding (CEFN with B.USOC as author): check the SOLAR commanding plan for upcoming 24 hours, should be compliant with the SOLAR Mission Tool and using the correct PODF

SAS12	CEFN : Open ARs : check AR that may affect the payload, if needed report to COL FCT
SAS13	CEFN : Ground Segment operations, Check for CEFNs regarding ground segment maintenance or so, check impact on the SOLAR operations and inform Ground Controller
SAS14	CEFN : Older CEFNs: check previous inputs
SAS15	MDB: check command stack for upcoming activities
SAS16	IPV/PODF: retrieve correct PODF for execution activities
SAS17	IOT-AR/IOT-SPR: check open actions on B.USOC side
SAS18	SOLAR Documentation: used as background information to successfully operate SOLAR
SAS19	OSTPV: check SAA passes (instrument constraint), KU-S band (TDRS→ TM/TC constraint), night-day cycle (SOLAR constraint), SOLAR resources, correct planning. ....
SAS20	SOLAR Predictor Tool (tool developed by JMW)/ SOLAR Beta angle: check for Sun Visbility Windows
SAS21	BUSOC Mail: check for PIs input, files that have been provided, requests from other operations
SAS22	Displays: check which parameters were off nominal
SAS23	Archive: Retrieve specific parameters of the occurrence/period to insert in an Xcell file
SAS24	Archive: Replay the occurrence using the displays
SAS25	MDB/Command History: Check which commands have been sent
SAS26	IPV/PODF: check which procedure has been used
SAS27	IOT-AR: check whether the anomaly has occurred before and the analysis or CARTs disposition, if so: a) CEFN: If Anomaly has occurred before check recovery procedure used back then b) IOT-SPRs: check the offline analysis if the anomaly has occurred before c) Check Console logs of previous occurrence
SAS28	Console Log: Check the logs of all involved parties during the occurrence
SAS29	Voice (VoCS): requests replay of the loops coordination (this has never happened for B.USOC)
SAS30	COL DMS: retrieve engineering documentation to support the analysis or recovery actions (Safety DP, User Manuals, ...)
SAS31	SOLAR Documentation: check reference documentation for analysis
SAS32	SOLAR operations Documentation: check reference on the agreed operations concept
SAS33	SOLAR shiftplanner Tool (JMW tool) : check which ground controller is on call
SAS34	SOLAR POC list: check numbers of PI/Payload Developer/.. to contact them for support
SAS35	SOLAR Mission Tool: retrieve current planning to update it
SAS36	SOLAR Mission Tool: retrieve filenames to be uploaded
SAS37	SOLAR dropbox: retrieve files to be uploaded
SAS38	Flight Rules/ PL Reg: retrieve which Flight Rule or PL Reg is applicable (if so)

ID	Covered by SPARQL 1.0	Covered by SPARQL 1.1	Existing extensions	CUBIST specific extension	Comments
SAS01	x				With simple timeline modelling, Date/time restrictions & ordering are possible in SPARQL 1.0
SAS02	x				Searching for error reports/abnormal events
SAS03	x	x			With simple timeline modelling, searching for abnormal or missing values
SAS04	x				
SAS05		x			Checking for missing facts regaring availability
SAS06	-	-	-	-	This is not a query
SAS07	x				simple timeline modelling
SAS08	x				
SAS09		x	x		Aggregation functions + complex SELECT expressions; FTS for CEFN data
SAS10	x	x			Checking for error reports and missing facts
SAS11	-	-	-	-	Not a query
SAS12	x	x			Checking for missing or out of range values
SAS13	x				without informing GC
SAS14	x				with simple timeline modelling
SAS15	x				
SAS16	x		x		FTS might provide additional flexibility
SAS17	x				
SAS18	x		x		FTS might provide additional flexibility
SAS19	-	-	-	-	Not a query. Requires custom pre-processing
SAS20					
SAS21	x		x		FTS might provide additional flexibility
SAS22	x	x			check for present or missing values/facts

SAS23	x				with simple timeline modelling
SAS24	x				simple timeline modelling
SAS25	x				if 'sent' status is presented/reflected in the store
SAS26	x				
SAS27					
SAS27a	x				with simple timeline modelling
SAS27b	x				with simple timeline modelling
SAS27c	x				with simple timeline modelling
SAS28	x				with simple timeline modelling
SAS29					
SAS30	x				
SAS31	x				
SAS32	x				
SAS33	x				with simple timeline modelling
SAS34	x	x			PI list retrieval, counting if necessary
SAS35	x				'current' should be marked somehow
SAS36	x				filenames related to current planning
SAS37	-	-	-	-	Not a query
SAS38	-	-	-	-	Not a query

Innovantage

Requirement ID	Requirement Content
INN01	Job Vacancy search by title, description, advertiser organization name or type (agency or direct employer), location hierarchy, category hierarchy, date posted, business sector, salary, jobtype, source(job boards or company website)
INN02	Quick Vacancy search through keywords using Lucene text search facility
INN03	Statistics from search results such as % of jobs from various sources, locations, job categories, industry sector etc.
INN04	Company lookup for the advertisers from 3rd party directories such as Market Location or DNB, jobs associated and various recruitment statistic about an advertiser
INN05	Yahoo news articles relevant to a particular advertiser by keyword match
INN06	Radius job search around a town, county or postcode or part of a postcode by number of miles using the geo-spatial extension of sparql
INN07	Skill, qualifications, company information within job description
INN08	Job categorisation based on description, title and company sector
INN09	Facts and sentiments about companies from external unstructured sources
INN10	Mining hidden concepts and relations within the data

ID	Covered by SPARQL 1.0	Covered by SPARQL 1.1	Existing extensions	CUBIST specific extension	Comments
INN01	x				
INN02			x		FTS support in OWLIM and others
INN03		x			implementable by: aggregates and complex SELECT clauses allowing expressions in them
INN04		x			implementable by: aggregates and complex SELECT clauses allowing expressions in them
INN05			x		FTS support
INN06			x		OWLIM geo-spatial support
INN07	x				
INN08	-	-	-	-	Not a query. Requires pre-processing and analysis
INN09	-	-	-	-	Not a query. Requires pre-processing and analysis
INN10	-	-	-	-	Not a query. Requires pre-processing and analysis

	Sources from WWW	Unstructured textdocuments (word,txt,pdf)	Structured textdocuments (word,txt,pdf)	XML	Excel	DB	Ontologies
Task to perform:	1) hyponymy and other relation extraction 2) extract common phrases extraction 3) Taxonomy building 4) sentiment analysis	Annotate documents with concepts/instances in the ontology	Extract facts from documents	Extract facts from documents	Extract facts from documents	Transform (parts) of DB into triples	Direct import
Technologies for harvesting	Crawler for harvesting and IE (based on GATE) for processing	Crawler for harvesting and IE (based on GATE) for processing	Crawler for harvesting and IE (based on GATE) for processing.	„XML2RDFParser“	Possibly an Excel-macro which models columns as attributes and columns as instances.	Triplifier, e.g. Triplify or D2R Server	
Sources	**From Innovantage** **web pages** **social streams from Twitter/LinkedIn** **company blogs and forums**	**From SAS:** **CEFN/ICN Columbus flight notes:** issue tracking web pplication **Consoloe logs :** short timestamped messages from operators **JOIP (Joint Operations Interface Procedures), FR (Flight Rules), PL Reg (Payload Regulations):** JOIP governs the operations of the payload andthe communications between various parties involved in the Columbus environment. FR and PL Reg are similar to JOIP and cover standard situations and operations tasks for Columbus. Most of these documents are created in Microsoft Word and distributed as PDF files. **eRoom:** a Documentum™ installation that serves as a repository of interface control documents, specifications, technical notes, procedures, protocols, minutes, presentations, reports, manuals, operational products, databases, and emails. Most files are kept in PDF or Office formats.	**From SAS:** **OSTP (onboard short term planner):** a web application that displays timeline information about ground and onboard procedures, schedules, and activities. Data can be received in a textual, structured format. **Telemetry data:** organized into packets. It contains temperature measurements, voltage and current readings, various operational states and reports. The metadata of telemetry is found in MDB. **Science data:** technically part of telemetry **Telecommands:** structured data sent to payload during the operations. A complete history of telecommands over the operating live of the payload is saved. **System and software logfiles:** Typical examples are apache logs and windows event logs. **Auxiliary data:** from public sources		**From SAS:** **IPV (International Procedure Viewer):** a database of onboard and ground procedures in XML format **PODF (Payload Operations Data File):** a set of documents governing the operations of the payload in the pre-defined situations. Available in IPV and retrievable as XML files.	**From SAS:** **DOR (daily operations reports):** They include the summary of the perations by ISS path, as well as reports on discrepancies and anomalies. **SOLAR Mission Tool:** An excel file that serves to generate DORs, implement configuration control, etc. **SPRdb (System Problem Report database):** an online issue tracking system for ground and onboard issues about Columbus, its payloads, and control centres. The contents of SPRdb can be extracted into XML files.	**From SAS:** **MDB (mission database):** machine-readable description of telemetry **Local bugs database:** running on Bugzilla, used to enter bug reports related to the software running inside B.USOC. **From HWU:** **EMAGE:** a database of in situ gene expression data in the mouse embryo. It contain information about the kind of experiments (assays) and their results in terms of image data and their annotations. **EMAP :** A resource describing the anatomy of the mouse embryo. It contains two main aspects - a set of virtual 3D embryos at different Theiler stages of development, and the associated EMAP anatomy ontology. **From Innovantage** **MySQL database** **geo-spatial data from Ordinance Survey** **company data from DNB**	**From HWU:** **EMAP anatomy ontology:** a structured list of 13,000+ terms that describe visible anatomical structures in the developing mouse embryo. **Gene Ontology:** a controlled vocabulary of terms for describing gene product characteristics and gene product annotation data (see www.geneontology.org)