Cardiff University School of Computer Science and Informatics

Initial Plan: Building the BBC Music Website

CM3203 One Semester Individual Project – 40 credits

Author: Ashley Harris (C1413555)
Project supervisor: Alia Abdelmoty

Project Description

Linked open data has made it easier for developers to build domain-specific web applications which offer enhanced scalability over traditional Web 2.0 applications. By 'mashing up' data from multiple linked sources, these applications can offer large amounts of useful information to their users. The BBC Music site is a good example which uses data from MusicBrainz and DBpedia, adding greater value to their own content.

My goal for this project is to develop a similar application which can automatically query the web for data that can be used to build a website such as BBC Music. The query results will be presented on automatically generated web pages.

Semantic web data offers many advantages, but there can be issues such as identifying data provenance to ascertain the quality of information and how accurately such information is interlinked with other sources. Therefore, before developing the application, I will carry out research into available data sources (such as DBpedia) to identify which sources would best help me fulfil the core functionality of my application.

As the Semantic Web is comprised of RDF triples, I can use SPARQL queries to extract the data I need for my application from the web. However, there are multiple ways of achieving this with different programming languages such as Python and Java, so I will also research the different methods and explain my chosen approaches in my final report.

Once I create the basic application which can successfully retrieve RDF data, I am aiming to create web pages which offer unique features unlike those exhibited by sites such as BBC Music. Creating a user interface which offers lots of functionality and feels unique, without being complicated to understand and difficult to use will be a challenge, but I would like to enhance the system as much as possible in the final stages of the project once all functional requirements have been met.

For instance, embedded YouTube videos for specific songs within the web pages generated would be unique. BBC Music does include videos on their artist pages, but they tend to be specific to the BBC, such as interviews and live performances broadcast by the organisation. Furthermore, the user is only provided with preview snippets of songs on the BBC Music site, not entire tracks. Other sites listing tracks released by an artist have links to Spotify, so that the user can stream tracks in their entirety, but the user will be asked to sign up/log in to Spotify in these instances. Therefore, I feel links to YouTube videos for songs are more appropriate as people will not be required to sign in to anything. I feel inclusion of such links could be the basis of a unique feature for my application.

Furthermore, I would also like to incorporate search functionality into my application so that users can find artists, albums or songs quickly. It will also be desirable to make the web pages of the application dynamic, modern and easy to use, so it is likely that I will be required to use JavaScript and/or PHP in development.

Project Aims and Objectives

Primary (Core system functionality)

- Study different approaches available for building a linked data driven web application and choose a suitable platform for implementation.
- Identify sources of data and build a data collection system to query multiple data sources, integrate and pre-process the results for publishing.
- Design and implement an interface component of the application which allows the filtering and extraction of specific data sets and the presentation of the data in an appropriate format.
- Test and evaluate the system with multiple data sets on the linked data cloud. Produce web pages that are automatically generated based on content the user is looking for.

Secondary (Desirable UI features)

- Incorporate links to external sites in the application which can offer the user more information based on their data search (e.g. song lyric websites when the application returns a list of songs).
- Use visualisation techniques such as maps, charts or graphs to represent data collected by the system.
- Embed YouTube videos into automatically generated web pages which vary depending on the data obtained by the data collection system.
- Facilitate data retrieval by including a search bar in the automatically generated web pages.
- Evaluate the system's usability by collecting feedback from potential users.

Work Plan

Week 1 (05/02/18 – 11/02/18): Initial research

- Investigate how to create a local RDF store.
- Research into libraries that will be suitable for creating the application (examples include: ARC, dotNetRDF, Jena, Sesame, SPARQL Wrapper)
    - Note the advantages and disadvantages of the different methods.
- Decide which programming languages and libraries I will use.
- Research Semantic Indexers and determine whether I need to incorporate one into my system.
- **Milestones:** Suitable platform for implementation established. All methods and libraries to be used will be documented with reasoning for my choices.

Week 2 (12/02/18 – 18/02/18): Further research and application requirements

- Research suitable data sources for my application.
- Identify system requirements, which should be a combination of:
    - Data collection system (Back-end) requirements.
    - User interface web page (Front-end) requirements.
- Meet with supervisor.

- **Milestones:** Select which data sources I will primarily use. Establish and document application requirements.
- **Deliverable:** Application requirements document.

Week 3 (19/02/18 – 25/02/18): Data collection system design

- Design the back-end of the system paying close attention to requirements.
- System requires local RDF store, Logical system (SPARQL) which is controlled by interaction with the UI and a data integration component for fetching the linked data from the web.
- **Milestones:** Complete back-end system design which is feasible and meets functional requirements.
- **Deliverable:** UML Class diagram for application's back end

Week 4 (26/02/18 – 4/03/18): Data collection system implementation

- Begin programming of the back-end application for retrieving RDF data.
- Create a local RDF data store.
- Meet with supervisor.
- **Milestones:** At least have some basic SPARQL queries implemented which retrieve data from the semantic web. Have RDF data store working.

Week 5 (05/03/18 – 11/03/18): User interface design and data collection system development

- Design the front-end user interface for the web pages by creating wireframes.
- Create two wireframe models, one for the most basic implementation (in case there is not time to develop all the additional features outlined in the Secondary Aims), and one including all Secondary Aim features.
- Include search bar, filter controls and chart space where possible on the second model.
- Continue development of the back-end application.
- Meet with supervisor.
- **Milestones:** Have complete wireframes on paper and/or online (Balsamiq).
- **Deliverable:** System design document and interface prototype

Week 6 (12/03/18 – 18/03/18): User interface implementation.

- Compare data collection system in its current state to the system requirements identified at the beginning of the project, prioritising any requirements that have not yet been met.
- Begin development of the front-end web pages of the application.
- First of two review meetings with supervisor.
- **Milestones:** Have the basics of at least one web page completed.

Week 7 (19/03/18 – 25/03/18): Complete implementation of functional requirements.

- Spend time developing any unfinished aspects of both the front- or back-end of the application that are necessary for ensuring functional requirements are met.
- Begin work on fulfilment of non-functional requirements.

- **Milestones:** All essential requirements/functionality of the application should be met by this stage. Complete front-end development.

Week 8 (26/03/18 – 1/04/18): Finish all development work

- Complete work on both data collection system and web pages.
- Meet with supervisor.
- **Milestones:** Complete programming of the back-end application.

Week 9 (02/04/18 – 8/04/18): Testing and introductory report sections

- Begin testing the application, using test cases and recording the results.
- Begin final report writing.
- Meet with supervisor.
- **Milestones:** Produce detailed test cases, complete Introduction and Background sections of report.

Week 10 (09/04/18 – 15/04/18): User feedback and report main sections

- Final report writing.
- Gather feedback from potential users of the application.
- Meet with supervisor.
- **Milestones:** Complete Approach, Implementation, Results and Evaluation sections of report.

Week 11 (16/04/18 – 22/04/18): Feedback analysis and report writing

- Analyse any feedback gathered and document it, creating visual representations such as charts if applicable/possible.
- Second review meeting with supervisor.
- Final report writing.
- **Milestones:** Complete Future Work, Conclusion and Reflection on Learning section.

Week 12 (23/04/18 – 29/04/18): Final report writing

- Final report writing.
- Contingency time.
- Meet with supervisor.
- **Milestones:** Ensure introductory elements (Acknowledgements, Table of Contents/Figures) and concluding elements (Glossary, Appendices, References) of the report are completed.

Week 13 (30/04/18 – 07/05/18): Report completion

- Complete final report.
- Final meeting with supervisor to identify any required amendments to the final report.
- Contingency time.
- **Milestones:** Read through report adding or removing content where necessary.

Week 14 (07/05/18 – 11/05/18): Report review and submission

- Review final report by proof reading it to check correct English has been used and all appendices I intend to submit have been referenced appropriately.
- Submit final report and appendices.
- **Milestones:** Complete project
- **Deliverables:** Final report and appendices such as code listings.

References

Bizer, Christian, Tom Heath, and Tim Berners-Lee. "Linked data-the story so far." *International journal on semantic web and information systems* 5.3 (2009): 1-22. Available at: https://eprints.soton.ac.uk/271285/1/bizer-heath-berners-lee-ijswis-linked-data.pdf

Hausenblas, Michael. "Exploiting linked data to build web applications." *IEEE Internet Computing* 13.4 (2009). Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5167270&isnumber=5167256

Hausenblas, Michael. "Linked data applications." *First Community Draft, DERI* (2009). Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.303.219&rep=rep1&type=pdf