Resource Description Framework (RDF) Models for Representing the Revision Process in Research Support Systems

Harriet Nyanchama OCHARO[a*] & Shinobu HASEGAWA[b]
[a]*School of Information Science, Japan Advanced Institute of Science and Technology, Japan*
[b]*Research Center for Advanced Computing Infrastructure, Japan Advanced Institute of Science and Technology, Japan*
*harriet.ocharo@jaist.ac.jp

Abstract: A general research support system essentially provides support for the daily research activities of students in higher education. One of the important research activities is the revision process that students go through to improve the quality of their research output after they receive feedback from their supervisors or reviewers. This concept paper presents the rationale for using the Resource Description Framework (RDF) to describe the revision process metadata in the form of a universal model that enables integration, interchange of information and standardized communication across various research support tools, processing and storage. With RDF, we can not only query the original metadata to find out the progress of the revision process, but we can also apply inference queries to discover relationships or patterns related to the overall research process. The inference queries will also enable students to reflect on and observe the changes the research drafts go through during revision leading to the final version. Such inferences cannot be realized when using a traditional relation database (RDB) or when using native file applications such as word processors. Using RDF, the logging and history of the revision process during daily research can be obtained for reflection, which is an important part of learning how to carry out research better. The design is currently in the process of being implemented and tested.

Keywords: RDF-based models, inference querying, revision process, research support system

1. Introduction

Students in universities and other institutions of higher education need a system to manage their daily research activities. Daily research activities include logging and tracking of the research activity process step by step, communication with supervisors, managing the research schedule etc. Also, a research support system helps the students with finding the relevant information, choosing the right tools and producing an effective presentation of research results (Yao, 2003). Tracking of research progress would also enable students to complete their research in time (Suhaily, Rozainun, & Azmi, 2015).

A typical research process is commonly divided into phases, such as problem definition, literature review, design, experimentation, testing and evaluation and presentation of results. There are currently many tools available to support various research activities. For example, there are reading tools that support the literature review process, and word processors that support the writing of research articles. However, there is limited research in a common framework that brings together the various sub-systems to support the common goal of research (Yao, 2003). A general goal of our research is to utilize a universal framework to provide research support throughout the entire research process.

One of the challenges when designing such a universal system to support the research activities of students is how to handle the various files produced as research output data to support the revision process in the system (Ocharo & Hasegawa, 2016). Depending on the field of research and the research phase, files or applications of different types may be produced. These files or applications may be unstructured, semi-structured or structured and this might present challenges when developing the system because of the need to write different routines to handle each file or application type.

In summary, there are two problems we encounter when developing a research support system. One is that the research output files are of different types depending on the research field and research phase. Thus it is difficult for a student who is trying to evaluate their overall research progress because of disjointed subsystems/tools that support different phases of research. It is a challenge to obtain information such as tracing a common link from the beginning to the end of research, calculating the overall time taken at each phase in comparison to other phases, total time taken, etc. However, using a common framework for logging research output files' metadata such as the Resource Description Framework, a W3C recommendation, can overcome this challenge.

The second problem is how to improve support for the revision process as there are limitations to the software tools or systems that students use when creating or editing their research output. For example, students would like to find out information such as the number and type of unresolved comments, what common mistakes they might be making, what comments take the longest time to resolve and why, etc. With RDF, we cannot only query the metadata of the output files to obtain this information, but we can also apply inference queries to discover relationships or patterns related to the overall research process. Such inferences cannot be realized using a traditional RDB (relational database) or by using native file applications such as word processors or spreadsheets. This is the originality of our contribution to the field of research about research support systems for students in higher education.

In this paper, we propose using RDF to overcome the challenges of handling file types in a universal framework to support the revision process for all the phases of research. RDF uses a linking structure called a triple to describe two resources and the relationship between them. This linking structure forms a directed, labeled graph, where the edges represent the named link between two resources. We seek to represent the metadata about the various research output files in RDF, while maintaining a link to the original file for reference. This is because file metadata is the main input needed to support the daily research activities of students, which include logging, tracking, scheduling, communication with the supervisor, and most importantly, revising the output file to improve its quality depending on the comments from supervisors. When the student needs to work on the original file, they to follow the URI (Uniform Resource Identifier) to access it with a suitable application. In this way, the research system can support any kind of files produced during research, because the metadata is described using a common XML(Extended Markup Language)-based RDF model which enables integration, interchange of information with other applications and standardized communication, processing and storage. The idea is to create an RDF-XML modeling environment as a service layer that sits between the application logic layer and the data storage layer. This layer will create suitable metadata models in RDF-XML that enables the interchange of information between the application and data storage layers, as well as facilitating exchange of information with any other applications outside of the research support system. This will make it easier to develop and extend applications that support revision during the entire activity research process, and will also make it possible to discover useful patterns using inference queries.

The rest of the paper is structured as follows – section 2 is an overview of the scientific research process and revision process, where we also identify the requirements of a system to support the revision process. Section 3 is a review of the related literature. In section 4, we discuss the rationale for RDB. Section 5 is the conclusion and future work.

2. Research Activity Process and Revision Process

2.1 The Research Activity Process

We base our research activity process on the research process model proposed by (Fankfort-Nachmias Nachmias, 1992). Most scientific research follows a more or less similar process (Lynch, 2013). It starts with the problem definition phase, where the research theme is set. This is followed by a literature review phase, which is in turn followed by the design of experiment or model phase, the development or experimentation phase, the testing and evaluation phase and the presentation of results phase, in that order. We abstract the basic scientific research process in an iterative waterfall model as shown in Figure 1 which represents an instance of research activity that might be typical & sequential but not necessarily the case for all research activities. The waterfall model implies that the researcher can always go back and revise any of the previous phases.

One of the important factors to consider when designing a system to support the daily research activities of students how to handle the various files produced as research output data. At different phases, students use different tools or subsystems to create research output. See Table 1 for examples of the research output from various phases and the corresponding tools used to generate the output by students in the information science research field. Handling each file type individually can be difficult as it will require building support for the many different file types that exist today. Our idea is to obtain the metadata about the files which includes a link to the original file location so that students can still use the native application to manipulate it. The metadata is what will be used to support the daily research activities of students.

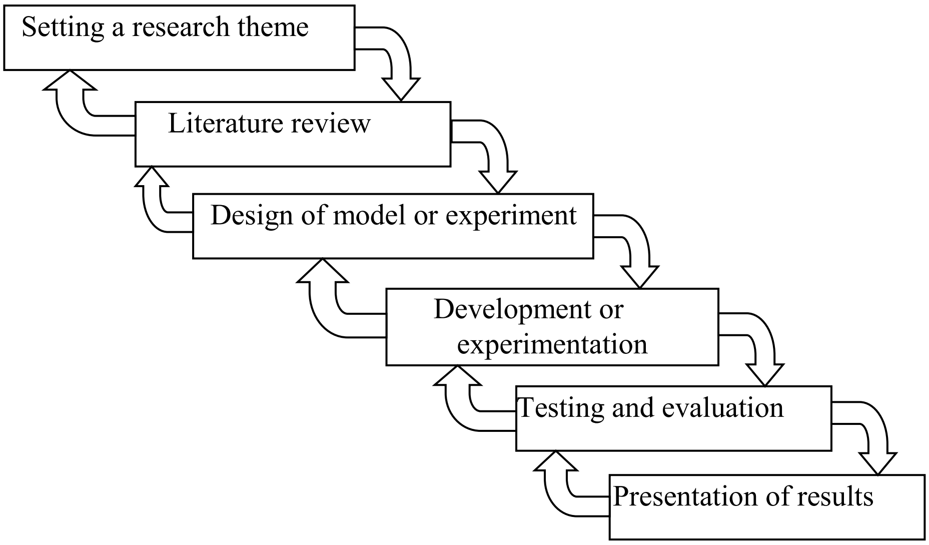

Figure 1. A typical representation of an instance of research activity

Table 1: Research phases and corresponding research output

Research Phase	**Research Output Examples**	**Description of output**	**Research subsystem/Tool**
Setting a research theme	Research proposal	Text	Word processor e.g. Microsoft Word
Survey of related literature	Literature review chapter/doc	Text	Reading and citation manager e.g. Readcube, word processor
Design and modeling of system	System model	Uml diagrams, graphical illustrations	Graphics applications e.g. Microsoft Visio
Development	System prototype	Raw code, executable application, web application	Integrated Development Environments (IDEs) e.g. IDLE for python
Testing and Evaluation	Summarized results, visualizations	Statistics - text, graphs, tables	Statistical software packages e.g. SPSS
Presentation/publication of results	Presentation slides, thesis, journal paper, conference paper, poster	Text, PowerPoint slides	Word processor, Slide software

2.2 The Revision Process

In section 2.1, we have discussed a general research process. However, depending on the field of research, laboratory style etc., there are variations in the actual process. The one aspect that is common to all is that students will produce some kind of research output, for example experiment results or files of code, and that they will receive feedback form their supervisor to improve the quality of this feedback. Hence all students undergo a revision process throughout their research. This especially applies to thesis/paper writing/presentation where students go through several drafts before arriving at a final copy. Presentation of research results is a very important stage that culminates the end of the research process. It is for this reason that our paper focuses on the revision process.

Even during daily research activities, students may take notes or undertake experiments to collect data, which they will eventually use to compile a report or some kind of output for a particular research phase. Once students produce research output, they undergo a revision process to improve the quality of their output after receiving feedback, in the form of comments for example, from their supervisor. Since a student's research is evaluated based on their quality of research output, then the revision process is an important part of the research process.

There may be differences in what constitutes output from each research activity but any revision process can be described a revision model that describes the daily research activity followed by the uploading of an initial draft, feedback from a supervisor and subsequent revision by the student based on this feedback. This revision process repeats itself until the final draft is satisfactory. See Figure 2, which is adapted from Hasegawa & Yamane, (2011).

We identified the followings self-describing metadata that we consider to be useful supporting the revision process: file author, file identifier, date created, due date, draft version, title, research phase, file description, file location (uniform resource identifier), and comments from the supervisor. The comments in turn contain more information such as comment identifier, comment author, date, due date, comment range, comment text, comment type and comment status.

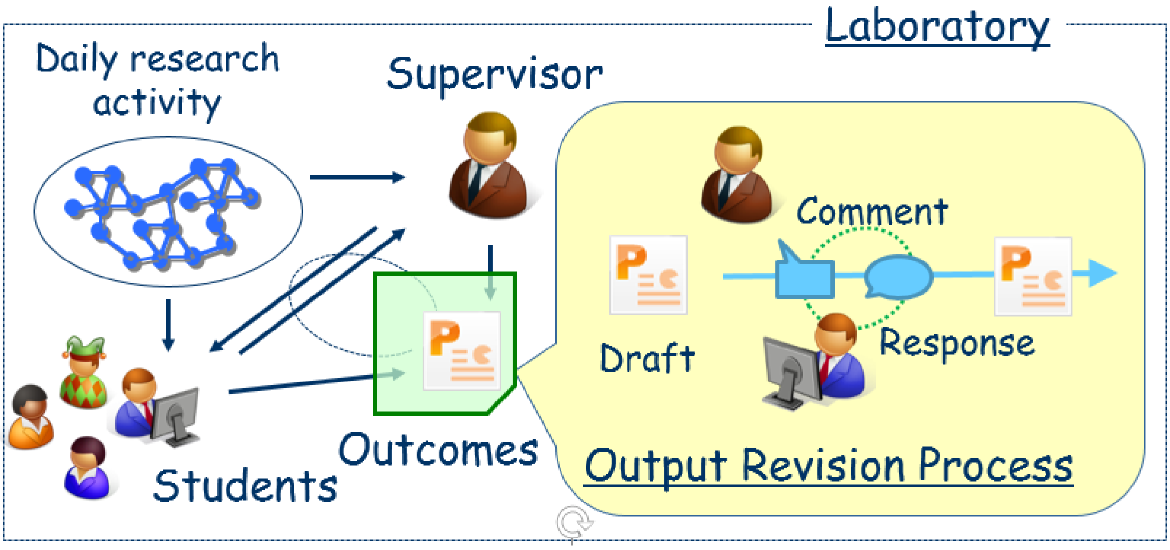

Figure 2. The research output revision process

2.3 Requirements for Supporting the Revision Process

In order to provide logging, tracking, and support for the revision process throughout the research process, we need to integrate, store and manage data from all the different research subsystems. Figure 3 shows the interoperability needs with other research subsystems such as word processors, presentation tools, or data analytic tools. We have identified the requirements below to support the revision process:

- Ability to handle metadata from different file types while maintaining a link to the original file so that students can edit it in the native application

- Ability to handle unstructured, semi-structured or structured data from the many different subsystems as it is not possible to anticipate all file types
- Ability to query information about the revision process including the duration of revision, the due date of the research output, the number of unresolved comments, the author, total number of comments etc.
- Ability to perform inferential queries to obtain information such as what could be making the duration of the revision process too long, what could be the most difficult comments to resolve etc.
- Identification of common patterns emerging during research
- Knowledge transfer to new students in a laboratory in the form of a checklist of the most common comments

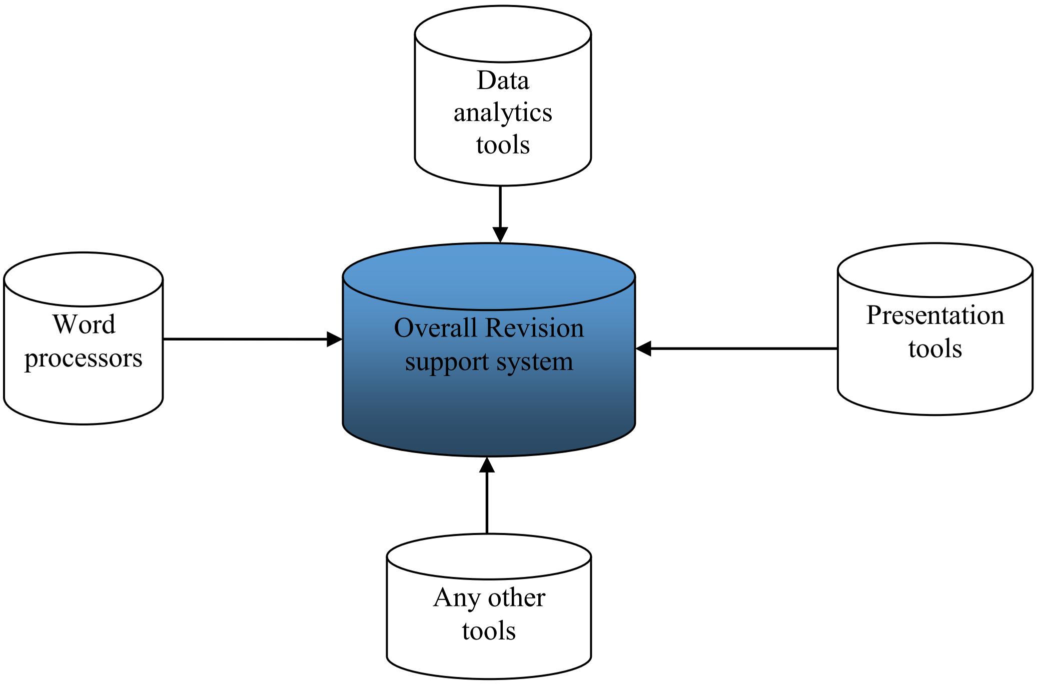

Figure 3. Interoperability with other research support subsystems

3. Related Literature

In the semantic web, information has well-defined meaning (Berners-Lee, Hendler, & Lassila, 2001). In other words, it can be thought of as a paradigm shift from a document-based web to a web of interlinked data (Open Data Support, 2014), where the data has meaning and relationships and is not just plain text. The semantic web is about enabling access to this data, by making it available in machine-readable formats and connecting it using URIs. The Resource Description Framework (RDF) is a W3C recommendation for representing data and resources on the web. In RDF, a resource is represented as a triple of a subject, predicate and object (W3Org, 2014). The subject is a URI identifying the resource, the object is another resource that the subject is related to, and the predicate describes the kind of relationship between the subject and object (Heath & Bizer, 2011).

There are several advantages of RDF discussed in the various literature. One important aspect of RDF is that the RDF data model is designed to enable integration of information that originates from multiple sources (Bergman, 2009). Information from different sources can be easily combined into a single graph. RDF data models allow for the representation of tightly structured data as well semi-structured data. It also enables the setting of RDF links between data from different sources (W3Org, 2014), which allows client applications to navigate between data sources and to discover additional data (Bizer, Heath, & Berners-Lee, 2009). Another advantage of using RDF models is the ability to support inference queries. Inference queries allow us to generate or discover new relationships based on existing ones, such as is the case with logic programs. We can query RDF models using SPARQL, a query language for RDF. SPARQL can be used to express queries across diverse data sources (The SPARQL Working Group, 2013). SPARQL query results can be data sets or RDF graphs, which enable us to visualize the relationships in the data models.

One thing we have to consider when using RDF models to represent data in our research support system is linked open data. Open data is publicly available under an open license to be freely used, reused and redistributed (Open Data Support, 2014). Our aim is to develop a web system based on RDF-models and enventually make a contribution to the open linked data sets when the data models mature and are stable. As Bizer, Heath, & Berners-Lee (2009) state in their paper, the challenge now is for researchers and developers to create domain-specific applications that exploit the potential of linked (open) data. We have identified a research gap in RDF specifications for research support tools or systems. As Yao (2003) note in their research, there is a lack of study of such research support systems in a common framework. There are various research tools out there to support the various phases of research, so integrating all the data from the different tools into one overall system is a challenge that can be overcome by RDF-based models.

4. Why RDF and not RDB (Relational Database)?

There are several metadata formats available for describing data in web systems, including plain XML, HTML (Hypertext Markup Language) etc. The most common way to store data and metadata for general web applications, including research support tools, is in relational databases (RDB) and that is why in this section we focus on the discussion of the advantages of RDF over RDB. However, the advantages we discuss can be extended to cover the advantages RDF over all other metadata formats, particularly the ability to do semantic searches and inference queries.

We illustrate with an example of a student who is working on phase 1 of the research (see Table 1) where the expected output is a research proposal, typically a text file. The student may be using a word processor such as Microsoft Word to create and revise the document. However, when the student is revising, he or she cannot query Microsoft Word to obtain an overview, at a glance, of information such as:

- The total number of comments, the resolved comments, the open comments, persistent comments in subsequent drafts etc.
- The total time spent on revision so far, because there is no direct link between two distinct files that are separate drafts of the same document
- Commonly occurring comments in research proposals of other students so the student can know which common mistakes to look out for
- Other metadata about the research phase such as due date, document and comment authors, etc.

Our system's aim is not to take over the functions of Microsoft Word but to act as a complementary tool in the revision process by utilizing research output file metadata. We could simply put the metadata in a relational database and query that. However, there are several reasons why we should consider an RDF-based application over a relational database (RDB) to fulfill the requirements of the revision system as discussed in section 2.3:

- RDF can handle metadata from any file type as it describes resources (research output) with the corresponding properties and property values. It can therefore handle metadata from different types of files whether structured or not. With RDBs, input data is always constrained. As it is not possible to anticipate all the different file types, marking up the data as RDF models is better.
- Interfacing or interchange of information with research support subsystems – for example if we require specific data in the research support system, we can automatically obtain information such as author, date etc. from some of the subsystems that have structured metadata. For instance, Microsoft Office Open XML (Ngo, E. C. M. A., TC45, 2008) is a zipped, XML-based file format developed by Microsoft for representing spreadsheets, charts, presentations and

word processing documents. With RDF-XML, it is easy to map these XML representations into RDF-based models. RDF-XML is a W3C standardized serialization format for RDF.

- When it comes to XML-RDF data models that facilitate interoperability, there are many open source and free tools to convert the models into various formats such as JSON, pdf, excel, csv, pdf etc. So it is possible to still have a conventional relational database to store the data or the data can be stored in a file-based format, and then mark-up that data into RDF models.
- Another reason is that nesting can be problematic for RDB especially when it comes to nested comments (feedback on research output). RDF is suitable for describing tree-like structures which are more intuitive for nested comments.
- With RDF, it is possible to use inference queries to identify common comments and similar patterns during research. Such inference queries are not possible with traditional RDB models. If we obtain RDF files from other students doing similar research, we can build an archive of knowledge and this can enable knowledge transfer to new students in a laboratory, for example in the form of a checklist of the most common comments for students to pre-check their research output.

Using our system, the new revision process can described as below (see Figure 4).

- The student will upload the document into the system, where the web application passes this document to the RDF modeling environment that maps the document metadata into RDF-XML model.
- The resulting file is then saved in the backend storage (can be any suitable storage system, the developer is free to choose). The data stored will be used for search and retrieval.
- The scheduler is then updated with the due date, and a notification is sent to the supervisor to review the uploaded document. The supervisor is also provided with a link to the location of the uploaded file.
- The supervisor can either directly add the comments to the word file, assuming it is a word file, or write their comments into the system.
- Once the supervisor has written the comments, the RDF model of the document metadata is updated as a 2nd version of the document, and a notification is sent to the student to improve the document based on the comments.
- Once the student revises their output and uploads the second version of the comment, the supervisor checks it and sets the status of the comments that have been adequately addressed as "closed".
- If there are any new or unaddressed comments, the student is notified of them. This goes on until the statuses of all the comments are set to close, at which point the document is archived, along with the previous versions which will enable the student to reflect and learn from the revision process.

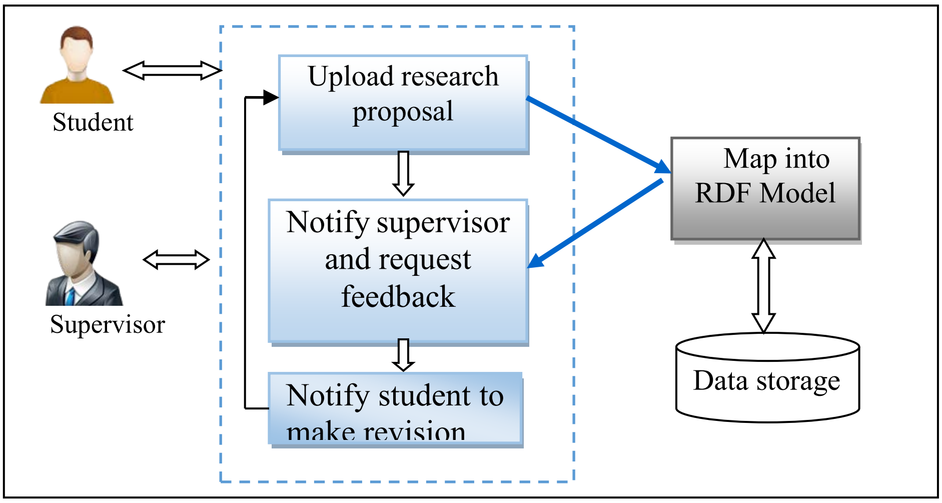

Figure 4. Example of a Revision Process

5. Conclusion and Future Work

In this concept paper, we presented the case for using RDF-based models to describe the metadata of research output files that can make it easier to develop and extend applications to support the daily research activities of students in higher education, especially the revision process that students go through to improve the quality of their output. RDF models can enable integration and exchange of information with the other existing subsystems that support various phases of research in a linked open data environment. Using RDF models enables us to collect all metadata from all different research phases, and using this data we can run inference queries to reflect and learn about the overall research process. We recommend using RDF over other metadata formats particularly because it is possible to do semantic searches and inference queries. Since research is a core activity of institutions of higher education, the importance of a research support system that aggregates data from various research in a common framework cannot be overestimated.

We are currently in the process of implementing and testing the usability and efficiency of the system. Future work will involve analyzing the research output metadata we collect using inference queries to discover common patterns in the research process, and sharing the lessons learned from inference querying. We eventually hope to make a contribution to the open linked data sets when the data models mature and are stable, thereby creating an open gateway for other developers or researchers to extend the system.

Acknowledgements

The work is supported in part by Grant-in-Aid for Scientific Research (C) (No. 17K00479) from the Ministry of Education, Science, and Culture of Japan.

References

Bergman, M. (2009, April 22). *Advantages and myths of rdf.* Retrieved May 2017, from Michael Bargman: http://www.mkbergman.com/

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American 284(5)*, 28-37.

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked Data - The Story so Far. *International Journal on Semantic Web and Information Systems*, 205-227.

Chatfield, D. C., Harrison, T. P., & Hayya, J. C. (2004). XML-Based Supply Chain Simulation Modeling. *Proceedings of the 2004 Winter Simulation Conference. Vol 2*, pp. 1485-1493. Washington DC: IEEE.

Dong, J., Du, H. S., Lai, K. K., & Wang, S. (2004). XML-Based Decision Support Systems: Case Study for Portfolio Selection. *International Journal of Information Technology & Decision Making, 3*(4), 651-662.

Fankfort-Nachmias, C., & Nachmias, D. (1992). *Research Methods in the Social Sciences, Fourth Edition.* New York: St. Martin's.

Hasegawa, S., & Yamane, K. (2011). An Article/Presentation Revising Support System for Transferring Laboratory Knowledge. *19th International Conference on Computers in Education* (pp. 247-254). Chiang Mai, Thailand: Asia-Pacific Society for Computers in Education.

Heath, T., & Bizer, C. (2011). *Linked Data: Evolving the Web into a Global Data Space,* (Vol. Synthesis Lectures on the Semantic Web: Theory and Technology). Morgan & Claypo. Retrieved May 2017, from http://linkeddatabook.com/editions/1.0/#htoc16

Lynch, S. M. (2013). *Using Statistics in Social Research: A Concise Approach.* New York: Springer.

Ngo, T., E. C. M. A., & TC45. (2008). *Office Open XML Overview.*

Ocharo, H. N., & Hasegawa, S. (2016). An Adaptive Research Support System for Students in Higher Education: Beyond Logging and Tracking. *Human Interface and the Management of Information: Applications and Services, Lecture Notes in Computer Science. Volume 9735*, pp. 178-186. Toronto: Springer.

Ocharo, H. N., Hasegawa, S., & Shirai, K. (2017). Topic-based Revision Tool to Support Academic Writing Skill for Research Students. *Proceedings of The Tenth International Conference* (pp. 102-107). Nice: ThinkMind.

Open Data Support. (2014). *Introduction to Linked Data.* Retrieved May 17, 2017, from European Data Portal: https://www.europeandataportal.eu/sites/default/files/d2.1.2_training_module_1.2_introduction_to_linked_data_en_edp.pdf

Suhaily , H., Rozainun, A. A., & Azmi, A. H. (2015). Postgraduate Tracking System: Student Research Progress Tracking Tool. *International Research in Education, Vol. 3*(No. 1), 47-53.

The SPARQL Working Group. (2013, March 26). *SPARQL Query Language for RDF*. Retrieved May 2017, from w3.org: https://www.w3.org/TR/rdf-sparql-query/

Tolk, A. (2004). XML Mediation Services Utilizing Model Based Data Management. *Proceedings of the 2004 Winter Simulation Conference* (pp. 1476-1484). Washington DC: IEEE.

W3Org. (2014, February 25). *Resource Description Framework Overview.* Retrieved May 17, 2017, from w3org: https://www.w3.org/RDF/

Yao, Y. Y. (2003). A framework for Web-based research support systems. *Computer Software and Applications Conference* (pp. 601-606). Dallas, Texas, US: IEEE.