A SEMI-AUTOMATED APPROACH TO SUPPORT LOGICAL FORMALISM FOR REQUIREMENTS ANALYSIS AND VALIDATION

RICHA SHARMA

AMAR NATH AND SHASHI KHOSLA SCHOOL OF INFORMATION TECHNOLOGY

INDIAN INSTITUTE OF TECHNOLOGY, DELHI

AUGUST 2016

A SEMI-AUTOMATED APPROACH TO SUPPORT LOGICAL FORMALISM FOR REQUIREMENTS ANALYSIS AND VALIDATION

by

Richa Sharma

Submitted in fulfilment of the

Thesis requirements for the degree of
Doctor of Philosophy

to the

Amar Nath and Shashi Khosla School of Information Technology
Indian Institute of Technology, Delhi
August 2016

Certificate

This is to certify that the thesis titled "**A Semi-Automated Approach to support Logical Formalism for Requirements Analysis and Validation**" being submitted by **Richa Sharma** to the Indian Institute of Technology Delhi, for the award of the degree of **Doctor of Philosophy** in Amar Nath and Shashi Khosla School of Information Technology, is a record of bona-fide research work carried out by her under my supervision. To the best of my knowledge, the work presented in this thesis has not been submitted to any other university or institute for the award of any other degree or diploma.

Prof. K.K. Biswas

Department of Computer Science and Engineering

Indian Institute of Technology Delhi

New Delhi 110 016

ACKNOWLEDGEMENTS

This thesis work would not have been possible without the support and cooperation of many people. I would like to take this opportunity to thank them all.

First of all, I would like to express my sincere and profound gratitude to my supervisor, Prof. K.K. Biswas for his invaluable guidance, continuous encouragement and wholehearted support in every stage of this research. His constant encouragement in exploring new things immensely increased my knowledge. His critical comments on my technical writing and eye for details strongly motivated me to strive hard towards excellence in completion of this thesis.

I am extremely grateful to the members of my Research Committee – Prof. Niladri Chatterjee, Prof. Saroj Kaushik and Prof. Parag Singla, who have been very helpful by offering suggestions and advice. A special mention goes to Prof. Niladri Chatterjee whose constructive reviews enabled me to give final shape to my work and thesis. I would like to thank Prof. V. Gervasi, who I happened to meet at the time of my first international presentation and who gave me the directions for future work. My cordial thanks to two professors from Computer Science and Engineering department, Prof. S. Arun Kumar and Prof. S.K. Gupta, constant moral support and encouragement enabled me to overcome disappointing phases.

I am also obliged to those who have agreed to participate in the empirical studies and annotation tasks carried out as part of this research work with a special mention for Jaspreet Bhatia. I thank her for the fruitful and interesting discussions we had while working on some parts of this thesis together. I would like to thanks students from Indian Institute of Technology - Delhi, Shiv Nadar University and Bharti Vidyapeeth's College of Engineering, Delhi for enthusiastically participating in empirical studies and annotations. I would also like to thank my colleagues from Shiv Nadar University and participants from Tech Mahindra without whose support, value-addition to empirical studies would not have been possible.

Most importantly, I would like to thank my parents for instilling me with a life-long love for learning and education. My mother Raj Rani Sharma and father K.C. Sharma showed immense patience and provided me great moral support during the course of my work. I would like to make special word of thanks for my mother who stood as a pillar of strength during all the ups and downs of the research. This thesis is dedicated to my parents.

Richa Sharma

Abstract

The growing complexity and size of software systems emphasize the need for capturing the requirements in a way that is amenable to automatic requirements analysis and validation so that the defects in the delivered software can be reduced considerably. Natural Language (NL) is the most preferred form of representing requirements in software industry. However, NL cannot be subjected to automated reasoning, analysis, and validation, thereby leaving inconsistencies unnoticed. NL requirements are usually analysed through manual reviews, inspections. Requirements analysis models like structured analysis diagrams (data-flow diagrams and structure charts) or object-oriented analysis diagrams (UML models) are also often drawn manually to assist the analyst while analysing the requirements. However, being manual in nature, these industry practices of requirements analysis are effort-consuming, time intensive, and dependent on analysts' expertise and experience. Moreover, NL is inherently ambiguous allowing the possibility of different interpretation of the requirements statements other than the intent of business users.

In the absence of tool support for automated reasoning, analysis, and validation, ambiguities and inconsistencies are often overlooked in requirements specification documents expressed in the form of NL. Formal Knowledge Representation techniques have been widely researched for representing requirements, each with varying degree of success. Such Knowledge Representation (KR) techniques include mathematical representations and logical representations. These representation techniques help in identifying and resolving the ambiguity and inconsistency issues in software requirements. These defects, if not controlled during requirements analysis, may percolate down to subsequent phases of software development like design and development. However, the earlier explored requirements representations are limited by their inability to maintain consistency between conflicting requirements. Secondly, they suffer from practicality and applicability concern from software industry point of view. The business users and stakeholders providing requirements are not comfortable with the logical representation of requirements.

This thesis work is an attempt to address ambiguity and inconsistency concern in the NL requirements documents as well as to address the practicality and applicability issues in the proposed solution approach. We propose Courteous Logic as a suitable form for representing requirements that not only supports automated reasoning and inferencing but also preserves consistency between conflicting requirements. Our key contribution lies in proposing a semi-automated approach that first addresses ambiguity concern in the requirements and then, translates unambiguous NL representation of requirements to Courteous Logic representations. The proposed semi-automated approach for translation has also been found effective for generating UML (class, activity, and sequence diagrams) models automatically. The work presented in this thesis is an attempt to bridge the gap between industry practice of capturing requirements, and the formal KR techniques. We make use of the expressive strength of NL and the reasoning capabilities of formal KR techniques.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI - Artificial Intelligence

BNC - Brown National Corpus

CL - Courteous Logic

DL - Description Logic

FOL - First Order Logic

GKP - Grammatical Knowledge Patterns

GT - Grounded Theory

KR - Knowledge Representation

ML - Machine Learning

NL - Natural Language

NLP - Natural Language Processing

NML - Non-monotonic Logic

OOA - Object oriented Analysis

RE - Requirements Engineering

SE - Software Engineering

SDLC - Software Development Life Cycle

SSL - Semi-supervised Learning

UML - Unified Modelling Language