SIT764 - Project Document

Visualising a complex organisational history

Kosta Polkalioukhine Team Leader

Victor Jayakody Arachchige Don Deputy Leader

Isaac Tesla Client Communicator Lead, Secretary

Ramu Katha Client Communicator Deputy

Ali Hassan Baig Documentation

Contents

[1.0 Business Case 4](#_Toc536804104)

[1.1 Strategic Context 4](#_Toc536804105)

[1.1.1 The need 4](#_Toc536804106)

[1.1.2 Background 4](#_Toc536804107)

[1.1.3 Past Attempts 4](#_Toc536804108)

[1.2 Benefits Realisation 4](#_Toc536804109)

[1.2.1 Benefits 4](#_Toc536804110)

[2.0 Project Charter 5](#_Toc536804111)

[2.1 High-level Goal 5](#_Toc536804112)

[2.2 Aims 5](#_Toc536804113)

[2.2 Governance Structure 5](#_Toc536804114)

[2.3 Key Stakeholders 5](#_Toc536804115)

[2.4 Assumptions 6](#_Toc536804116)

[2.5 Constraints 6](#_Toc536804117)

[2.6 Risks 6](#_Toc536804118)

[3.0 Integration Management Plan 7](#_Toc536804119)

[4.0 Scope 8](#_Toc536804120)

[4.1 Product Description 8](#_Toc536804121)

[4.2 Dependency Linkages 8](#_Toc536804122)

[4.3 Measures of Project Success 8](#_Toc536804123)

[5.0 Objectives and Deliverables 8](#_Toc536804124)

[6.0 Project Approach 9](#_Toc536804125)

[6.1 Project timeline 9](#_Toc536804126)

[6.2 Requirement specifications 9](#_Toc536804127)

[6.3 Description 9](#_Toc536804128)

[7.0 Project Milestones 11](#_Toc536804129)

[7.1 Project Schedule 11](#_Toc536804130)

[7.2 Sprint 1 12](#_Toc536804131)

[7.3 Sprint 2 13](#_Toc536804132)

[7.4 Sprint 3 16](#_Toc536804133)

[7.5 Sprint 4 20](#_Toc536804134)

[8.0 Technical details 22](#_Toc536804135)

List of figures

[Figure 1 - Project architecture 12](#_Toc536804057)

[Figure 2: Database site hosted in AWS 13](#_Toc536804058)

[Figure 3: Visual graph of tested dataset 13](#_Toc536804059)

[Figure 4: Website front-end 14](#_Toc536804060)

[Figure 5:Tomcat 15](#_Toc536804061)

[Figure 6: Vertices and Edges to dataset 16](#_Toc536804062)

[Figure 7:Vertices and edges to dataset 16](#_Toc536804063)

[Figure 8: Vertices and Edges to dataset 17](#_Toc536804064)

[Figure 9: Vertices and Edges to dataset 17](#_Toc536804065)

[Figure 10: Query tool integrated onto website 18](#_Toc536804066)

[Figure 11: rdf4j server 19](#_Toc536804067)

[Figure 12:Visualization tool integrated onto website 20](#_Toc536804068)

[Figure 13: Sankey diagram 20](#_Toc536804069)

[Figure 14:Force graph 21](#_Toc536804070)

# Business Case

## 1.1 Strategic Context

### 1.1.1 The need

With continuous improvement comes the requirement to trial various methodologies and technologies. This project is to find opportunities through the exploration of graph databases for future use across several organisational departments and public and commercial sectors.

### 1.1.2 Background

CSIRO is Australia's national research agency. Its focal role is to enhance the social and economic performance of the industry for the greater good of the community. It is an enterprise level organisation which maintains more than 50 sites across multiple countries, employing about 5500 people.

Since its foundation in 1916, it has evolved and had multiple operating models and organisational structures, therefore it has a complex organisational history. The project focuses on developing a formal representation of the CSIRO’s organisational history using the W3C Organisational Ontology and developing a tool to visualise and query it.

### 1.1.3 Past Attempts

Database construction has occurred previously with a partially complete dataset for organisational entities built from the Encyclopaedia of Australian Science.

## 1.2 Benefits Realisation

### 1.2.1 Benefits

**Interactive historical web application**

An improved ability to rapidly visualise and comprehend the complexity of CSIRO organisational history and query this history in a dynamic way. The current measure includes various charts and hyperlinked web pages on the Encyclopaedia of Australian Science. The target measure is to have an interface which enables interaction to be able to view information in a more easily identifiable way. The timeframe to achieve this is 6 months.

**Organisation Ontology**

This benefit is to gain understanding of the application domain for the W3C Organisation Ontology. The current measure is a partial understanding of applicability for the W3C Organisational Ontology against the information domain. The target measure is to understand the applicable uses and whether the W3C Organisational Ontology has further application for the CSIRO. The timeframe to achieve this is 6 months.

# 2.0 Project Charter

## 2.1 High-level Goal

Complete the formal representation of CSIRO’s organisational history and generate visualisations that can be queried through an API.

## 2.2 Aims

* The immediate project goal is to develop a formal representation of the organisational history of CSIRO, and tools to visualise and query it.
* The scientific goal is to evaluate the capability of the W3C Organisation Ontology for an evolving organisational structure.

## 2.2 Governance Structure

CSIRO (the client) has authorised the project in conjunction with Deakin University. The project manager is Simon James. The team consists of university students completing Information Technology Master’s degrees. The project manager is the escalation point for issues.

## 2.3 Key Stakeholders

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Email** |
| Simon Cox | CSIRO project owner | [Simon.cox@csiro.au](mailto:Simon.cox@csiro.au) |
| Jonathan Yu | CSIRO project owner | [Jonathan.yu@csiro.au](mailto:Jonathan.yu@csiro.au) |
| Simon James | Deakin University Project Manager | [simon.james@deakin.edu.au](mailto:simon.james@deakin.edu.au) |
| Kosta Polkalioukhine | Team Leader | [kpokalio@deakin.edu.au](mailto:kpokalio@deakin.edu.au) |
| Victor Jayakody Arachchige Don | Deputy Leader | [vjayakod@deakin.edu.au](mailto:vjayakod@deakin.edu.au) |
| Isaac Tesla | Client Communicator Lead, Secretary | [itesla@deakin.edu.au](mailto:itesla@deakin.edu.au) |
| Ramu Katha | Client Communicator Deputy | [rkatha@deakin.edu.au](mailto:rkatha@deakin.edu.au) |
| Ali Hassan Baig | Documentation | [ahbaig@deakin.edu.au](mailto:ahbaig@deakin.edu.au) |

The following role definitions are being applied to the resources assigned to this project:

|  |  |
| --- | --- |
| **Role** | **Description** |
| Project Manager | Oversees the project, provides guidance. |
| Project Owner | Individual/company that initiates a project, finances it, contracts it out, and benefits from its outputs. |
| Team Lead | A team leader is someone who provides guidance, instruction, direction and leadership to a group of individuals (the team) for the purpose of achieving a key result or group of aligned results. |
| Stakeholder | Key provider of requirements and recipient of project deliverable and associated benefits. |
| Team Member | Working project team member who analyses, designs and ultimately improves or replaces the business processes. This includes collaborating with teams to develop high level process designs and models, understanding best practices for business processes and partnering with team members to identify appropriate opportunities, challenging the old rules of the business and stimulating creating thinking, and identifying organisational impact areas. |
| Client manager | Serve as liaisons between the team/company and its client. |
| Secretary | Takes minutes of all meetings. Circulates to all team members. |

## 2.4 Assumptions

1. All required information to create a complete dataset is freely available.
2. Software exists to achieve the desired outcome.
3. Relative difficulty of the project is well-aligned with the chosen team.
4. Free-tier cloud provision services are available to run the required server instances.
5. Resources will be available to adequately staff the project.
6. Customer requirements will not significantly deviate from the initial project proposal document.

## 2.5 Constraints

1. Hardware requirements for server instances vary, coupled with hardware requirements for different graph database software installations.
2. Time to complete the project is limited to SIT764 for initial stages of development. Limited time – Each team member has a limited amount of time that they can dedicate to the project over the trimester period. This may mean that parts of the project that are especially time consuming, such as manual data entry, remain partially complete.
3. Limited skills – the team has not utilized most of the technologies required for this project previously. This may have an effect how sophisticated the deliverables are.

## 2.6 Risks

1. Allocated time insufficient to complete requirements.
2. Communication breakdown due to team spread across three States.
3. Redundancy of work if architecture simplified.
4. Products do not match client expectations – this risk is mitigated by following an agile  
   method of development. The client will be able to provide feedback regularly, so that  
   corrective action can be taken (if and when required).
5. Products are not delivered in a timely manner – this risk is mitigated by following an agile method of development. If a team member responsible for a specific aspect of the project is lagging behind be the end of a sprint, an extra resource will be allocated to assist them.

# 3.0 Integration Management Plan

|  |  |
| --- | --- |
| Project lifecycle | * G1 = Phase Gate 1 Approval of Team allocation required before the project may proceed to the next phase. * G2 = Phase Gate 2 Completion of Kick-off meeting required before the project may proceed to the next phase. * G3 = Phase Gate 3 Approval of final Status Report required before the project may proceed to the Closure Phase. |
| Project Management methodology & tools | * Agile method; using a parallel model, integration to conclude implementation phase. * GitHub – store working files * Trello – Track sprints * Slack – Team communication * Outlook – Email communications * Skype for Business – Communicate with client, team meetings * OnTrack – Project monitoring tool |
| Integrated Change Control | * All changes to be discussed by team. * Escalation point of Project Manager. |
| Status Reports | * To be completed in accordance with OnTrack project monitoring tool. |
| Closure | * + Completed CSIRO organisational history dataset in RDF (pre-2000 and post 2000)   + Graph database prototype completed   + Website front end prototype completed   + Visualization prototype completed   + API prototype completed   + Documentation competed |

# 4.0 Scope

Includes: The final product includes a GraphDB database containing CSIRO’s organisational history datasets. The database is accessible through a web front end by way of query and visualisation. External accessibility will be provided through a REST API. These will all be delivered in prototype form.

Does Not Include: The project does not attempt to deliver a finished product (prototype only). The project is standalone and will not be integrated with CSIRO systems.

## 4.1 Product Description

The product solution is a Resource Description Framework (RDF) format database containing CSIRO’s organisational data. The database is connected to a web interface with visualisation and query functionality. The stretch goal is to create a rich REST API, improving accessibility.

## 4.2 Dependency Linkages

The visualisations, query tool and API can only be fully functional, once the CSIRO organisational dataset is completely loaded to GraphDB.

* + W3C Organisation Ontology https://www.w3.org/TR/vocab-org/
  + W3C SPARQL RDF Query Language <https://www.w3.org/TR/sparql11-query/>

## 4.3 Measures of Project Success

The project will be successful if the client is able to query the database (visually and via text inputs) and successfully extract useful chronological information on CSIRO’s organisational structure.

# 5.0 Objectives and Deliverables

To test the concept of utilizing the W3C Organisational Ontology when applied to CSIRO. The product will be a useful resource to track and query organisational change over time but is designed to be a microcosm that will inform the feasibility of creating further functionality by incorporating HR and research output data for a richer organisational view.

**Deliverables (included):**

* Consolidated CSIRO RDF datasets (pre-2000 and post 2000)
* CSIRO RDF datasets stored in Graph DB
* Web interface with a simple front-end query tool for the GraphDB
* D3.js visualisation of the CSIRO organisational history
* REST API

**Deliverables (excluded):**

* Finalised products (prototypes only)
* Integration with CSIRO systems (stand-alone solution)
* A complete history of the organisational structure and changes of CSIRO (1916-2018), formalised using the W3C Organisation Ontology, and linked to standard classifications of research functions
* A service to visualise and query this dataset

# 6.0 Project Approach

The project will be implemented in 4 sprints. Each sprint will provide incremental improvements/new features on previous deliverables.

## 6.1 Project timeline

Key Project milestones relative to project start are as follows:

|  |  |
| --- | --- |
| **Project Milestones** | **Target Date** |
| Sprint 1 | Nov 5, 2018 |
| Sprint 2 | Nov 26 to Dec 10, 2018 |
| Sprint 3 | Dec 10 to Dec 17, 2018 |
| Sprint 4 | Jan 7 to Jan 14, 2018 |
| Project completion | Feb 4, 2018 |

## 6.2 Requirement specifications

This section consists of requirements outlined in the document provided by the project owner at the start of the project.

## 6.3 Description

CSIRO is Australia's national research agency. Since it was founded in 1916 it has had many operating models, with corresponding organisational structures. Teams, labs and capabilities contributing to research topics such as animal health, soils, forestry, mineral exploration, remote-sensing, and polymers have been based in various units (divisions, flagships, business units) over the lifetime of CSIRO.

Sometimes it is necessary to track down the capability responsible for a piece of work long after the original organisational unit that appears in its citation has ceased to exist. To support this function, a query able organisational history of CSIRO is required, and a visual and programmatic query interface constructed.

A preliminary dataset was acquired from the Encyclopaedia of Australian Science (EOAS) and has been converted into a 'semantic web' representation using the W3C Organisation Ontology. This covers the period from 1916 until the early 2000s, at a limited granularity. Subsequent organisational structures for CSIRO are partially available at CSIROPedia, with details in printed (PDF) reports. This data is currently only available visually without an external query capability.

The project will:

* complete the formal representation of CSIRO’s organisational history
* (if possible) link it to standard research function classifications
* generate visualizations (e.g. Sankey diagrams) and an API to allow the organisational structures to be linked to other knowledge services
* develop some useful graph queries
* (stretch goal) design an organisational structure API

# 7.0 Project Milestones

Details of what was achieved by the team for each sprint.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Milestones | Planned date | Current estimated date | Variance | Status  👍 = On track  ⌛ = Behind  💣 = Issue  ✓ = Complete | Comments |
| Project Plan |  |  |  |  |  |
| Sprint 1 |  |  |  |  |  |
| Sprint 2 |  |  |  |  |  |
| Sprint 3 |  |  |  |  |  |
| Sprint 4 |  |  |  |  |  |
| Close Project |  |  |  |  |  |

## 7.1 Project Schedule

## 7.2 Sprint 1

The following outputs were delivered during sprint 1 of the project:

1. Research on graph database structures + Test 3 different products:
   1. ArangoDB: multi-format for presenting information, problems with RDF (potential future risk)
   2. Neo4j: Works with RDF, uses Cypher in place of SPARQL.
   3. GraphDB: Native with both RDF and SPARQL. (recommend)

1. The development of a draft product architecture (see figure 1)

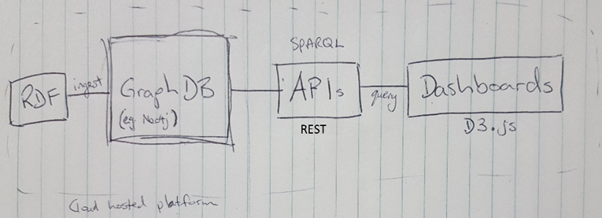


Figure 1 - Project architecture

1. The development of a [scope document](https://deakin365-my.sharepoint.com/:w:/g/personal/kpokalio_deakin_edu_au/EXi5yeJECVRPtmY3zZJfIaIBkXQnLMWy9crPuGB-leEF8A?e=Ndam7x) - a document describing project deliverables and major objectives

## 7.3 Sprint 2

The following outputs were delivered during sprint 2 of the project:

1. Hosting Database site in AWS (see figure 2)

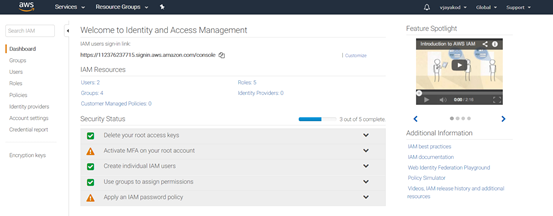


Figure 2: Database site hosted in AWS

1. Provisioned GraphDB on Azure, tested dataset pulled from GitHub and SPARQL query on dataset (see figure 3)

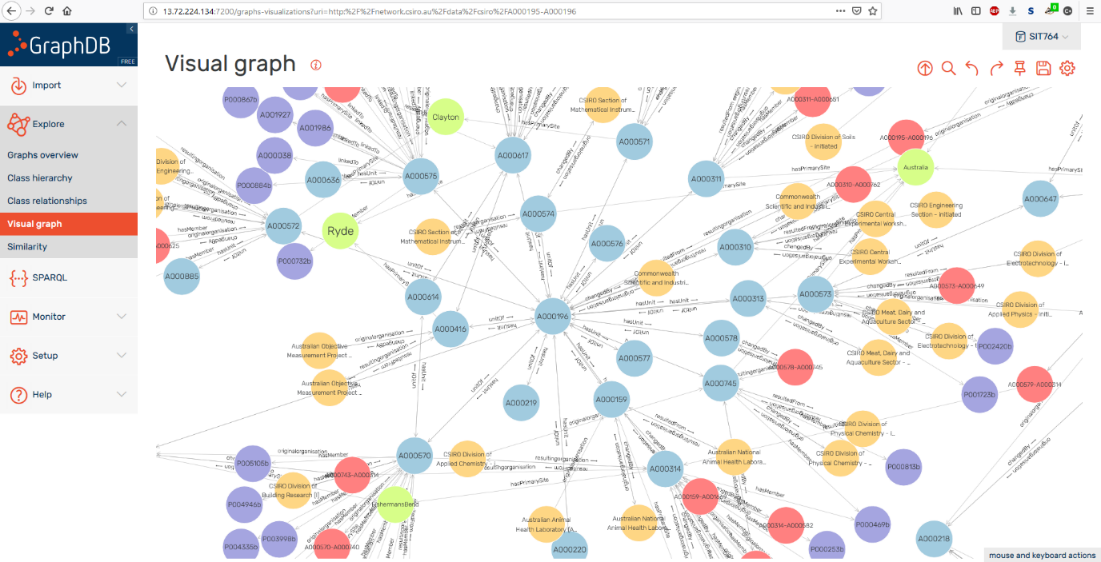


Figure 3: Visual graph of tested dataset

1. Draft database front-end created using HTML and Java Script (see figure 4)



Figure 4: Website front-end

1. Tested a prototype D3.js Sankey visualisation with dummy data, testing is done on local tomcat due to its Java support (see figure 5)

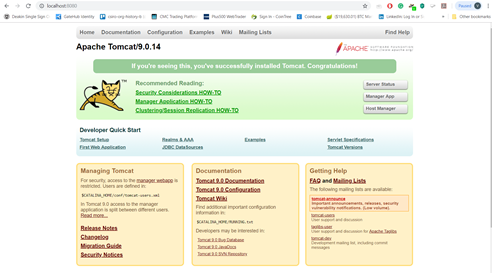


Figure 5:Tomcat

1. Running SPARQL queries on Apache Fuseki server to get XML/JASON outputs. Apache Jena Fuseki is a component within the Jena framework from Apache organisation. It consists of SPARQL server with support of REST-style querying over http. An RDF file with .ttl file extension format can be uploaded to create the dataset. All the queries will be running on local machine with options to download or see the results in various file formats such as JSON, CSV or XML.

## 7.4 Sprint 3

The outputs of sprint 3 are listed below:

1. Added vertices and edges to dataset using CSIROpedia diagrams and CSIRO website (see figures 6,7 and 8)

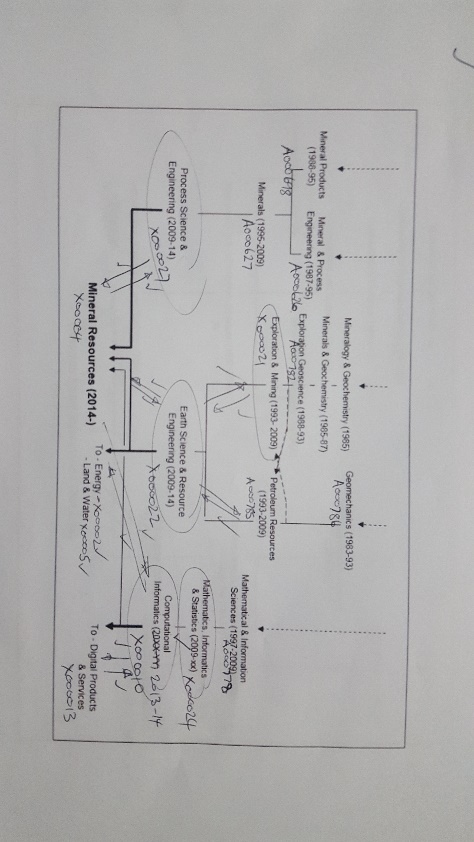
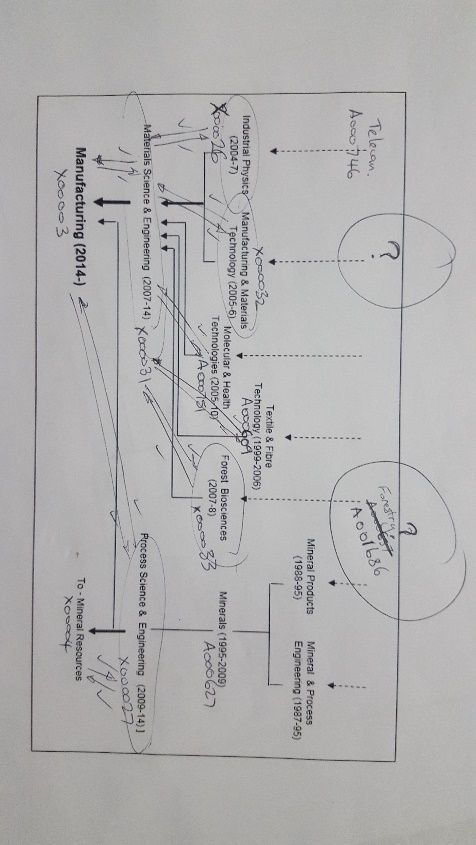


Figure 6: Vertices and Edges to dataset

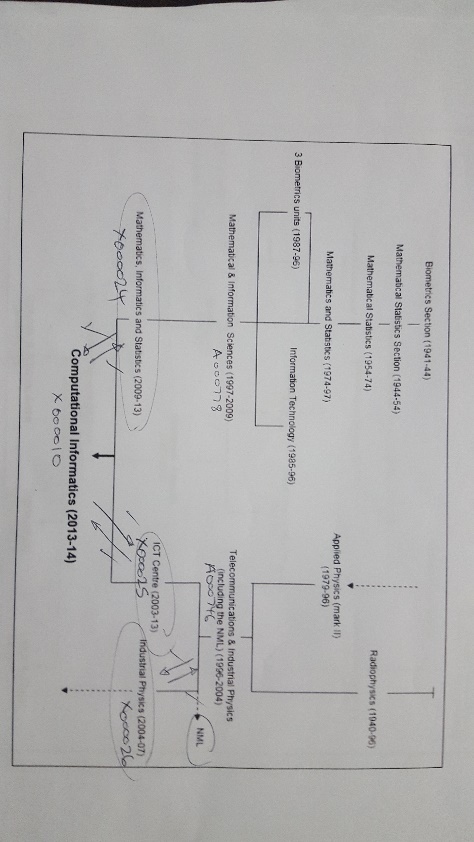


Figure 7:Vertices and edges to dataset

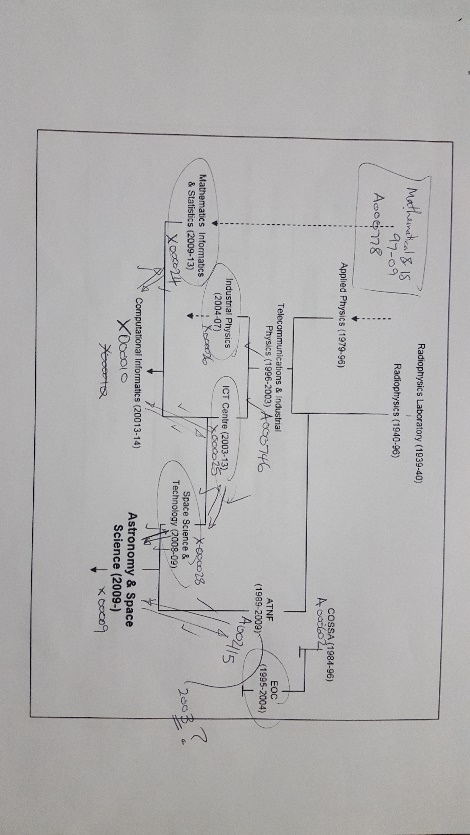
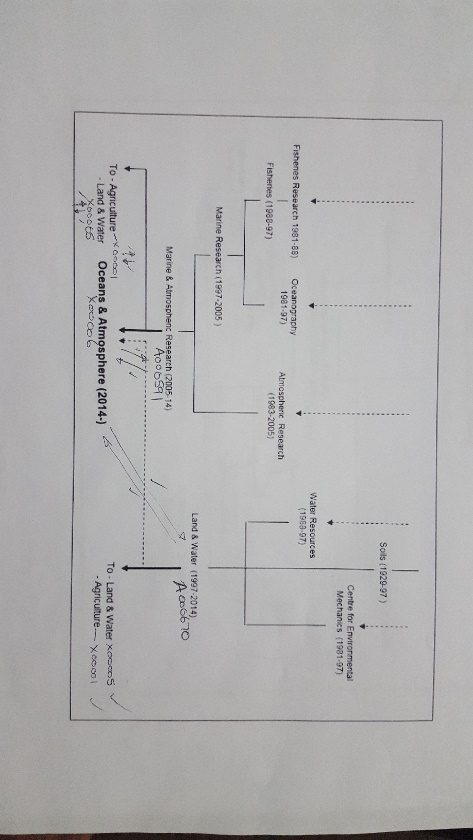
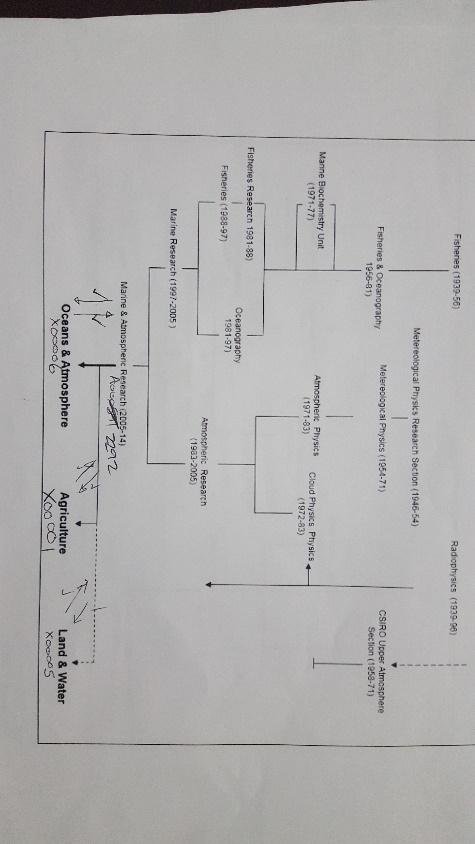


Figure 8: Vertices and Edges to dataset

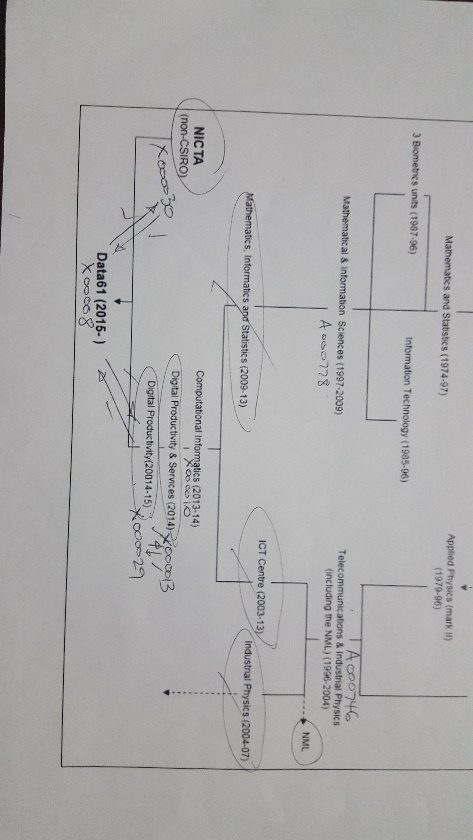
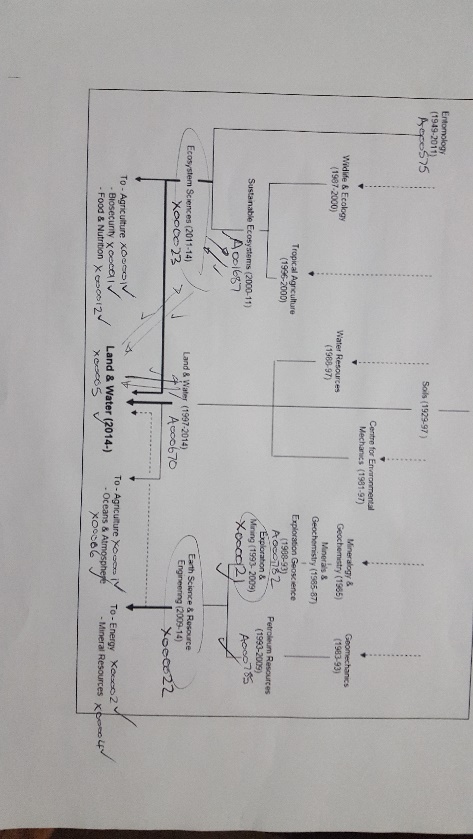
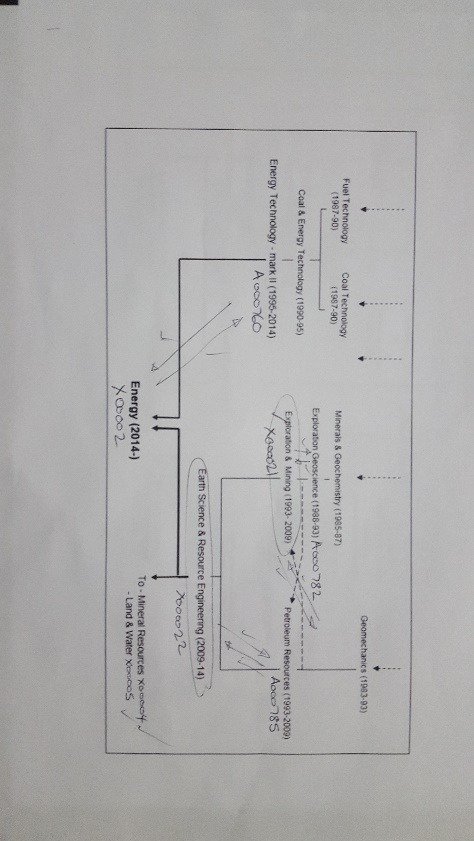


Figure 9: Vertices and Edges to dataset

1. Creation of working query tool and integration into website (see figure 9)

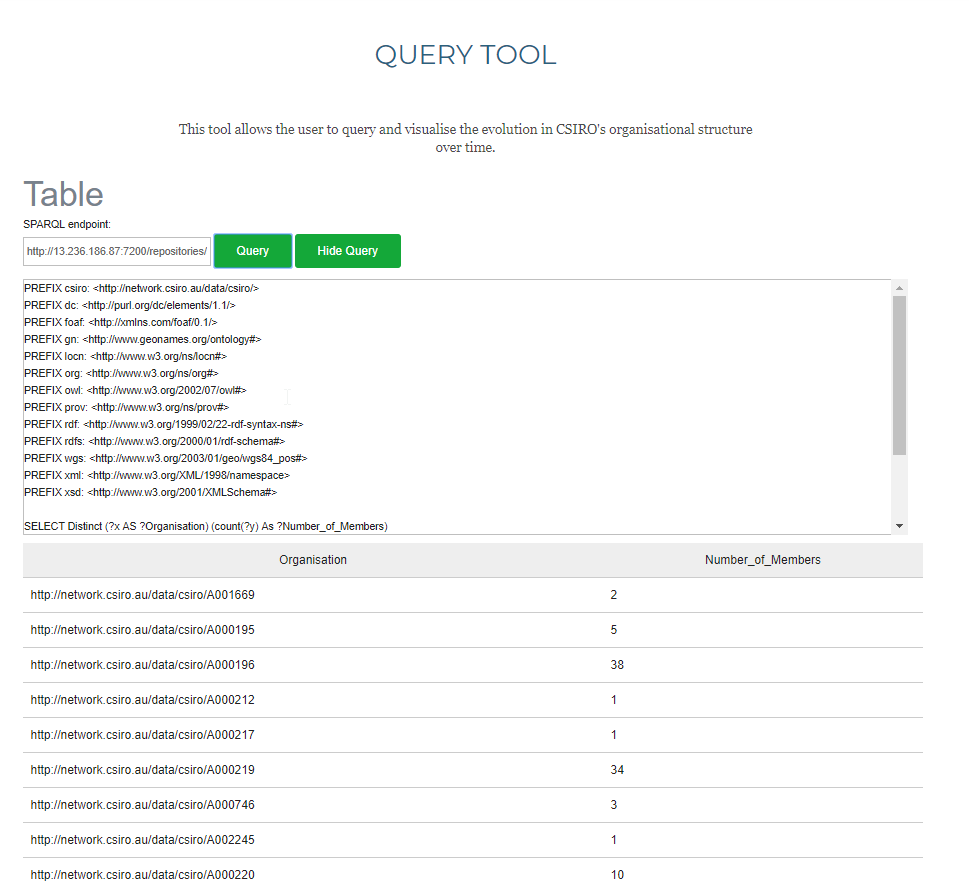


Figure 10: Query tool integrated onto website

1. Exploring hosting solution on Tomcat server with RDF4J to take advantage of built in REST API. Solution successfully deployed on a localhost.

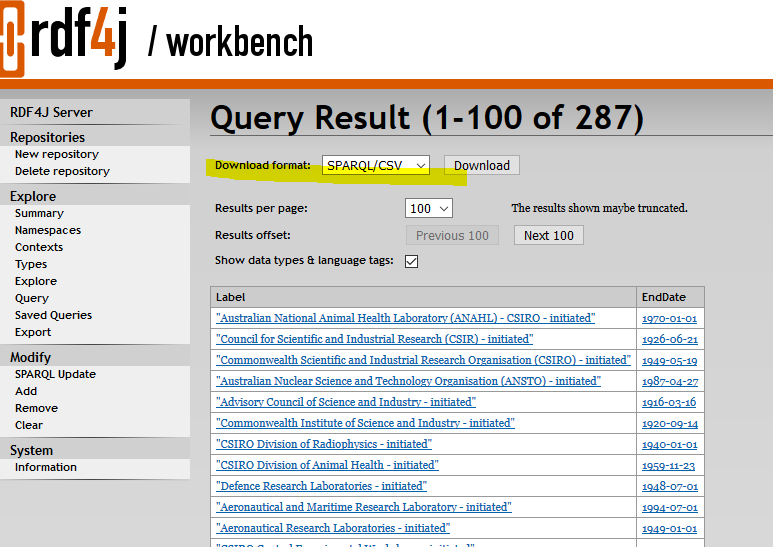


Figure 11: rdf4j server

## Sprint 4

This sprint output is creation of working graphical query tool and integration into website. Operational graphs include: Sankey, Pie chart and Force graph.



Figure 12:Visualization tool integrated onto website

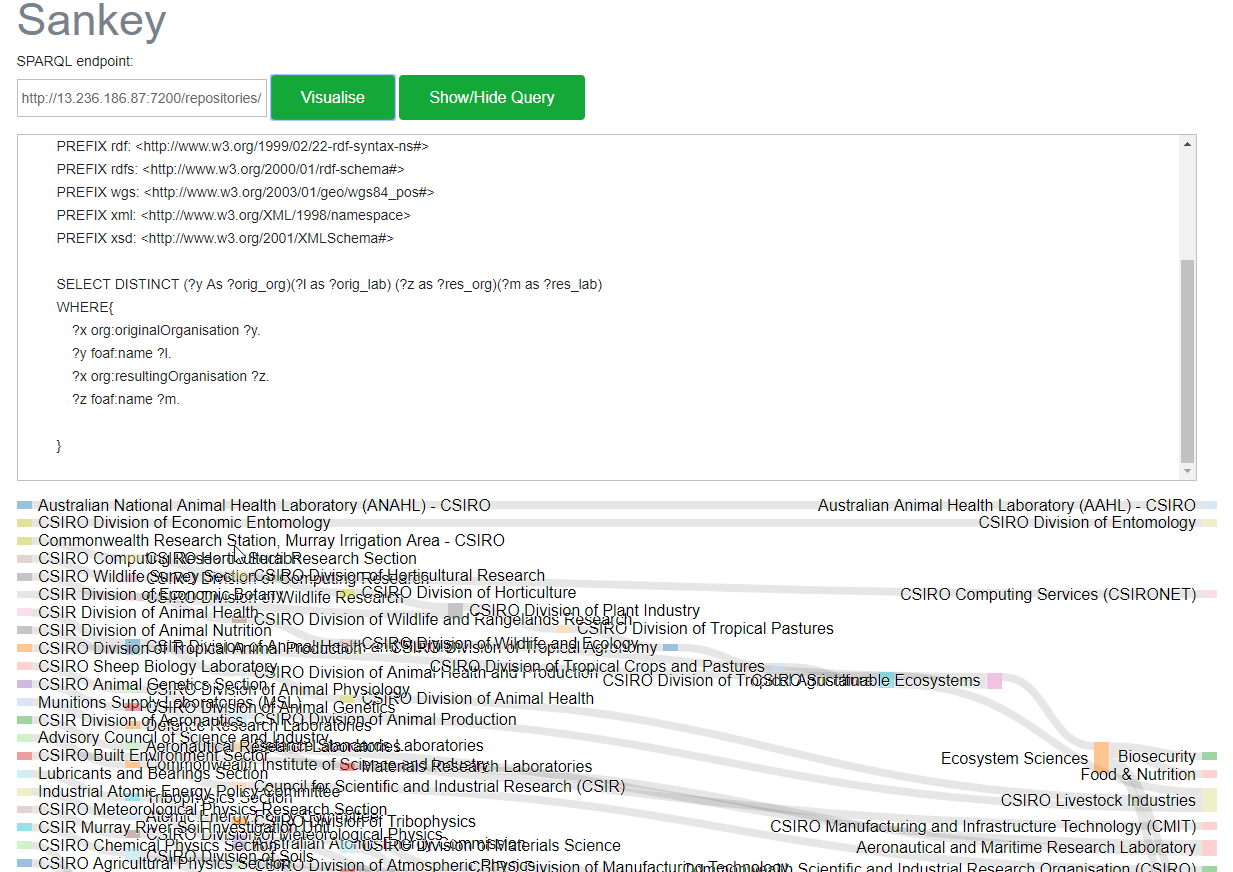


Figure 13: Sankey diagram

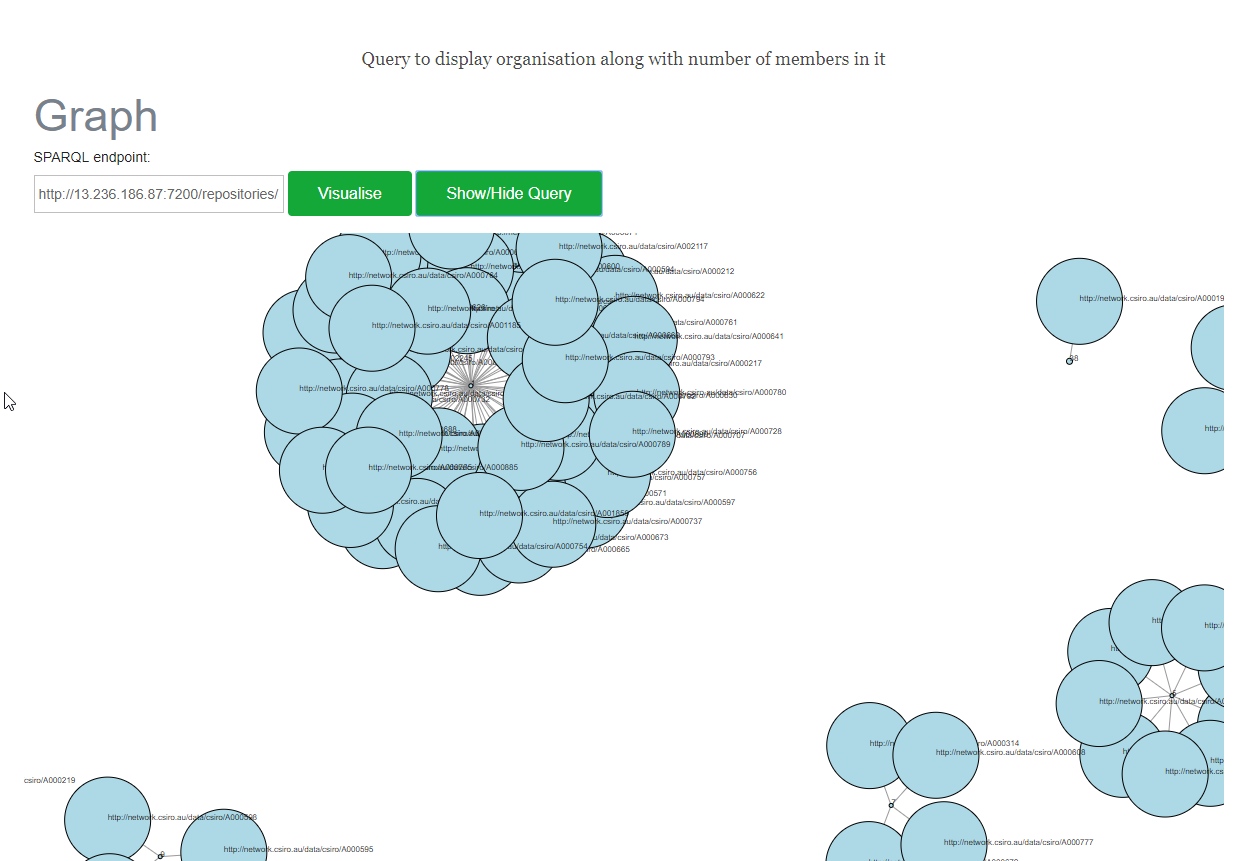


Figure 14:Force graph

# 8.0 Technical details

This section contains technical details of processes carried out for the completion of the project, the sole reason for making a separate section for technical details was to ensure that technical audience gets a better insight on the project.

1. **SPARQL for querying the graph database:**

In this project, we used Resource Description Framework (RDF) to create nodes and vertices for the project. For executing the SPARQL queries independently, we used two options. They are

* GraphDB – a semantic graph database from Ontotext.
* Apache Jena Fuseki- a JAVA web application (WAR file) based SPARQL server.

Installations:

1. *GraphDB*

Free basic edition of GraphDB is available for all. It has options to explore, visualise and query a given graph data repository. The installation file can be downloaded from <https://www.ontotext.com/free-graphdb-download/>

Minimum requirements: 2 GB Memory, 2 GB Disk space, JDK 1.8 or higher version (Optional for free edition)

1. *Apache Jena Fuseki:*

It is a component within the Jena framework from Apache organisation. It consists of SPARQL server with support of REST-style querying over http. It can be downloaded from <https://jena.apache.org/download/index.cgi>

Minimum requirements: 2 GB Memory, 2 GB Disk space, JDK 1.7 or higher version

Uploading the data:

* The RDF data can be downloaded in Turtle format (ttl format) from GitHub link <https://github.com/teslaImpertior/csiro-org-history-b/blob/master/complete.ttl>
* For GraphDB, a repository along with properties such as name, data type and constraints need to specified before creation.
* For Fuseki server: A RDF file with .ttl file extension format can be uploaded to create the dataset.

Running and saving the queries:

All the queries will be running on local machine with options to download or see the results in various file formats such as JSON, CSV or XML.

1. **Workbench REST API**

GraphDB Workbench REST API can be called as curl commands for test scripting Curl http://<base\_url>/repositories/{repositoryId}?query={query} -H "Accept: application/sparql-results+json"

*Example:*

*Curl* http://192.168.1.10:7200/repositories/SIT764?query=select+\*+where+%7B+%0A%09%3Fs+%3Fp+%3Fo+.%0A%7D+limit+10+%0A -H "Accept: application/sparql-results+json"

Using GraphDB Workbench REST API data can also be imported in to a database more information

<http://graphdb.ontotext.com/documentation/standard/devhub/workbench-rest-api/curl-commands.html>

*GraphDB Workbench REST API with Java script*

var request = new Request('http://192.168.1.10:7200/repositories/SIT764?query{query} ', {

headers: new Headers({

'Accept':'application/sparql-results+json'

})

});

fetch(request)

1. **Content request using HTTP headers**

Using http request heads content types like “text/plain”, “text/html”, “application/xml”, “application/json” can be requested

*Example:*

</div>

<script>

var request = new Request('http://192.168.1.10:7200/repositories/SIT764?query=PREFIX%20prov%3A%20%3Chttp%3A%2F%2Fwww.w3.org%2Fns%2Fprov%23%3E%0APREFIX%20org%3A%20%3Chttp%3A%2F%2Fwww.w3.org%2Fns%2Forg%23%3E%0APREFIX%20rdfs%3A%20%3Chttp%3A%2F%2Fwww.w3.org%2F2000%2F01%2Frdf-schema%23%3E%0APREFIX%20foaf%3A%20%3Chttp%3A%2F%2Fxmlns.com%2Ffoaf%2F0.1%2F%3E%0ASELECT%20%3Flabel%20%3FendDate%0AWHERE%20%7B%0A%20%20%20%20%3Forg%20prov%3AendedAtTime%20%3FendDate%20%3B%20%20%0A%20%20%20%20%20%20%20%20%20org%3AoriginalOrganization%20%3Forigin%20%3B%0A%20%20%20%20%20%20%20%20%20org%3AresultingOrganization%20%3Fend%20%3B%0A%20%20%20%20%20%20%20%20%20rdfs%3Alabel%20%3Flabel%20.%0A%20%20%20%20%23%3Forg%20foaf%3Aname%20%3Forganisation%20.%0A%7D', {

headers: new Headers({

'Accept':'application/sparql-results+json'

})

});

fetch(request)

.then(data => {

console.log(data) });

</script>

</body>

</html>

1. **d3sparql.js Library**

JavaScript library for quiring using SPARQL and visualize resulted JSON in D3.js. The library is licensed under the same license as D3.js's (BSD license) more information <https://github.com/ktym/d3sparql>

*Sankey Diagram*

function exec() {

var endpoint = d3.select("#endpoint").property("value")

var sparql = d3.select("#sparql").property("value")

d3sparql.query(endpoint, sparql, render)

}

function render(json) {

var config = {

// for d3sparql.graph()

"key1": "orig\_org",

"key2": "res\_org",

"label1": "orig\_lab",

"label2": "res\_lab",

// for d3sparql.sankey()

"width": 1200,

"height": 1200,

"margin": 10,

"selector": "#result"

}

d3sparql.sankey(json, config)

}

*Force Graph*

function render(json) {

var config = {

"charge": -500,

"distance": 50,

"width": 1200,

"height": 1200,

"margin": 10,

"selector": "#result"

}

function exec() {

var endpoint = d3.select("#endpoint").property("value")

var sparql = d3.select("#sparql").property("value")

d3sparql.query(endpoint, sparql, render)

}

*d3sparql.forcegraph(json, config)*

1. **Same-origin policy**

Same origin policy of all major browser block accessing AJAX requests across domains.

“Cross-Origin Resource Sharing (CORS) is a mechanism that uses additional HTTP headers to tell a browser to let a web application running at one origin (domain) have permission to access selected resources from a server at a different origin. A web application makes a cross-origin HTTP request when it requests a resource that has a different origin (domain, protocol, and port) than its own origin.”

<https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>

Recommended workaround is to host the database and the website using the same domain but for testing Google Chrome can be opened with

*"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --disable-web-security --user-data-dir="C:/ChromeDevSession"rossSiteDocumentBlockingIfIsolating*

to disable CORS policy.