ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 9 , September 2016

Creating a Software Requirements Specification Document Using an Ontology Based Methodology

Robert A. Elliott Sr. Ph.D, Edward B. Allen Ph.D.

.

Assistant Professor, Department of Computer Information Systems, Southern University at New Orleans, USA
Associate Professor(retired), Department of Computer Science& Engineering, Mississippi State University, USA

ABSTRACT: Software requirements engineering typically includes a variety of manual activities.Manual activities are vulnerable to causing errors. Our long range goal is to provide waysto mitigate the likelihood of such errors, especially during requirements elicitation, verification,and documentation. This paper presents a methodology with automated supportfor producing a software requirements specification that conforms to IEEE Standard 830-1998, *Recommended Practice for Software Requirements Specifications*. The methodologyis also includes requirements engineering data elements recommended in the *Software EngineeringBody of Knowledge* (SWEBOK). The methodology is presented as seven usecases and an ontological framework. This paper also presents three empirical retrospective case studies that demonstrated the practicality of the methodology. The case studies alsodemonstrated that the ontology is readily customized for various application domains. Weconclude that ontological support is a promising way to enhance the processes that producea software requirements specification.

KEYWORDS: ontology, requirements engineering, SWEBOK, requirements specification

I. INTRODUCTION

Requirements engineering includes many manual activities, such as information sharing, creation of a software requirements specification document, and provenance tracking. An (SRS) is a formal document completely describing the required software solution in stake-holder terms. Requirements engineering is a process for obtaining, defining, analysing and verifying what is required of a proposed software system. Manual processes are often slow and vulnerable to errors that are costly to find and correct [16]. Standards are available that guide these activities. The Institute of Electrical and Electronics Engineers (IEEE) Standard 830-1998 Recommended Practice for Software Requirements recommendations [9], provides guidelines for the format and creation of the SRS. The Software Engineering Body of Knowledge (SWEBOK) [10] contains generally accepted guidelines about software engineering processes and methods, including recommended requirements engineering data elements. However, in practice IEEE Standard 830-1998 recommendations and SWEBOK guidelines are often neglected by practitioners [1].

This research was motivated by the lack of automated assistance for software requirements engineering processes to facilitate information gathering, document creation, and provenance tracking. This paper proposes a methodology with ontological support for producing a software requirements specification document that conforms to the IEEE Standard 830-1998 recommendations and the SWEBOK Software Requirements Knowledge Areas.The proposed methodology includes an associated ontology with a reasoner application to improve accuracy in information sharing and to provide provenance tracking capabilities [17]. Information sharing improves productivity and efficiency. Provenance tracking of requirements and underlying documents is necessary for managing the evolution of requirements,and is also neglected in practice [2].

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 9 , September 2016

This research used the following software components: Protégé 3.4[1],OWL/RDF[2], XSL/XSLT[3],Jess 7.1[4], SWRL[5], and Pellet Reasoner 1.5.2[6]. A general framework of the research in this paper was previously entered as a poster presentation at the 24th Annual Consortium for Computing Sciences in Colleges, South Central Conference [5] and is based on research presented in Elliott's dissertation [4]. A more detailed paper was also published in the Proceedings: 24th Annual Consortium for Computing Sciences in CollegesSoutheastern Conference [6].

II. RELATED WORK

Researchers have used ontologies to support a variety of goals in requirements engineering. Sardinha et al. [18] used an ontology based tool to help identify conflicting requirement during the elicitation stage of requirements engineering. The elicited requirements may come from existing requirements that are well-defined or from hazy, unclear, and even conflicting requirements from many sources. Denaux et al. [3] describe an approach using an ontology to discover requirements, to describe users, and to record other relevant data during requirements elicitation. Lee et al. [12] also describe a methodology for elicitation and analysis. Zhang [21] advocates using an interactive application while employing an ontology-based system to elicit requirements. Miura et al. [15] propose building specification documents solely from elicitation meetings in the development process. Mala et al. [14] propose using software requirements specifications as input to a methodology that would build an application domain ontology for later use.
In summary, ontologies are proposed in the requirements engineering literature to aid in requirements elicitation [3], requirement analysis [12], and analysis of software complexity [19], requirements analysis [13], requirements specification [7], requirements conflict resolution [8], stakeholder description [20], and problem domain definition [11]. Each of the cited works above focuses on one or a few aspects of the requirements engineering process. None include SWEBOK [10] attributes, tracking of provenance, use of general automated reasoning capability, or automated production of a software requirements document. This paper proposes a comprehensive requirements engineering process that incorporates these
aspects.

III. PROPOSED METHODOLOGY

This section presents a methodology for the requirements engineering process as depictedin Figure 1. This ethodology is depicted as the following uses cases that practitionerscan perform specific activities during requirements engineering.

- Requirements Elicitation Use Case— The requirements engineer will capture softwarerequirements in a raw, informal format and store stakeholder requirements within an ontology.
- Requirements Analysis Use Case— The requirements engineer will (a) take theelicited requirements and employ methods to formally analyze the requirements,(b) update and categorize the requirements as user and system requirements, and (c)update the ontology accordingly.

[1]http://protege.stanford.edu/
[2]http://www.w3.org/2001/sw/wiki/OWL
[3]http://www.w3.org/Style/XSL/
[4]http://herzberg.ca.sandia.gov/
[5]http://www.w3.org/Submission/SWRL/
[6]http://clarkparsia.com/pellet/

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 9 , September 2016

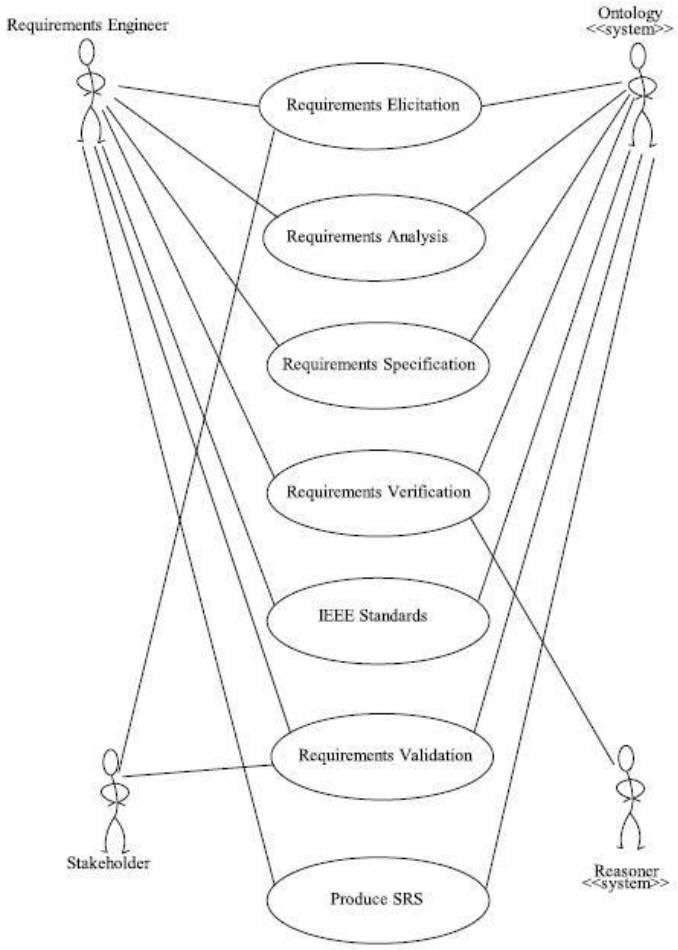


Fig. 1 An Ontology based Requirements Engineering Approach

- IEEE Standards Use Case— The requirements engineer will populate the IEEE Standard830-1998 and SWEBOK data items in the ontology with domain specific information.
- Requirements Validation Use Case— The requirements engineer will run scripts to produce requirement reports which will be checked and approved by stakeholders. The requirements engineer will validate the requirements in the ontology to reflect the approved stakeholder documents.
- Produce SRS Use Case— The requirements engineer will use scripts to extract requirements from the ontology and produce the software requirements document.

In our case studies below, the Requirements Verification use case takes advantage of the Jess and Pellet rules engines which were part of the the Protégé ontology IDE. These engines provide for real time reports and data creation in the requirements engineering process. Data files can be created outside of the ontology and requirements can be updated within the ontology via either Jess or Pellet.

All requirement data is kept in the ontology. Figure 2 shows the data flow, of our reference implementation, SRSOntology, within the ontology and the scripts used for reports and ontology class population. The left column of rectangles in Figure 2 consists of ontology classes. The right column of rectangles consists of reports that are automatically generated by scripts (arrows). Figure 2 also shows the relationship of the use cases to data flow within the ontology.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 9 , September 2016

When the Requirements Elicitation use case is executed, it populates the Elicited Requirements class as depicted in Figure 2. After the Requirements Analysis use case is executed, it populates the Analyzed Requirements class. The execution of each use case populates the ontology with data.

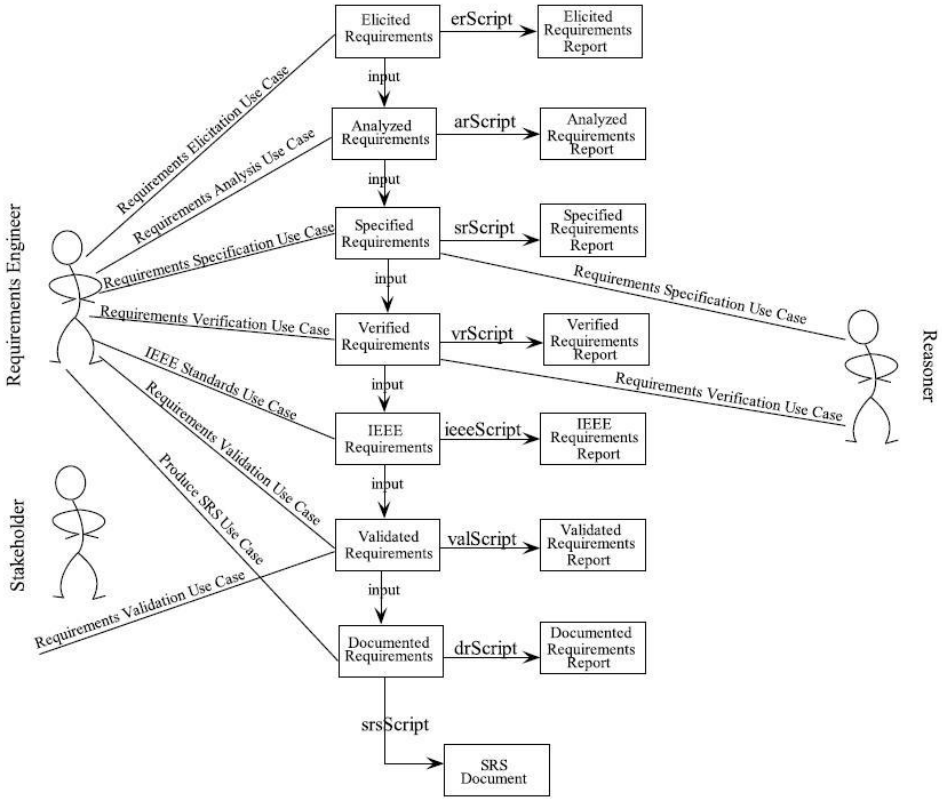


Fig. 2 SRSOntology Data Flow

IV. ONTOLOGY

The baseline Software Requirements Ontology (SRSOntology) contains classes and relationships that are modeled from the SWEBOK . In contrast to prior research, SWEBOK recommended data attributes are included in the ontology. Goals, domain knowledge, stakeholders, operational environment, organizational environment, requirements classification
and conceptual model data are directly stored in the ontology. A detailed list of all SWEBOK recommended data can be found in Appendix A of Elliott's dissertation [4].

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 9 , September 2016

Range Class	Relationship	Domain Class
Stakeholder	hasStakeholderIDNumber	StakeHolderID
RequirementsEngineer	hasRequirementEngineerIDNumber	RequirementsEngineerID
ElicitedRequirement	hasRequirementsElicitationIDNumber	RequirementsElicitationID
AnalyzedRequirement	hasRequirementsAnalysisIDNumber	RequirementsAnalysisID
SpecifiedRequirement	hasSoftwareRequirementIDNumber	SoftwareRequirementID
ElicitedRequirement	hasStakeholder	Stakeholder
ElicitedRequirement	hasRequirementsEngineer	RequirementsEngineer
AnalyzedRequirement	isAnalyzedBy	ElicitedRequirement
SpecifiedRequirement	hasBeenAnalyzedBy	AnalyzedRequirement
IEEEStandard830	hasSpecifiedRequirement	SpecifiedRequirement
ValidatedRequirement	hasIEEEStandards	IEEEStandard830
DocumentedRequirement	hasValidation	ValidatedRequirements
		DocumentedRequirement

Table 1 SRSOntology Major Relationships

Provenance tracking was facilitated by the ontology. All requirements engineering activities can be traced back to any stakeholder or any requirements engineer, by date, time and project. The attributes: StakeholderIDNumber, RequirementsEngineerIDidentified participants. The following classes were used to support provenance tracking:

Stakeholder, RequirementsEngineer , ElicitedRequirementID, and ElicitedRequirement.

The ElicitedRequirements class is the component which is used to manage softwarerequirementselicitation. The ElicitedRequirements class members are under ontology constraints that force each member of the class to be in class relationships with a stakeholder, a requirements engineer and a unique requirements elicitation number. These constraints provide for provenance of requirements. Requirement extraction and maintenance can be performed referencing stakeholder, requirements engineer, or requirements ID number.

The AnalyzedRequirements class sets bounds on the software to resolve software requirement conflicts, to describe the problem, and to elaborate system requirements [10,p. 6]. All user requirements must be transformed from high-level desires to formal system definitions. The analysis phase of requirements engineering serves that purpose. This class is also used to contain concept models or use cases which describe user requirements and detailed information about the requirement.

The SpecifiedRequirements class allows for the declaration of a formal document or electronic equivalent of the software requirements. These requirements now contain more specific details that have been included after the elicitation and analysis processes. They are now in a form that can be used by software developers to design and code the software system [10, p.8].

The IEEEStandard830 1998 class allows data items specifically recommended by the IEEE Standard 830-1998 to be included in the ontology. This is a way to systematically include IEEE Standard 830-1998 data items into the requirements-engineering process.

The ValidatedRequirements class keeps track of requirements that have met stakeholdersrequirement needs. Requirements engineers, directed by stakeholders, can individually select the requirements which stakeholders feel meet their needs. The requirement engineers can then note this validation of requirements using the ontology.

The DocumentedRequirements classes contain data that populate the three sections of the SRS, namely, the Introduction, Overall Description, and the Specific Requirements Sections.

The classes in the Range Class column of Table 1 are all restricted classes within the ontology and those restrictions are enforced for each class by the relationships listed in the Relationship column.

V. CASE STUDY—FLOODVIZ

This retrospective case study focused on a scientific application that aided scientists in processing hydrology data. The research applied software requirements engineering principles as it related to the proposed methodology. The ultimate goal of the case study was to produce section 3.1 of the software requirements document which contained the specific requirements in full detail.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

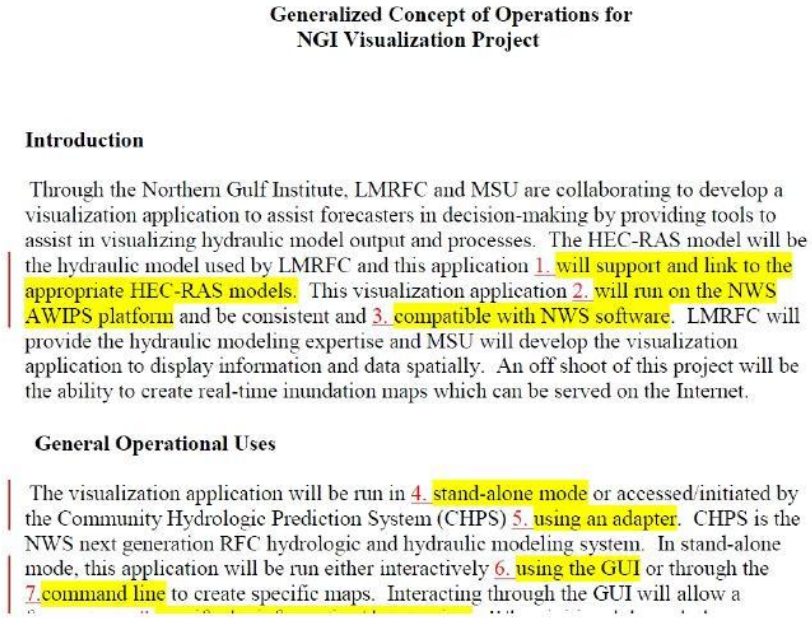
Vol. 3, Issue 9 , September 2016

A. Research Method

This case study performed the six use cases described in Section 3 and executed the scripts which generated the files and reports listed in Figure 2. The baseline ontology in this case study was an empty SRSOntology described in Section 4, which was used to create a new ontology to model and capture the FloodViz project requirements. This new ontology, named FloodViz, specified the stakeholders and requirements engineers, and elicited, analyzed and specified requirements. Major data descriptions were taken from the SWEBOK Software Requirements Knowledge Area and were modeled into the ontology. These descriptions were modeled as classes and the classes contained much of the elicited requirement information from the two source documents shown in Figure 3 and Figure 4 as snippets.
Figure 3 shows a snippet of the concepts document and Figure 4 shows a snippet of the requirements traceability matrix(RTM). The yellow highlights in the Conops document were numbered and each highlighted entry was created as a requirement in the ontology. The yellow highlights in the RTM were created as data items in the ontology. The methodology allows for repetitive execution of the activities within the ontology as necessary and the execution of the scripts listed in Figure 2 were run when appropriate. The following are the activities that were done in this case study.
The input source documents were received from the stakeholders. The Prot´eg´e IDE was started and the requirements were entered into the FloodViz ontology. Each use case was executed and scripts were run to produce requirement reports. Details of those items and scripts are documented in Appendix A of Elliott's dissertation [4]. After the last use case was executed, the srsScriptwas run to produce the SRS document.

B.The SWRL and Jess Tools: Automated Data Creation

The Semantic Web Rule Language (SWRL) is a language for the Semantic Web that expresses
rules and logic using the Ontology Web Language (OWL). Java Expert Shell System (Jess) is a rule engine and scripting environment written in Java.[7]
SWRL rules can be used to infer new knowledge from existing OWL knowledge basesand the SWRL specification does not restrict how reasoning against ontologies is performed.SWRL rules reason about OWL individuals by using OWL classes and propertiesdefined within OWL ontologies. We used SWRL to create new facts within the ontologyas needed. This automation of data creation improved the requirement elicitation process since new requirements were created with software.



Generalized Concept of Operations for NGI Visualization Project

Introduction

Through the Northern Gulf Institute, LMRFC and MSU are collaborating to develop a visualization application to assist forecasters in decision-making by providing tools to assist in visualizing hydraulic model output and processes. The HEC-RAS model will be the hydraulic model used by LMRFC and this application 1. will support and link to the appropriate HEC-RAS models. This visualization application 2. will run on the NWS AWIPS platform and be consistent and 3. compatible with NWS software. LMRFC will provide the hydraulic modeling expertise and MSU will develop the visualization application to display information and data spatially. An off shoot of this project will be the ability to create real-time inundation maps which can be served on the Internet.

General Operational Uses

The visualization application will be run in 4. stand-alone mode or accessed/initiated by the Community Hydrologic Prediction System (CHPS) 5. using an adapter. CHPS is the NWS next generation RFC hydrologic and hydraulic modeling system. In stand-alone mode, this application will be run either interactively 6. using the GUI or through the 7.command line to create specific maps. Interacting through the GUI will allow a

Fig. 3 Concepts of Operations Document

[7]http://www.jessrules.com/

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 9 , September 2016

The following is an example of a Jess script which checks a requirement variable to make sure the data value of that variable is in a required range.

```
(defrulerangeCheck
"This rule will check numeric requirements for valid numeric
ranges"
(bind ?requirement numeric)
(bind ?lowRange http://pluto.cse.msstate.edu/rae20/SRSontology#RequirementsElicitationLowRange)
(bind ?highRange http://pluto.cse.msstate.edu/rae20/SRSontology#RequirementsElicitationHIGHRange)
(object (http://pluto.cse.msstate.edu/rae20/SRSontology#RequirementsElicitationDataType = = ?requirement
AND
http://pluto.cse.msstate.edu/rae20/SRSontology#RequirementsElicitationRequireRange = = "YES"
AND
((http://pluto.cse.msstate.edu/rae20/SRSontology#RequirementsElicitationDataValue < ?lowRange)
OR
http://pluto.cse.msstate.edu/rae20/SRSontology#RequirementsElicitationDataValue > ?highRange)))
=>(printout t "Alert: A numeric value is out of RANGE at requirement number ElicitedRequirementID" crlf))
```

In this research, Jess was used to infer knowledge about requirement individuals in the SRSOntology based on SWRL rules. This ability to automatically and correctly add data to an ontology may improve requirements maintenance over strictly human interaction. There were several different Jess rules defined and executed within seconds. These rules were defined to execute when certain requirement conditions were met and entered into the ontology.

C.Results

The FloodViz case study provided evidence that an ontology could be modeled to a specific problem domain and could be used in the requirements engineering process. The ontology framework was modeled to accept requirements, provided for maintenance of the requirements, and for requirements reporting. A concepts of operation document and a requirements traceability matrix were used for requirements elicitation and the ontology facilitated storage of these documents made available to requirements engineers.The requirements were also used as input to rules engines that provided real time status feedback of requirements to requirements engineers. A primary purpose of this case study was to produce the most important part of a software requirements document, Section 3.1. The research methodology was run to completion and Section 3.1 of the software requirements document that adhered to IEEE Standard 830-1998 was produced.The researchers met with Dr. Phil Amburn, of the Geosystems Research Institute (GRI) at Mississippi State, to provide the results of the research. Dr. Amburn reviewed the document and was satisfied with the requirements that were captured and the printed results.

The following is a snippet of the SRS document that was produced.

Software Requirements Specifications
3. Specific Requirements
3.1 Features
Specified Requirement = #SoftwareRequirement_1
3.1.1 IEEE_EI_Name = Graphical User Interface
3.1.2 IEEE_EI_Purpose = create a graphical user interface module
3.1.3 IEEE_EI_IO = Interface to Application
3.1.4 IEEE_EI_Range = N/A
3.1.5 IEEE_EI_Units = N/A
3.1.6 IEEE_EI_Timing = N/A
3.1.7 IEEE_EI_Relationship = Input module
3.1.8 IEEE_EI_ScreenFormat = standard 16" monitor
3.1.9 IEEE_EI_WindowFormat = top left corner
3.1.10 IEEE_EI_DataFormat = character and decimal input
3.1.11 IEEE_EI_CommandFormat = N/A
3.1.12 IEEE_EI_Message = this GUI is the main application GUI
...

VI. CASE STUDY—FACULTY TRAVEL

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 9 , September 2016

This retrospective case study performed the six use cases described in Section 3 and executed the scripts which generated the files and reports listed in Figure 2. The case study examined a business application that provided administrators and system users the ability to manage financial resources allocated for travel expenses. The system under study software interfaced with several other systems in an academic environment. The case study was focused on creating the entire software requirements document but only populating Section 3 of the requirements document.

	A	B	C	D	E	F	G
1		REQUIREMENTS TRACEABILITY MATRIX			REQUIREMENTS TRACEABILITY MATRIX		
2	Project Name: Visual Analytics for Assessment and Interpretation of Simulated Rive				Project Name: Visual Analytics for Assessment and Interpretation		
3							
4	ID	Requirement Description	Status	Software Module(s)	Test Description	Test Results	Comments
5							
6		Prototype Research Software					We will create prototype software, not commercial quality software. Create testbed for experimentation and investigation. Testing will be key to quality
7							
8		User Interface Requirements					Tailor features to users needs. Prototype in cross-platform GUI builder. Routinely subject to change.
9	UI 1	Easy to use					
10	UI 1.1	Point and click interface					
11	UI 1.2	Drop down menus					Possibly use HEC-RAS GUI features to allow some transfer of experience
12	UI 1.3	Match workflow					
13	UI 1.4	Effective use of icons, keyboard shortcuts					
		Coded to allow this GUI to be run as a					

Fig. 4 Requirements Traceability Matrix

A. Research Method

The baseline ontology used in this case study was the previously created FloodViz Ontology. The individuals from the FloodViz Ontology were deleted and script headings were changed to reflect the new ontology. This modification phase was a requirement's engineer manual task and it involved new ontology design and old ontology deletion. Deletion of
the old ontology data took approximately 1.5 hours while new ontology design took fortyeight hours. Modification time and ontology design are factors that must be considered in our research methodology. This new modified ontology was named FacultyTravel and the following figures below: Figure 5, Figure 6, and Figure 7 were the source documents used
to populate the ontology.The use cases were executed and the SRS document was produced.

B. Results

A primary purpose of the Faculty Travel case study was to improve upon the results from the previous FloodViz case study results by producing a full software requirements document framework. All sections of the document were produced. The software requirements document contained placeholders for requirements for all sections of the software requirements document, including Section 1 Introduction, Section 2 Overall Description and Sections 3.1 through 3.7 which presented detailed requirements.
The software requirements document adhered to IEEE Standard 830-1998. Section 1 and Section 2 were not populated with data in this case study but placeholders for the actual data. This case study improved upon the results from the FloodViz Application by producing all of the required sections of the software requirements document and confirmed that a complete software requirements document could be produced by our research methodology.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 9 , September 2016

NetID _____ Name _____ Title _____

Request for:

Academic Department [^] in College of _____

University Level Center [^] _____

(HR query to only show Centers. Home Org is 19%)

Funding Program Appliying for: http://www.research.msstate.edu/funding

- Quick Grant Program (Includes travel to visit sponsors)
- Faculty Travel Program

Purpose: _____

Destination: _____

Dates of Travel (DD-MON-YY): _____ TO _____

Rationale: _____

_____

(Limit size of entry to 250 characters ?)

Amount of funding requested:

Conference	$_____	
Hotel	$_____	
Registration	$_____	
Rental Car	$_____	
Airfare	$_____	
Other	$_____	Please Specify _____
Total	$_____	

Fig. 5 Faculty Travel and Quick Grants Document

Amount of Funding to be provided by:

If Quick Grant Program
(radio button selected above, show these options)

ORED	$_____	
Other	$_____	Please Specify _____
Total	$_____	

If Faculty Travel Program AND person is an academic department employee
(radio button selected above, show these options)

Department	$_____	
Dean's Office	$_____	
Univ Level Center	$_____	Center _____
ORED	$_____	
Other	$_____	Please Specify_____
Total	$_____	

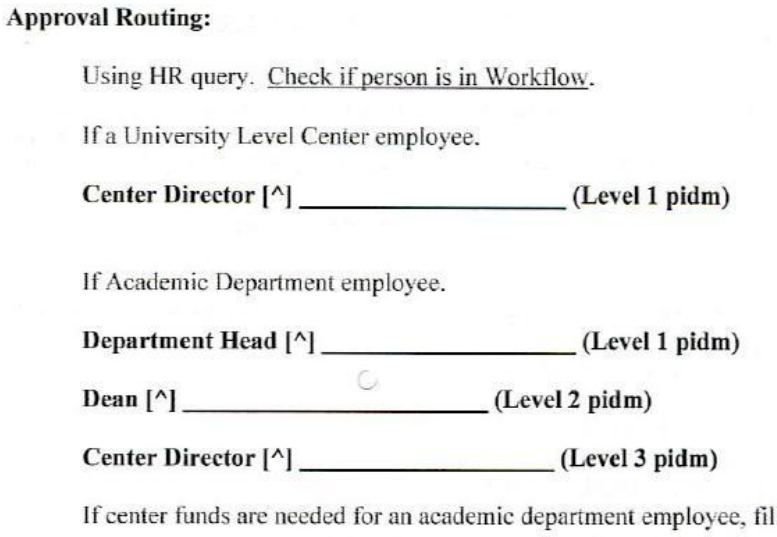
If Faculty Travel Program AND person is a University Level Center employee
(radio button selected above, show these options)

Univ Level Center	$_____	Center _____
ORED	$_____	
Other	$_____	Please Specify_____
Total	$_____	

Total of funding entered on these must match Total amount of funding requested from ORED from above.

Fig. 6 ORED Funding Request Document

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 9 , September 2016



Approval Routing:

Using HR query. Check if person is in Workflow.

If a University Level Center employee.

Center Director [^] _____ (Level 1 pidm)

If Academic Department employee.

Department Head [^] _____ (Level 1 pidm)

Dean [^] _____ (Level 2 pidm)

Center Director [^] _____ (Level 3 pidm)

If center funds are needed for an academic department employee, fill out level 3.

Fig. 7 ORED Travel Funding Approval Document

The following is a snippet of the SRS document that was produced. The snippet shows the IEEE recommended format and organization of the Software Requirements Document. Section 1 and Section 2 contain the header information but no data has been provided for those sections. Section 3 is formatted with headings and data which was a goal of executingthe methodology in this case study. Section 3.1 lists a requirement that was processed in the case study. All of the data was captured and stored within the Faculty Travel Ontology. Section 3.8 captures provenance information which is a contribution of this methodology. All requirements can be traced throughout the requirements engineering process in this methodology.

Faculty Travel and Quick Grants Program
Software Requirements Specification
Version 6.0
Produced by:
Robert A. Elliott, Sr.
Dr. Edward B. Allen
Dr. Allen Ulmer
Rene Hunt
Table of Contents
1. Introduction
1.1 Purpose
1.2 Scope
1.3 Definitions, abbreviations, acronyms
1.4 References
1.5 Overview
2. Overall Description
2.1. Product Perspective
2.1.1 System Interfaces
2.1.2 User Interfaces
2.1.3 Hardware Interfaces
2.1.4 Software Interfaces
2.1.5 Communication Interfaces
2.1.6 Memory
2.1.7 Operations
2.1.8 Site Adaptation Requirements.
….

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 9 , September 2016

3.1. System Features
3.1.1 Requirement Name = accessEmployeeStatus
3.1.2 Elicitation Technique = Document Review
3.1.3 Requirement Description = The software shall have access to employee status(faculty, staff, fulltime, partime, etc...)
3.1.4 Requirement Goal = This requirement's ultimate goal is to allow the proposed software to access all information that pertains to individuals that will request funds from the Office of Research and Economic Development(ORED).
3.1.5 Requirement Priority = High
3.1.7 Elicitation Location = ITS--Office 215
3.1.8 Elicitation Address = Mississippi State University - Main Campus
3.1.9 Operational Environment Requirement = .
3.1.10 Organizational Environment = .
3.3 Functions
3.3.1 Functions: Responses = CommunicationFacility
3.3.2 Functions: Validity_Check? = TRUE
3.3.3 Functions: Relationship = interface software
3.8.2 Validator Date = 2011-12-26
3.8.3 Documentor = Dr. Susan N. Dixon
3.8.4 Documentor Date = 2011-12-26

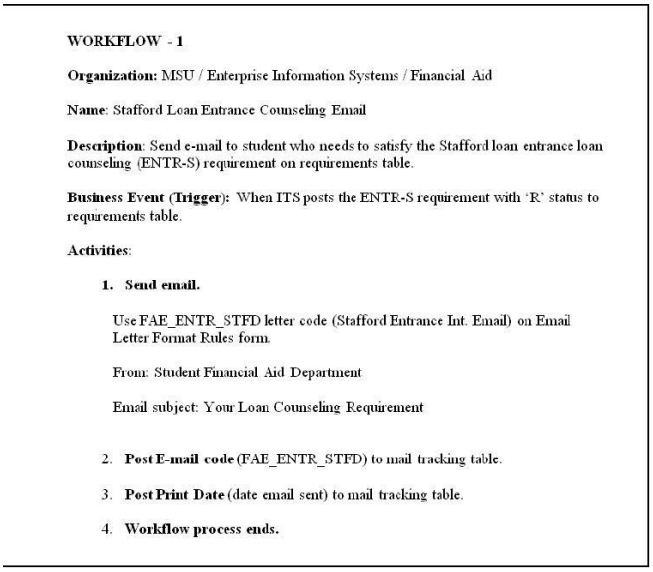
VII. CASE STUDY—WORKFLOW

This retrospective case study examined requirements for software for a financial aid application developed at Mississippi State University that supplemented the administrative system. This application provided software to automatically notify staff, students and users of information about financial aid transactions including loans. This case study focused on creating an entire software requirements document populating all sections of the document. This case study confirmed that the research methodology could produce an entire software requirements specification document.
The requirements for this case study were captured on forms that were designed to notify all entities at the university involved in the financial aid loan process.

A. Research Method

The baseline ontology used in this case study was the previously created Faculty TravelOntology. This retroactive case study was a financial aid application within an educationalenvironment. The Faculty Travel Ontology was modified to include the data items for automaticemail notification to staff and students concerning financial aid. This new modified ontology was namedWorkFlow and was formed with additions and deletions of data items in the Faculty Travel Ontology. Deletion of the old ontology data took approximately 1.5 hours while new ontology design took forty-eight hours. Data items for this case study are listed in Appendix A of Elliott's dissertation [4].
The source information for this case study was provided in ten workflow items. Figure 8 below lists the contents for a workflow item. Each workflow item contained certain conditions. When student data records matched conditions in a workflow, activities within the workflow were performed.
Table 2 below displays the requirements of the application and lists each condition and the associated workflow or action. Seven of the workflows or actions were associated withjust one condition while three were associated with two or more trigger conditions. Theconditions were codes that when met, triggered an action to be performed.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 9 , September 2016



WORKFLOW - 1

Organization: MSU / Enterprise Information Systems / Financial Aid

Name: Stafford Loan Entrance Counseling Email

Description: Send e-mail to student who needs to satisfy the Stafford loan entrance loan counseling (ENTR-S) requirement on requirements table.

Business Event (Trigger): When ITS posts the ENTR-S requirement with 'R' status to requirements table.

Activities:

1. **Send email.**

   Use FAE_ENTR_STFD letter code (Stafford Entrance Int. Email) on Email Letter Format Rules form.

   From: Student Financial Aid Department

   Email subject: Your Loan Counseling Requirement

2. **Post E-mail code** (FAE_ENTR_STFD) to mail tracking table.
3. **Post Print Date** (date email sent) to mail tracking table.
4. **Workflow process ends.**

Fig. 8 WorkFlow 1

Condition Number	Condition	WorkFlow
1	ENTR-S = "R"	1
2	FSLSIR = "R"	2
3	LDCD-S = "R"	3
4	Subsidized Stafford Loan Payment	4
5	Unsubsidized Stafford Loan Payment	4
6	Graduate PLUS Loan Payment	5
7	PLUS Loan Payment	6,7
8	ENTR-G = "R"	8
9	EXIT-S = "R"	9
10	Enrollment_Status = "WITHDRAWN"	9
11	Current_Term = "YES"	9
12	Fund Code = "STFDS"	10
13	Fund Code = "STFDU"	10
14	Loan Status = "ACPT"	10
15	Application Status = "Sent"	10
16	Current date = Submission Date + 14	10
17	Disbursement = "NO"	10
18	Loan Disbursement Number = Application Loan Number	10
19	Letter Code = "FAE_NO_DISB_XX"	10

Table 2 WorkFlow Ontology Requirements Table

The workflows were also associated with input and output external interfaces. There were seven input files, forms and records and three output external interfaces. These external interfaces were defined as external interface requirements in the ontology.

The actions to be carried out by the workflows were modeled in the ontology as functions. Four distinct functions were defined that would do all of the necessary actions thateach workflow was designed to fulfill. Each of the four functions were defined in the ontology as requirements. Table 3 displays the analysis results of those requirements.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 9 , September 2016

Condition Number	Condition	WorkFlow
1	ENTR-S = "R"	1
2	FSLSIR = "R"	2
3	LDCD-S = "R"	3
4	(LoanName ="Subsidized Stafford Loan Payment" OR "Unsubsidized Stafford Loan Payment")	4
5	Graduate PLUS Loan Payment	5
6	PLUS Loan Payment	6,7
7	ENTR-G = "R"	8
8	EXIT-S = "R"	9
9	(Fund Code = "STFDS" or "STFDU") AND Loan Status = "ACPT" AND APPL_STATUS ="Sent" AND (currentDate=submissionDate+14) AND Disbursement = "NO" AND LDNumber=APPLNumber)	10

Table 3 WorkFlow Ontology New Requirements Table

The research method was executed in this case study and the requirements were captured and stored in the ontology.

B. Results

The execution of the methodology produced a complete requirements specification document for this case study. A progressive approach was used to produce this document by using lessons learned from the previous two case studies. The methodology could accommodate multiple workflow processes and accommodated different problem domains.
This research prototype document did have a readability issue. Standard data and lists were stored in ontology variables.

VIII. DISCUSSION

The following discusses empirical evidence generated by the case studies and threats to validity. This research devised a new methodology for creating software-requirements specifications, based on an ontology. The practicality of this methodology was demonstrated by three retrospective case studies that simulated some aspects of real-world use of the methodology. We found that only minor modifications to the ontology were needed even though the case studies were in different application domains.

A. Empirical Evidence

All of the case studies were based on documentation of past projects rather than active. The researcher played the role of requirements engineer. This approach was adequate for an initial demonstration of the practicality of the methodology and of the usefulness of ontological support.
The first case study, FloodViz, produced the detailed requirements section of the SRS and used an automated reasoner to verify some requirement properties. This case study also demonstrated the recording of provenance data in the ontology. The first case studywas in the scientific visualization domain.The second case study, Faculty Travel, produced an SRS with detailed requirement plus placeholders for other sections, namely, the Introduction and Overall Description.
This case study demonstrated that the methodology was suitable for a business application domain with a forms oriented style, and that customization of the ontology was readily achieved.
The third case study, Workflow, produced a complete SRS. It demonstrated the suitability of the methodology to business application domain with a workflow style of requirements.
In summary, the three case studies showed that the proposed methodology's use cases are practical and useful over a variety of application domains and requirement styles. Ontological support of requirements engineering has the potential to support verification using an automated reasoner, and to facilitate tracking of provenance of requirements.

B. Threats to Validity

Conclusion validity, *internal validity*, and *construct validity*were not applicable to the case studies because our case studies did not involve statistical analysis. However, this research had the following threats to external validity.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 9 , September 2016

Although the proposed methodology and ontology are presented as generally applicable to requirements engineering, the case studies did use a specific set of software tools for the reference implementation. Alternative tools might have different capabilities.

The case studies were retrospective, based on artifacts of past projects and not the performance of people in active projects. Use of the proposed methodology and ontology in practice might have other practical issues. The FloodViz project was performed in the past by university researchers in collaboration with relevant government agencies. The Faculty

The case studies were based on two application software domains. The FloodViz domain was scientific visualization. The Faculty Travel and Workflow projects were in the business administration domain. The case-study results did not address other domains.

IX. CONCLUSION

A method to produce a software requirements specification based on an ontology was the primary contribution of this work. The underlying ontology-defined data was recommended by the IEEE Standard 830-1998: Recommended Practice for Software Requirements Specifications. The ontology also encoded data elements in the Software Requirements Knowledge Area of the SWEBOK. A method to extract software requirements from an OWL/RDF ontology and format them in simple text files was created. The research produced scripts to automatically create a software requirements specification document

in conformance with IEEE Standard 830-1998. The method also facilitates capture of provenance data during the requirements engineering process.

We conducted three empirical retrospective case studies which showed that requirements could be defined, stored and extracted from an ontology. All three case studies were based on artifacts from real world projects. The case studies also showed that a software requirements specification document could be produced from extracted requirements. The FloodViz case study provided evidence that characteristics of ontology based requirements could be automatically verified by a logic engine (Pellet).

Acknowledgments

This work was supported in part by the ”Rapid Prototyping Capability for Earth-Sun System Sciences” project at Mississippi State University, sponsored by the NASA via the Mississippi Research Consortium. This work was also supported in part by the School of Business of Southern University at New Orleans.

We thank the following people for helpful discussions: Philip Amburn, Rene Hunt, Charles O’Hara, David Dampier, Nan Niu, Patricia Robertson, Samuel Eweni and David Alajahni.

REFERENCES

[1] Assawamekin, T. Sunetnanta, and C. Pluempitiwiriyawej, “Resolving MultiperspectiveRequirements Traceability through Ontology Integration,” *Proceedings: The5th IEEE International Conference on Semantic Computing*, Santa Clara, California, August 2008, pp. 362–369.

[2] B. H. C. Cheng and J. M. Atlee, “Research Directions in Requirements Engineering”, *Proceedings: The Future of Software Engineering*. IEEE, May 2007.

[3] R. Denaux, L. Aroyo, and V. Dimitrova, “An Approach for Ontology-based Elicitation of User Models to Enable Personalization on the Semantic Web,” *Proceedings:The 14th International Conference on World Wide Web*, New York, 2005, pp. 1170–1171, ACM.

[4] R. Elliott, *Software requirements elicitation, verification, and documentation: An ontology based approach*, doctoral dissertation, Mississippi State University, MSState, Ms, December 2012.

[5] R. Elliott, “Creating An IEEE Standard 830-1998 Software Requirements Specification Document,” *Proceedings: 24th Annual Consortium for Computing Sciences inColleges, South Central Conference*, LSU-Shreveport, La, April 2013, Consortium for Computing Sciences in Colleges, Poster.

[6] R. Elliott, “A Methodology for Creating An IEEE Standard 830-1998 Software Requirements Specification Document,” *Proceedings: 24th Annual Consortium forComputing Sciences in Colleges, Southeastern Conference*, Greenville, SC, November 2013, Consortium for Computing Sciences in Colleges.

[7] M. Georgiades and A. Andreou, “Automatic Generation of a Software Requirements Specification (SRS) Document,” *Proceedings: The 10th International Conference onIntelligent Systems Design and Applications (ISDA)*, Cairo, Egypt, December 2010,IEEE, pp. 1095–1100.

[8] N. Gu, J. Xu, X.Wu, J. Yang, andW. Ye, “Ontology based semantic conflicts resolution in collaborative editing of design documents,” *Advanced Engineering Informatics*,vol. 19, no. 2, April 2005, pp. 103–111.

[9] IEEE Computer Society, *IEEE Recommended Practice for Software Requirements Specifications*, Standard 830-1998, IEEE, New York, June 1998.

[10] IEEE Computer Society, *Guide to Software Engineering Body of Knowledge(SWEBOK)*, Standard ISO/IEC TR 19759-2005, International Organization forStandardization, Geneva, Switzerland, February 2005.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 9 , September 2016

[11] S.-W. Lee, R. Gandhi, D. Muthurajan, D. Yavagal, and G.-J. Ahn, "Building problem domain ontology from security requirements in regulatory documents," *SESS '06:Proceedings of the 2006 international workshop on Software engineering for securesystems*, New York, NY, 2006, pp. 43–50, ACM.

[12] Y. Lee andW. Zhao, "An Ontology-Based Approach for Domain Requirements Elicitation and Analysis," *Proceedings: The 1st International Multi-Symposiums on Computerand Computational Sciences*, Shanghai, China, 2006, IEEE, pp. 364–371.

[13] Y. Liu, L. Gao, and X. Feng, "An Ontology Modeling Methodology in Requirements Analysis," *Proceedings: The 3rd International Conference on Biomedical EngineeringAnd Informatics (BMEI)*, Yantai, China, October 2010, IEEE, pp. 2739–2742.

[14] G. A. Mala and G. Uma, "Design and Development of Ontology for Natural Language Requirement Specification by Eliciting Object Oriented Elements," *AsianJournal of Information Technology*, vol. 5, no. 9, November 2006, pp. 985–989.

[15] N. Miura, H. Kaiya, and M. Saeki, "Building the Structure of Specification Documents from Utterances of Requirements Elicitation Meetings," *Proceedings: The 2nd Asia Pacific Software Engineering Conference*, Brisbane, Australia, December 1995, IEEE, p. 64.

[16] D. O'Leary, "Special Issue on Verification and Validation Issues in Databases,Knowledge-based Systems, and Ontologies," *International Journal of Intelligent Systems*,vol. 16, no. 3, March 2001, pp. 263 – 264.

[17] M. Sabou, M. d'Aquin, and E. Motta, "Using the Semantic Web as Background Knowledge for Ontology Mapping," *Proceedings: The 5th International SemanticWeb Conference ISWC 2006 Workshop Ontology Matching*, Athens, Georgia, 2006,International Semantic Web Organization.

[18] A. Sardinha, A. Rashid, R. Chitchyan, N.Weston, and P. Greenwood, "EA-Analyzer:A Tool for Identifying Conflicting Dependencies in Requirements Documents," *Proceedings:The 24th IEEE/ACM International Conference on Automated Software Engineering*,Auckland, New Zealand, November 2009, IEE/ACM, pp. 530–534.

[19] A. Sharma and D. S. Kushwaha, "Natural Language Based Component Extraction from Requirement Engineering Document and its Complexity Analysis," *ACM SIGSOFTSoftware Engineering Notes*, vol. 36, no. 1, January 2011, pp. 1–14.

[20] H. Sharp, A. Finkelstein, and G. Galal, "Stakeholder Identification in the RequirementsEngineering Process," *Proceedings: The 10th International Workshop onDatabase and Expert Systems Applications*. IEEE, September 1999, pp. 387–391.

[21] X. Zhang, *An Interactive Approach of Ontology-based Requirement Elicitation*, doctoral dissertation, University of Windsor, Windsor, Canada, 2011.