Ontology Development and Utilization in Product Design

Xiaomeng Chang

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State
University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Grado Department of Industrial & Systems Engineering

Committee Chair: Dr. Janis P. Terpenny
Committee Member: Dr. Patrick Koelling
Committee Member: Dr. Eileen Van Aken
Committee Member: Dr. Terry R Rakes

April 23, 2008
Blacksburg, VA

Keywords: Knowledge Management, Ontology, Data Integration, Decision Support

Ontology Development and Utilization in Product Design

Xiaomeng Chang

ABSTRACT

Currently, computer-based support tools are widely used to facilitate the design process and have the potential to reduce design time, decrease product cost and enhance product quality. PDM (Product Data Management) and PLM (Product Lifecycle Management) are two types of computer-based information systems that have been developed to manage product lifecycle and product related data. While promising, significant limitations still exist, where information required to make decisions may not be available, may be lacking consistency, and may not be expressed in a general way for sharing among systems. Moreover, it is difficult for designers to consider multiple complex technical and economical criteria, relations, and objectives in product design simultaneously.

In recent years, ontology-based method is a new and promising approach to manage knowledge in engineering, integrate multiple data resources, and facilitate the consideration of the complex relations among concepts and slots in decision making. The purpose of this research is to explore an ontology-based method to solve the limitations in present computer-based information systems for product design.

The field of Design for Manufacturing (DFM) is selected for this study, and three primary aspects are investigated. First, a generalized DFM ontology is proposed and developed. The ontology fulfills the mathematical and logical constraints needed in DFM, as well as ontology editor capabilities to support the continuous improvement of the ontology. Second, the means to guide users to the proper information and integrate heterogeneous data resources is investigated. Third, based on the ontology and information integration, a decision support tool is developed to help designers consider the design problem in a systematic way and make design decisions efficiently based on accurate and comprehensive data.

The methods and tools developed in this research are refined using example cases provided by the CFSP (The NSF Center for Friction Stir Processing). This includes cost models and a decision support environment. Errors that may occur in the research are categorized with management methods. An error ontology is built to help root cause analysis of errors and further reduce possible errors in the ontology and decision support tool. An evaluation methodology for the research is also investigated.

Acknowledgement

I would like to first thank my advisor, Dr. Terpenny, for her support during my Ph.D. study. Three years ago, it was my first time to go abroad alone and study in a new environment. Dr. Terpenny gave me bunches of advice and helped me conquer many difficulties in both my study and my life. I offer my sincere appreciation and gratitude to her patient advice, warm help, funding, and edits of the documents. To my other committee members, Dr. Koelling, Dr. Van Aken and Dr. Rakes, I am very grateful for their probing questions and validation of the worthiness of my research. Thank you very much for your help and support.

I would also thank you to all my friends in the SMART lab as well as in the NSF e-Design Center for the study and happy time together. Among them, Dr. Asli Sahin, who has graduated from Virginia Tech, coauthored two of my research papers (one conference paper and one journal paper), and Dr. Seung Ki Moon in Penn State University collaborated with me in two of his papers (one conference paper and one book chapter). It is a wonderful time for me to make a progress with these friends in my graduate study. I am also most grateful to the collaborators in NSF CFSP (Center for Friction Stir Processing), Kumar Tipaji Pradeep, Harish Sadanand Bagaitkar and other professors, for providing the research background, cost models and decisions support environment in my research. In addition, thanks to the REU (Research Experiences for Undergraduates) students, Brandon Brisbon, Johnna Esposito and Nikita Sharakhov, at Virginia Tech for the realization of my research.

I am inspired by and thankful to all my friends in Blacksburg and all over the world. Although we are in different places pursuing our own dreams, your encouragement and friendship in these years make me realize my goals in study confidently.

This dissertation is dedicated to my parents, Guodong Chang and Jing Lv, and all the other family members. As your single daughter, I know how hard for you to be apart with me. Thank you for consistently encouraging me to be brave, independent and optimistic, and I appreciate your support and love in these years. 至此，由衷感谢我的父母，常国栋和吕晶，以及其他家人对我的爱护、鼓励和帮助。因为你们，才使我能够勇敢、

独立而乐观地面对生活中的挑战，才使平凡的我变得幸运，才使我有了我美好的过去，今天和未来。

Finally, but not lastly, I am grateful for boyfriend, Yanfeng Li, for the happy and hard time together during our undergraduate and graduate study in these ten years. Thank you very much for your understanding, patience, help and collaboration. I am so happy to grow up with you.

Table of Contents

ABSTRACT ii
Acknowledgement iii
Table of Contents v
List of Figures viii
List of Tables x
Chapter 1 Introduction 1
1.1 Background 1
1.1.1 Present Situation in Product Design 1
1.1.2 Motivation 2
1.2 Research Purpose and Objectives 4
1.3 Research Questions 4
1.3.1 Primary Research Questions 4
1.3.2 Research Sub-questions 5
1.4 Research Approach and Methods 6
1.5 Research Contributions 6
1.6 Outline of the Dissertation 7
Chapter 2 : Review of Literature 9
2.1 Present Product Design 9
2.1.1 Need in Product Design 9
2.1.2 Concurrent Engineering 11
2.2 Semantic Web 14
2.3 Ontology and its Present Development 15
2.3.1 Reasons of Using Ontologies 16
2.3.2 Ontology Editors 17
2.3.3 Some Present Ontologies related to Product Design 18
2.4 Information Integration Based on Ontology 19
2.4.1 Integration and Collaboration of Ontologies 20
2.4.2 Information Integration Methods 21
2.5 Multiple Criteria Decision Making (MCDM) in Product Design 24
2.5.1 Choice of Criteria and Measurement Scale 25
2.5.2 Classification to Form the Tree Structure for Decision Making 25
2.5.3 Analysis Techniques 26
2.5.3.1 Single-Dimensioned Analysis 26
2.5.3.2 Full-Dimensioned Analysis 29
2.5.4 Ontology-based Decision Support Systems 29
2.6 Management and Analysis of Errors in the Research 30
2.6.1 Types of Errors related to Ontology 30
2.6.1.1 Experimental Science 31
2.6.1.2 Engineering 31
2.6.1.3 Computer Programming 32
2.6.1.4 Numerical Analysis 33
2.6.2 Error Collection 33
2.6.3 Error Reason Analysis 34
2.7 Literature Related to Evaluation of Information Systems 35

2.8 Summary 39
Chapter 3 : Approach and Methodology 41
3.1 Overview of the Research Approach 41
3.2 Ontology Development 46
3.2.1 Concept Categorization and Class Hierarchy Development 46
3.2.2 Slot Categorization and Development 48
3.2.3 Identification and Realization of Relations among Slots in the Ontology. 49
3.2.3.1 Identification of Slot Relations 49
3.2.3.2 Realization of Relations among Slots in the Ontology 50
3.2.4 Strategy of Ontology Development 54
3.3 Ontology-Based Data Integration 55
3.3.1 Data Query and Retrieval by Using Query Tab in Protégé 55
3.3.2 Data Reasoning Based on JTP Reasoner 57
3.3.3 Information Integration based on Ontologies and Heterogeneous Data Resources 59
3.4 Decision Support Tool Based on Ontologies and Heterogeneous Data Resources 61
3.5 Test and Refine Ontologies and Ontology-based Information Systems 63
3.5.1. Errors and Causal Factors in Ontology Development, Reuse and Maintenance 64
3.5.1.1 Errors and Causal Factors in Ontology Development and Reuse.... 64
3.5.1.2 Errors and Causal Factors in Ontology Maintenance 67
3.5.2 Errors and Causal Factors in Ontology-based Data Integration 68
3.5.3 Errors and Causal Factors in Ontology-Based Information Systems 69
3.5.4 Ontology-Based Method for Root Cause Analysis (RCA) 70
3.6. Evaluation Methodology for the Research 72
3.6.1 Related Dependent Variables 72
3.6.2 Related Independent Variables 73
3.6.3 Evaluation of the Research 74
Chapter 4 : Research Results 77
4.1 Overall Framework of the Ontology-Based System 77
4.2 DFM Ontology and Other Ontologies for Data Integration and Decision Support 79
4.2.1 DFM Ontology 80
4.2.1.1 DFM Ontology Hierarchy and Important Classes 80
4.2.1.2 Relations Realization in DFM Ontology 84
4.2.2 Local Ontologies for Data Integration 88
4.2.3 DFM e-Design Ontology for Decision Support 89
4.2.3.1 Ontology Structure 89
4.2.3.2 SWRL Rules with its Usage in the Jess Rule Engine 90
4.3 Ontology-Based Data Integration 93
4.4 Local Multi-Attribute Decision-Making Tool (MADMT) Using Integrated Data 97
4.5 Ontology-Based Web System for DFM 99
4.6 Ontology-Based Root Cause Analysis for Error Control in the Research 107
Chapter 5 : Ontology-Based Welding Design Examples 114

5.1 Background of Case Studies 114
5.2 Case Studies 115
5.2.1 Case Study 1: Cost Analysis of MIG Welding Process............................115
5.2.2 Case Study 2: Filler Material and Machine Selection in MIG design Using SWRL 117
5.2.3 Case Study 3: Pin Tool Selection in FSW Design Using SWRL............. 122
5.2.4 Case Study 4: Ontology-Based Data Integration and Decision Support Platform.. 126
5.3 Conclusions 132
Chapter 6 : Contributions and Recommendations 135
6.1 Research Summary 135
6.2 Contributions of the Research.. 139
6.3 Limitations of the Research and Opportunities for Future Research................ 141
References 143
Appendix A : Introduction of Ontology 157
Appendix B : Background of FSW and the TIE Project 159
Appendix C : Cost Model for FSW 161
Appendix D : Cost Model for MIG (Metal Inert Gas) Welding................................ 168
Appendix E : FSW e-Design Process 175

List of Figures

Figure 1-1: A pictorial overview of the dissertation 8
Figure 3-1: Flowchart of the development and utilization of the ontology 43
Figure 3-2: Representation method for relations among classes 47
Figure 3-3: Method to represent class and its properties 48
Figure 3-4: Representation method for relations in slots 49
Figure 3-5: Query tab in Protégé with its query result and query library 56
Figure 3-6: Query result and its detail information in Protégé 57
Figure 3-7: Working process of data reasoning based on the JTP reasoner 58
Figure 3-8: Framework of Ontology-Based Heterogeneous Data Integration 60
Figure 3-9: Working process of the decision support tool 62
Figure 3-10: Elements in the root cause map and its work process 71
Figure 4-1: Framework of ontology-based method for product design 78
Figure 4-2: Hierarchy of the DFM ontology 81
Figure 4-3: MIG Welding related slots and their relations in the DFM ontology 85
Figure 4-4: Relations among classes and slots in local ontologies 88
Figure 4-5: Screenshot of the reasoning result in the DFM e-Design ontology 93
Figure 4-6: Working process of ontology-based data integration and analysis 94
Figure 4-7: Relations among central ontology, local ontologies and data resources 94
Figure 4-8: Flowchart of local MADMT 97
Figure 4-9: Working mechanism of MADMT 98
Figure 4-10: Working mechanism of the decision support platform on B/S mode 99
Figure 4-11: Flowchart of the ontology-based web system for DFM 100
Figure 4-12: Variables and their relations in the ontology-based web system for DFM 102
Figure 4-13: Ontology-based query and reason process for output increase 106
Figure 4-14: Ontology-based query and reason process for output decrease 106
Figure 4-15: Hierarchy of the Error Ontology 107
Figure 4-16: Classes and their slots in the EO 108
Figure 4-17: Root cause map for RCA in ontology development and utilization 110
Figure 4-18: An example of the reasoning result in EO by SWRL and Jess rule engine 112
Figure 4-19: Screenshot of EO in Protégé to find the root cause 113
Figure 5-1: Design of a new welding alternative using Java Function Calls programs .. 116
Figure 5-2: Classes and slots used in case study 2 118
Figure 5-3: Relations among slots used in case study 2 118
Figure 5-4: A new instance in the MIG_Design_cases class of the DFM ontology 120
Figure 5-5: SWRL rules and Jess rule engine used to infer new knowledge 121
Figure 5-6: Inferred result from SWRL and Jess rule engine for material selection 121
Figure 5-7: Allowed fillers and machines inferred in a design case 122
Figure 5-8: Using the DFM ontology in Protégé to select suggested pin tool 125
Figure 5-9: Using the DFM ontology in Protégé to select the allowed pin tool 125
Figure 5-10: Screenshot of Step 3 of the ontology-based web system 127
Figure 5-11: Screenshot of integrated data resources based on ontology 128
Figure 5-12: Screenshot of integrated data from one data resource 128
Figure 5-13: Integrated data from the ontology-based web system for DFM 129
Figure 5-14: Screenshot of MADMT on a local computer 130

Figure 5-15: Outputs of the ontology-based web system for DFM 131
Figure 5-16: Screenshot of parameter adjustment suggestion 132
Figure 5-17: Screenshot of possible result for decreasing the dimension of Component 1 and 2 132
Figure A-1: An example of using ontology in the wine agent 158
Figure E-1: Component geometry options in FSW 175
Figure E-2: Joint category in FSW 176

List of Tables

Table 2-1: Comparison of three ontology-based information integration methods 22
Table 2-2: Comparison of single and full dimensioned analysis 26
Table 2-3: Difference of statistical error and systematic error... 31
Table 2-4: Summary of error collection methods 34
Table 3-1: Comparison of rule-based method and Java Function Calls 54
Table 3-2: Relationship between dependent and independent variables in evaluation..... 75
Table 4-1: Java Function Calls Programs to realize the cost model for MIG welding..... 86
Table 4-2: Java Function Calls Programs to realize the cost model for FSW 87
Table 4-3: Restrictions and their related classes in the DFM ontology 87
Table 4-4: Important properties and sub-properties in the DFM e-Design ontology........ 89
Table 4-5: Classes and their properties in the DFM e-Design ontology 90
Table 4-6: SWRL rules in the DFM e-Design ontology to infer root influence factors ... 91
Table 4-7: An instance in the local_databases class in the DFM ontology....................... 96
Table 4-8: Input variables and their influence factors.. 103
Table 4-9: Output variable category and influence factors 104
Table 5-1: Comparison of two design alternatives in MIG welding design 117
Table 5-2: Properties in the DFM ontology for the FSW tool selection 123
Table C-1: Average hourly wage of a labor for FSW 162
Table C-2: MARC's depreciation table.. 164
Table C-3: Average electric power cost in different states of USA 165
Table C-4: Criteria for FSW tool selection 167
Table C-5: Currently available tools at UMR 167
Table D-1: Relation between metal thickness and filler or wire diameter...................... 170
Table D-2: Relation between metal thickness and current... 170
Table D-3 : Few MIG welding machines costs and capabilities.................................... 170
Table D-4: Commonly used filler material types and associated price.......................... 172
Table D-5: Commonly used shielding gas proportion, quantities and associated price . 172
Table D-6: Average hourly wage of a labor for MIG welding....................................... 172
Table D-7: Price list of generally used tools and weld preparation equipment 173
Table D-8: Price list of generally used safety equipment 173

Ontology Development and Utilization in Product Design

Chapter 1 Introduction

1.1 Background

1.1.1 Present Situation in Product Design

Product design can influence a large portion of the product cost, so it is critical to obtain the cost estimates as early and quickly as possible during the design phase (Hyeon et al., 1993). Moreover, 80% to 90% of the time to market is used in the product planning and development phase (Charney, 1991) due to the complexity of the product, and 40% of all the quality problems are caused by bad design (Saaksvuori and Immonen, 2005). For example, the bad design can be the one that is not technically feasible. If a company wants to shorten significantly the time to market and enhance the quality of its product, it should try to enhance the efficiency and quality of the product design.

Computer-based approaches are generally used to facilitate designers in reducing design time, decreasing product cost and enhancing product quality. In the past twenty years, several CAX (Computer Aided X) and DFX (Design for X) tools have been developed for this purpose. All DFM systems share a common goal, which is to minimize the total product life-cycle costs (O' Flynn and Ahmad., 1993) through more systematic and efficient decision making, when considering the design in light of manufacturing,.

The data produced by existing information systems, such as CAD and DFM systems, are already in electronic format. But all the information required to make a decision may not be available, lack consistency, and may not be expressed in a general format. According to a study by Coopers and Lybrand, 59% of the design time can be saved if the information system can help share, retrieve and reuse the information efficiently. PDM and PLM systems are aimed at helping users allocate data and facilitate design.

A PDM System (Product Data Management System) has been defined as "a systematic, directed method by which to manage and develop an industrially manufactured product (Saaksvuori and Immonen, 2005, p.243)." PLM (Product Lifecycle Management) is defined as a systematic, controlled concept for managing and developing products and product related information. It helps connect separate information data systems, process and automation islets. Both PDM and PLM are information systems to manage product lifecycle and product related data (Saaksvuori and Immonen, 2005), but PLM has a wider frame of PDM, especially to the life cycle perspective of information management (Saaksvuori and Immonen, 2005).

However, since PLM needs to support different databases, the use of several different database types brings a greater challenge to the PLM system and naturally to the integration and data transfer between different systems. How to utilize the data integrated and retrieved is also a key problem for PLM, since no matter how data are retrieved, it must be usable and aid designers in order to be of value.

In product design, there are several technical and economical factors that need to be considered at the same time and these factors are related with each other. Facing the abundant data integrated from multiple data resources and the complex relations among them, designers also need support to make the decision in product design.

1.1.2 Motivation

In recent years, the concept of ontology has been used in the field of knowledge management and Computer Supported Cooperative Work (CSCW). **Ontology** is a formal specification of domain knowledge and has been used to define a set of data and their structure for experts to share information in a domain of interest. It is good at the representation and utilization of relations among data, and is efficient in knowledge reasoning.

Consider McGuinness et al.’s KSL wine agent as an example. It is a test bed application for semantic web technologies. It combines a logical reasoner (called JTP (Frank, et al., 2004)), an object-oriented modular reasoning system with an OWL ontology. In the wine agent, after a meal description is received, a selection of matching wines available on the web is retrieved. This is realized by consulting the ontology, performing queries, and outputting results. The wine agent demonstrates that at least the simple task can be accomplished within semantic markup technology.

At present, product designers are separated in different places all over the world and use multiple computer support tools to complete design tasks. The data created is saved in isolated and heterogeneous data resources, making it difficult to integrate information from data resources and obtain optimal designs.

Similar to the wine industry, which has restaurants, customers, retailers, and so on, semantic web technologies can also be used in product design to relate product, manufacturing process, machinery, material, etc. Methods in the wine agent can be used to build up the information system for product design based on ontology. For example, engineers can mark each product with standardized machine-readable definitions, manufacturers can mark activities of manufacturing, and providers can mark resources that those activities need. Rather than the traditional approach of trying to build an enormous database of products, resources, activities, etc., the definitions would be distributed across the participating parties with concepts that are connected through an ontology. Utilizing the semantic relationships in the ontology, appropriate information can be located after querying the ontology. For example, cost data can be obtained through a sequence of queries from item cost to activity cost, and finally to resource cost with other related information, such as details about the item, activities, material and resource. Ontology can also be queried to get the design suggestions like the wine selection suggestion in the wine agent.

The purpose of this research is to explore the ontology-based method for product design. The field of Design for Manufacturing (DFM) is selected as the context for this study, since it impacts and relates several life cycle phases, and it is important to consider representation and decision support in the field of DFM. The DFM ontology is developed and the information is queried in a manner similar to the wine agent. After the proper information is located, detail data can also be retrieved from heterogeneous data resources. Based on the ontology and information integration, the proposed decision support tool is developed to help designers consider the design problem in a systematic way and make design decisions efficiently based on accurate and comprehensive data.

1.2 Research Purpose and Objectives

The purpose of this research is to explore the ontology-based method to enhance the consistency and availability of data, increase the efficiency of data utilization, and support the decision making of product design for product life cycle. The field of Design for Manufacturing (DFM) is used in the study, which includes the overall process of manufacturing, and focuses on the technical and economical criteria in manufacturing. The DFM ontology and other assistant ontologies can be used:

- to capture the domain knowledge in DFM,
- to define a set of data and their structure and relations for product designers who need to share information in a domain of interest,
- to retrieve the feasible design alternatives from heterogeneous data resources,
- to help designers make decisions based on technical and economic characteristics and relations.

Utilizing advantages of the ontology in knowledge capture, storage, sharing and retrieval, the method can be expanded to the whole product design to manage and reuse a large number of different data related to each phase of the product life cycle, and support the decision making in product design from a broad perspective.

Based on the research purpose and objectives, primary research questions and research sub-questions are defined in Section 1.3.

1.3 Research Questions

Primary research questions, supported by sub-questions, provide the basis and outline for the work.

1.3.1 Primary Research Questions

How can the ontologies and ontology-based system help designers to capture, manage and reuse different data related to the early phases of the product life cycle, to make better decisions in product design, and to solve the significant limitations that exist in present computer-based support tools, where information required to make decisions may not be available, maybe lacking consistency and may not be expressed in a general way?

This topic has been studied in the context of DFM in this research and laid the foundation for a generalized and comprehensive framework for a broadly defined system to support design across lifecycle phases.

1.3.2 Research Sub-questions

(1) Questions on Ontology Development

Q1. What concepts, properties and constraints related to DFM, such as the product, cost, manufacturing requirements, etc. need to be structured in the ontologies? What ontologies need to be developed and what existing ontologies can be reused to represent these data in DFM?

Q2. Which ontology editor (such as Protégé, Oiled, Onto-Builder, etc.) should be used? How can the ontology be developed based on the chosen ontology editor? How can the concepts and properties in DFM and the complex relations among them be represented and utilized?

(2) Questions on Ontology Utilization

Q3. What mode is suitable for an ontology-based information system capable of integrating and utilizing data on cost, manufacturing process, product, material, etc. for DFM? How can the system guide users to the proper information?

Q4. Which criteria (such as cost, time and quality) and constraints (such as feasibility, flexibility) do product designers consider in DFM? What is the suitable working mechanism for an ontology-based decision support tool that can help designers consider the DFM problem step by step, compare the design alternatives and make decisions based on the criteria?

(3) Questions on Refinement and Evaluation of the Research Approach

Q5. How can errors in ontology development, maintenance and utilization be reduced? The utilization includes information integration and decision support tools.

Q6. How can the benefits to a DFM ontology and decision support tool be evaluated?

1.4 Research Approach and Methods

Three primary aspects of this research have been investigated. First, a generalized ontology development and utilization process is summarized. DFM ontology is proposed and developed. After classifying the concepts and properties in the field of DFM, the DFM ontology is developed with the relations realized in it by using Java Function Calls programs and SWRL (Semantic Web Rule Language). Other ontologies related to the second and third part of the research are also developed. Second, the means to integrate heterogeneous data resources is investigated. Third, based on the ontology and information integration, a decision support tool is designed and developed using JSP (JavaServer Pages) to help designers consider the design problem in a systematic way and make design decisions efficiently based on accurate and comprehensive data.

Errors that may occur and their causal factors are categorized, and an ontology-based root cause analysis method has been put forward and utilized to refine the ontologies and ontology-based system through CFSP (The NSF Center for Friction Stir Processing) and industrial partners (GM and Boeing). Currently CFSP and CED (the NSF Center for e-Design) work collaboratively on an NSF project where e-Design method and tools for friction stir welding and processing are being investigated.

1.5 Research Contributions

The research on ontology development, ontology-based information integration and ontology-based decision making method in DFM is expected to:

- Offer a formal specification of knowledge in DFM, which can facilitate the sharing of knowledge and collaboration between designers in a distributed and flexible organization;
- Enhance the accuracy and consistency of data used;

- Explore a suitable ontology-based information integration system to enhance the availability of data from heterogeneous data resources;

- Support decision making in product design by simultaneously considering the technical and economical constraints and relations and making tradeoffs among them;

- Enhance product design quality;

- Expedite the growth of ontology-based methods by exploring a management and control method for errors in the development, maintenance and utilization of ontologies;

- Provide an evaluation methodology for the ontology-based knowledge management and decision making method.

1.6 Outline of the Dissertation

As shown in Figure 1-1, following the first introductory chapter, the second chapter includes a review of literature related to product design, semantic web, ontology development, ontology-based information integration, multiple-attribute decision making in product design, error management and analysis, and related evaluation. Chapter 3 presents the research approach and methods. Chapter 4 is the realization of the Chapter 3, and Chapter 5 gives some case studies to illustrate research results. In Chapters 2-4, after the background and overview of the chapter, ontology development related to the chapter is first described, followed by ontology-based data integration, ontology-based decision support, then error control in the research and evaluation. Contributions and future work are summarized in Chapter 6.

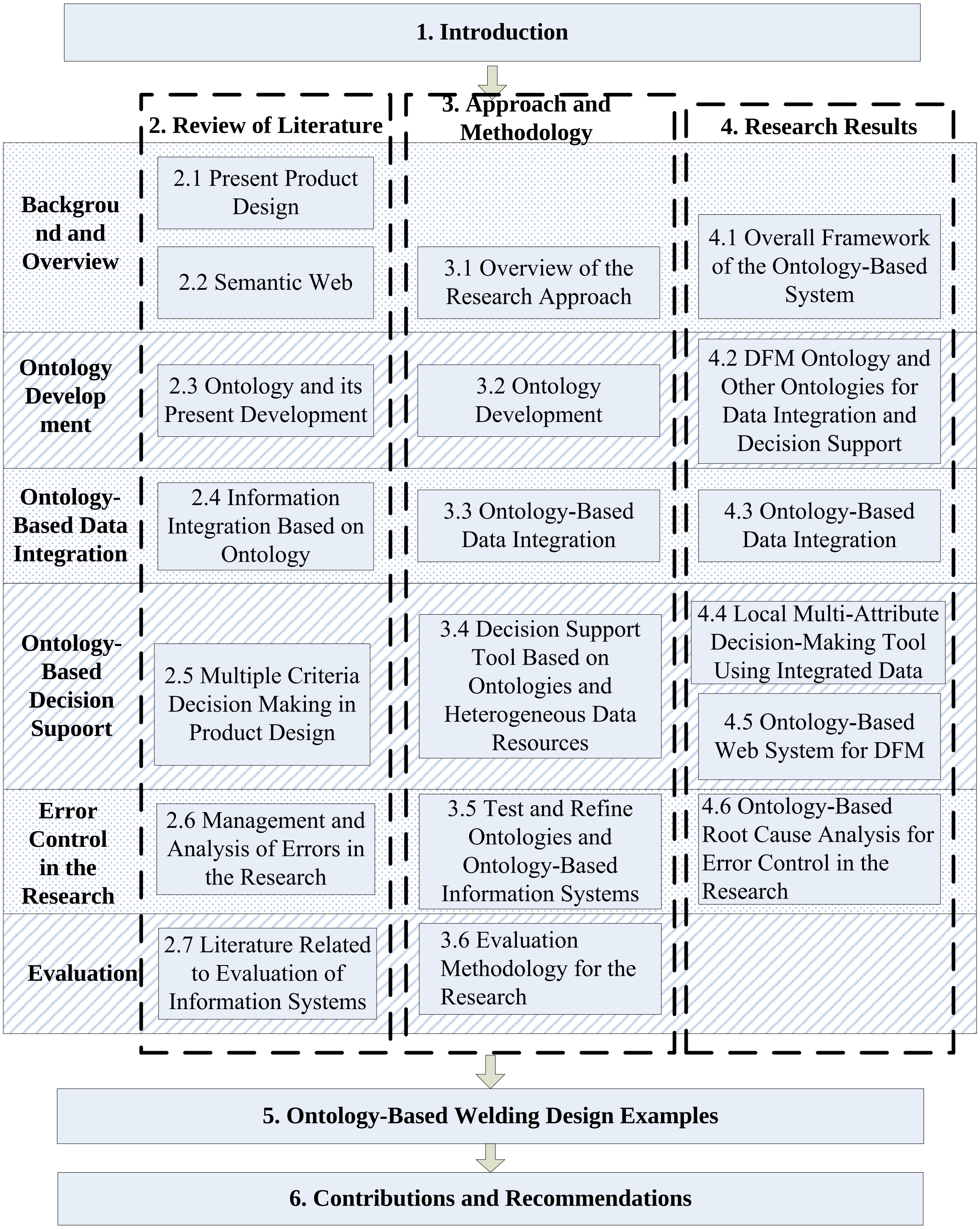


Figure 1-1: A pictorial overview of the dissertation

Chapter 2: Review of Literature

In this chapter, after considering the present development of product design, the idea of using ontology in knowledge representation, data retrieval and decision support tool in product design is put forward. Since ontology plays a key role in the semantic web, and some layers of metadata in the semantic web are also used in the research, the basic knowledge of semantic web is introduced. Ontology is the basis of the method, so the ontology editors are first compared in order to choose the most suitable one, and the previous related ontologies are analyzed in order to build the DFM (Design for Manufacturing) ontology. Based on ontologies, three kinds of ontology-based information integration approaches are compared and the hybrid approach is selected to be used in the research. After the integration of data, designers can utilize different criteria and analysis techniques to analyze the integrated data and make decisions in product design, so the analysis criteria and techniques to be used in the decision support tool for product design are summarized. In the development of ontology and ontology-based decision support tool, errors might happen. Types of errors and the collection and reason analysis methods for these errors are presented in order to reduce the effect of errors in the research as much as possible. To evaluate the research in the future, previous evaluation of information system are summarized at the end.

2.1 Present Product Design

In this section, needs in product design are introduced first. Concepts such as Concurrent Engineering (CE), CAX (Computer Aided X), DFX (Design for X), PLM (Product Lifecycle Management) and PDM (Product Data Management) are explained in detail. The original idea of the research is stated at the end.

2.1.1 Need in Product Design

Product design can influence a large portion of the product cost, so it is critical to obtain the cost estimates as early and quickly as possible during the design phase (Hyeon et al., 1993). It has been found that 80 percent of a product’s cost is committed during the design phase, whereas design only absorbs 8 percent of incurred costs (Pawar, Menon, and Riedel, 1994).

Besides the economical factors, other characteristics, such as manufacturability, feasibility, and time to market, also need to be considered in order to get the global optimal result. It is reported that 80% to 90% of the time to market is used in the product planning and development phase (Charney, 1991) and 40% of all the quality problems are caused by bad design (Saaksvuori and Immonen, 2005). If a company wants to shorten significantly the time to market and enhance the quality of its product, it should try to enhance the efficiency and quality of product design.

In Sullivian et al. (2005, p.9), the engineering design process is simplified into the following steps:

1) Problem/need definition, formulation and evaluation;

2) Synthesis of possible solutions (alternatives);

3) Analysis, optimization and evaluation, which include substeps, such as developing the outcomes and cash flows for each alternatives, selection of criteria and analysis and comparison of alternatives;

4) Decision making to select the preferred alternative;

5) Communication, such as performance monitoring and post-evaluation of results.

The decisions that need to be made in product design (step 4 above) involve information that comes from different sources and disciplines and are governed by multiple measures of merit and performance (Mistree et al., 1993). In the engineering process shown above, decision making is based on the first three steps. Designers always want to solve problems such as how to ensure the quality, consistency and availability of the data, how to retrieve the heterogeneous data resources, and how to analyze, optimize and evaluate the data retrieved in order to support the decision making in product design.

2.1.2 Concurrent Engineering

Product design needs to consider all its related manufacturing process in order to satisfy the corporate objectives through an optimized product design and reduce the cost caused by the changes in the later phases of the product lifecycle. Concurrent Engineering (CE) was developed as a viable approach to realize the simultaneous design of a product and all its related processes in a manufacturing system. CE is expected to reduce the work of change in the later phases of the product, such as preliminary design, manufacturing and product usage. CE is expected to not only reduce the research and development (R&D) cost and time, but also enhance the product design quality.

In concurrent engineering, decisions often need to be made in product design with respect to the market analysis, material procurement, product cost estimation, machining, assembly, inspection, service, maintenance and disposal (Hyeon et al., 1993). Design decisions made early in the product development cycle can have a significant effect on manufacturability, quality, product cost, production introduction time, and thus on the ultimate marketplace success of the product (Hyeon et al., 1993). Product designers must include manufacturing consideration as early as possible along with the structural, functional and aesthetic requirements (Burling et al., 1987).

Computer-based approaches are generally used in recent years to facilitate designers in reducing design time, decreasing product cost and enhancing product quality. The underlying prerequisite to the computer-based approach is systematically to acquire, represent, integrate, and coordinate the requisite concurrent engineering knowledge with which computers can perform required analyses (Hyeon et al., 1993). In these twenty years, several CAX (Computer Aided X) and DFX (Design for X) tools have been developed for this purpose.

Computer Aided Design (CAD) tools need to interact and cooperate to find a globally optimized or compromised design with the consideration of the product’s life cycle. A CAD framework is “a software infrastructure that supports the application environment, providing a common operating environment for CAD tools (Graham, 1991).”

Design for Manufacturability (DFM) is developed with the expectation that it will address the time-to-market, quality and cost issue, and a Design for Assembly (DFA) system considers the ease of handling and assembling of component parts and the number of parts used in the product (Hyeon et al., 1993). All DFM systems share a common goal, which is to minimize the total product life-cycle costs (O' Flynn and Ahmad, 1993) through more systematic and efficient decision making.

The data produced by existing information systems, such as CAD and DFM systems, are already in electronic formats. But all the information required to make a decision may not be available, may lack consistency and may not be expressed in a general way. The available data resources are often isolated and inconsistent, which makes it hard to reuse them in a systematic way. According to Saaksvuori and Immonen (2005), it is difficult to allocate the information in the scattered information islands to purchase orders, work orders, production lots and individual products or vice versa. In practice, the worst problems are often caused by different modes of operation, the wide spectrum of differences in software, and the numerous interfaces between information systems (Saaksvuori and Immonen, 2005).

According to Coopers & Lybrand study from the year 1994, 14% of an engineer's time is spent on meetings, mainly to share information, 21% on the work that somebody has done earlier, and 24% on information sharing and retrieval (Saaksvuori and Immonen, 2005). Only 29% of his/her time is spent on actual engineering work. That means 59% of the time can be saved if the information system can help engineers share, retrieve and reuse the information efficiently. PDM and PLM systems are aimed at helping users allocate data and facilitate design.

PDM is "a systematic, directed method by which to manage and develop an industrially manufactured product (Saaksvuori and Immonen, 2005, p.243)." With the help of PDM, the product process can be well managed and all product data can be handled in a systematic manner. Today, the term PDM can also be considered as an older name for PLM (Saaksvuori and Immonen, 2005).

PLM is a systematic, controlled concept for managing and developing products and product related information. It helps connect separate information data systems, process and automation islets. Both PDM and PLM are information systems developed to manage product lifecycle and product related data (Saaksvuori and Immonen, 2005). But PLM has a wider frame of reference than PDM, especially to the life cycle perspective of information management (Saaksvuori and Immonen, 2005).

However, since PLM needs to support different databases, the use of several different database types brings a greater challenge to the PLM system and naturally to the integration and data transfer between different systems. Ontology is good at not only representing the concepts and relations, but also retrieving data through the relations. Relations can be hierarchical relations between classes, subclasses and superclasses, the slots that connect multiple classes, the facets (properties of slots) and hierarchy of slots, or the axioms and rules between classes, slots and instances. Ontology is expected to help represent, manage and utilize knowledge in the PLM system, consider different criteria and relations in the decision making, and enhance the quality and efficiency of product design.

In the research, ontology and semantic relations in the ontology have been used to integrate design data in DFM and help designers share, manage, retrieve and reuse data (related to welding cost, welding material, machine, etc.), retrieve and compare feasible design alternatives, and support decision making in product design. It is expected to reduce the time and cost of product design, and enhance the quality of design.

In the rest of Chapter 2, basic principles, characteristics and present development of semantic web and ontologies are first introduced. Utilizing ontologies and the semantic relations in ontologies, information integration methods are described. Based on the integrated information, a decision support tool is built. So the criteria, classification for the structured decision and multi-attribute decision making methods are summarized. Types of errors that might happen in the research, as well as error collection and reason analysis methods, which might be used in refining the ontology and the decision support tool, are explained. Finally, the evaluation used in the related systems are summarized.

2.2 Semantic Web

The semantic web is essentially about machine-understandable web pages. Berners-Lee et al. (2001) described the semantic web as a web that can understand and interpret web pages and manage activities (Bhavani et al., 2002). A model is promised to be used in order to connect different information sources such as web pages, databases and even flow of daily life data (Tjoa1 et al., 2005).

Metadata are data about data (Aufaure et al., 2006), such as structure and relationship. The semantic web is composed of several layers of metadata and comprises the standards, such as XML (eXtensible Markup Language), XML Schema, RDF (Resource Definition Framework), RDF Schema (RDFS) and OWL (Web Ontology Language). McGuinness and van Harmelen (2004) explained the function and relationship of each component in the semantic web from bottom to the top:

- URI (Uniform Resource Identifier) identifies resources on the web and uses a global naming convention. The primary facilitators of the semantic web are URIs along with XML and namespaces.

- XML (EXtensible Markup Language) provides a surface syntax for structured documents, but imposes no semantic constraints on the meaning of these documents.

- XML Schema is a language for restricting the structure of XML documents and also extends XML with data types.

- RDF (Resource Definition Framework) is a data model for objects and relations between them. It provides simple semantics for this data model in XML syntax. RDF triple describes a subject, a predicate and an object. The predicate is also known as the property of the triple (Klyne and Carroll, 2004).

- RDF Schema (RDFS) is used to describe properties and classes of RDF resources, with a semantics for generalization-hierarchies of such properties and classes. RDFS makes semantic information machine accessible (Stuckenschmidt and van Harmelen, 2005).

- OWL (OWL Web Ontology Language) adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. exactly one), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes. Compared with RDF, RDFS, DAML+OIL, OWL has more facilities for expressing meaning and semantics, and is better at the representation of machine-readable content on the Web (Kim, 2005).

Web identifier (called URI) is used to identify things for RDF (Aufaure et al., 2006) and OWL. XML Schema constrains the structure of XML documents, and RDFS is used to create vocabularies that describe groups of related RDF resources and the relationships between those resources. OWL is built upon W3C standards XML, RDF and RDFS, and extends them to express class properties (Brickley and Guha., 2003; Lassila and Swick, 1999). It incorporates lessons learned from the design and application of DAML+OIL (DARPA Agent Markup Language, Ontology Inference Layer) and is a revision of DAML+OIL (Horrocks, 2002).

The World Wide Web Consortium (W3C) has approved RDF and OWL, provided a solid base to establish enterprise semantic applications and implied a significant leverage of the semantic web from a research level to an industry standard for building next generation applications (Tjoa1 et al., 2005).

2.3 Ontology and its Present Development

Ontology plays a pivotal role in the development of semantic web, since it not only provides a representation of a shared conceptualization of a particular domain that can be communicated between people and applications, but also captures knowledge about concepts in the domain and relationships among these concepts (Horridge et al., 2004). Relations can be hierarchical relations among classes, subclasses and superclasses, the slots that connect multiple classes, the facets and hierarchy of slots, or the axioms and rules between classes, slots and instances. Appendix A of this dissertation introduces the basic knowledge of ontology and includes a simple example of an ontology and ontology-based system. Ontology is more expressive than RDF and allows a better specification of constraints on classes. It helps realize reasoning methods (forward chaining and backward chaining). The aim of ontologies is “to formalize domain knowledge in a generic way and provide a common agreed understanding of a domain, which may be used and shared by applications and groups (Aufaure et al., 2006, p.267).” Ontologies (Gomez-Perez et al., 2003) play a key role in information retrieval from Internet and heterogeneous data sources.

In practice, developing an ontology includes defining classes in the ontology, arranging classes in taxonomic hierarchy (classes, subclasses and super classes), defining slots (attributes of classes or relations between classes) and specifying allowed values for these slots, creating instances of classes and inputting/editing values of their slots (Noy and McGuinness, 2001). There are seven steps in ontology development, including: 1)Determine the domain and scope of the ontology, 2) Consider reusing existing ontologies, 3) Enumerate important terms in the ontology, 4) Define the classes and the class hierarchy, 5) Define the properties of classes’ slots, 6) Define the facets of the slots, 7) Create instances (Noy and McGuinness, 2001).

One of the most important advantages of ontology is that it is good at the representation and utilization of relations. Relations can be the hierarchical relations between classes, subclasses and superclasses, the slots that connect multiple classes, the facets and hierarchy of slots, or the axioms and rules between classes, slots and instances.

In the research, ontologies have been developed to capture the concepts and relations in the welding process and provide the basis for information integration and the decision support tool. In this section, reasons for using ontologies, present suitable ontology editors, and present ontologies for product design will be summarized and explained in detail.

2.3.1 Reasons of Using Ontologies

According to Noy and McGuinness’ ontology development 101 (2001), reasons for developing an ontology are:

- “To share common understanding of the structure of information (p.1),” which includes concepts and their relationship, among people or software agents;

- “To enable reuse of domain knowledge (p.1);”
- “To make domain assumptions explicit (p.1);”
- “To separate domain knowledge from the operational knowledge (p.1);”
- “To analyze domain knowledge (p.1).”

Moreover, the critical advantage is that it aids reasoning. Reasoning can be useful at many stages during the design, maintenance and deployment of ontologies. An ontology can be proved to be consistent because autonomous software agents can perform their reasoning and come to logical conclusions without human supervision (Dong et al., 2002).

Ontology and semantic web can be used to capture, store, and retrieve a large amount of information related to product design, to facilitate sharing of the common understanding in the field among different types of engineers during product design, and to support the consideration of complex relations among data in decision making of product design.

2.3.2 Ontology Editors

Ontology is the basis of the research, and it should be built through an ontology editor. In Lambrix et al. (2003), four ontology editors are compared, which are Protégé-2000, Chimaera, DAG-Edit and OilEd, and it was found that all four systems have their strengths and weaknesses. The main strengths of Protégé-2000 compared to the other systems are its user interface, the extendibility using plug-ins, the functionality that the plug-ins provide (such as merging) as well as the different formats that can be imported and exported. Protégé-2000 is an old version of Protégé, and up till now, the latest version of Protégé also holds the advantages of the other three ontology editors in Lambrix et al. (2003). Moreover, from an ontology editor survey results (Denny, 2004), 94 ontology editors are compared in terms of 16 characteristics, such as release data, base language, web support, import/export format and other characteristics. Among these editors, Protégé has more advantages than others.

Protégé is a free, open source ontology editor and a knowledge acquisition system. It supports ontology developers to think about domain models at a conceptual level without having to know the syntax of the language ultimately used on the Web (Noy et al., 2001). It can develop ontology in its own format, and can import or export ontology in RDF, RDFS, DAML+OIL, XML, OWL, Clips and UML. It can browse classes and properties via plug-ins (OntoViz, TGViz) and its query tab allows searching. There are many plug-ins available for extending ontology construction, constraint axiom, inferring and integration functions. Ontologies in the research have been built through Protégé to capture and represent concepts, their relationship, and instances in product design.

2.3.3 Some Present Ontologies related to Product Design

At present, more and more ontologies are built for product design with the consideration of manufacturing process, cost, geometry relation between parts, etc. Kim et al. (2006) proposed a collaborative assembly design framework that offers a shared conceptualization of assembly design modeling. The proposed framework encapsulates assembly knowledge about a product, and has the flexibility to represent assembly constraints and spatial relationships among features, explicitly in a computer-interpretable manner. By utilizing the assembly constraints captured in the Assembly Design (AsD) ontology (Kim et al., 2006), sharing of assembly information along with the assembly geometry can be realized at the same time. However, the AsD ontology focuses on the geometric relations among parts and lacks other information such as manufacturing process and cost, which is emphasized in the TIE project (Terpenny, 2006). The background of FSW and TIE project is explained in greater detail in Appendix B. Based on the AsD ontology, the DFM ontology has been developed in the dissertation in order to facilitate manufacturing process design with the consideration of cost, feasibility, availability, etc.

Besides the manufacturing information, cost information for a product family can be represented by the relations between the specifications of the product family and its activity cost (Park and Simpson, 2005). Activity-based costing (ABC) identifies the costs of activities, such as receiving raw material, traces the activities to particular product or customer that use the activity, and assigns cost to products or customers based on resources that they consume (O’Guinn, 1991). The ABC ontology (Park et al., 2006) represents ABC processes, formalizes concepts to describe different types of costs, and classifies the design features that affect costs for a particular product family domain. The ABC ontology (Park et al., 2006) addresses activities and their relations with resources and costs, and provides a more holistic picture of costing process.

Moreover, Fernandes et al. (2007) created people ontology, project ontology, functional basis ontology and design ontology, and used SWRL and the Jess rule engine to generate design recommendation.

The purpose of using these ontologies is to represent knowledge in product design, and help designers find useful knowledge more easily through the semantic relations in the ontologies. At present, these ontologies are designed for some special purposes, and some of them are not built up from a comprehensive view. The classes, properties, and relations need to be refined, and information in these ontologies should be integrated to ensure the consistency and enhance the efficiency for DFM. Semantic web technologies in ontology integration and collaboration play a key role in integrating, retrieving and utilizing these distributed heterogeneous data in product design, which was explained in the next section.

2.4 Information Integration Based on Ontology

Ontology and semantic web can be used to capture and represent the common understanding in the field among different types of engineers during product design, store and retrieve a large amount of concepts and relations in product design, and provide proper and comprehensive semantic information for product design by utilizing the semantic relationships in the ontologies. In the following sections, ontologies integration and collaboration (Section 2.4.1) and ontology-based information integration (Section 2.4.2) are explained in detail. The integrated data in this section is used to support decision making in product design (Section 2.5) in order to enhance the efficiency and quality of designers’ work.

2.4.1 Integration and Collaboration of Ontologies

The success of semantic web is based on the expansion of ontologies. With the proliferation of ontologies, many personalized and small-scale ontologies appear. With the development of several ontologies which represent the corresponding fields and have some intersection with each other, reconciling multiple ontologies is the only way to access the available information on the web (Aufaure et al., 2006). Relevant methods used in integration of ontologies include merging, integration, alignment and mapping, as explained below. The ontology integration method used depends on the structure, concepts and semantic relations of the ontologies.

- **Ontology merging** aims at creating a new ontology from several ontologies. The new ontology contains all the information from different sources.

- **Ontology integration** creates a new ontology containing only parts of the source ontologies.

- **Ontology alignment** makes several ontologies consistent through a mutual agreement.

- **Ontology mapping** defines equivalence relations between similar concepts or relations from different ontologies. (Aufaure et al., 2006, p.276).

To realize the ontology integration, reuse, and other ontology management activities, there are several systems developed, such as Chimaera, an ontology-merging tool, McGuinness et al.(2000), ONION, an ontology-mapping tool, Mitra et al. (2000), and OntoView, an ontology-versioning tool, Klein et al. (2002). These tools accomplish a variety of tasks independently, but cannot meet the complex requirements required several different tasks, such as merging, aligning, versioning, etc. Noy and Musen (2004) described an ontology management framework to manage multiple ontologies by leveraging data and algorithms developed for one tool in another. It includes PROMPTDIFF to find a structural difference and iPROMT to find similarities and differences between different ontologies from different sources (Noy and Musen, 2004). These systems are not included in the following chapters in the dissertation. In future research, the ontology management framework by Noy and Musen (2004) might be used to integrate and manage ontologies.

2.4.2 Information Integration Methods

The heterogeneity in the way of structuring and interpreting information causes conflicts, and makes it difficult to retrieve information from different sources. Wache (2003) summarized those conflicts into structural conflicts and semantic conflicts. Structural conflicts arise because the same objects and facts in the domain can be described in different ways. For example, engineer A puts product geometric and material information in the same table in one database, and engineer B puts these data in separate tables in another database. Semantic conflicts occur when two systems do not use the same interpretation of information. For instance, designer A uses parts and designer B uses components to represent the items that form the product.

There are three ways of using ontologies in integration: single-ontology approaches, multiple-ontology approaches and hybrid approaches (Stuckenschmidt and van Harmelen, 2005), as explained below.

In single-ontology approaches, a global ontology is used in order to provide a shared vocabulary for the specification of the semantics. In multiple-ontology approaches, each part of data is described by its local ontology. In hybrid approaches, a shared vocabulary (central ontology) is connected with each local ontology that correspond to each part of data. It can be used to conquer the shortcomings of the first two kinds of ontology-based approaches, but it is only suited to complex problems. Referring to the comparison in (Stuckenschmidt and van Harmelen, 2005), Table 2-1 summarizes some benefits and drawbacks of these three methods.

Table 2-1: Comparison of three ontology-based information integration methods

	Multiple-ontology approaches	**Single-ontology approaches**	**Hybrid approaches**
Implementation effort	Might be costly since it needs some repeated work.	Straight forward (shared data do not need to be rebuilt).	Reasonable
Semantic heterogeneity	Supports heterogeneous views.	Similar views of a domain.	Supports heterogeneous views.
Adding a source	Providing a new local ontology and relating it to all other local ontologies.	Needs adaptation in the ontology	Providing a new local ontology and relating it to the shared vocabulary.
Comparing multiple ontologies	Difficult because of the lack of a common vocabulary.	---	Simple because ontologies use a shared vocabulary.

Currently, data in product design are in heterogeneous formats. Most are saved in databases, while others are in other formats. Information conflicts (Wache, 2003) are obstacles to information integration in product design. The hybrid method can be utilized in the research in order to retrieve appropriate heterogeneous data from different resources for product design. For example, there are heterogeneous data resources that record data about product items, activities, materials and machines respectively. Each local ontology represents concepts and relationships for each data resource, and the DFM ontology can be built and used as the shared vocabulary (the central ontology). The hybrid approach can utilize the relationship in the central ontology and local ontologies to retrieve proper data in these distributed data resources in order to facilitate designers in product design.

Several semantic data schema integration tools have been developed for information integration, such as COG/Unicorn, PROMPT, OBSERVER, etc. Their characters have been compared by Alexiev et al. (2005). Unicorn Workbench is a two-layered approach for integrating disparate heterogeneous data resources. Its external assets layer contains the mappings to all the disparate data sources, and its model layer contains the central ontology (Arens et al., 1996). The central ontology has rich semantics to integrate the information assets. The central ontology and the mapping jointly offer information about where the data are stored. A query can be translated by Unicorn Workbench into correct

SQL-like queries based on the central ontology. After the user formulates a query requirement in terms of the elements in the ontology, the tool identifies the data resource that needs to be queried, reformulates the global query into proper query sentences for the data resource, collects and combines the query results and returns the result. However, according to Alexiev et al. (2005), there are problems in the Unicorn Workbench. Unicorn Workbench is still not able to query the data resources by itself. There is no query execution engine in Unicorn Workbench. It is also impossible to use a single query for multiple databases or automatically retrieve the results of a query from a data source. In the process of design, designers need guidance to locate the proper items, and then get the related details, but Unicorn Workbench cannot achieve the same function. For example, at the early stages of product family design, cost information needs to be retrieved in order to estimate the cost effect of developing product families. When a designer wants to reduce the cost of a product, he/she needs to drill down the data from feature-based task to operation activity cost, then to resource pool, and at last to resource cost. When the real factor of high cost is located, the designer needs related information about the factor from different resources, such as databases belong to the inventory, to the manufacturing process, etc. The ontology can be used here to help locate the useful information, such as the factors of high cost, and then the method in the Unicorn platform can be used to query the details from heterogeneous data resources.

Moreover, there has been some research in ontology-based methods for information retrieval. Hwang et al.(2006) designed the architecture of search systems based on the ontology using web services. Since ontology can provide the inferred and associated information between data, ontologies in the system are exploited by different semantic web applications to provide information to search clients. In order to search the increasing medical text documents, a medical information system with a semantic search function was developed using a medical ontology that represents medical terminology semantic structure (Taki et al., 2003). Vasilecas and Bugaite (2006) put forward a method for ontology transformation into business rules, which are implemented by information-processing rules. In Vasilecas and Bugaite (2006), the ontology axioms can be used to create a set of information-processing rules. They can be transformed into ECA rules and then to active DBMS triggers. Stojanovic N. and Stojanovic L. (2004)

presented a logic-based approach for query refinement in ontology-based information portal. The query refinements are ranked according to their relevance to user's needs. Kim (2005) designed and implemented an ontology-based web retrieval (ONTOWEB) system. In the ONTOWEB system, the ontology-based search engine is used to query the information that has been loaded into the database. This study showed that ontologies can be used not only to improve precision, but also to reduce the search time. However, none of these works focus on how to guide users to the proper information especially when they are not very familiar with the process or domain, and when corresponding detailed data from different data resources belongs to multiple departments.

In this research, a framework using the hybrid approach in ontology-based integration is built. The framework can help designers utilize the ontology to first locate the proper factor, then the framework can apply the method in Unicorn Workbench to retrieve rich data from the data resources, and finally analyze the integrated result. In Section 3.3.3, the structure of the framework is explained in detail.

2.5 Multiple Criteria Decision Making (MCDM) in Product Design

According to Sen and Yang (1998), MCDM can be broadly classified into multiple attribute decision making (MADM) and multiple objective decision making (MODM). All decision problems can be classified into one of these two broad classes. When there is a set of alternatives to choose from, the choice is most conveniently made by comparing the prioritized attributes or weighted value of the alternatives(MADM). If an attribute has direction, it is an objective. MODM synthesizes an alternative or alternatives on the basis of prioritized objectives.

After data is integrated and retrieved by using the ontology, the decision support tool should utilize some analysis methods to compare the multiple design alternatives that meet the requirements according to some criteria. A criterion can be any measure of performance for an alternative. In this section, criteria choosing methods, classification for the structured decision and design alternative analysis methods that might be used in MADM in the decision support tool are introduced.

2.5.1 Choice of Criteria and Measurement Scale

In multi-attribute decision making, in order to choose the design alternatives that meet the requirements, questions such as how the criteria (attributes) are chosen and what analysis techniques are required (a rank ordering of alternatives, for instance) need to be answered first.

According to Canada et al. (2005), "each attribute should distinguish at least two alternatives, and capture a unique dimension or facet of the decision problem, i.e., attributes are independent and non-redundant (p.546)." Differences in value to each attribute are assumed to be meaningful and distinguishing among feasible alternatives, and all attributes together should be sufficient for selecting the best alternative.

The number of criteria should also be considered. Too many criteria can increase the difficulty to manage the analysis, while too few attributes can limit the discrimination among alternatives (Canada et al., 2005).

2.5.2 Classification to Form the Tree Structure for Decision Making

Classification is basically one of the methods for partitioning the feature space into regions, one region for each category of input. It assigns every data point in the entire feature space to one of the possible classes (Bandyopadhyay et al., 2005).

Flat classification and hierarchical, tree-structured classification are described and compared in Freitas and Carvalho (2007). In the flat classification, all classes are at the same level, and in the hierarchical classification, classes are at different levels, which means some classes may have subclasses. Hierarchical classification offers more flexibility to the specification of the level of a class to an example, and it is quite well suited for the hierarchy of the ontology. The hierarchical classification can be transformed to the flat classification by using one flat classification problem for each level of the class hierarchy and using a flat classification algorithm to solve each of these problems independently. And there are several hierarchical classification methods, such as big-bang and top-down hierarchical classification (Freitas and Carvalho, 2007).

Further, there are two main types of structure for a class hierarchy, tree structure and a direct acyclic graph (DAG) structure (Freitas and Carvalho, 2007). In the tree structure, each class node has at most one parent, and in a DAG structure, each class node can have more than one parent. Considering the character of decision making in product design and the decision making in the TIE project, the tree structure is used.

2.5.3 Analysis Techniques

After getting the abundant alternatives, analysis methods in this section are used in the decision support tool in order to help designers compare alternatives and make decisions.

There are two ways to deal with data. One is single-dimensioned analysis, and the other is full-dimensioned analysis (Canada et al., 2005; Sullivan et al., 2005). The dimension here is the number of metrics used to represent the attributes that discriminate among alternatives. According to Canada et al. (2005) and Sullivan et al. (2005), the comparison of these two analysis methods is shown in Table 2-2.

Table 2-2: Comparison of single and full dimensioned analysis

Analysis method	Single-dimensioned analysis	Full-dimensioned analysis
Concepts	➢ Collapses all information to a single dimension. ➢ Changes in the value of an attribute can be offset by, or trade off against, opposing changes in another attribute.	➢ Does not collapse attributes to a common scale ➢ Trade-offs among attributes are not allowed.
Model	Compensatory models ➢ Nondimensional scaling, ➢ Hurwicz procedure, ➢ Additive weighting technique	Noncompensatory models ➢ Dorminance, ➢ Satisfying, ➢ Disjunctive resolution, Lexicography
Methods	➢ Ordinal scaling, ➢ Weighted evaluation of alternatives, ➢ Analytic hierarchy process.	Alternatives attributes score card.

2.5.3.1 Single-Dimensioned Analysis

In single-dimensioned analysis, all information is collapsed into a single dimension. This is popular in practice with compensatory models corresponding to this kind of analysis (Sullivan et al., 2005).

In single-dimensioned analysis, all attributes must be converted to a common measurement scale first. The converted attribute values are weighted according to the weights provided by users, and are then summed together. The assumption is that good performance in one attribute can compensate for the poor performance in another.

I. Conversion to a common measurement scale

In Sullivan (2005, p.591), the nondimensional scaling is calculated after considering whether or not large numerical values are desirable. When a large numerical value is undesirable, the rating is

$$\text{Rating}=\frac{\text{worst outcome-outcome being made dimensionless}}{\text{worst outcome-best outcome}} \tag{1}$$

When the large numerical value is desirable, the rating is

$$\text{Rating}=\frac{\text{outcome being made dimensionless-worst outcome}}{\text{best outcome-worst outcome}} \tag{2}$$

II. Weight values

Weights represent the relative importance of the criteria in the overall consideration. Users can input the weight or the relative rank to the decision support tool. Relative rank = ordinal ranking+1. Value 1 represents the most unimportant attribute. Relative rank is used in the additive weighting technique, which is probably the most popular single-dimensioned method (Sullivan et al., 2005).

By utilizing the additive weighting technique (Sullivan et al., 2005), the normalized weight can be used to calculate the weighted sum.

$$\text{Normalized weight of the attribute} = \frac{\text{relative rank of the attribute}}{\text{sum of the relative ranks}} \tag{3}$$

Moreover, there are several other methods to normalize weight value in Canada et al. (2005). For uniform or equal weights: $W_i = 1/N \times 100\%$, for rank sum weights, $W_i = \frac{N - R_i + 1}{\sum_{i=1}^{N} N - R_i + 1} \times 100\%$, and for rank reciprocal weights, $W_i = \frac{1/R_i}{\sum_{i=1}^{N} (1/R_i)} \times 100\%$, where N is the number of attributes, and R_i is the rank position of attribute i, and W_i is the weight for attribute i.

III. Weighted sum and value index

There are several ways to find the best alternative. The most commonly used method is to calculate the weighted sum of all the attributes in consideration, and choose the alternative with the biggest value.

The Hurwicz procedure (Sullivan et al., 2005) is another method to reach an intermediate level between the pessimism of maximin and the optimism of the maximax. As shown in the following expression,

$$\begin{aligned}&\text{weighted sum for each alternative}\\&= \alpha(\text{value of best attribute})+(1-\alpha)(\text{value of worst attribute})\end{aligned} \tag{4}$$

where α is between 0 and 1 and reflects the intermediated attitudes, and the values of best and worst attributes are the dimensionless rating calculated by Formulas (1) and (2).

Besides the weighted sum, value index (Sullivan et al., 2005) can also be used to help designers choose the best design alternatives.

$$\text{Value Index} = \frac{\text{Worth}}{\text{Cost Factor}} \tag{5}$$

$$\text{Cost Factor} = \frac{\text{the maximum allowable cost-cost extimate of the alternative}}{\text{the maximum allowable cost-the minimum attainable cost}} \times 100 \tag{6}$$

Worth is the weighted value from the additive weighting technique, as shown in Formula (3). The design alternative with the biggest value index will be chosen.

2.5.3.2 Full-Dimensioned Analysis

In full-dimensioned analysis, there is no attempt to collapse attributes into a common scale. In Sullivan et al. (2005), there are several noncompensatory models, which are dominance, satisficing, disjunctive resolution and lexicography. In dominance, one alternative is selected as the best if it is better than any other alternative with respect to all attributes. It is useful to screen inferior alternatives from the analysis. Satisficing model requires alternatives that are included in further consideration to have acceptable values for each attribute. Compared to satisficing model, disjunctive resolution evaluates each alternative on the best value achieved for any attribute. In lexicography, the alternative having the highest value for the most important attribute is chosen. Moreover, in Canada et al. (2005), attribute score card and ordinal scaling are described, which also belong to full-dimensioned analysis. In the decision support tool in the research, the method for full-dimensioned analysis is similar to the lexicography model.

2.5.4 Ontology-based Decision Support Systems

The focus of this research is on utilizing the advantages of ontologies in knowledge representation and reasoning for product design. There has been some work on ontology-based decision support systems by other researchers.

Grosse et al. (2005) used ontologies and developed a prototype engineering analysis modeling knowledge base ON-TEAM to facilitate knowledge exchange and improve reuse, adaptability, and interoperability of analysis models for supporting engineering analysis models. In Witherell et al. (2007), an optimization ontology is used in a prototype knowledge based tool for knowledge sharing and exchange in engineering design optimization.

There also has been some work on ontology-based decision support systems other than engineering design. For example, OntoWEDSS is a decision support system for the management of wastewater treatment plants, which augments classic rule-based reasoning and case-based reasoning with a domain ontology. OntoWEDSS makes use of the WaWO ontology in domain modeling and for the clarification and reduction in confusion found in existing terminology in the wastewater domain (Ceccaroni, 2001; Ceccaroni et al., 2002). In Li et al. (2001a) and Li et al. (2001b), goal ontology is used with first order logic reasoning to provide the capability of describing and reasoning the relationship of goal decomposition, changeability, redundancy and consistency for partner selection of virtual enterprises. In Loaiza and Wartik (2005), multiple agents process the basic information from the knowledge base of ontologies and provide automated decision support to accelerate decision making and improve joint war fighting.

These systems mainly use ontologies to represent and exchange knowledge, reduce confusion, and provide the knowledge base for decision support functions in specific areas. In this research, in order to facilitate systems thinking and enhance the data availability and quality in product design, an ontology-based method utilizes ontologies as well as an ontology-based rule engine to provide decision support for DFM in the following chapters.

2.6 Management and Analysis of Errors in the Research

In the ontology development and maintenance, as well as the ontology-based applications, errors might happen. For example, the structure of the ontology is not consistent with the categorization of the field, the data input into the ontology is inaccurate, or the computer programs for the ontology-based decision support tool have logical bugs. In this section, types of errors that might occur in ontology development and utilization are summarized, and their collection and reason analysis methods are described.

2.6.1 Types of Errors related to Ontology

In ontology development and utilization in product design, different kinds of errors from different views can be encountered. In this section, from the views of experimental science, engineering, computer programming, and numerical analysis, different types of errors that are related to ontology development and utilization are described along with possible solutions. Knowing the type of a specific error is helpful in reason analysis and solution development.

2.6.1.1 Experimental Science

In experimental science, errors are classified into two types: statistical error and systematic error. Many things can lead to statistical errors, such as errors of measurement, analysis and sampling (Box et al., 2005). The differences of these two types of error are summarized in Table 2-3.

Table 2-3: Difference of statistical error and systematic error

Statistical error	**Systematic error**
“Caused by random (and therefore inherently unpredictable) fluctuations in the measurement apparatus (Box et al., 2005, p.17).”	Caused by unknown but nonrandom fluctuations. Some systematic error can be removed if the cause of the systematic error can be found.
Varies randomly from observation to observation, due to factors which are not controllable or not controlled.	Always occurs (with the same value) when we use the instrument in the same way, and with statistical bias.
Causal factor cannot be analyzed (Meister, 1989).	Causal factor can be analyzed (Meister, 1989).
May be too expensive or may be too ignorant of these factors to control them each time they are measured (Box et al., 2005).	The effect of systematic errors usually can be reduced or even eliminated (Box et al., 2005). But not all the systematic error can be eliminated in this way, and some of them will remain (Rabinovich, 2005).

Statistical errors and systematic errors can exist in the slot value of an ontology. One purpose of using ontology and an ontology-based system for product design is to offer designers proper and correct data in order to enhance the quality of design. Incorrect data in the ontology can largely influence this purpose.

Currently, there are two common methods to mitigate the influence of experimental errors: Randomization and Blocking (Gad, 1999). If the causes of systematic errors can be identified, systematic errors that are observed and estimated will be eliminated from measurement through correction, such as adequate experimental design and analysis.

2.6.1.2 Engineering

In engineering, two common errors are involved. One is human error, which comes from the involvement of human beings, and the other is design-facilitated errors. Human errors exist in ontology development and in ontology and data resources integrated, while the design-facilitated errors happen in the ontology-based information system. For example, in ontology development, the wrong data type for a slot belongs to the human error. And the working process of the ontology-based information system may not prevent a user's mistake, which can be a design-facilitated error. Human errors can cause several bad results, such as wrong output to users, and the design-facilitated errors can lead users to the wrong usage of the information system or can lead to the wrong design result.

Human error, to some extent, is related to the cognitive bias of human beings. A human error is defined as a discrepancy between the operator's actual performance and the performance desired of him (Meister, 1989). Depending on the degree of human involvement in the system, the human component is responsible for 20%-90% of the failures in many systems (Lee et al., 1988, p.24). This type of error is often minimized by applying human factors engineering to designs through making systems more forgiving or error-tolerant.

Design-facilitated errors (DFEs), another type of engineering error, can be prevented by refining procedures and equipment to compensate for innate human error tendencies. DFE should be solved if possible, since failures to account for DFEs in the design process can be expensive. DFEs can be predicted during the design phase by utilizing the following methods, listed with their advantages and disadvantages (Freed, 1996):

- Analytical methods: inexpensive but not effective for complex tasks.
- Empirical methods: effective but potentially very expensive.
- Computer simulation: inexpensive; retains some of the effectiveness of empirical methods.

2.6.1.3 Computer Programming

In computer programming, an error may be a piece of incorrectly written program code. It can be a syntax error or a logic error. An error may also be an exception, which arises during program execution due to an unexpected event. Ontologies are developed using ontology editors, such as Protégé. So errors in this view are mainly in the ontology usage, not its development. That means errors in computer programming are in the programs of the information system supported by ontologies. Since the information system is closely connected with users, errors in this view will give users unexpected output and negatively influence design efficiency and quality. Developers should use some methods, such as improving the working process or writing related code, to handle them.

2.6.1.4 Numerical Analysis

Errors related to numerical analysis include approximation error, relative error, absolute error and percent error. They may exist in the slot value of an ontology, and wrong slot value can directly influence the accuracy of design result. The approximation error is in the mathematical subfield of numerical analysis. According to Oden and Prudhomme (2002), approximation error is “due to the inherent inaccuracies incurred in the discretization of mathematical models of the events (p.496).” It is the discrepancy between an exact value and some approximation to it. Relative error, absolute error and percent error can be used to describe how large the approximation error is.

2.6.2 Error Collection

In Meister (1989), several ways for error and success/failure data collection are described. The definition and advantages/disadvantages of each way are summarized in Table 2-4 below. All these methods can be used in the collection of errors in ontology development and utilization for product design.

Table 2-4: Summary of error collection methods

Methods	Definition	Advantages	Disadvantages
Manual collection in the real world	Has a human data collector who observes and records certain events (Meister, 1989).	Human controllable (Consistent with the following error analysis)	● Expensive because data collectors should be physically present during task performance and deal with a range of situations. ● Data collectors may fail to recognize errors .
Automatic data collection	Develops instrumentation that automatically records operator actions and other events (Meister, 1989).	Weaknesses of the manual method are overcome High efficiency	● Effective only with physical responses. ● Almost impossible to record incorrect perceptions of visual stimuli.
Self -report	"The person who made the error reports the error (Meister, 1989, p.22.2.2)."		Even if the report is anonymous, few operators will report an error voluntarily.
Experimental studies	An experimental study that compiles error data (Meister, 1989).	Less expensive compared to the real design environment.	Rarely been done because ● Most error data result from studies investigating other variables and one is at the mercy of the original study. ● The return on investment is very small.
Expert judgment	Predicts error likelihood and is used primarily in system development (Meister, 1989).	● Consistent and exhibit convergent reliability. ● Experts are usually available when other data sources are not.	Experts may not be what people expect them to be.

2.6.3 Error Reason Analysis

Knowing what is wrong is important, while knowing where the error comes from is relatively more important. The reasons of errors can be analyzed from both the nature of the errors and contextual circumstances. If the error can be recognized and the reasons of an error can be found, this error, such as human error, might be corrected.

Root Cause Analysis (RCA) is "a process designed for use in investigating and categorizing the root cause of events with safety, health, environmental, quality, reliability and production impacts (Rooney and Heuvel, 2004, p.45).” According to Andersen and Fagerhaug (2000, p.10), “root cause analysis is a structured investigation that aims to identify the true cause of a problem, and the actions necessary to eliminate it.” It helps identify what, how and why something happened and facilitates prevention and recurrence. It can be traced to the broader field of Total Quality Management (TQM) and is one of the core building blocks in an organization’s continuous improvement efforts (Andersen and Fagerhaug, 2000). It is expected to help improve the ontology-based product design continuously.

The Root Cause Map™ technique came from the management oversight and risk tree (MORT) for the Department of Energy's Savannah River Laboratory. Its map structures the reasoning process for identifying root causes by identifying detailed root causes for each major root cause category. It is helpful to ensure consistency across all root cause investigations and supports trending of root causes and categories (Boniface, 1997).

To find the reason for errors in ontology development and utilization, errors that can happen at different stages of ontology-based work can be classified first (Sections 3.5.1-3.5.3), and in each stage, Root Cause Map can be used in order to help people find the root reason and possible solutions (Section 3.5.4).

2.7 Literature Related to Evaluation of Information Systems

Smithson and Hirschheim (1998) identified five levels at which the evaluation of information systems is performed. The levels are: macro, sector, firm, application and stakeholder. The macro level refers to a national or international perspective; the sector level refers to an industrial sector; the firm level usually focuses the impact of a firm’s IS (Information System) on its performance; the stakeholder level recognizes that different stakeholders have different concerns and different value systems which strongly influence their evaluation of a particular IS (Land, 1976). The research in this dissertation can be located at the application level, which attempts to evaluate the impact of a particular application.

In evaluating systems for MIS (Management Information System) and DSS (Decision Support System), DeLone and McLean (1992) classified evaluation criteria under six categories: system quality, information quality, use, user satisfaction, individual impact and organizational impact. Costs and benefits are also frequently politically charged at the stakeholder level (Lederer et al., 1990). Irani (2002) described taxonomies of benefits in a company's cost benefit analysis: strategic, tactical, and operational benefits. The survey for the research here is on the tactical and operational benefits. The measures that can be used in the tactical aspect are improved flexibility, improved response to changes, improved product quality, increased productivity, improved data management, improved accuracy of decisions, and improved data availability.

Smithson and Hirschheim (1998) proposed a framework to classify the literature on IS evaluation. It includes three zones, which are efficiency, effectiveness and understanding. In the efficiency zone, outsourcing evaluation metrics would focus on the lower level metrics associated with meeting objective measures on the productiveness of the outsourcing arrangement. Effectiveness, is more concerned with 'doing the right things' rather than the efficiency view of 'doing things right' (Drucker, 1970). Most current approaches in this zone are development of traditional cost-benefit analysis. According to Hamilton and Chervany (1981), MIS effectiveness, defined by the accomplishment of objectives, is of concern not only to the management function, but also to user, developer, and internal audit personnel involved in MIS implementation. Understanding regards evaluation as problematic and seeks to understand more about evaluation within the particular organizational context. Farbey et al. (1993) argue that the following factors related with the efficiency and effectiveness zone are important in deciding which evaluation approach to adopt: the role of the evaluation, the decision environment, system characteristics, organization characteristics, and the cause and effect relationships between the investment and its benefits.

Shannon and Warren (1949) summarized that the output of an information system (IS) could be measured at three levels, including the technical level, the semantic level and the effectiveness level. The technical level is defined as the accuracy and efficiency of the IS, the semantic level focuses on whether the transmitted symbol precisely convey the desired meaning, and the effectiveness level is defined as the effect of the IS on the receiver. Effectiveness is a series of influence events that change in recipient behavior and in IS performance, such as the receipt of the information, an evaluation of the information and the application of the information (Mason, 1978). According to Le Blanc and Rucks (1996), the utilization of an IS is often assumed as a surrogate measure of its effectiveness or success. System usage is often used as an explanatory variable associated with recognized measures of system performance or success, such as profit or market share. Computer-based approaches for product design are generally used to facilitate designers in reducing design time, decreasing product cost and enhancing product quality. This research focuses on initial design and development of ontologies and ontology-based systems for product design. Outcomes, such as profit, market share, cost, and other cost-related outcomes, are not considered as dependent variables here, because there is no mature or commercial platform or product involved.

Two general views can be taken concerning what system effectiveness means and how it should be measured: the goal-centered view and the systems-resource view (Hamilton and Chervany, 1981). In the goal-centered view, the way to assess system effectiveness is first to determine the *task objectives* of the system, or of the organizational units utilizing the system, and then to develop *criterion measures* to assess how well the objectives are being achieved. Effectiveness is determined by comparing performance to objectives. In the system-resource view, system effectiveness is determined by attainment of a normative state. In terms of technological resources, system effectiveness might be indicated by the quality of the system or service levels. The distinction between the two views is similar to the distinction drawn between "summative" and "formative" evaluation approaches in the evaluation research literature (Schriven, 1972). Summative evaluation determines whether the system has accomplished objectives and provides information on the system outcomes, or ends, to support decisions to continue, adopt, or terminate the system. Formative evaluation assesses the quality of the system as well as related support, and provides information throughout the implementation process to help improve the means, or process, to accomplish objectives and aid interpretation of summative evaluation results. The choice of goal-centered view or systems-resource view to measure effectiveness depends on the evaluation context. As the research is not for a specific organization, the effectiveness part of evaluation method in Section 3.6 is on the systems-resource view. System effectiveness might be assessed at three levels of objectives (Hamilton and Chervany, 1981), which are listed below.

- Level 1: The information provided by the information system, and the support provided by the MIS function to the users of the system.
- Level 2: The use of the information system and the effect on user organizational processes and performance.
- Level 3: The effect of the information system on organizational performance. System effectiveness is ideally assessed in terms of the information system's contributions to accomplishment of organizational objectives, i.e., its effect on organizational performance (Level 3).

Moreover, Kumar (1990) summarized the most frequently used summative or post implementation evaluation of computer-based information systems. The 17 criteria are accuracy of information, timeliness and currency of information, user satisfaction and attitudes towards system, internal controls, project schedule compliance, adequacy of information, appropriateness of information, system security and disaster protection, hardware performance, system performance versus specifications, system usage, user friendliness of system-user interface, quality and completeness of system documentation, system's impact on users and their jobs, net operating costs-savings of system, quality of programs, systems' fit with and impact upon organization.

Sprague (1980) articulated and integrated major concerns of several "stakeholders" in the department of DSS: executives and professionals who use them, the MIS managers who manage the process of developing and installing them, the information specialists who build and develop them, the system designers who create and assemble the technology on which they are based, and the researchers who study the DSS subject and process. There are two kinds of surveys, one is for system users, the other is for system supporters who manage and support the system. The criteria to be used in future evaluation, which are summarized in Section 3.6.3, will be evaluated through these two kinds of surveys.

2.8 Summary

In this chapter, the present situation of product design and its problems are stated in Section 2.1 first. The basic knowledge and development of semantic web is introduced in Section 2.2. Ontology plays a pivotal role in semantic web. Its principle, advantages, editors, and its present development in product design are analyzed in Section 2.3. Based on the semantic web and ontologies, ontology-based information integration approaches, as well as the decision making criteria and methods in product design are presented in Sections 2.4 and 2.5.

In order to avoid some errors in the development and utilization of the research, types of errors related to the research, error collection methods and reason analysis methods used in Section 3.5 are summarized in Section 2.6. Moreover, the evaluation criteria and methods used in related work are summarized in Section 2.7.

From the literature review, it can be concluded that the following points are not solved, which provide the motivation for the focus of this work.

1. There are some ontologies and ontology development methods, but in order to represent the common understanding of DFM and use it for design with the consideration of complex technical and economical relations, a systematic process of ontology development and utilization process as well as detailed methods in the process need to be investigated. The DFM ontology and other related ontologies should be built or edited by following the process and these methods.

2. Although there are several PDM and PLM systems developed, there is needed work to improve data consistency, availability and quality. Moreover, while there are several ontology integration and ontology-based data integration methods that have been studied, these have not been adopted in product design. To date, the main focus has been on how to locate the data resource and transform query sentences into the proper format by utilizing the semantics of ontologies. The ontology-based method to guide users to the proper information, locate the proper resources, integrate heterogeneous data and utilize the integrated data in product design needs to be

investigated in order to solve the problems in data availability and suitability. The means to utilize the advantages of ontologies in relation representation and utilization also needs to be studied in order to enhance the data consistency and quality.

3. In concurrent engineering, decisions often need to be made in product design with respect to several aspects, such as market, manufacturing, etc. In concurrent engineering, systems engineering and systems dynamics, multiple factors and relations among factors are also considered. It is a big challenge for designers to consider these factors and constraints at the early phase of product design and make trade-offs among design parameters, especially when they are not very familiar with the area and the new technologies. Although ontology can be used to solve this problem, ontology-based methods and tools for decision support in product design are not well developed.

4. Ontology development and utilization for knowledge management in product design is an important topic under development. Avoiding errors in each phase of the ontology development and utilization process, the methods for error control and process improvement are critical for the continuous development and usage of ontology-based methods used in product design and other fields. Moreover, the evaluation method and criteria for this topic also need to be summarized and investigated.

Chapter 3: Approach and Methodology

This chapter provides the approach and methods of the research. The ontology-based method for information retrieval, reuse and decision making in Design for Manufacturing (DFM) is put forward in Sections 3.1-3.4, followed by the test and refinement of the research (Section 3.5) as well as the evaluation methodology of the research (Section 3.6). The study is based on the TIE project being conducted by the NSF e-Design Center at Virginia Tech. In Appendix B, the background of the FSW and the TIE project are described.

3.1 Overview of the Research Approach

In concurrent engineering environments, the time to develop a new product can be reduced significantly by avoiding design errors, which are caused by considering the problem locally instead of globally, and features difficult to machine. Design for Manufacturability (DFM) addresses not only the time-to-market issue but also those of quality and cost (Parsaei and Sullivan, 1993).

According to Hatamura (2006), what designers want is a hierarchical display of data. A designer will be lost not knowing where to start in searching through information if all of the information is displayed together. This forces the designers to manually arrange the information in a mental hierarchy. Ontology has the ability to capture concepts, and represent these in a hierarchical manner. The DFM ontology is built to categorize concepts in the design, record and represent these in a hierarchical way, consider the concepts in parts, manufacturing, machine, material, etc., and constrain the slots using built-in logical and mathematical relations. It aims to facilitate the data retrieval, provide the basis for the decision support tool, and help designers get the best trade-off results after considering the constraints and relations among slots saved in the DFM ontology.

Ontology is a way of dealing with the order and structure of reality. The main focus of constructing an ontology is to create a common language for sharing and reusing knowledge about phenomenon in the world of interest. For example, an ontology for knowledge based organizations would provide terms and definitions for concepts (objects, events, processes, entities, results and goals) that are of importance in characterizing these organizations at a preferred level of detail. Because of the importance of ontologies in the emergent area of knowledge management, better support for their development and utilization is needed.

As described previously, this research is motivated by the shortcomings of current computer-based methods for product design in light of life cycle considerations and modern engineering practices which are collaborative and distributed across time, place and cultures. Difficulties persist in locating data, in considering the complex relations among data and in making design decisions in a systematic way. The ontology development and utilization methods for knowledge management and decision making in product design proposed herein are put forward to address these difficulties and advance the field of engineering design. The field of Design for Manufacturing (DFM) is selected as the context for this study, since it impacts and relates several life cycle phases, and addresses important challenges that persist for representation and decision support in the field of DFM.

For ontology development, there are three representative methodologies, including: TOVE (Gruninger and Fox, 1995; Gruninger, 1996), ENTERPRISE (Uschold and King, 1995; Uschold, 1996) and METHONOTOLOGY (Fernandez et al., 1997; Fernandez et al., 1999). In practice, developing an ontology includes defining classes in the ontology, arranging classes in taxonomic hierarchy (classes, subclasses and super classes), defining slots and specifying allowed values for these slots, creating instances of classes and input/edit values of their slots (Noy and McGuinness, 2001, p.3).

Based on the seven step development processes by Noy and McGuinness (2001), the process of ontology development and utilization of this work is put forward, as shown in Figure 3-1. This process can not only be used in DFM, but also in other fields, such as design for maintenance or conceptual design.

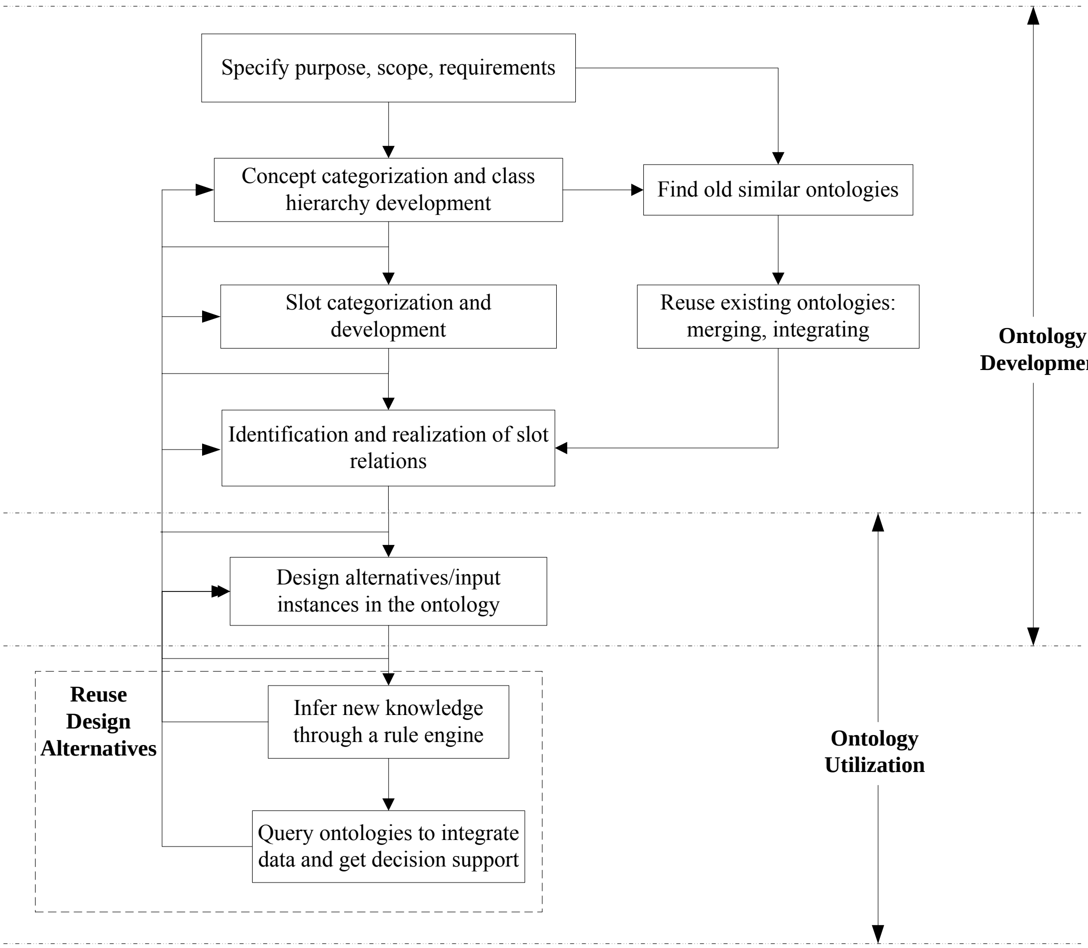


Figure 3-1: Flowchart of the development and utilization of the ontology

There are two parts in this process, which are ontology development and ontology utilization. In the ontology development, the structure, relations and instances of the ontology are developed and proper existing ontologies are reused. In the ontology utilization, new design alternatives are input as instances into ontologies, new knowledge can be reasoned out based on the relations in ontologies, rule engines infer feasible design alternatives in ontologies, other data resources can be retrieved, compared and reused through querying ontologies, and the design suggestions can also be queried out from ontologies.

During the first step in the ontology development process, the motivation, purpose, scope and requirements for the proposed ontology should be specified according to its intended applications. The purpose of the DFM ontology is to capture the domain knowledge in

DFM, to share information in a domain of interest for product designers by defining a set of data and their structure and relations, to retrieve the feasible design alternatives from heterogeneous data resources, and to compare the design alternatives and help designers make decisions based on technical and economic characteristics and relations. One kind of classification of ontologies considers the degree of representation of conceptualization and the closeness of the ontology (Guarino, 1998), which classifies ontologies into fine-grained ontology and coarse ontology. A coarse ontology consists of a minimal number of axioms while a fine-grained ontology gets closer to the intended meaning of the vocabulary of the knowledge base. Based on the motivation, purpose, scope and requirements of ontology development and utilization, the degree of representation and the level of details should also be considered.

Secondly, the concepts in the domain of endeavor are categorized. The classes and relations among classes and their subclasses form the class hierarchy as a tree structure, as described in Section 3.2.1. The properties of each class (slots) are then categorized as described in Section 3.2.2. Ontology editors, such as Protégé, can be used for the ontology development. Based on the classes and slots, the relations between properties in the area can be captured and realized. At present, there are two major ways to realize mathematical and logical relations among slots in ontology development. First is using a rule engine (such as Jess) and rules language (such as SWRL (Semantic Web Rule Language)), and the second is using Java Function Calls programs. In Section 3.2.3, these two methods are explained and compared to meet different users' requirements. The steps above mainly focus on developing a new ontology to meet the requirements.

With the rapid development of ontologies in Computer Integrated Manufacturing Systems (CIMS), more and more existing ontologies can be reused. Gruber (1995) proposed a preliminary set of design criteria for ontologies whose purpose is knowledge sharing and interoperation among programs based on a shared conceptualization. The criteria are clarity, coherence, extendibility, minimal encoding bias, and minimal ontological commitment, which can be used not only to select existing ontologies for reuse, but also to develop ontologies with tradeoffs among these criteria. After finding the old existing ontologies which can partly meet the present requirements and have similar categorization of concepts and relations, they should be reused in the present ontology. There are two major approaches to ontology reuse, which are fusion/ merging and composition/ integration (Pinto and Martins, 2004). In fusion/merging, source ontologies are unified into a single ontology; while in composition/integration, source ontologies are aggregated, combined, and assembled together. After reusing ontologies in the present ontology, the structure and relations among concepts and properties still need to be adjusted or improved to properly meet the present requirements.

After the concepts, slots and relations are identified and realized in the DFM ontology, new design alternatives can be created with the support of the ontology. A new design alternative is a new instance in the corresponding class in the ontology. This step can also be seen as the enrichment of the knowledge base of the DFM ontology. Slot relations realized in Section 3.2.3 help to calculate and infer some advanced information based on the basic inputs from the designers. With the support of relations in the ontology, in this step, the designers' input work can be largely reduced, and the data consistency and data accuracy can be enhanced. The first case in Chapter 5 illustrates this step. The strategy of ontology development in the research is explained in Section 3.2.4.

In Protégé, the query tab can realize searching instances in the ontology model according to the query requirements, as explained in Section 3.3.1. Yet Protégé is a very basic ontology editor, and has shortcomings for use in product design such as internet connection ability and heterogeneous data integration. To query the knowledge in ontologies, the approach used in this work uses the data reasoning methods based on the JTP reasoner (Section 3.3.2), creates programs using JSP to connect ontology with data resources for data integration (Section 3.3.3).

After the above steps, designers not only can use the ontology to create and add new design alternatives into the knowledge base of the ontology and infer new knowledge through the relations in the ontology, but can also retrieve and compare the existing design alternatives from different data resources (Section 3.3) and get design suggestions through querying ontologies (Section 3.4), as shown in the last two steps in Figure 3-1.

After finishing each step in this working process, evaluation and refinement is implemented, and with the change of domain, conceptualization and specification in the knowledge sources, ontologies need to be maintained and evolved (Noy and Klein, 2004). All of this work focuses on the improvement of the ontology. Possible errors in each phase of the research are summarized and the ontology-based method for root cause analysis for continuous improvement of the ontology-based method is put forward in Section 3.5. And in Section 3.6, there is evaluation methodology for the research.

3.2 Ontology Development

3.2.1 Concept Categorization and Class Hierarchy Development

Before using the ontology editor, Protégé, concepts, slots and relations need to be identified and categorized. The Unified Modeling Language (UML) (Object Management Group, 2004) is an object-oriented modeling language, which supports object-oriented principles and provides the modeling diagrams that are needed to develop software using these principles. The advantages of using UML in ontology development are that it can aid developers to visualize complex ontologies, manage the ontology development process and reuse existing UML files in ontology development. But according to Knublauch (2003), UML is basically a language for object-oriented programming that few domain experts fully understand, and it consists of a fixed set of modeling constructs (such as classes and attributes) that are not very useful when domain experts would rather talk about specific business processes, and products. Moreover, Ontology UML Profile uses boxes to illustrate different concepts, and requires text in these boxes to mark different objects, such as <<OWLClass>>, <<OWLObjectProperty>>. This makes the ontology UML Profile diagram overcrowded. Although Dragan et al. (2006) uses standard graphical symbols, such as marking classes, instances and properties in unique colors, the process and the diagram are still complex. What's more, the relations among properties are not focused. Further, although it is expected to accomplish the transfer between the ontology and the UML file, tools for these two often cannot communicate well. Additionally, the Ontology UML Profile considers the instances in the ontology. But for some large ontologies, developing all the instances in the UML will be very time consuming and not very useful for ontology development. Instances can be input after the ontology structure is built and relations are realized in the ontology. And with rule engines, the relations among slots can also facilitate the input of the property values of the instances.

A more simple and concise visual representation method is presented here to illustrate the categorization of concepts and properties in this field and to facilitate the understanding and communication of the steps. It is explained in Sections 3.2.1 and 3.2.2.

The class hierarchy is represented as classes and their subclasses in the ontology model. The representation method shown in Figure 3-2 is used to categorize the concepts and identify some relations among these classes. The ontology editor, Protégé, can be used to develop the ontology structure and finish the following steps. In Figure 3-2, C has four subclasses, C1, C2, C3 and C4. A is the union of A1, A2, A3 and A4. XY1 is the intersection of X and Y.

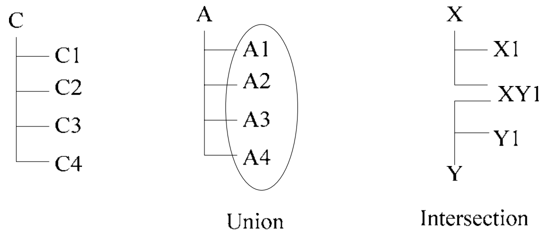


Figure 3-2: Representation method for relations among classes

OWL ontologies can represent union, intersection, disjointness and complement through logical OWL class expressions. For the conciseness of the hierarchy diagram, other relations in classes, which are disjointness and complement, are not represented. The OWL Plugin in Protégé provides direct access to reasoners such as Racer (Haarslev and Moeller, 2003), and supports two types of DL (Description Logic) reasoning, which are consistency checking and classification (subsumption) for classes (Knublauch, Musen and Rector, 2004).

After concepts are categorized and the class hierarchy is developed, some existing similar ontologies can be reused and edited to develop the expected ontology. Kim et al. (2006) proposed a collaborative assembly design framework that offers a shared conceptualization of assembly design modeling. By utilizing the assembly constraints captured in the Assembly Design (AsD) ontology (Kim et al., 2006), sharing of assembly information along with the assembly geometry can be realized at the same time. AsD ontology mainly focuses on the spatial relations and the manufacturing methods in the ontology are not comprehensive enough for designers. The Design for Manufacturing (DFM) ontology presented in this dissertation builds upon the AsD ontology, enriches the feature classification according to Lin et al. (1999) and Song et al. (1996), adds more manufacturing methods according to Olson et al. (1993), considers the environment and condition restriction, records the management feature of the ontology, and includes more manufacturability criteria in the ontology. The classes in DFM ontology cover the terms in the manufacturing ontology (Zhou and Dieng-Kuntz, 2004) except the method and value. The realized DFM ontology is shown in Section 4.2.1.

After the class hierarchy of DFM ontology is developed in Protégé, slots for each class are categorized and developed by methods in Section 3.2.2.

3.2.2 Slot Categorization and Development

Slots are properties of classes. In the representation method, classes are marked with box and arrows represent the relation between properties and classes. In Figure 3-3, the representation on the left side shows that class 1 has two properties: property 1 and property 2. The representation on the right side shows that the value of property 1 is an instance (or several instances) of class A, and property 1 is a property of class B, so the range of Property 1 is Class A and the domain of Property 1 is class B. The two classes are connected through Property 1.

Figure 3-3: Method to represent class and its properties

After the slots are identified and categorized in DFM, they are built in the corresponding classes in the DFM ontology, and their facets (restrictions on slots, such as cardinality, value type) as well as the slot hierarchy are specified in Protégé at the same time. After that, the relations among slots are identified and realized in Section 3.2.3.

3.2.3 Identification and Realization of Relations among Slots in the Ontology

In the Semantic Web, an ontology is defined as a formal representation of the relationship among terms (Berners-Lee, et al., 2001). One of the most important advantages of ontologies is that it is good at the representation and utilization of relations. In engineering, there are a great number of mathematical and logical relations, and most of these relations are among slots.

After slots are categorized and built up, the relations among the slots besides the slot hierarchy are identified (Section 3.2.3.1), and methods described in Section 3.2.3.2 realize and utilize them in the ontology. These relations can facilitate the reuse of existing data, find the inconsistency and errors in data, reduce the input work into the knowledge base of the ontology, and help designers make decisions by considering complex technical and economical criteria.

3.2.3.1 Identification of Slot Relations

There are two ways to facilitate the identification of the relations in the knowledge base. One is to get the generic relations from manuals, reports and papers from the knowledge source. These relations can be the calculation formula approved in this field, technical constraints that define the feasible region of the design alternatives, etc. The visual representation method in Figure 3-4 is used to identify the relations in the first method. As shown in Figure 3-4, slot A determines the value of slot B. Arrows are used to indicate that the start point determines the value of the end point.

Figure 3-4: Representation method for relations in slots

The other way of getting relations is using data mining methods to get the potential relations in the data resources. With the accumulation of historical design data in the databases, relations can be found. For example, Chang et al. (2007) derived the function groups from the graphical modeling databases by using clustering algorithm and saved these relations in their central ontology; and Chen et al. (2006) retrieved the association rules and selected the relation among concepts.

3.2.3.2 Realization of Relations among Slots in the Ontology

There are several methods to realize and utilize the relations in the ontology. PAL (Protégé Axiom Language) is a constraint language beyond slot facets that helps to enforce the semantic properties of knowledge bases encoded in Protégé (Tu, 2001). The PAL query language can be used for searching instances that satisfy or do not satisfy certain relations, but PAL can not be used to infer new knowledge and is not good for the representation of mathematical relations. More research is needed to capture relations among slots and their use to infer new knowledge and support decision making in design. This work introduces two approaches to satisfy this need. One approach is using a rule language (such as Semantic Web Rule Language, SWRL) with a Rule engine (such as the Jess rule engine), and the other is using Java Function Calls by Protégé. Both methods have advantages and disadvantages. In this section, these two methods are explained and compared to meet different users' requirements.

(1) Rule Language with a Rule Engine to Realize Relations

OWL (OWL Web Ontology Language) is built upon W3C standards XML (Extensible Markup Language), RDF (Resource Description Framework) and RDFS (RDF Schema), and extends them to express class properties and relations between terms (Brickley and Guha., 2003; Lassila and Swick, 1999). The ontology used in this method is built in OWL. Both OWL and SWRL (Semantic Web Rule Language) provide a way to represent knowledge as a series of related terms which are machine operable (Berners-Lee, et al., 2001), and SWRL implements rules that operate on OWL knowledge representation.

Similar to many other rule languages, SWRL rules are written as antecedent-consequent pairs, where the antecedent is referred to as the rule body, and the consequent is referred to as the head (O'Conner et al., 2005). According to Horrocks et al. (2003), a rule can be read as meaning that if the antecedent holds (is true), then the consequent must also hold. According to SWRL rules defined in the OWL ontology, Jess rule engine can infer new knowledge, which is the consequent part of the rules. The head and body consist of a conjunction of one or more atoms, where atoms can be of the form C(x), P(x,y), sameAs(x,y) or differentFrom(x,y), C is an OWL description, P is an OWL property, and x,y are either variables, OWL individuals or OWL data values. SWRL also supports a range of built-in predicates, such as mathematical built-ins, which greatly expand its expressive power.

There is an SWRL Rule tab in Protégé-OWL to support the SWRL edition. The SWRL Editor itself has no inference capabilities, since it simply allows users to edit SWRL rules and save and load them to and from OWL knowledge bases, but Protégé SWRL Factory mechanism is used in the SWRL editor to integrate an existing rule engine(such as Jess rule engine) with the SWRL Editor (O'Conner et al., 2005). The Jess system consists of a rule base, a fact base, and an execution engine. The execution engine matches facts in the fact base with rules in the rule base, and asserts the new facts into the fact base or executes Java functions (O'Conner et al., 2005). The Jess rule engine helps to run SWRL rules interactively to create new OWL concepts and inserts them into an OWL knowledge base. SWRL also has built-ins for comparisons, math, Boolean values, strings, date, etc. that expand its expressive power. These built-ins are useful to realize a large number of mathematical relations in DFM. Compared with ontology reasoning, such as Pellet, Jess rule engine has longer initiation time, but better query answering time (Dellas et al., 2007).

There have been some applications of SWRL and the Jess Rule engine. Fernandes et al. (2007) created people ontology, project ontology, functional basis ontology and design ontology, and used SWRL and the Jess rule engine to generate design recommendation. Kim et al. (2006) represented Assembly Design constraints in the AsD ontology by using OWL (Web Ontology Language) and SWRL (Semantic Web Rule Language), which can capture assembly and joining intents.

The process of developing and using SWRL rules are as follows:

1) Develop the SWRL rules in SWRL Rule tab of Protégé;
2) Input value of slots related to the antecedents of the rules;
3) Use the Jess rule panel in SWRL Rule tab to transfer OWL knowledge and SWRL rules to Jess, perform inference using those knowledge and rules, and then transfer the resulting Jess facts back to Protégé-OWL as OWL knowledge. Interaction between the SWRL Editor and the Jess rule engine is user-driven (O'Conner et al., 2005).

The OWL ontology is saved and used as the knowledge base for the decision support tool. Details are provided in Section 4.2. The inferred knowledge can be used to support design, as illustrated in Sections 5.2.2-5.2.4.

(2) Java Function Calls by Protégé to Realize Relations

Although some rule engine approaches such as CLIPS, Jess, Jade, PAL and Algernon are available as tab widget plug-ins for Protégé, none of them could cover the entire desired functionality. They work only "on demand" after activating and focusing their tab and the rules are applied on the entire knowledge base, rather than on a certain class or slot, regardless of specific scopes of rules. In contrast, using an object-oriented programming language, such as Java Function Calls, can get a more efficient integration in the host program Protégé (e.g. by developing slot widgets to be applied in normal forms instead of separate tab widgets), and has the chance to program just anything, not limited by constraints of a rule engine (Reiner, 2003).

The steps of building and using Java Function Calls Programs are shown below.

1) Make Java programs and create the java package;
2) Select project->manage included projects, and include the J-Calls.jar into the ontology project in Protégé;
3) Under :SYSTEM-CLASS->:EXTERNAL->:J-PACKAGE, create new instance, and specify the jar file, the related class that the jar file packed will be shown under the

path;

4) Under :TEMPLATE->:JAVA-ATTACHMENT->:J-INPUT-INTERFACE->:J-FUNCTION-SLOT, create a new instance *A*, which is a slot in another class in the ontology. Specify the attributes in the instance(slot):
   - Name, which is the name of slot *A*;
   - Value type, which is the type of the slot *A*;
   - Domain, which is the class that the slot *A* belongs to;
   - :J-INPUT-SLOTS: specify the inputs of the java program;
   - Java Class: the class that the jar file packed.
5) Under :TEMPLATE->:JAVA-ATTACHMENT->:J-OUTPUT-INTERFACE->:J-INOUT-INTERFACE-> :J-ACTION-SLOT, buttons with some specific actions can be specified;
6) Create instance in the class (specified as domain in step 4), after the input slots are all filled, the java program will be run automatically and the result will be output to the slot *A*.

In the DFM ontology, Java Function Calls (Reiner, 2003) programs are used to force the ontology frames (classes, slots, and instances) to realize cost models and facilitate the input of instances into the knowledge base of the DFM ontology. For example, a slot called "total_weld_cost" is the sum of other five slots in an instance of the MIG class. Java Function Calls can calculate the result automatically after getting the input. This is illustrated in Case study 1 in Section 5.2.1. Through slot relations built in Java Function Calls, data consistency and ontology quality can be enhanced and errors related to calculation and data input can also be avoided to some extent.

The shortcoming is that the functions of Java Function Calls cannot be created and edited within the Protégé user interface in a logical language (Crubézy, 2002). It must be edited and compiled in a JAVA editor, and the java package need to be included as an instance of its metaclass.

(3) Comparison of the Two Methods

Both of the two methods to represent relations in engineering have advantages and disadvantages, and can be used in different situations. Table 3-1 below summarizes their differences. Both of the methods can be used directly for decision support, which is illustrated in case studies found in Section 5.2.

Table 3-1: Comparison of rule-based method and Java Function Calls

Comparison items	**Rule language and rule engine**	**Java Function Calls by Protégé**
Functions	inference and judgement	Realize flexible functions and cover the entire desired functionality
Ontology format	OWL/RDF Files	Call Java Attachments
How to develop it	use the SWRL tab to input the rules; easier to build.	● Make java programs, ● export them into jar package, ● build the input and output in the ontology with the java package
How to get the result	Need to use a separate tab in the ontology editor to get the result, work on demand, control the step of this inference process	Get a more efficient integration in the host program Protégé and bring result automatically
What is retrieved	Logical judgement and new inferred knowledge	Any result, such as calculation, logical judgement, an action, etc.

3.2.4 Strategy of Ontology Development

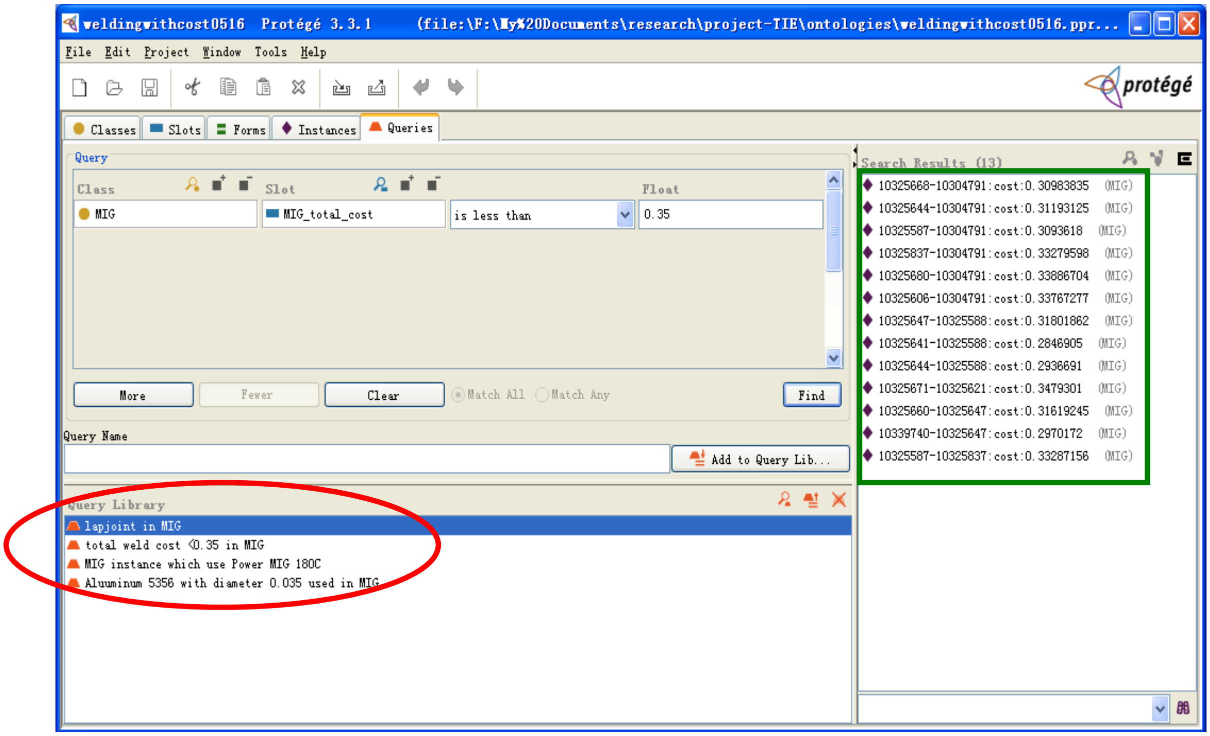
The DFM ontology of this work is built with consideration of ontology reuse criteria in Gruber (1995) and the AsD ontology (Kim, et al., 2006). The AsD ontology has good clarity and coherence. It mainly focuses on the spatial relations between parts in assembly design, and focus on cost and manufacturing requirements is absent. Based on the hierarchy of Kim et al.'s (2006) AsD ontology, after categorizing the concepts and specifying the relations by using the method in Figures 3-2, 3-3 and 3-4, the DFM ontology can be built by adding more concepts, as well as technical and economical relations and data, such as welding conditions and features related to cost, to give a good basis for the information integration and data analysis. Java Function Calls programs are used to facilitate the input of instances into the ontology, reduce the work and enhance the accuracy in enrichment of the knowledge base of the ontology. For example, a program of Java Function Calls calculates the total weld cost after the design information is input into a case. SWRL rules and the Jess rule engine are mainly used to infer new knowledge in a specific case. For example, in a company, after considering the thickness of the parts that will be welded, allowed diameter of filler material, voltage and currency, and available machines, the cheapest design alternative can be inferred.

The DFM ontology is developed in Section 4.2.1. In addition, in order to integrate data from heterogeneous data resources (Section 3.3) and give design suggestions (Section 3.4), other ontologies besides DFM ontology also need to be built. They are described in detail in Sections 4.2.2 and 4.2.3.

3.3 Ontology-Based Data Integration

3.3.1 Data Query and Retrieval by Using Query Tab in Protégé

Utilizing the query function in Protégé, instances that fit the query requirements can be found. Query results in Figure 3-5, marked by the box, are obtained from the query in the MIG class with the specification that MIG_total_cost should be less than 0.35 dollars/joint.


weldingwithcost0516 Protégé 3.3.1 (file:\F:\My%20Documents\research\project-TIE\ontologies\weldingwithcost0516.ppr...

Class: MIG
Slot: MIG_total_cost
is less than
Float: 0.35

Search Results (13)

```
10325668-10304791:cost:0.30983835  (MIG)
10325644-10304791:cost:0.31193125  (MIG)
10325587-10304791:cost:0.3093618  (MIG)
10325837-10304791:cost:0.33279598  (MIG)
10325680-10304791:cost:0.33886704  (MIG)
10325606-10304791:cost:0.33767277  (MIG)
10325647-10325588:cost:0.31801862  (MIG)
10325641-10325588:cost:0.2846905  (MIG)
10325644-10325588:cost:0.2936691  (MIG)
10325671-10325621:cost:0.3479301  (MIG)
10325660-10325647:cost:0.31619245  (MIG)
10339740-10325647:cost:0.2970172  (MIG)
10325587-10325837:cost:0.33287156  (MIG)
```

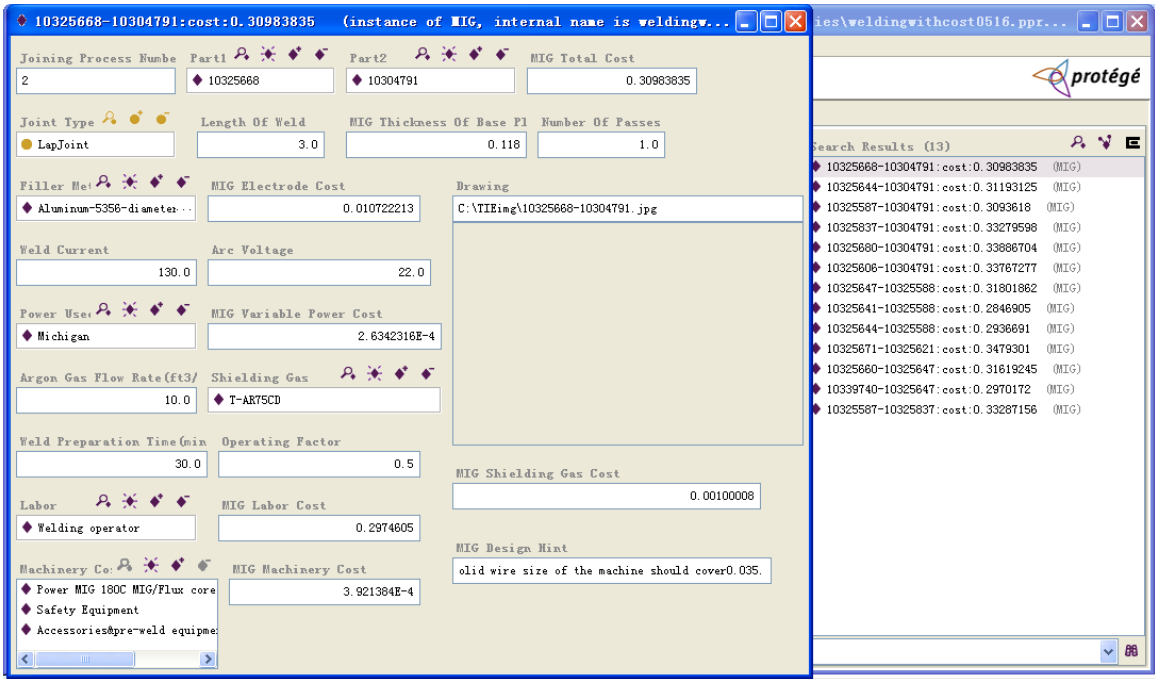
Query Library

- lapjoint in MIG
- total weld cost <0.35 in MIG
- MIG instance which use Power MIG 180C
- Aluuminum 5356 with diameter 0.035 used in MIG

Figure 3-5: Query tab in Protégé with its query result and query library

Frequently used query sentences can be saved for reuse. As shown in the left bottom portion of Figure 3-5, some query sentences have been saved. For example, the first query retrieves all the MIG instances whose joint type is lap joint. The frequently used query requirements for DFM can be summarized and saved in the query tab. This is expected to help designers enhance the efficiency and quality of their work.

The query result is shown on the right side of the queries tab. After clicking the instance, the detailed information of this instance will pop up as shown in Figure 3-6. Designers can compare different instances and change the parameters in this way to reuse and edit exiting MIG welding data. The query result from this tab can also be output to a text file in the local computer. This file can be used for the evaluation of design alternatives in the local MADMT (Multi-Attribute Decision-Making Tool), which is described in Sections 3.4 and 4.4.


10325668-10304791:cost:0.30983835 (instance of MIG, internal name is weldingw...

Joining Process Numbe	2
Part1	10325668
Part2	10304791
MIG Total Cost	0.30983835
Joint Type	LapJoint
Length Of Weld	3.0
MIG Thickness Of Base Pl	0.118
Number Of Passes	1.0
Filler Me	Aluminum-5356-diameter···
MIG Electrode Cost	0.010722213
Drawing	C:\TIEimg\10325668-10304791.jpg
Weld Current	130.0
Arc Voltage	22.0
Power Use	Michigan
MIG Variable Power Cost	2.6342316E-4
Argon Gas Flow Rate(ft3/	10.0
Shielding Gas	T-AR75CD
Weld Preparation Time(min	30.0
Operating Factor	0.5
MIG Shielding Gas Cost	0.00100008
Labor	Welding operator
MIG Labor Cost	0.2974605
MIG Design Hint	olid wire size of the machine should cover0.035.
Machinery Co:	Power MIG 180C MIG/Flux core Safety Equipment Accessories&pre-weld equipme
MIG Machinery Cost	3.921384E-4

ies\weldingwithcost0516.ppr...

Search Results (13)

- 10325668-10304791:cost:0.30983835 (MIG)
- 10325644-10304791:cost:0.31193125 (MIG)
- 10325587-10304791:cost:0.3093618 (MIG)
- 10325837-10304791:cost:0.33279598 (MIG)
- 10325680-10304791:cost:0.33886704 (MIG)
- 10325606-10304791:cost:0.33767277 (MIG)
- 10325647-10325588:cost:0.31801862 (MIG)
- 10325641-10325588:cost:0.2846905 (MIG)
- 10325644-10325588:cost:0.2936691 (MIG)
- 10325671-10325621:cost:0.3479301 (MIG)
- 10325660-10325647:cost:0.31619245 (MIG)
- 10339740-10325647:cost:0.2970172 (MIG)
- 10325587-10325837:cost:0.33287156 (MIG)

Figure 3-6: Query result and its detail information in Protégé

3.3.2 Data Reasoning Based on JTP Reasoner

This section presents a data reasoning process that is used to query information from ontologies. In the research, the working process of design is combined with the process for information integration (Section 3.3.3) and decision support to help designers get the information from heterogeneous data resources and to provide the basis for the decision support tool (Section 3.4).

This process was also once used in the wine agent by Stanford University and in the knowledge management system for ABC (Activity-based Costing) by CED at Virginia Tech. As shown in Figure 3-7, in the system, the query engine uses a reasoner (called JTP) to query the knowledge base which is in a web-based ontological mark-up language (OWL).

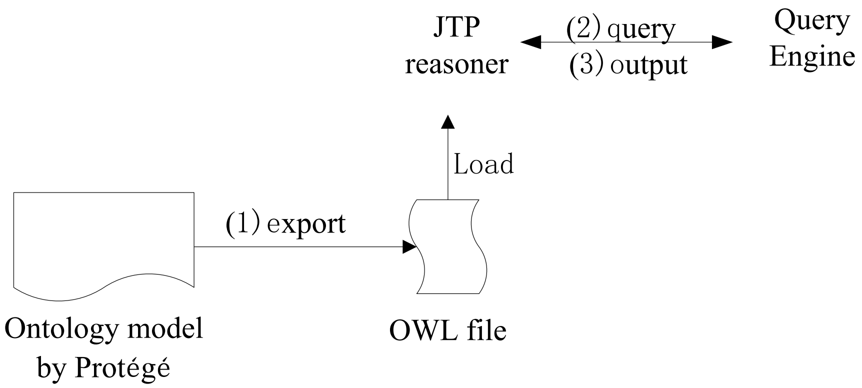


Figure 3-7: Working process of data reasoning based on the JTP reasoner

If the ontology is not built in OWL format, the first step toward developing the knowledge-based system is to convert the ontology built in Protégé into OWL format.

At the second step, JTP is used as a reasoner to infer pairing within the ontology. JTP is an object-oriented modular reasoning system based on a very simple and general reasoning architecture. It can conduct various assertions and queries for a knowledge base. In JTP, two main reasoning methods, backward chaining and forward chaining, can be used to make inference from the knowledge base (Solow, 2005). The backward chaining starts with a list of goals and works backwards to find proof for the goals. An inference engine for the backward chaining would search the inference rules until it finds one which has a **Then** clause that matches a desired goal. The forward chaining starts with the available data and uses inference rules to draw conclusions. An inference engine for the forward chaining searches the inference rules until it finds one where the **If** clause is known to be true. Once found, it can conclude, or infer, the **Then** clause, resulting in the addition of new information to its dataset.

At the final step, a JavaServer Pages (JSP) program manipulates the outputs of the JTP reasoner in a readable format to generate Browser/Server (B/S) mode. Since the ontology is a common vocabulary that different groups of engineers can use to store and query product information, B/S mode is an appropriate mode for visualization of knowledge. The advantages of using this mode include:

- It does not require users to install any kind of client programs. Web browsers can be used to connect to the query engine and get the information needed;

- The information is stored and managed in the server, which can provide consistent and updated data that different users share;

- It does not require that users learn any specific tool or programming language, such as Protégé, JTP, and so on.

3.3.3 Information Integration based on Ontologies and Heterogeneous Data Resources

In traditional tools, databases are directly connected to the interface. They save data that was input from the user interface, and give users corresponding results according to the query. But with the usage of multiple computer-supported tools for design, such as various CAX and DFX tools, more and more data are accumulated in different databases. Some useful information, such as the similar functions of parts or products, is saved in different places. This research integrates and uses information to enhance data availability and designers' working efficiency.

In this method, ontologies are used to capture the common understanding in conceptual design and help to utilize the relations among concepts and properties. The framework is shown in Figure 3-8. It has three main parts, including: web portal, data process part and heterogeneous data resources. The web portal retrieves the input from users, gives users guidance and reports the proper information to users. The heterogeneous data resources can be SQL databases, XML files, Excel files, etc. The data process part has four modules (data query guidance, data resources specification, querying sentences formation, and query result analysis) and two kinds of ontologies (one central ontology and several local ontologies). It belongs to the hybrid approach in ontology-based information integration, as described in Section 2.4.2. The advantages of the hybrid approach are that it is relatively easy to build and maintain for complex problems and support the heterogeneous data resources. The system in the research will be used for large manufacturing enterprises, so the hybrid approach, which contains the central ontology and several local ontologies, is more suitable than the other two approaches in Section

2.4.2 to integrate heterogeneous data resources in this work.

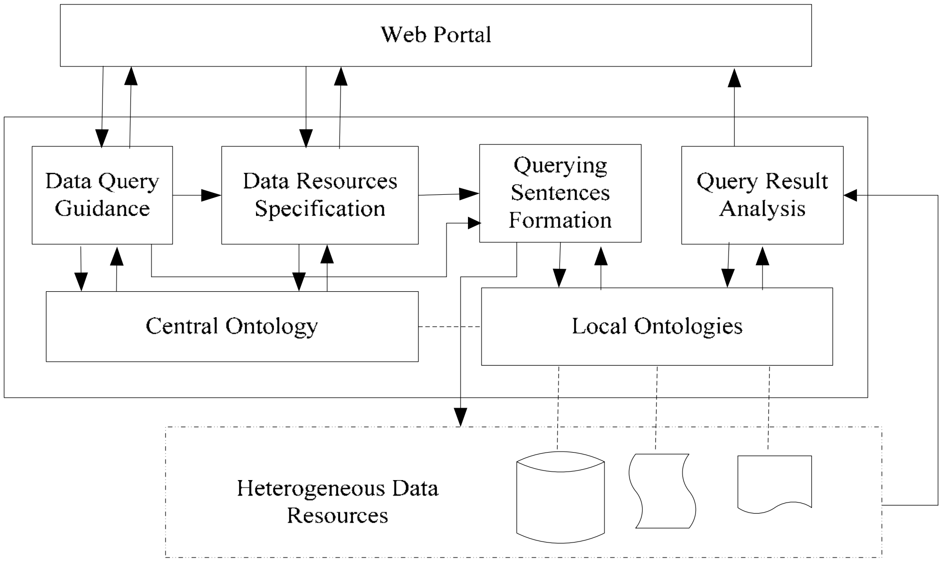


Figure 3-8: Framework of Ontology-Based Heterogeneous Data Integration

Li et al. (2007) proposed the design and implementation method for the ontology-based information retrieval system, which parses users' requests, tests the validity, reasons and expands the semantics of query terms based on ontology, forms sub-queries and finally retrieves the query results from different data resources. Their work mainly focuses on SQL and XML data resources and accomplishes the query process based on users' query requirements without guidance, which is like Unicorn mentioned in Section 2.4.2. The data query guidance module in the research here is added to conquer the shortcoming of these previous systems. It gets input information from users, queries the knowledge base of the central ontology by using the method in Section 3.3.2, and helps designers drill down to desired information. After the data query guidance locates the proper factor in which designers really need to get the related details, the data resources specification module finds the related data resources that save the data. Based on the central ontology, local ontologies and the mapping relations within, the query sentences formation module forms the query sentences for these data resources. The working method in this step is like the Unicorn Workbench (Alexiev et al., 2005). Then after the related data is queried out from multiple data resources, the query result analysis module can help designers analyze the query results from the data resources, such as evaluating the design alternatives with economic and technical criteria, find the root factors of the problems, etc. The local multi-attribute decision making tool (MADMT) in Sections 3.4 and 4.4 is a query result analysis module. It provides the comparison of results and help designers make decisions.

3.4 Decision Support Tool Based on Ontologies and Heterogeneous Data Resources

Alternatives are "a particular set of controllable design variables, which will lead to particular attribute performance (Krishnamurty, 2006, p.21)." The purpose of engineering design is to find the alternative with the highest "value" or utility. In the decision process, design alternatives are first formulated, and then the attribute space and features, which are critical to evaluate alternatives and their performance, are identified. After that, preference assessment enables the comparison of each attribute, and a multi-attribute function might be used to make a trade-off among attributes. Ranking of the alternatives is done to help designers get the final optimal alternative from an overall perspective (Krishnamurty, 2006).

Decisions involve information that comes from different sources and disciplines, and are governed by multiple measures of merit and performance (Parsaei and Sullivan, 1993). Time, cost and quality are three factors that designers care most about. In Huang (1996), sample performance characteristics in these three areas are introduced. According to the methods and models to deal with data in Section 2.5, criteria and measurement scales are chosen with the consideration of product cost, product life cycle time and product quality. They are considered in the case studies found in Chapter 5.

According to the report from CFSP (Center for Friction Stir Processing), there are six input nodes in the DFM process. The sequence of these six steps is (1) welding technique, (2) component geometry and material, (3) joint category, (4) mating/align faces and the base component, (5) joint strength, and (6) weld category. Choices for each step of the process have been offered by CFSP (Bagaitkar, 2007). See Appendix E for more detail.

According to Hatamura (2006), there are three types of decisions: (1) go or no go, (2) single selection, when there are several choices on this node, and (3) structured decision, which has multiple nodes needing a decision, and there are multiple choices on each node. The decision making in DFM relates to multiple steps (nodes), and each node has multiple choices. So the decision made here belongs to the structured decision type.

The working process is shown in Figure 3-9 below. Based on the ontology-based information integration, design alternatives can be retrieved and grouped, and the tree structure can be formed. Since the six-input-step process has been offered by CFSP, these steps are used to form the tree structure. In the future, when the classifier is not offered and large amounts of data exist, hierarchical classification methods (Section 2.5.2) can be used to form the tree structure.

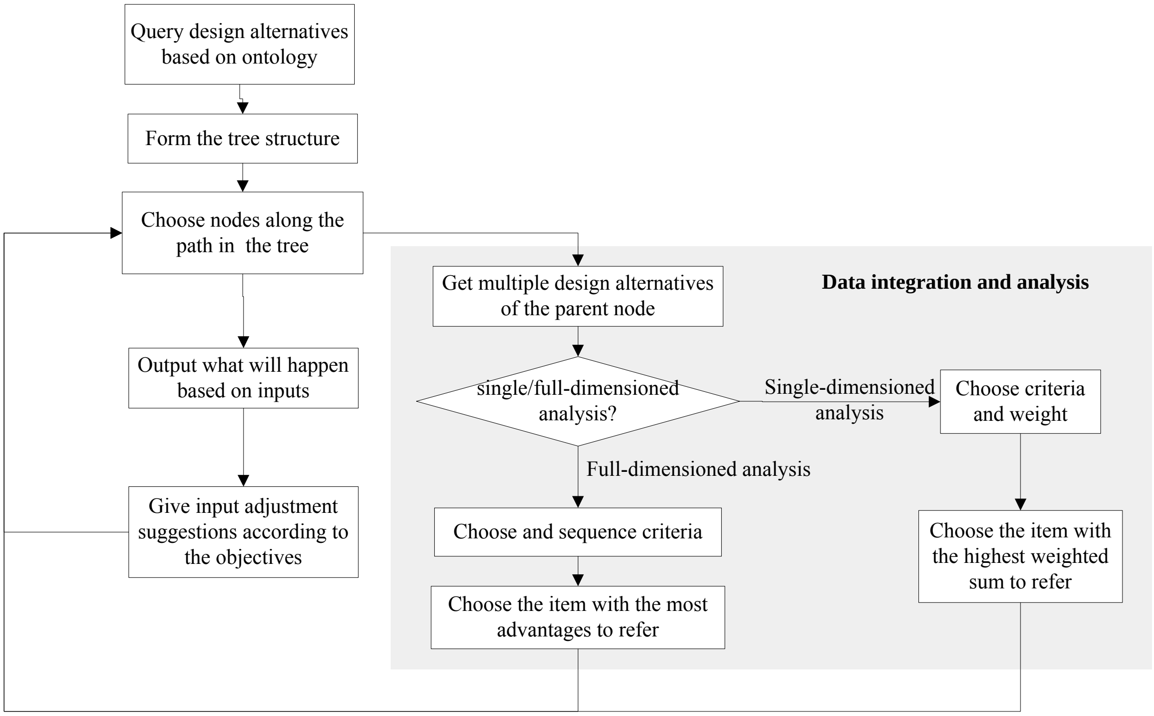


Figure 3-9: Working process of the decision support tool

In the next step, before a selection is made, integrated historical data from other data resources can be referred. They are alternatives in the previous selected node. Single-dimensioned analysis and multiple-dimensioned analysis can help the multiple attribute decision making (MADM) and get the best reference.

The ontology-based web system for decision support in DFM is developed by using JSP

(Section 4.5), and the multi-attribute decision making tool for the comparison of design alternatives is built by using Java (Section 4.4). The web system utilizes relations in the ontology to help designers consider complex relations and constraints in the choices along the path of the tree. Users are able to go through one path of the tree and find the node which contains multiple design alternatives. Design alternatives in the same node can be compared by the multi-attribute decision making tool using single-dimensioned or full-dimensioned analysis as described in Section 2.5.3.

After the input, the tool can calculate the results according to DFM formulas and parameters. After specifying the objective (attribute with direction), such as increasing the length of weld, the tool can query an ontology (DFM e-Design ontology here) and get the related parameter adjustment suggestions.

The working process of ontology-based parameter adjustment suggestion is as follows.

1) DFM e-Design ontology obtains the variables and the direct influence relations among variables in the system;

2) Jess rule engine infers the root influence factors for each variable according to SWRL rules, and store the inferred results to the corresponding slots. The knowledge base is ready for use;

3) In the step Give input adjustment suggestions according to the objectives of Figure 3-9, according to the adjusted parameter, the DFM e-Design ontology is queried through the way described in Section 3.3.2, and the root influence factor can be reasoned out from the knowledge base, which is the parameter adjustment suggestion, and the possible influenced factors after the adjustments can also be reasoned out when the users drill down the information in the tool.

The DFM e-Design ontology is developed in Section 4.2.3 and the realized web system is explained in Section 4.5, and illustrated in Section 5.2.4.

3.5 Test and Refine Ontologies and Ontology-based Information Systems

In the research of ontology development and utilization for product design, work mainly focuses on:

1) Ontology development, reuse and maintenance (related to Section 3.5.1),

2) Ontology-based information integration (related to Section 3.5.2), and

3) Ontology-based information systems for decision support in product design based on ontology (related to Section 3.5.3).

Errors may occur in any of the three aspects. In Sections 3.5.1-3.5.3, errors that might appear in the important phases of ontology development and utilization in product design are described with their types, causal factors, collection and control methods. In Section 3.5.4, an ontology-based method for Root Cause Analysis (RCA) is put forward to continuously improve the ontology-based method and tool in the future. Generally speaking, after detecting the errors and finding their root causes, some errors, such as user's misunderstanding in ontology development or the ontology aging problem, can be eliminated. An example DFM problem based on a collaborative project between two centers, CFSP (NSF Center for Friction Stir Processing) and CED (NSF Center for e-Design), provides the context to support the validation, testing, and refinement of the proposed DFM ontology and decision support tool (Terpenny, 2006).

3.5.1. Errors and Causal Factors in Ontology Development, Reuse and Maintenance

Modeling error is "due to the natural imperfections in abstract models of actual physical phenomena (Oden and Prudhomme, 2002, p.496)." Modeling error can exist in both the ontology model development, reuse and maintenance. The assumption before modeling might not be suitable and might be outdated during the usage. The difference between the ontology model and common understanding in the field should also be considered as a type of modeling error.

3.5.1.1 Errors and Causal Factors in Ontology Development and Reuse

At present, there are various tools to build up ontologies, such as Oiled, Protégé, etc. One of the advantages of some ontology tools is that they can help find logic errors in the ontology, reduce, and possibly eliminate syntax errors and logic errors. Set relations among classes can be captured by logical OWL class expressions in OWL plugin in Protégé, and Racer (Haarslev and Moeller, 2003) can check the consistency among classes and avoid logic errors. Constraints between slots can be realized by rule languages and rule engines, as well as Java Function Calls programs embedded into the ontology editor, Protégé. The rule engines can help users check the logical consistency of instances in the knowledge base (ontology), and the Java Functions Calls by Protégé can be made to realize more flexible constraints, such as mathematical relations among slots. Through these approaches, errors in data inconsistency can be largely reduced.

Another source of errors comes from different developers choosing different methods to express the same concept. There is no standard for the ontology being built. And there are differences such as culture, language, law, and so on. The discrepancies lead to some errors during each step or phase of ontology development which is dependent on human understanding. These errors belong to human errors.

Approximation errors and measurement error can also appear in the ontology being developed. For example, one slot value of an instance can be an approximation of reality, and accordingly it causes an approximation error. Further, if measurement error occurs in data acquisition, it will also be presented in the ontology model directly.

In Step 2 of ontology development (Noy and McGuinness, 2001), ontology reuse should consider the difference between the place where the original ontology is being used and the present place where the ontology will be reused. Besides, in ontology reuse, some ontologies need to be aligned or merged. When ontologies are merged, several tools, such as Ontomorph (MacGregor et al., 1999), Chimaera (McGuinness, 2000) and PROMPT (Noy and Musen, 2000), can be used. These tools determine linguistic similarity among concept names and some of these tools use algorithms to give merging suggestions based on the comparison. The merging based on these suggestions might still be wrong, classes or slots with the same name are merged as one, but they represent different concepts; on the other hand, some classes or slots with the same semantics are not merged together. This kind of error can be found by experts or artificial intelligent programs. In these programs, semantics will be compared to find the consistency and discrepancy among items.

The causal factors in this phase can be wrong data type, wrong relations between slots, wrong slot value, wrong structure of the ontology, improper reuse of existing ontologies, and improper merging of items in ontologies.

Minimizing human errors, approximation errors and measurement errors requires analyzing the shortcomings of equipments, processes and people involved in data collection and recording. In order to reduce and find errors, ontology development can be implemented collaboratively by several developers with different roles. For example, in order to find errors in DFM ontology development, one developer in CED is in charge of building the ontology structure, its classes, properties and relations, and another developer in CED is in charge of inputting detail information and checking whether the ontology is correct and whether the instances offered by CFSP are consistent with these relations. Refinement should be done immediately after any error found by those checking accuracy.

After the initial ontology development is complete, it needs to be sent back to the end users, CFSP in the example. The feedback can be executed in formats, such as interview, focus group, survey, or through methods described in Section 2.6.2, such as self-report, expert judgement, etc. Refinement is based on feedback. Questions used to get feedback are listed below.

- Do the classes and slots in the ontology use correct naming conventions?
- Is the structure of the ontology right? Does the subclass really belong to the class or should it exist on the same level?
- Do the properties of one class really belong to this class?
- Is there any other class or subclass that can be added? (Is there any other aspect that is useful but is not covered here? Is there any subclasses that can be added to the class?)
- Is there any other property that should be added to some classes? In the feasibility

and economic analysis, has the information used been covered in the slots (properties) in the present ontology?

The real quality of an ontology can be assessed only by its use in real applications (Tjoal, et al., 2005). During the usage of the ontology, that means in the development and usage of the ontology-based information system, errors in the ontology should continue to be checked. This refinement process is iterative and should last throughout the entire lifecycle of the ontology (Noy et al., 2001).

3.5.1.2 Errors and Causal Factors in Ontology Maintenance

During the lifecycle of ontologies, based on the need of the semantic-based system, ontology aging needs to be detected and the ontologies need to evolve according to the changes of environmental parameters. That is why the ontology should be flexible enough to adapt itself with the latest status (Tjoal, et al., 2005). If the ontology is not updated in a timely manner, errors will occur because the ontology model cannot represent the present reality. Possible causal factors and their collection/detection methods in this phase include the following:

- Incorrect or inconsistent slot value: Outdated information in the existing instances are common causal factors of errors in this phase. Updated data may also have numerical error or contain inconsistency with the history data. Manual data collection, self report and expert judgement can be implemented periodically by ontology designers and experts. Rule languages and rule engines can also be used to detect the inconsistency among data.
- Repeated or missed concepts in classes: In this phase, nodes can be split or merged according the development of related concepts in the domain. Gargouri et al. (2003) used data mining methods to facilitate the process. Similar nodes might appear after splitting a node, and differences between two merged nodes might be dismissed. It is influenced by the setting of parameters in data mining, such as the value of k in K-mean approach used in clustering to split one classes and its subclasses. Adjusting parameters and then comparing different maintenance results is one of the methods that can be used to detect these errors.

- Improper slots and relations: The value type of some slots are Instance. With the adding and deleting of classes and slots, some of these slots become useless because their domain or range is empty. Java Function Calls programs can be made here to detect the useless slots.

Moreover, manual data collection, self-report and expert judgement can be implemented periodically by the ontology designer and experts.

3.5.2 Errors and Causal Factors in Ontology-based Data Integration

In today's collaborative distributed design environment, it is beneficial to have different ontologies collaborate with each other in order to integrate data from heterogeneous data resources, such as XML files, databases, etc. There are several ways of ontology-based data integration, as described in Section 2.4.2. Causal factors in this phase can be bad data resources, bad central ontology, and wrong or improper relations among data resources and ontologies. Causal factors in this phase and the related methods are:

- Improper or incorrect data in data resources: Errors may already exist in a data resource before the resource is chosen and the data resource selected might not be the one really needed in the system. Checking the quality and pertinence of data resources is a required step before use, and only high-quality data resources should be chosen.

- Wrong relations between data resources and ontologies: Errors might occur in the creation of relationships. This involves not only concepts mapping among ontologies, but also the development of relationships between ontology and data resources. It can be human error. These errors can be detected through expert judgement, manual data collection and self report, which are described in Section 2.6.2.

- Improper information from the ontology for data integration: If the central ontology (Alexiev et al., 2005) is used here, errors in development and maintenance of the central ontology can also occur. Errors and their management methods to find and control this kind of errors have been described in Section 3.5.1.

3.5.3 Errors and Causal Factors in Ontology-Based Information Systems

The ontology-based information system for product design can have multiple formats. The system can be based on Browser/Server or Client/Server mode for visualization of knowledge to multiple designers. It has been used in the cost management system to control cost during product family design. The preliminary work provides the context for discussion of errors to consider in ontology-based information system for product design.

According to the mechanism, errors that might appear in this phase are the differences between what are recorded in the original ontology and what are presented in the information system. The causal factors of these errors and the related methods can be:

- Distortion or dismissing of information from the ontology model built in Protégé to the OWL file. Exporting ontology to OWL is a function of Protégé. According to past experience, errors in the information system can be: disorder of the sequence of the original list of classes, and information missing because some functions are not included in OWL. This is caused by errors or deficiencies in computer programming. For example, there are several types of ontology that can be chosen when creating an ontology in Protégé. When the ontology created by using the type "Protégé Files" is exported as an OWL file, the information of FORM (representation format of instances) is lost. If the ontology is directly built in OWL instead of other types, this causal factor will not exist.

- Improper query sentence. The query sentences are sent to JTP (Frank, et al., 2004) or other reasoners/agents. If the query sentences do not meet the structure or format of the OWL ontology, the querying result will show "no querying result" or "system error," even though these elements were present in the file. These errors can be design-facilitated error (DFE) or logic error. During system development, several exception conditions should be considered and prevented beforehand. After development, some error collection methods, such as manual collection, experimental studies and expert judgement, can also be used to detect errors and then these errors can be eliminated by editing the program of the information system, changing working flow of user query manipulation, refining procedure and equipment, etc.

- Improper algorithms and approximation during calculation, representation, etc. in data analysis. These can be collected through manual/automation collection, and expert judgement might be used to detect the cause of these errors.

Errors taking place in the ontology-based information system should be tested by developers, the knowledge source, and by the end users during the test period. The questions that can be used in the test are listed below. There are six choices of responses to the questions ranging from the level of 'strongly agree' to the level of 'strongly disagree.'

- Is the data retrieved useful and accurate?
- Is the category of the retrieved data proper?
- Does the user get what was desired from the analysis of the tool? Does it help to make design decisions?
- Are the reported results easy to read and understand?

If necessary, survey and self-report in Sections 3.5.1-3.5.3 can be realized in a computer system. The advantages of using a computer system to get data are that it can enhance the working efficiency, reduce errors which are caused by manual operation, and facilitate statistical analysis and data mining.

3.5.4 Ontology-Based Method for Root Cause Analysis (RCA)

Knowing what is wrong is important, while knowing where the error comes from is relatively more important. For example, the improper integrated data is retrieved. When system developer investigates the data integration part, wrong relations between an ontology and data resources are found. After this specific error is eliminate, this kind of error can happen again frequently if the root cause of the error is not investigated and the working mechanism of ontology-based product design is not improved. What is the root cause of this kind of error? In order to minimize errors in ontologies and their usage as well as to reduce the reoccurrence of the errors, root cause of the errors needs to be found and solved with a proper method.

Root Cause Analysis (RCA) is “a process designed for use in investigating and categorizing the root cause of events with safety, health, environmental, quality, reliability and production impacts (James and Lee, 2004).”

Four steps in RCA are: data collection, causal factor charting (to find causal factor), root cause identification (identify root cause for each causal factor), and recommendation generation and implementation (James and Lee, 2004). The first two steps of RCA for ontology related error analysis can be accomplished by methods in Sections 3.5.1-3.5.3. In the 3rd step, root cause identification, five kinds of elements should be identified. They are primary difficulty source, problem category, root cause category, near root cause and root cause, as shown in Figure 3-10. These elements for the root cause map form a tree structure. When errors happen, users can go through one path on the tree and find the root cause or near root cause (James and Lee, 2004).

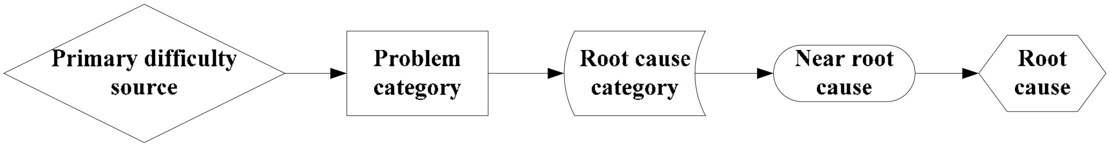


Figure 3-10: Elements in the root cause map and its work process

According to Boniface (1997), items associated with hardware and engineered systems appear toward the left side of the Root Cause Map, while items associated with personnel appear toward the right side of the map. The primary difficulty source at the top of the map for a causal factor is first identified, and then the root cause path is stepped down to each level of the map, working to a root cause.

Shen et al. (1997) proposed some error taxonomies to support root cause analysis. Following a similar approach, according to the expression method and the work process in Figure 3-10, Root Cause Map for RCA in ontology development and utilization is developed in Section 4.6. To determine the root cause of errors in ontology development and utilization, in Section 4.6, an error ontology is developed to represent the error taxonomy and possible management methods for errors, and it can facilitate to find the root cause. By using the ontology with SWRL rules and the associated rule engine, the ontology-based RCA systematically determines the root causes by using a structured approach and uniform reasoning process. Root cause can be easily inferred and located, and working process is expected to be improved in time through the ontology-based method. Over time, the root causes can be used to identify trends for the type of root causes that are occurring (Boniface, 1997).

3.6. Evaluation Methodology for the Research

3.6.1 Related Dependent Variables

The focus of the research is mainly on the ontology development and utilization for knowledge management in product design. Accuracy and efficiency are very important dependent variables since the research objective is to help design products more accurately and efficiently. The accuracy can be divided into the accuracy of the query results and the accuracy of decision making; and the efficiency can be divided into the efficiency of using query results and the efficiency of decision making. Moreover, on the technical level, data consistency of the system outputs and product design quality are closely related to the accuracy, while design cycle time and amount of designers' work are associated with efficiency. Considering the literature reviewed in Section 2.7, dependent variables that are important and need to be measured are listed below.

- Data consistency of the system outputs: Data resources and ontologies would be managed under the uniform information integration framework, and one of the advantages of ontologies is that relations among data and the reasoning capability reduce the logical inconsistency in the ontology (Horridge et al., 2004). These can help assure the data consistency of the output of the system.

- Timeliness and currency of information.

- Accuracy, adequacy and appropriateness of the query results, and improved data availability: Query in the system is based on the semantic relations in the ontology. Since it would facilitate the query engine to locate appropriate information by using relations, the quality of the query result can be guaranteed, and the accuracy of the query results can be enhanced.

- Efficiency of using query results: Good quality of the query results can lead to high efficiency of their usage. With the support of decision support tools in the system, which analyzes the query results, efficiency of using query results can also be improved.

- Accuracy and efficiency of decision making: Since the decision support tool is supposed to facilitate the designers to make decisions, the accuracy and efficiency of decision making are expected to be enhanced.

- Product design quality: With the enhancement of consistency and accuracy of the query results, and with the support of the decision support tool in the system to facilitate the system thinking, the quality of the product design result would be improved.

- Design cycle time: With the efficiency and accuracy of decision making, designers can complete the design faster, and hence the design cycle time would be reduced.

- Amount of designers' work: Previously, data retrieval, filter, analysis and decision making should be done by designers themselves; in the future, ontology-based information systems with the decision support tools will help designers obtain data, analyze data and make decisions. Some repeated work, originally manually done by the designers, will be reduced by the participation of the ontology-based systems and tools.

- User satisfaction and attitudes towards system

- System usage

- System documentation

3.6.2 Related Independent Variables

Martin et al. (2005) made a survey of 129 IS project managers on a broad range of project management issues, and found that IS project size influenced budget and project quality, while project complexity influenced the use of specific project management practices.

Therefore, both project size and the complexity of the problem are chosen as independent variable in the project.

Ontology is used here to retrieve and analyze data in data resources for product design. Therefore, the status of data resources is an important input. It can be represented as changing speed and update level, quality of data resource, and degree of heterogeneity of the data resource space. Independent variables in the research are project size, problem complexity, changing speed of data resources, update level of changes in data resources, quality of data resources and heterogeneity degree of the data resources.

3.6.3 Evaluation of the Research

In the evaluation, questions will focus on the dependent and independent variables in Section 3.6.1 and Section 3.6.2. There are two kinds of surveys, one for system administrators and one for system users. Each survey will have two types of questions: selective questions and open-ended questions. The survey begins with some warm-up questions that retrieve the basic information. The main part of the survey is comprised of several selective questions. In each question, there are 6 options that range from strongly disagree to strongly agree. At the end of the survey, the open-ended question obtains important feedback that is not covered by the previous questions.

Evaluation will be done according to the value of the dependent variables with the consideration of the relationships between dependent and independent variables, which are represented by the method shown in Table 3-2. In the table, “+” means positive effect and “-” means negative effect. As shown in Table 3-2, the bigger the project size, the more aspects related; and the more complex the problem, the more relations among ontology elements and among ontologies need to be considered. Project size can affect project quality (Martin et al., 2005), which corresponds to accuracy of the query results, accuracy of decision making, and product design quality. Problem complexity can affect all the dependent variables except the data consistency. Data consistency should not be influenced by project size or problem complexity, but by variables related to data resources, such as changing speed, update level, and quality and heterogeneity of the data resources. These variables, which are related to data resources, will also influence the accuracy of the query results, efficiency of query results usage, etc. All the independent variables have effect on design cycle time and the amount of designers' work.

Table 3-2: Relationship between dependent and independent variables in evaluation

Independent Variables **Dependent Variables**	**Project size**	**Problem complexity**	**Speed and level of changes in data resources**	**Data resources quality**	**Heterogeneity of the data resources**
Data consistency			-	+	-
Accuracy of the query results	-	-	-	+	-
Efficiency of using query result		-	-	+	-
Accuracy of decision making	-	-	-	+	-
Efficiency of decision making		-	-	+	-
Product design quality	-	-	-	+	-
Design cycle time	+	+	+	-	+
Amount of designers' work	+	+	+	-	+
Data management and improved data availability	-	-	-		-

Other dependent variables in Section 3.6.1 do not have apparent relations with independent variables in Section 3.6.2. During the evaluation, regression, Apriori algorithm and other data mining methods might be used to find the relations between them.

Scoring approach (Power, 2002) can be used to do the numerical comparison. It works similarly to the single dimensioned analysis and involves weighting factors. After the scores are summarized, the projects are ranked.

At present, outcomes like cost are not considered in the evaluation. In the future, if they are to be considered, cost-benefit analysis (CBA) (Power, 2002) can be implemented during evaluation which considers the cost-saving of current processes. Examples of DSS cost factors can be direct hardware and software costs, project personnel cost, support service, process change cost and incremental infrastructure cost, and benefits can be the dependent variables proposed above. Cost–effectiveness analysis (Power, 2002) can also be used, which is a special case of CBA, where alternatives have either the same benefits or the same costs.

Chapter 4: Research Results

This chapter provides results for 4 primary aspects of the work, including: 1) the DFM ontology development for product design and other related ontologies for data integration and decision support 2) the ontology-based data integration, 3) the ontology-based decision support tool, and 4) the error ontology and the error control in this research. In Section 4.1, the overall framework of the ontology-based system is described, and the four aspects are explained in detail in Sections 4.2-4.4.

4.1 Overall Framework of the Ontology-Based System

The framework of the ontology-based method for product design in the research is shown in Figure 4-1. It is divided into three sections, including the ontologies and databases, the web-based e-Design system and the local multi-attribute decision making tool (MADMT). The purpose of the ontologies and databases is to represent knowledge and relations in the domain, get potential relations in databases that come from different software, integrate data from heterogeneous data resources and find possible solutions by using ontologies and ontology-based rule engines; the purpose of web-based e-Design system is to guide user to the proper information during design, provide integrated data based on ontologies and support decision making with the consideration of complex relations in design; and the purpose of the local MADMT is to help compare integrated historical data by using single dimensioned or full dimensioned analysis and find the best historical data as a reference during design.

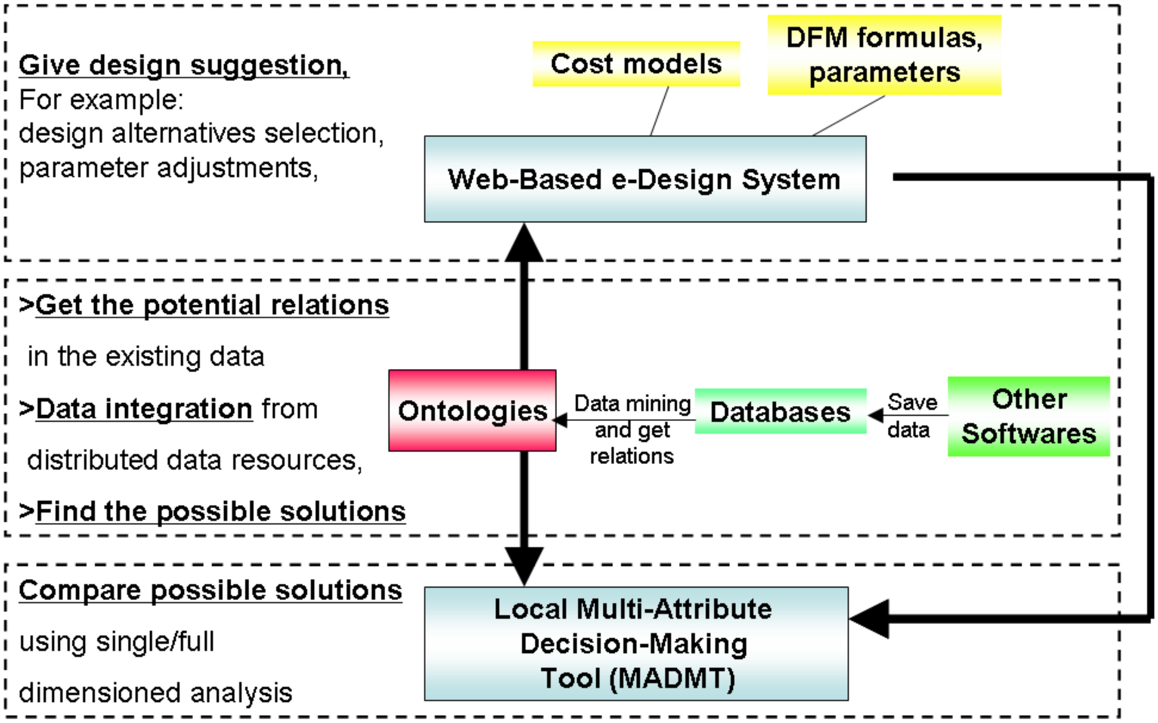


Figure 4-1: Framework of ontology-based method for product design

Ontologies can be used to represent the common understanding in design, integrate heterogeneous data resources and help to find the potential relations hidden in the existing data through data mining. It can represent, store and reuse the relations for further use in product design, and help to consider the problem comprehensively. The DFM ontology has been built to represent the concepts and relations in DFM, support the design process and help integrate data. Other ontologies include the local ontologies for data integration and the DFM e-Design ontology for decision support in the web-based e-Design system. These ontologies will be explained in Section 4.2.

At present, designers are using different computer-based applications, creating more and more data, and saving models, analysis results and data in distributed heterogeneous resources. The ontology-based data integration is explained in Section 4.3. In each design step in the web-based e-Design system, possible solutions from ontologies and different data resources will be retrieved by using the ontology-based data integration method so that integrated data can be exported to a local computer. The local Multi-Attribute Decision-Making Tool (MADMT) can be run on a local personal computer, and help designers finish the comparison and analysis through full dimensioned and single dimensioned analysis, which is described in Section 4.4. By recommending the best alternative based on the comparison, designers can avoid the failure of previous designs and finish the current design more easily and in less time.

The web-based e-Design system for DFM has been built in Section 4.5. It can utilize the relations and existing design alternatives in the ontologies, integrate the cost models and DFM formulas. It helps obtain the final result of the design and offer design suggestions, such as parameter adjustments.

Potential relations in existing data can also be mined from the data resources and saved in the ontology. The ontology and ontology-based tools will further utilize the relations (Chang et al., 2007). At present, the work of this part mainly focuses on the methodology, not the development and realization.

In each part of the work, there might be possible errors existing. The ontology based Root Cause Analysis (RCA) for error control in the research is explained in Section 4.6 to help find the real reason of errors and support the continuous improvement of the system.

4.2 DFM Ontology and Other Ontologies for Data Integration and Decision Support

According to Power (2002, p.144), “a knowledge base is a collection of organized facts, rules and procedures. A knowledge base has a description of the elements in the process along with their characteristics, functions, and relationships. It also contains rules about the actions to implement as a result of certain events.” An ontology with a set of instances of the classes in the ontology is a knowledge base, where an ontology provides a framework for reasoning and a knowledge base allows reasoning about specific data (Loaiza and Wartik, 2005). The developed ontologies in this section are used as the knowledge base of the method.

In the next section, the DFM ontology is first developed, which is the main ontology used in the system and serves as the central ontology for data integration. After that, the local ontologies for data integration are described (Section 4.2.2) and the DFM e-Design ontology, which supports the decision making in the web-based system is explained

(Section 4.2.3).

4.2.1 DFM Ontology

The DFM ontology presented here can be used to capture various manufacturing and assembly concepts and to share joining information along with the assembly geometry at the same time. It is also used as the central ontology in data integration. Local databases can be located by querying the DFM ontology since the DFM ontology serves as the central ontology in data integration. It can also facilitate collaborative design teams in performing various cost and technical evaluations of different facets in DFM for shortened product development cycles. In the following subsections, its hierarchy, important classes, as well as realized relations are explained in detail. The slots in these classes are described with classes and relations realized in this section, and instances in the ontology are described with case studies in Section 5.2.

4.2.1.1 DFM Ontology Hierarchy and Important Classes

Protégé was used in the development of the DFM ontology. The hierarchy of the DFM ontology classes is illustrated in Figure 4-2. In the DFM ontology, there are 10 main classes, and six of the classes are similar to the classes in the AsD ontology (Kim et al., 2006) to represent the knowledge in assembly. These six classes include Product, Part_material, Feature, SpatialRelationship, DegreeofFreedom and Manufacturing.

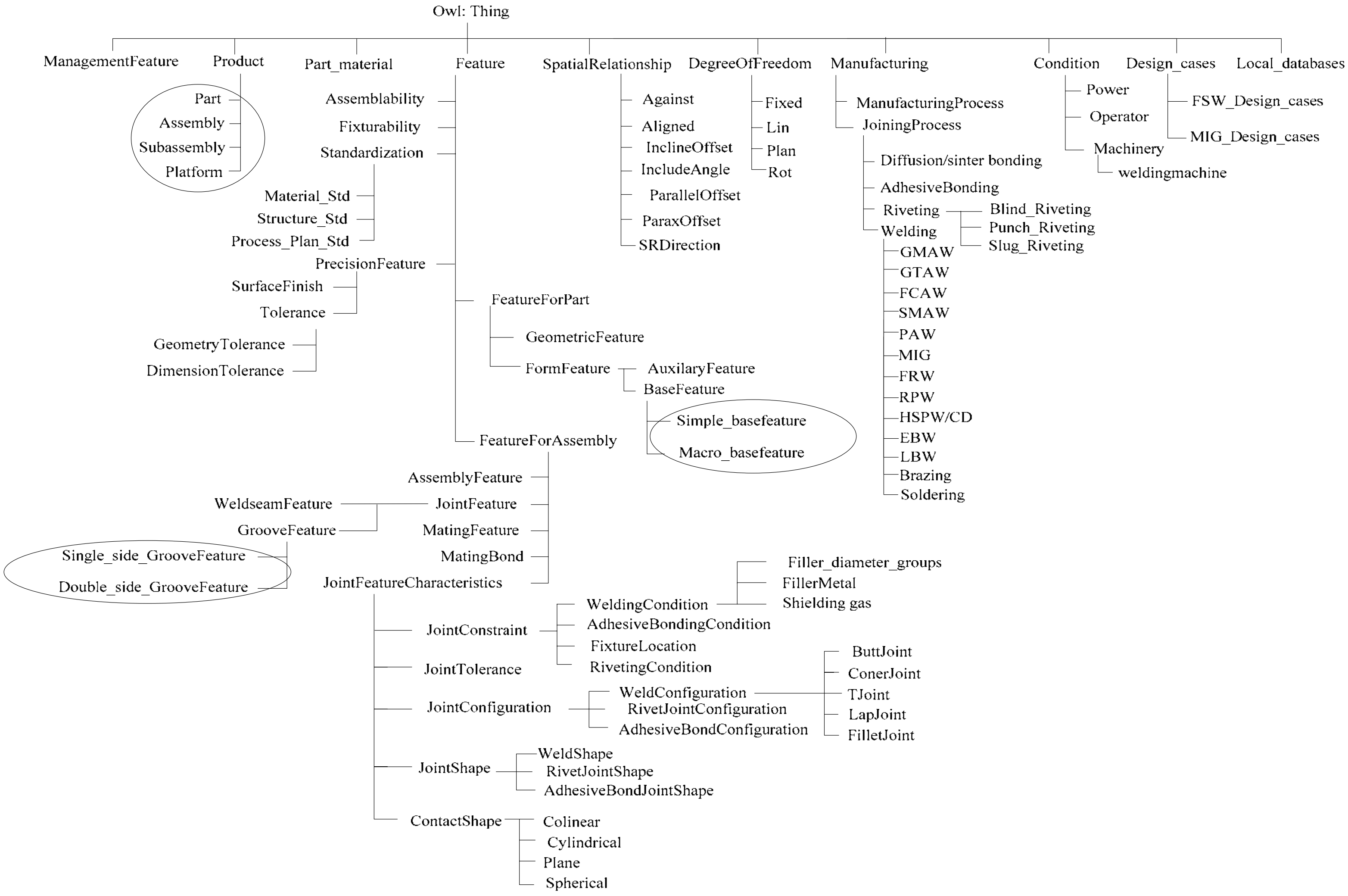


Figure 4-2: Hierarchy of the DFM ontology

Besides the assembly information, other information such as the assembly condition and requirements also needs to be considered in Design for Manufacturing. In the DFM ontology, four new classes, ManagementFeature, Condition, Design_cases and Local_databases, have been added to deal with different aspects of Design for Manufacturing:

- The ManagementFeature class: This class records the basic information of DFM, such as the design time, designer, function of the design (Lin et al., 1999).
- Condition class: Different manufacturing environments can lead to different assembly procedures and assembly costs. The Condition class records information pertinent to the manufacturing environment where the manufacturing will be done, such as power, labor and machinery.
- Design_cases class: The DFM ontology can be used for product design in the ontology editor, Protégé. By using the SWRL rules and Jess rule engine, some design suggestions can be inferred within the SWRL Rules tab. Design_cases class is used to record the specific design condition and inferred result. It will be illustrated in the case study in Sections 5.2.2 and 5.2.3.
- Local_databases class: Since the DFM ontology is also used as the central ontology for data integration, the information related to a data resource, such as database type and database location, will be recorded as an instance in this class.

The overall structure of the DFM ontology is shown in Figure 4-2. In the developed DFM ontology, some classes have a complex hierarchy. There also exist unique interrelationships among classes. These classes and relationships among them are described below.

Feature Class

In AsD ontology (Kim et al., 2006), Feature class is divided into feature for part and feature for assembly. FeatureForPart is then divided into geometric feature and form feature. In Lin et al. (1999), form feature is divided into base feature and auxiliary feature, and the base feature is further divided into simple base feature and macro base feature. In the FeatureForAssembly class in AsD ontology (Kim et al., 2006), there are several subclasses as shown in Figure 4-2. JointFeature is one of the subclasses. According to Lin et al. (1999), joint feature can be categorized into groove feature and weldseam feature.

Besides the feature for part and feature for assembly, precision feature is also considered in Lin et al. (1999). Precision feature subclass is included in the feature class of DFM ontology. It can be used to determine the precision that affects the weld quality during the assembling process. The precision feature is further divided into tolerance and surface finish, and tolerance is divided into geometry tolerance and dimension tolerance.

The technical criteria of manufacturability in Song et al. (1996) are encapsulated by standardization, assemblability, and fixturability subclass under the feature class of the DFM ontology. Feature class of the DFM ontology not only helps collaborative designers to store and extract feature data, but also helps in reducing the difficulty of mapping, converting and exchanging feature and product data.

Manufacturing Class

Manufacturing is classified into manufacturing process and joining process. Joining process is further subdivided into adhesive bonding, riveting and welding.

In the AsD ontology, five classes, GMAW, GTAW, FCAW, SMAW and PAW, belong to the welding class (Kim et al., 2006). According to the ASM Handbook (Olson et al., 1993), there are various other welding types besides the five kinds of welding in the AsD ontology, such as MIG, FRW, RPW, HSPW/CD, EBW, LBW, Brazing and Soldering. In order to comprehensively include various welding and joining technologies and support the technical and economic comparison among different welding methods for a given assembly, the above mentioned welding and joining processes are included under the welding class of the DFM ontology.

In addition, three types of Riveting are added under the Riveting class to make the information more specific. These subclasses are blind riveting, punch riveting and slug riveting.

Local_databases Class

In order to locate the data resources and continue the query transformation process in the data integration part, the properties in this class include the local database type, local database name, local database location, local ontology name, local ontology location, and related instances in the central ontology.

4.2.1.2 Relations Realization in DFM Ontology

There are several formats of ontology that can be built in Protégé. Java Function Calls can be used in "Call Java Attachments" project type, and SWRL rules and Jess rule engine can be built and used in "OWL/RDF files" project format. The ontology in one format can be changed to another. In order to input instances easily, "Call Java Attachments" project type is used to develop the DFM ontology initially, and Java Function Calls programs are plugged into the ontology to help the calculation of some slots during instance input. After the knowledge base is developed, the DFM ontology is exported to OWL format. In the OWL format, the SWRL rules are further developed to facilitate the design, such as feasibility and availability analysis, etc. In this section, the developed Java Function Calls programs are described. SWRL rules will be explained in Chapter 5 with two case studies in Sections 5.2.2 and 5.2.3.

Figure 4-3 represents MIG welding related slots and their relations in DFM ontology, which follows the representation method in Section 3.2. For example, the total cost is obtained through the sum of five cost items. The cost model can be found in Appendix D.

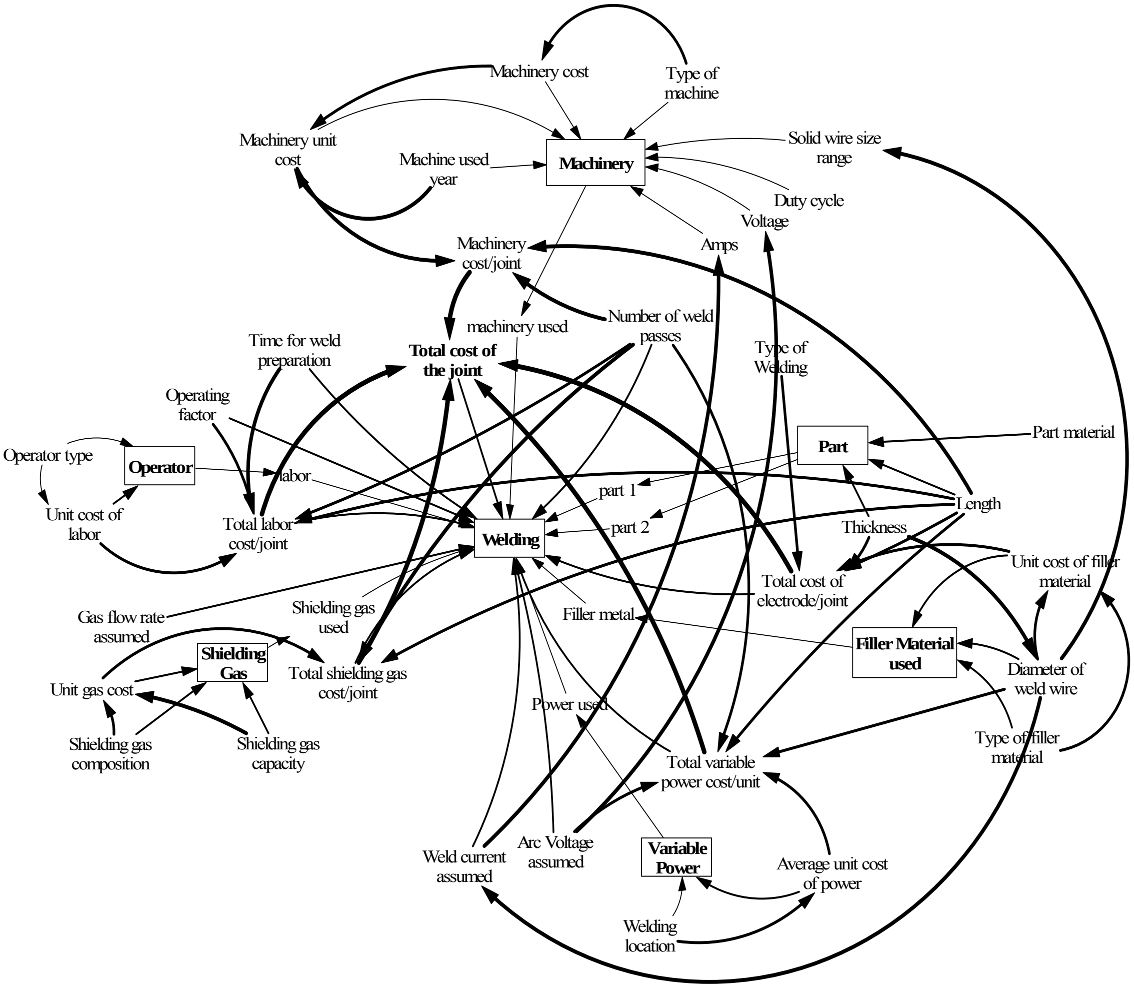


Figure 4-3: MIG Welding related slots and their relations in the DFM ontology

In the current DFM ontology, for the Manufacturing->Joininigprocess->Welding->MIG class, by using the Java Function Calls programs, the total cost per joint can be obtained through the 5 cost items, which are electrode cost, power cost, labor cost, machinery cost and shielding gas cost. In addition, cost items, thickness of base plate, etc. can be calculated automatically according to the input of design parameters.

Moreover, some arrows in Figure 4-3 are the restrictions of slot inputs. During the slot definition in Protégé, users can specify the slot value in a range in general. That means, no matter what the slot value might be for others, the value of this slot should be fixed in this range. But the restrictions here require the variable to be specified in a specific range according to the inputs of other variables. So these restrictions are realized by creating and attaching functions to slots through the Java Function Calls method.

In order to realize the cost model in MIG welding, the programs in Table 4-1 are developed. Similarly, to realize the mathematic relations in the cost model of FSW, programs in Table 4-2 are developed. They are compiled and exported in one java archive file. The cost models for MIG welding and FSW are introduced in Appendix C and D.

Table 4-1: Java Function Calls Programs to realize the cost model for MIG welding

Program	**Realized function**	**Slot where it is used**
eletrodecost.java	Calculate the filler material cost of MIG welding	Total_cost_of_electrode_per_joint($)
laborcost.java	Calculate the labor cost of MIG welding	Total_labor_cost_per_joint($)
machinecost.java	Calculate the machine cost of MIG welding	Machinery_cost_per_joint ($)
powercost.java	Calculate the power cost of MIG welding	Total_variable_power_cost_per_joint($)
shieldinggascost.java	Calculate the shielding gas cost of MIG welding	Total_shielding_gas_cost_per_joint($)
sum2.java	Calculate the total weld cost of MIG welding	Total_weld_cost($)
thickness.java	Get the thick of base plate, which is the lower thickness of the two parts	Thickness_of_base_plate
Divide.java	Get the unit gas cost	Unit_gas_cost($/ft3)
Machineryunitcost.java	Get the unit cost of machine	Calculate the tool cost
Migrequirement.java	Check the requirement of metal thickness, filler material and current as required in Tables D-1 and D-2 of Appendix-D	MIG Design Hint

Table 4-2: Java Function Calls Programs to realize the cost model for FSW

Program	**Realized function**	**Slot where it is used**
FSWlaborcost.java	Calculate the labor cost of FSW	FSW_labor_cost.java
FSWmachinerycost.java	Calculate the machine cost of FSW	FSW_machinery_cost.java
FSWpowercost.java	Calculate the power cost of FSW	FSW_power_cost.java
FSWtimetoweld.java	Calculate the time to weld of FSW	FSW_time_to_weld.java
FSWtoolcost.java	Calculate the tool cost of FSW	FSW_tool_cost.java
FSWtoollife	Calculate the tool life of the machine in FSW	FSW_tool_life
Sum.java	Calculate the total cost of FSW	FSW_total_cost

Moreover, during the welding design, some restrictions, such as electricity unit cost, machinery unit cost, etc. need to be considered with the environment, such as the location, life of machine, etc. In Appendix C and D, there are several tables that record these restrictions, as shown in Table 4-3, column 1. In traditional databases, these relations can be saved in several tables. In the ontology-based method, information in one table can be saved in several instances in a class. Each column of these tables corresponds to a slot in the class, and each line of the tables corresponds to an instance in the class. The restrictions and their related classes (where the instances are saved) are shown in Table 4-3.

Table 4-3: Restrictions and their related classes in the DFM ontology

Restrictions	**Classes in the DFM ontology**
Table C-3 Average electric power cost in different states of USA	Condition/Power
Table C-2 MARC's depreciation table	Don't save in class, but realized in Java Function Calls programs.
Table D-3 Few MIG welding machines costs and capabilities	Condition/Machinery
Table D-4 Commonly used filler material type and their price	Feature/FeatureForAssembly/JointFeatureCharacteristics/JointConstraint/WeldCondition/Filler Metal
Table D-5 Commonly used shielding gas proportion, quantities and their price	Feature/FeatureForAssembly/JointFeatureCharacteristics/JointConstraint/WeldCondition/Shielding_Gas
Tables C-1 and D-6 Average hourly wage of a labor	Condition/Operator

In the MIG and FSW cost model, the choices of parameters, such as filler material, current and machines, are limited to some limited number of values. So Java Function Calls Programs in Protégé can be used for the enumeration of these parameters to find the best design alternative with the lowest total welding cost. This method is efficient and easy to be used in cases with limited number of choices in each parameter. For other cases with parameters in some ranges or in a series of integers, models and methods in Operations Research, like linear programming, nonlinear programming or integer programming can be used.

4.2.2 Local Ontologies for Data Integration

In its current form, the ontology-based data integration has three kinds of data resources integrated, including: an Excel file, an Access database and a Mysql database. Each local ontology corresponds to one data resource. The task of these local ontologies is to map the attributes searched to the proper tables and proper attributes in these tables in the data resource. These local ontologies have the same structure and include three classes, which are:

- Central_ontology_attribute class: an instance in this class records a property in the central ontology;
- database_attribute class: an instance in this class records a property in one or some of the tables in the data resource;
- Table class: an instance in this class is a table in the data resource.

The relations between classes and slots in local ontologies are shown in Figure 4-4.

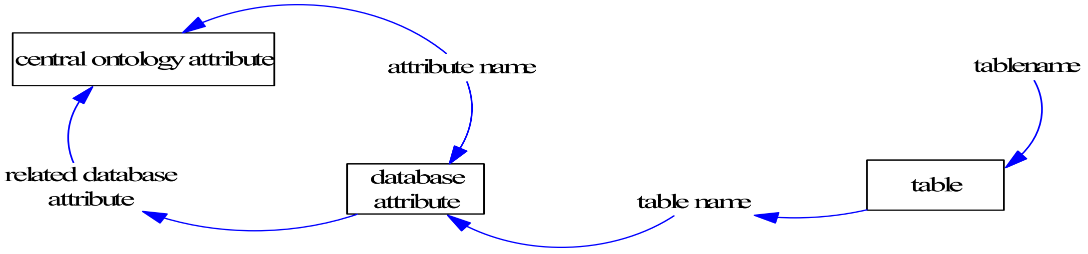


Figure 4-4: Relations among classes and slots in local ontologies

4.2.3 DFM e-Design Ontology for Decision Support

4.2.3.1 Ontology Structure

In a system, a variable can be affected directly or indirectly by other factors, so influence factors can be divided into direct influence factors and root influence factors. Direct influence factors of variable x are factors that directly affect x, and root influence factors of variable x are the input variables that are the root cause of the changing both in x and in the direct influence factors of x. Among the direct influence factors, some can increase x when they are increased (positive relation), others can decrease x when they are increased (negative relation). Some factors have other logical relations with variable x. For example, the density is decided by the component material, and this relation is not a positive or negative relation. So as shown in Table 4-4, in the DFM e-Design ontology, direct_influence_factor and root_influence_factor are two properties. Sub-properties in the direct_influence_factor are positive_direct_influence, negative_direct_influence and other_direct_influence. Similarly, sub-properties in the root_influence_factor are positive_root_influence, negative_root_influence and other_root_influence. Moreover, there is another slot complex_root to deal with complex relations during inference of the Jess rule engine. The usage of these slots is described later in this section.

Table 4-4: Important properties and sub-properties in the DFM e-Design ontology

Property	**Sub-property**	**Domain**	**Range**
direct_influence_factor	positive_direct_influence	Output, Input	Output, Input
	negative_direct_influence		
	other_direct_influence		
root_influence_factor	positive_root_influence	Output, Input	Input
	negative_root_influence		
	other_root_influence		
Complex_root		Output	Input

In addition, there are two classes in the DFM e-Design ontology, which are Input and Output. Their properties are shown in Table 4-5.

Table 4-5: Classes and their properties in the DFM e-Design ontology

Class	**Property**	**Type of the property**
Output	Var_name	String
	direct_influence_factor	Instances of Output or Input
	positive_direct_influence	
	negative_direct_influence	
	other_direct_influence	
	root_influence_factor	Instances of Input
	positive_root_influence	
	negative_root_influence	
	other_root_influence	
	complex_root	
Input	*Same as the class input except does not have complex_root slot.*	

4.2.3.2 SWRL Rules with its Usage in the Jess Rule Engine

When instances are input into the DFM e-Design ontology, it is assumed that any independent input variable has just one root influence factor, which is itself, because it is not influenced by other variables. In addition, the numerical independent inputs can increase or decrease with itself, i.e., it is defined to have a positive relation with itself. There is no input that has itself as negative_direct_influence or negative_root_influence. Moreover, it is also assumed that no output of the system can influence an input of the system, so the influence factors of inputs should not be any output variable.

After the variable name and related directed influence factors are input in the ontology, the root influence factors can be reasoned out by SWRL rules and the Jess rule engine. The related SWRL rules are shown in Table 4-6. The information of the direct and root influence factors for each variable will be used in the ontology-based web system for DFM in Section 4.5 to give users suggestions of design input adjustment.

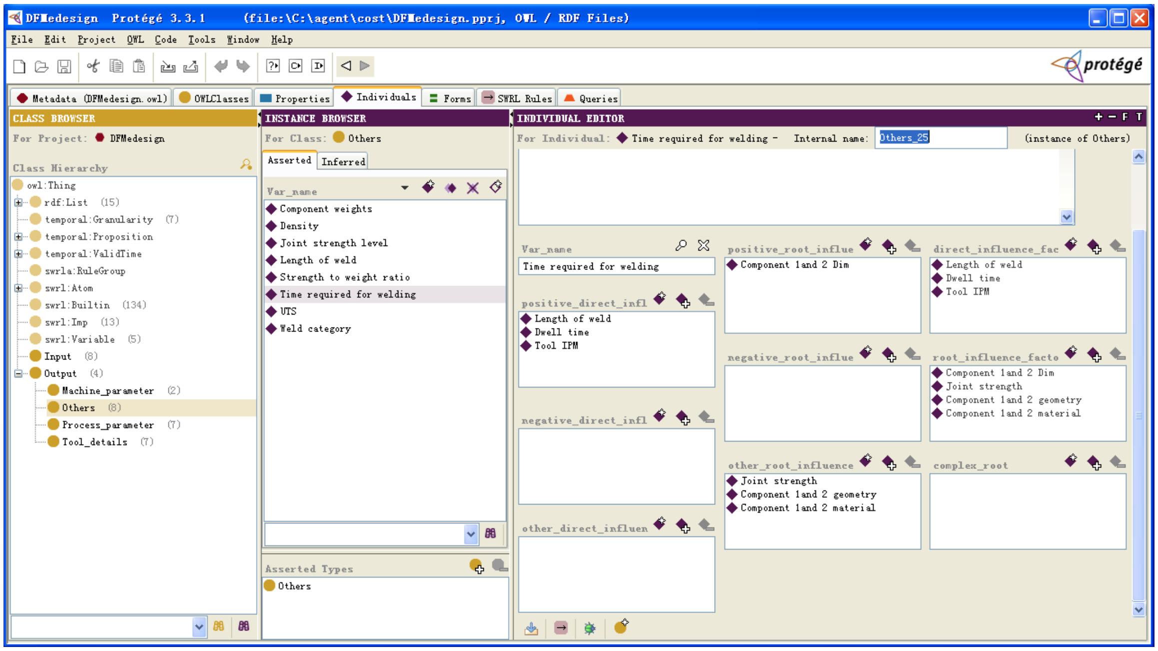
Table 4-6: SWRL rules in the DFM e-Design ontology to infer root influence factors

Name	**Rule**
Rule-1	Input(?x) ∧ positive_direct_influence(?x, ?y) ∧ Var_name(?x, ?x1) ∧ Var_name(?y, ?y1) ∧ swrlb:equal(?x1, ?y1) → positive_root_influence(?x, ?y)
Rule-2	Input(?x) ∧ other_direct_influence(?x, ?y) ∧ Var_name(?x, ?x1) ∧ Var_name(?y, ?y1) ∧ swrlb:equal(?x1, ?y1) → other_root_influence(?x, ?y)
Rule-3	Input(?x)∧positive_direct_influence(?x, ?y)∧positive_root_influence (?y, ?z) → positive_root_influence(?x, ?z)
Rule-4	Input(?x)∧positive_direct_influence(?x, ?y)∧negative_root_influence (?y, ?z) → negative_root_influence(?x, ?z)
Rule-5	Input(?x)∧positive_direct_influence(?x, ?y)∧other_root_influence (?y, ?z) → other_root_influence(?x, ?z)
Rule-6	Input(?x)∧negative_direct_influence(?x, ?y)∧positive_root_influence (?y, ?z) → negative_root_influence(?x, ?z)
Rule-7	Input(?x)∧negative_direct_influence(?x, ?y)∧negative_root_influence (?y, ?z) →positive_root_influence(?x, ?z)
Rule-8	Input(?x)∧negative_direct_influence(?x, ?y)∧other_root_influence (?y, ?z) → other_root_influence(?x, ?z)
Rule-9	Input(?x)∧other_direct_influence(?x, ?y)∧root_influence_factor(?y, ?z) → other_root_influence(?x, ?z)
Rule-10	Output(?x) ∧ other_direct_influence(?x, ?y) ∧ root_influence_factor(?y, ?z) → other_root_influence(?x, ?z)
Rule-11	Output(?x) ∧ positive_direct_influence(?x, ?y) ∧ positive_root_influence(?y, ?z) → positive_root_influence(?x, ?z)
Rule-12	Output(?x) ∧ positive_direct_influence(?x, ?y) ∧ negative_root_influence(?y, ?z) → negative_root_influence(?x, ?z)
Rule-13	Output(?x) ∧ positive_direct_influence(?x, ?y) ∧ other_root_influence(?y, ?z) → other_root_influence(?x, ?z)
Rule-14	Output(?x) ∧ negative_direct_influence(?x, ?y) ∧ positive_root_influence(?y, ?z) → negative_root_influence(?x, ?z)
Rule-15	Output(?x) ∧ negative_direct_influence(?x, ?y) ∧ negative_root_influence(?y, ?z) → positive_root_influence(?x, ?z)
Rule-16	Output(?x) ∧ negative_direct_influence(?x, ?y) ∧ other_root_influence(?y, ?z) → other_root_influence(?x, ?z)
Rule-17	Output(?x) ∧ positive_root_influence(?x, ?y) ∧ negative_root_influence(?x, ?y) → complex_root(?x, ?y)
Rule-18	Output(?x) ∧ complex_root(?x, ?y) ∧ Output(?z) ∧ direct_influence_factor(?z, ?x) → complex_root(?z, ?y)

Rules 1-9 obtain the corresponding root influence factors of inputs, according to the positive, negative and other direct influence factors of the variable. Rules 10-16 infer the corresponding root influence factors of the outputs. Among them, Rules 1 and 2 retrieve the root influence for independent inputs. Rules-3, 4, 6, 7, 11, 12, 14, 15 follow a similar principle. For example, y is the positive influence factor of x, and y has a positive root influence factor z. Increasing z can increase y and then increase x. So z is a positive root influence factor of x (Rule-3 and Rule-11). Rules-5, 8, 13, 16 mean that the other root influence factor of a positive or negative direct influence factor for x will be the other root influence for x. Rules 9 and 10 express that the root influence factor of the other direct influence factor for variable x will be the other root influence for x. Rule-17 and Rule-18 are built to identify a complex situation. Once a root influence becomes both positive and negative root influence factors of variable x, the ultimate relation between the root influence factor and x becomes complex. It can be positive in one situation and negative in another. Rule-17 identifies this situation and saves the root influence factor into the complex_root slot in the instance. In addition, output ?x may have a complex root and also affect other connected outputs. Hence, the complex_root of output ?x is also the complex_root of the outputs that are directly or indirectly affected by output ?x (Rule-18).

The developer for the knowledge base of this ontology can search instances whose complex_root slot is not empty by using the method in Section 3.3.1. After that, the developer can manually delete the positive and negative root influence that also exist in complex_root, and add it into the other_root_influence slot, which represents complex relations.

SWRL (Semantic Web Rule Language) rules are used with the Jess rule engine to reason out the possible root influence factors of inputs and outputs, as shown in Figure 4-5. Based on the ontology with inferred result, the reasoner can query the knowledge base of the ontology and give designers suggestions on how to adjust an output and what is the effect of this adjustment. This is explained in Section 4.5.



For Project: DFMedesign

Class Hierarchy
- owl:Thing
  - rdf:List (15)
  - temporal:Granularity (7)
  - temporal:Proposition
  - temporal:ValidTime
  - swrla:RuleGroup
  - swrl:Atom
  - swrl:Builtin (134)
  - swrl:Imp (13)
  - swrl:Variable (5)
  - Input (8)
  - Output (4)
    - Machine_parameter (2)
    - Others (8)
    - Process_parameter (7)
    - Tool_details (7)

For Class: Others

Var_name
- Component weights
- Density
- Joint strength level
- Length of weld
- Strength to weight ratio
- Time required for welding
- UTS
- Weld category

Asserted Types
- Others

For Individual: Time required for welding - Internal name: Others_25 (instance of Others)

Var_name: Time required for welding

positive_direct_infl
- Length of weld
- Dwell time
- Tool IPM

negative_direct_infl

other_direct_influen

positive_root_influe
- Component 1and 2 Dim

negative_root_influe

other_root_influence
- Joint strength
- Component 1and 2 geometry
- Component 1and 2 material

direct_influence_fac
- Length of weld
- Dwell time
- Tool IPM

root_influence_facto
- Component 1and 2 Dim
- Joint strength
- Component 1and 2 geometry
- Component 1and 2 material

complex_root

Figure 4-5: Screenshot of the reasoning result in the DFM e-Design ontology

4.3 Ontology-Based Data Integration

After the central and local ontologies are built, the ontology-based data integration can be realized. In the cost management system built by our previous work, the data query guidance module in Section 3.3.3 queries the central ontology and guides users from activity cost to the resource cost at the end to find the high cost reason, and then the following modules in Section 3.3.3 work to locate data resources and integrate data. In the present DFM e-Design system, according to the requirements from CFSP, the guidance has been specified to the seven-step inputs and outputs, so the query guidance module is not connected with central ontology here.

The overall working process of ontology-based data integration and analysis can be summarized as shown Figure 4-6. The first five nodes in Figure 4-6 are developed by using JSP, and the last node is the local MADMT (Multi-Attribute Decision-Making Tool) developed by using Java, which is described in Section 4.4.

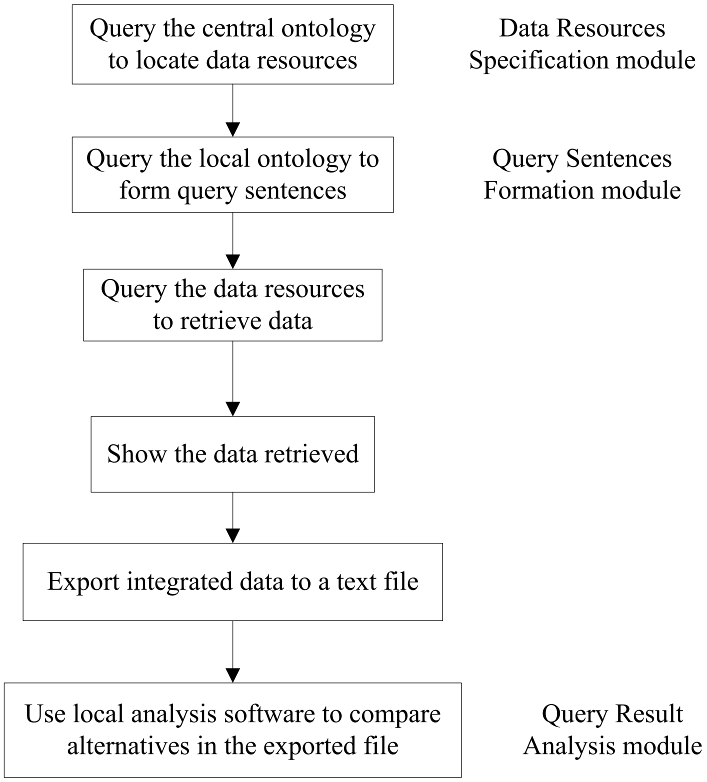


Figure 4-6: Working process of ontology-based data integration and analysis

In order to search the data in the heterogeneous data resources, relations between the central ontology and local ontologies, as well as the relations between local ontologies and the corresponding data resources should be built up. These relations are shown in Figure 4-7.

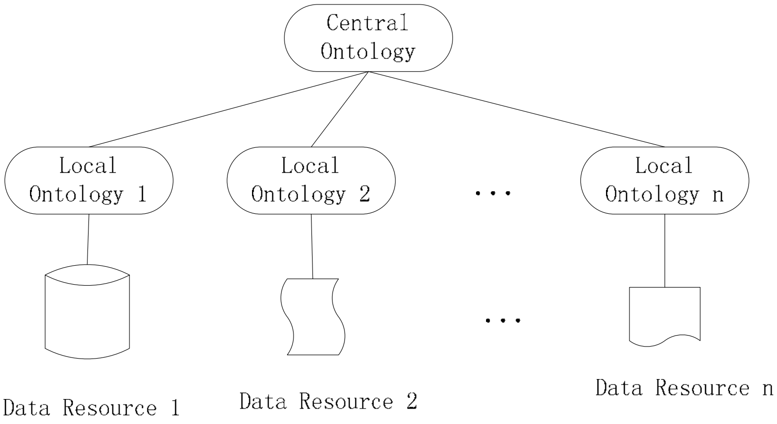


Figure 4-7: Relations among central ontology, local ontologies and data resources

In the ontology-based data integration, the DFM ontology is used as the central ontology. The related class in the DFM ontology is local_databases and its slots are: Related_attributes_in_central, Related_instances, ontology_location, ontology_name, databasetype, and databaselocation.

At present, the realized system can integrate data from Excel files, Access databases and mysql databases. For other formats, such as text documents, ontology-based information retrieval system by Varga et al. (2003) can be used to mine knowledge content of unstructured textual information into XML structures (content objects), and ontology-based framework for XML semantic integration by Cruz et al. (2004) can be used to integrate the data from XML documents. According to the requirements of the present collaborative project, text documents and XML documents integration are not integrated in the realized system, but it can be added into the system in the extended project in the future.

The central ontology is first queried to find the information about the related local ontologies and data resources. And the local ontologies are queried to get the related tables and attribute information in the tables of the data resources. The information is queried by the JTP reasoner and sent to the query sentences formation module.

The query sentence formation module forms the query sentences according to the type of databases or files.

- For Excel files: SELECT <names of the slots or * for all attributes> FROM <[table$]> WHERE <additional condition>;
- For mysql: SELECT <names of the slots or * for all attributes> FROM <table> WHERE <additional condition>;
- For Access: SELECT <names of the slots or * for all attributes> FROM <table> WHERE ([<table>].<additional condition>);

The database query sentences are formed by replacing the variables in <> from the original query sentence with the information retrieved from the corresponding local ontologies. A database query sentence has the same semantics as its original query. The working mechanism is similar to the operators for query reformulation in Arens et al. (1996).

For example, an instance in the local_databases class in the DFM ontology is shown in Table 4-7.

Table 4-7: An instance in the local_databases class in the DFM ontology

Slot name	**Slot value**
Related_attributes_in_central	Material_name
Related_instances	AA6016-T4, AA6082-T4, 6111-T4, 6061-T6, 6063-T7, 6063,6111, 6005-T6, 6061
ontology_location	http://www.enge.vt.edu/terpenny/Smart/localonDFM.owl
ontology_name	localonDFM
databasetype	Access
databaselocation	C:\DFMtool.mdb

For instance, when the information related to 6061-T6 is required, the JTP reasoner will query the DFM ontology, find 6061-T6 is a slot value of the slot “Material_name” in the central ontology. As shown in Table 4-7, the data resource specification module finds that the related local ontology is localonDFM, the location of the ontology is http://www.enge.vt.edu/terpenny/Smart/localonDFM.owl, and the related date resource is at C:\DFMtool.mdb and the type of the database is Access. Then the local ontology localonDFM.owl is loaded and queried.

In the local ontology, there is an instance in central_ontology_attribute class with Material_name as its attribute_name and Comp_Mat as its related_database_attribute, and the attribute Comp_Mat is in the 6000series table. To summarize, the reasoners retrieved the following information:

Database: C:\DFMtool.mdb,

Table: 6000series

Condition: Comp_Mat=’6061-T6’.

Since the material has the unique name, the value of the material, ‘6061-T6’, does not need to be replaced by another similar value. Since the database is Access, the query sentence formed at the end is:

Select * from 6000series where 6000series.Comp_Mat=’6061-T6’;

According to the type of resources, the format of the connection has little difference. For example, the mysql database has the username and password in order to access it.

After query the corresponding data resources, the related data can be output to the local computer as a text file. Then the local Multi-Attribute Decision-Making Tool (MADMT), as explained in the next section, can be run on a local computer to import this text file and finish the single-dimensioned and full-dimensioned analysis for users. The case study of retrieving and analyzing integrated data is provided in Section 5.2.4.

4.4 Local Multi-Attribute Decision-Making Tool (MADMT) Using Integrated Data

The flowchart of local Multi-Attribute Decision-Making Tool (MADMT) is shown in Figure 4-8. It can analyze data integrated from heterogeneous resources (Section 4.3) or data exported from Protégé (Section 3.3.1). The import module reads the imported text file and put it into a matrix. Then the data can be normalized in the next module. MDAMT assumes that in an attribute, the bigger the better. It uses formula (2) in Section 2.5.3.1 to convert the column to a common measurement scale. If a smaller value in an attribute represents a better value, after specifying it, MADMT can also normalize it using the same method. The best normalized value in an attribute is 1, and the worst is 0. They are marked by MADMT with different colors.

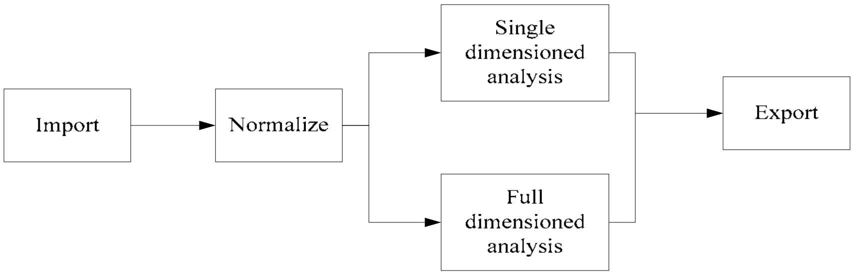


Figure 4-8: Flowchart of local MADMT

After the imported data is converted to a common measurement scale, if single dimensioned analysis module is selected to be used, the weight value, rank or the α value introduced in Section 2.5.3.1 can be input, and MADMT will calculate the weighted sum and output the result from best to the worst. If full dimensioned analysis is selected, after specifying the sequence of the criteria which represents the relative importance to the problem, MADMT can order the alternatives from best to worst according to the normalized value of imported data as well as the importance of the attributes. Both single and full dimensioned analysis modules put the best alternative at the first line to let designers refer to it in their design.

At the end, the original data, normalized data, single dimensioned analysis result and full dimensioned analysis result can be exported to the local computer for further use.

The working mechanism of MADMT on a local computer is shown in Figure 4-9. MADMT can import the integrated data that comes from heterogeneous data resources, as described in Section 4.3, as well from an ontology. If the imported data is from an ontology, the ontology can be on a remote server or on the local computer. In Protégé, the ontology editor, the query tab can search the instances according to the query requirements. These query requirements can also be saved in the query library of the ontology, as explained in Section 3.3.1. In this query tab, feasible design alternatives can be queried out and output to a text file in the local computer. The case in Section 5.2.4 illustrates the work in Sections 4.3-4.5.

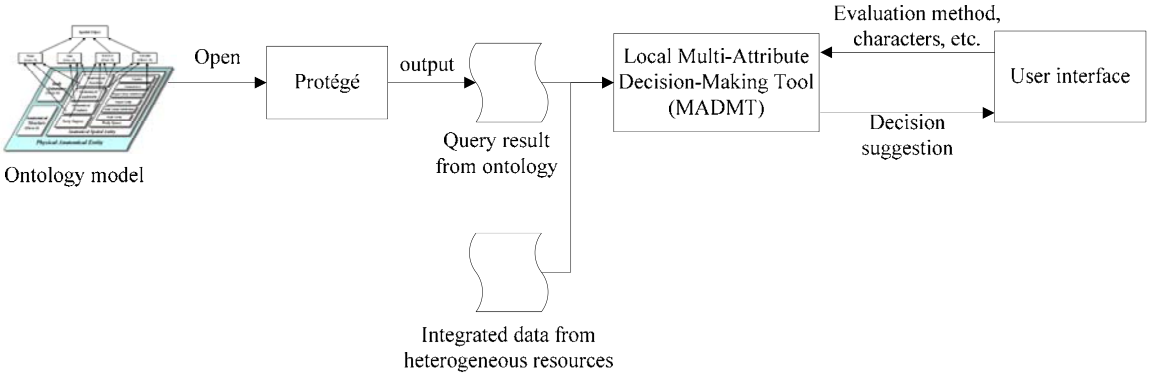


Figure 4-9: Working mechanism of MADMT

4.5 Ontology-Based Web System for DFM

This system is based on the Browser/Server (B/S) mode, and it is developed by using JSP. According to Bennett (1983), for a nontechnical user, the design of an appropriate user interface is the most important determinant of the success of a decision support implementation. The B/S mode platform in the research is supported by the ontologies, and it provides information to users in a more flexible format. The advantage of the B/S mode is that users do not need to be familiar with the ontologies and the structure of the domain knowledge. This helps ensure the implementation of the system in the future. The working mechanism of this platform is shown in Figure 4-10. The JTP reasoner is used to query the knowledge base of the ontologies and obtain the proper information, as explained in Section 3.3.2. The case in Section 5.2.4 is the usage of this working mechanism.

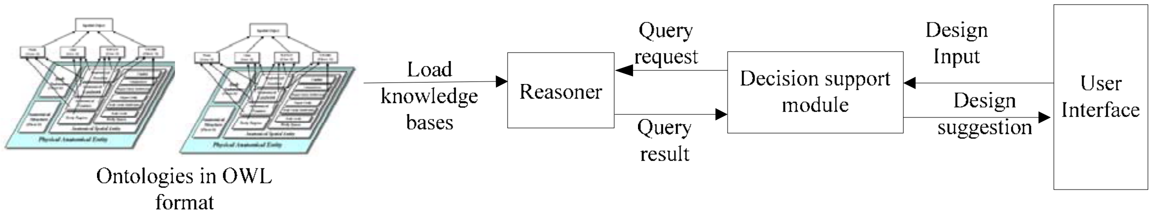


Figure 4-10: Working mechanism of the decision support platform on B/S mode

The flowchart of the ontology-based web system for DFM is shown in Figure 4-11. The first eight nodes are given by Bagaitkar (2007), which include six input steps, one calculation step and one output. In Figure 4-11, the input nodes are all marked with shaded background, and the calculation and other processing by the JSP programs are marked with an underline. Please refer Appendix E for formulas of output calculations.

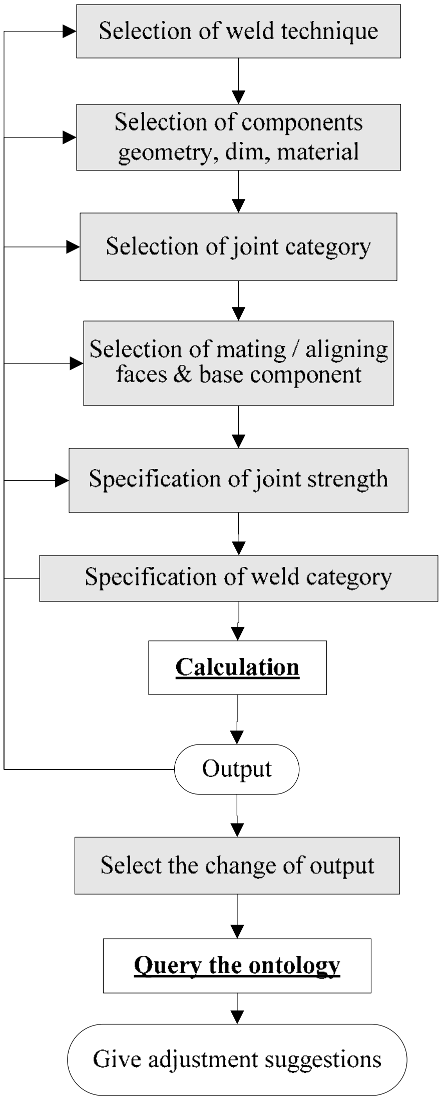


Figure 4-11: Flowchart of the ontology-based web system for DFM

After the output variables are calculated according to the inputs, some of the variables are expected to be adjusted, which is the last three nodes in Figure 4-11. Since complex relations are expected to exist in this situation, according to system dynamics, changing one input can influence multiple outputs in the unexpected direction. So the ontology described in Section 4.2.3 is used to give users hints on parameter adjustment during the design. Designers can go back to the previous steps and make the corresponding adjustment after considering the results and adjustment suggestions.

In order to give design suggestions, such as increase or decrease a variable, the relations between variable should first be analyzed. Figure 4-12 represents the relations among inputs and outputs variables. A->B in Figure 4-12 means variable A has an influence on variable B, and the +/- on the arrows represent the positive and negative impact of A on B. If there is no +/- mark on an arrow, the two variables connected by the arrow have a relation other than positive or negative impact.

For example, in appendix E, the joint strength level is categorized as high strength joints (HSJ), medium strength joints (MSJ) and low strength joints (LSJ) based on fraction of UTS of the parent material.

- HSJ: Joint strength >= 75% of parent material UTS
- MSJ: 75 % of parent material UTS > Joint strength >= 50% of parent material UTS
- LSJ: Joint strength < 50% of parent material UTS)

Joint strength level is decided by the ratio of joint strength and UTS, and the higher the ratio the higher the joint strength level. So joint strength has positive direct influence on joint strength level, and UTS has negative direct influence, as shown in Figure 4-12.

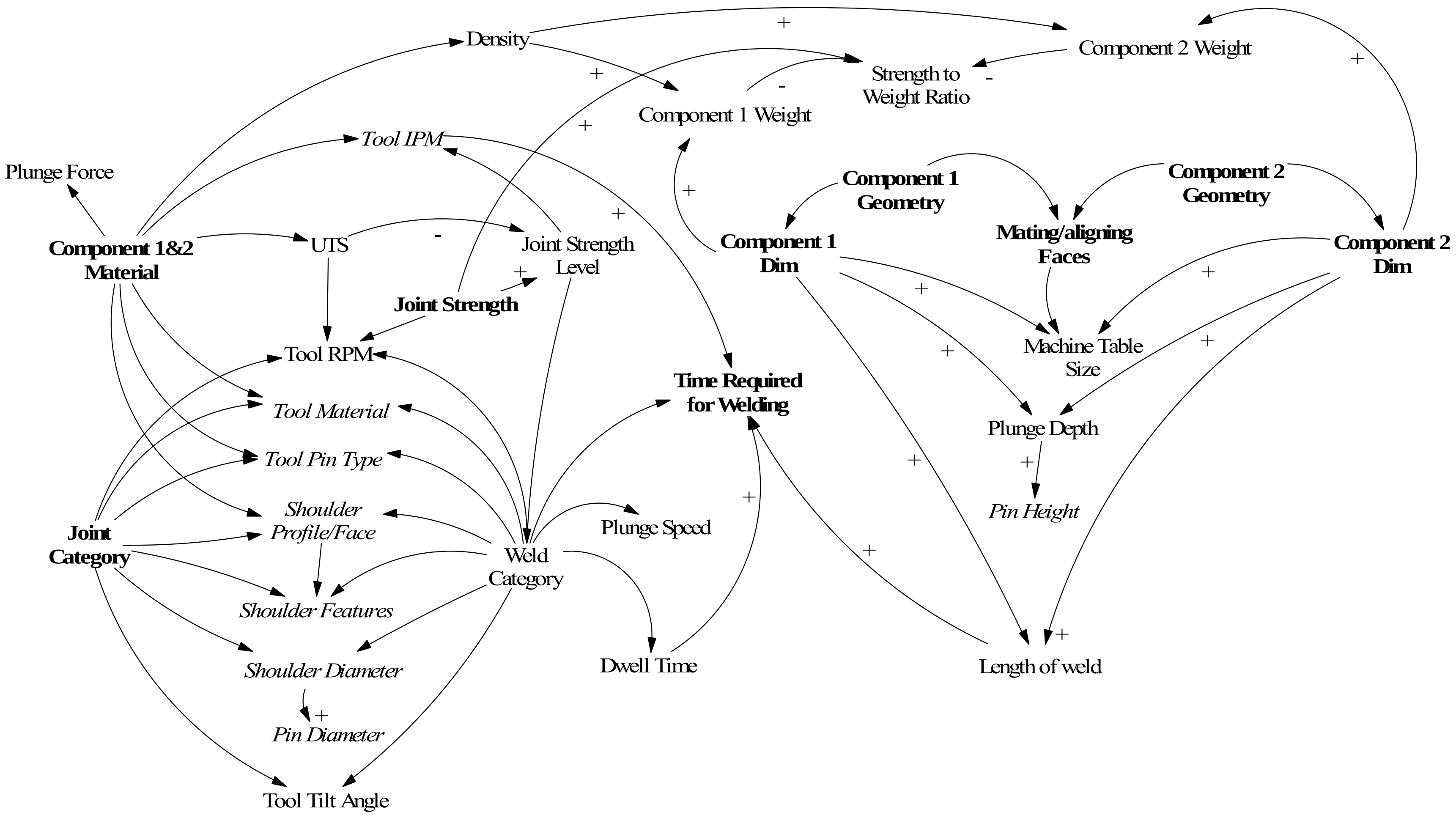


Figure 4-12: Variables and their relations in the ontology-based web system for DFM

As stated in Appendix E, these variables are categorized as input variables, tool details, process parameter, machine parameter, and others. They are described in Table 4-8 and Table 4-9, Column 2. As shown in Figure 4-12, some variables directly influence some other variables, as described in Table 4-8 and Table 4-9, Column 3. And when the variables are traced in the reverse direction of the arrows, the root influence factors of the beginning variable can be found, as shown in Table 4-8 and Table 4-9, Column 4. As assumed in Section 4.2.3.2, any independent variable in input has a root influence factor, which is itself, so any independent variable has itself as the root influence factor in Table 4-8.

Table 4-8: Input variables and their influence factors

Category	**Variables**	**Direct influence factor**	**Root influence factor**
Input	Component 1 and 2 geometry	Component 1 and 2 geometry	Component 1 and 2 geometry
	Component 1 and 2 Dim	Component 1 and 2 Dim, Component 1 and 2 geometry	Component 1 and 2 Dim, Component 1 and 2 geometry
	Component 1 and 2 material	Component 1 and 2 material	Component 1 and 2 material
	Joint Category	Joint Category	Joint Category
	Mating/aligning faces selection	Component 1 and 2 geometry	Component 1 and 2 geometry
	Joint strength	Joint strength	Joint strength

Table 4-9: Output variable category and influence factors

Category	**Variables**	**Direct influence factor**	**Root influence factors**
Tool details	Tool Material	Component 1 and 2 material, Joint category, weld category	Component 1 and 2 material, Joint category, joint strength
	Tool pin type		
	Shoulder profile/face		
	Shoulder feature	Joint category, weld category	
	Shoulder diameter		
	Pin diameter	Shoulder diameter	
	Pin height	Plunge depth	Component 1 and 2 dim, geometry
Process parameter	Tool tilt angle	Joint category, weld category	Component 1 and 2 material, Joint category, joint strength
	Tool RPM	Joint strength level, weld category, joint category	
	Plunge depth	Component 1 and 2 dim	Component 1 and 2 dim, geometry
	Plunge speed	Weld category	Component 1 and 2 material, Joint strength
	Tool IPM	Component 1 and 2 material, joint strength level	
	Dwell time	Weld category	
	Plunge force	Component 1 and 2 material	Component 1 and 2 material
Machine parameter	Machine degrees of freedom (constant)		
	Machine table size	Component 1 and 2 dim	Component 1 and 2 dim, geometry
Others	UTS	Component 1 and 2 material	Component 1 and 2 material
	Joint strength level	UTS, joint strength	Component 1 and 2 material, joint strength
	Weld category	Joint strength level	
	Density	Component 1 and 2 material	Component 1 and 2 material
	Component 1 and 2 weight	Density, component 1 and 2 dim	Component 1 and 2 material, dim, geometry
	Strength to weight ratio	Component 1 and 2 weight, joint strength	Component 1 and 2 material, dim, geometry, joint strength
	Length of weld	Component 1 and 2 dim	Component 1 and 2 dim, geometry
	Time required for welding	dwell time, Tool IPM and length of weld	joint strength, component 1 and 2 material, dim and geometry

The query and reason process in the parameter adjustment part is shown in Figures 4-13 and 4-14. The nodes with text underlined are the querying and reasoning process based on ontology, and nodes with grey background are those output to the user interface. After specifying the output to be adjusted and the direction (increase or decrease), the DFM e-Design ontology (Section 4.2.3) will be queried to obtain the positive, negative and other root influence factor, and the adjustment suggestion can be given. For example, when an output is to be increased, the system will query the ontology and suggest to either increase the positive root influence, decrease its negative root influence or adjust other root influence. When users select the parameter they intend to adjust, variables that might be influenced will also be identified. For example, an output is to be increased, and one of its positive root influence x is selected, the system will query the ontology and find that this adjustment will increase variables with x as the positive influence root.

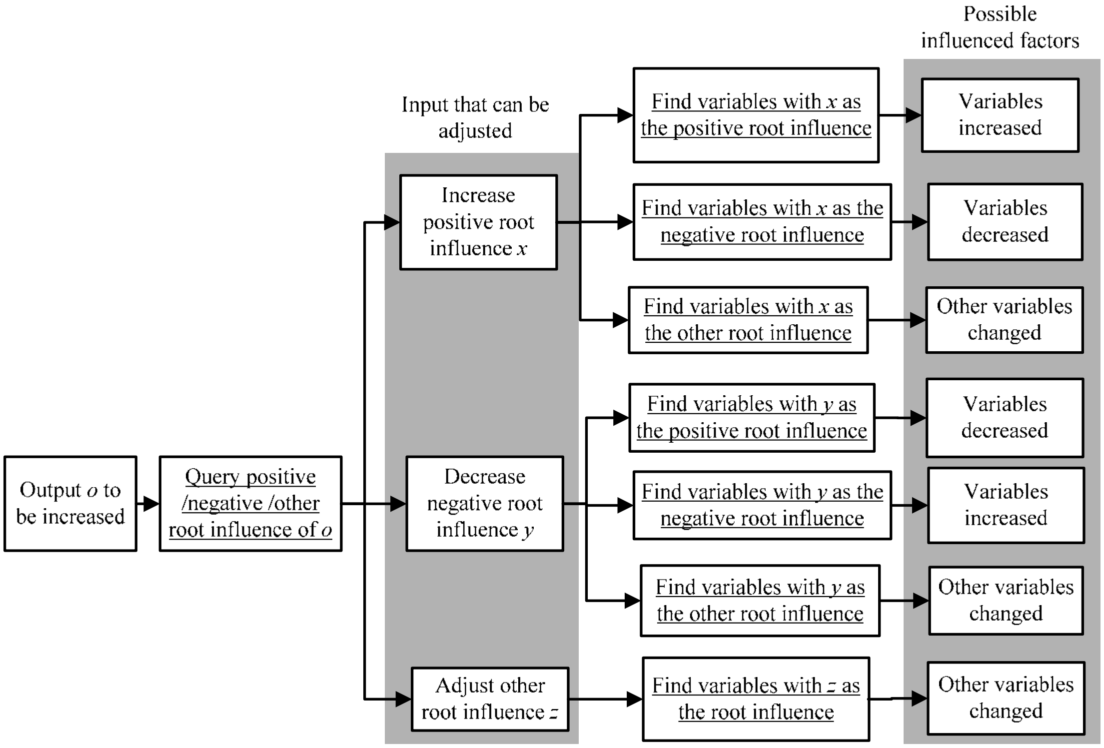


Figure 4-13: Ontology-based query and reason process for output increase

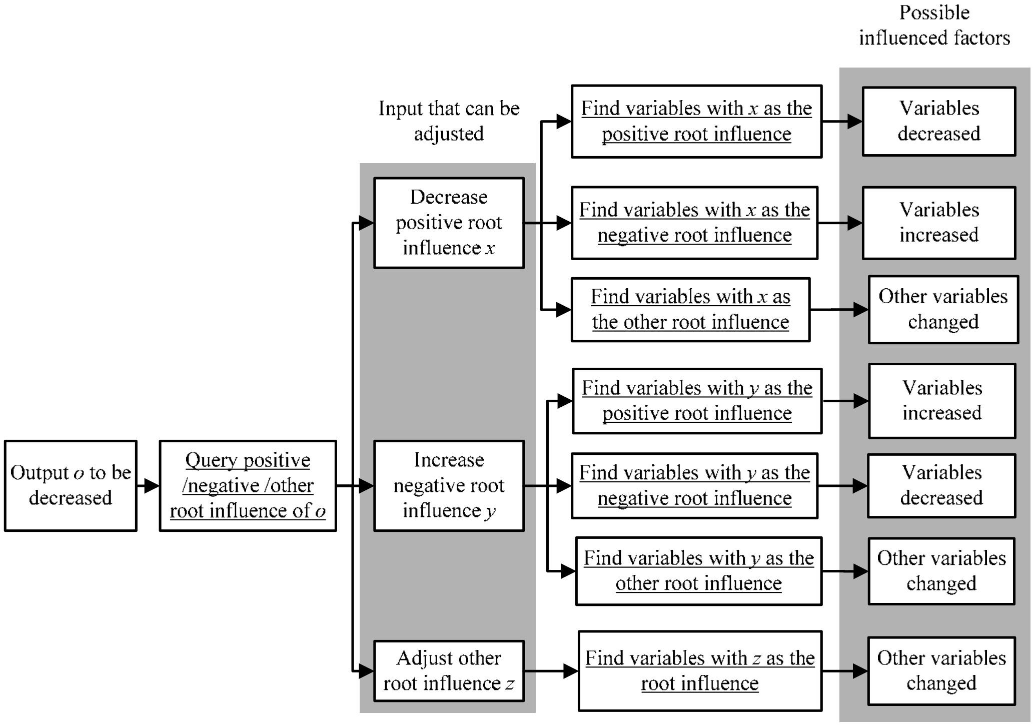


Figure 4-14: Ontology-based query and reason process for output decrease

The ontology-based data integration, MADMT and the ontology-based web system developed in Sections 4.3-4.5 will be illustrated in Section 5.2.4.

4.6 Ontology-Based Root Cause Analysis for Error Control in the Research

The ontologies and ontology-based methods in Sections 4.1-4.5 need to be refined continuously and the root cause analysis for these parts of the work can be facilitated by using the ontology-based root cause map in this section. An error ontology is developed to represent the error taxonomy, possible management methods for errors and to facilitate the RCA. This ontology is called EO (Error Ontology) and is built by using the ontology editor, Protégé. By utilizing the advantage of ontology in concept and relation representation, categorization, data reuse and reasoning, this ontology is expected to help users master and eliminate errors taking place in ontology development and utilization.

The EO structure is shown in Figure 4-15, which follows the representation method in Section 3.2.1. In the error ontology, the classes under the root are Causal_Factors, Collection_methods, Error_instance, Root_Cause_Map, Solutions_of_errors, and Types_of_errors. Under Causal_Factors and Root_Cause_Map class, there are several subclasses. After the classes are categorized and the hierarchy of the EO is developed, the slots are identified and developed, as shown in Figure 4-16.

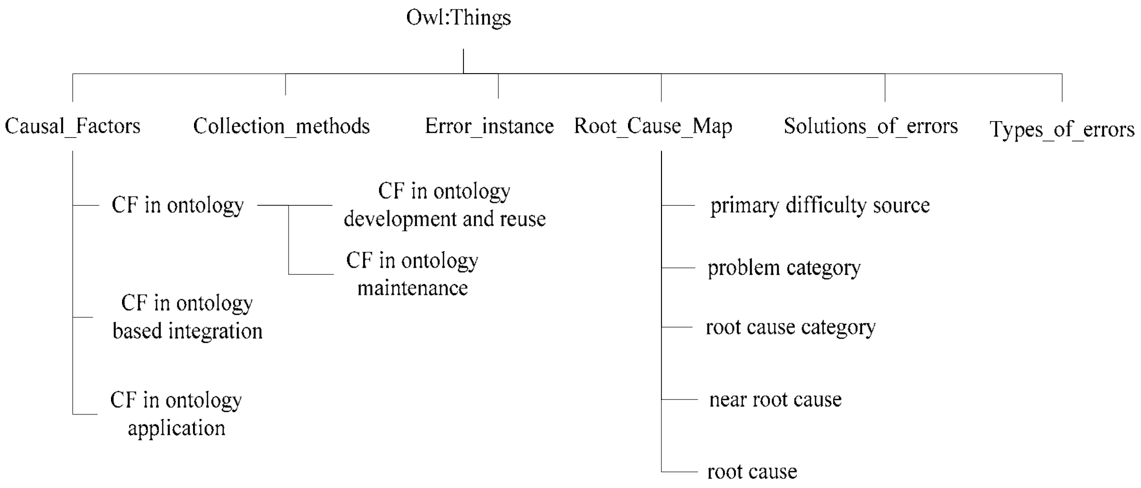


Figure 4-15: Hierarchy of the Error Ontology

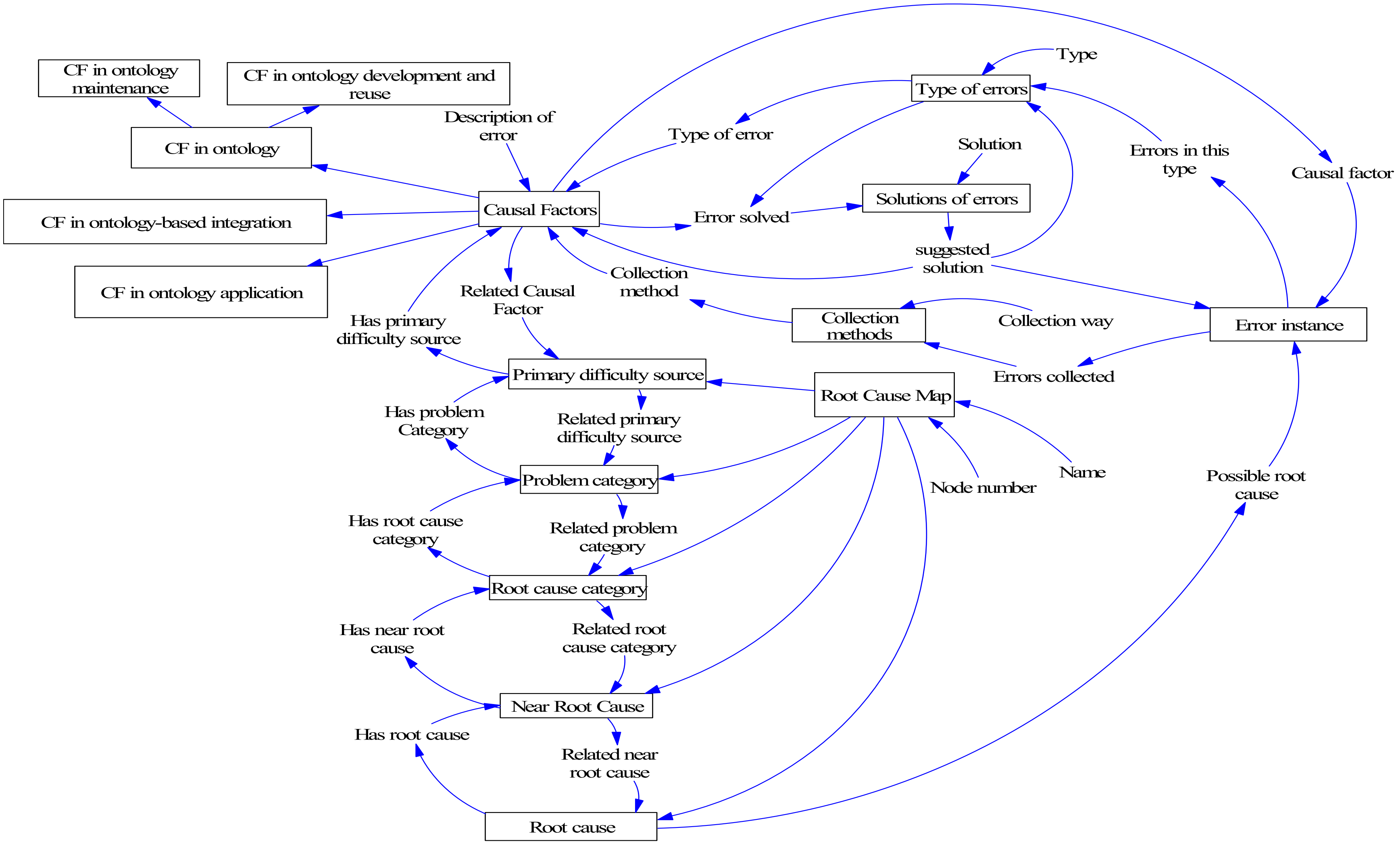


Figure 4-16: Classes and their slots in the EO

In Figure 4-16, classes are variables marked with boxes, and other variables are slots. The representation method in Section 3.2.2 is used. In addition, arrows between two boxes represent the relations of class and its subclasses.

The root cause map for RCA in ontology development and utilization is developed, as shown in Figure 4-17. The representation of primary difficulty source, problem category, root cause category, near root cause and root cause follows the rules in Figure 3-10.

After the classes and slots are developed, the nodes in Figure 4-17 are developed in the Error Ontology (EO) as instances to the corresponding subclasses of the class Root Cause Map. Errors and possible causal factors in each phase of ontology development and utilization in Section 3.5 are input as instances in the subclasses of Causal Factors with other related information (e.g. type of errors, solutions, collection methods and primary difficulty source).

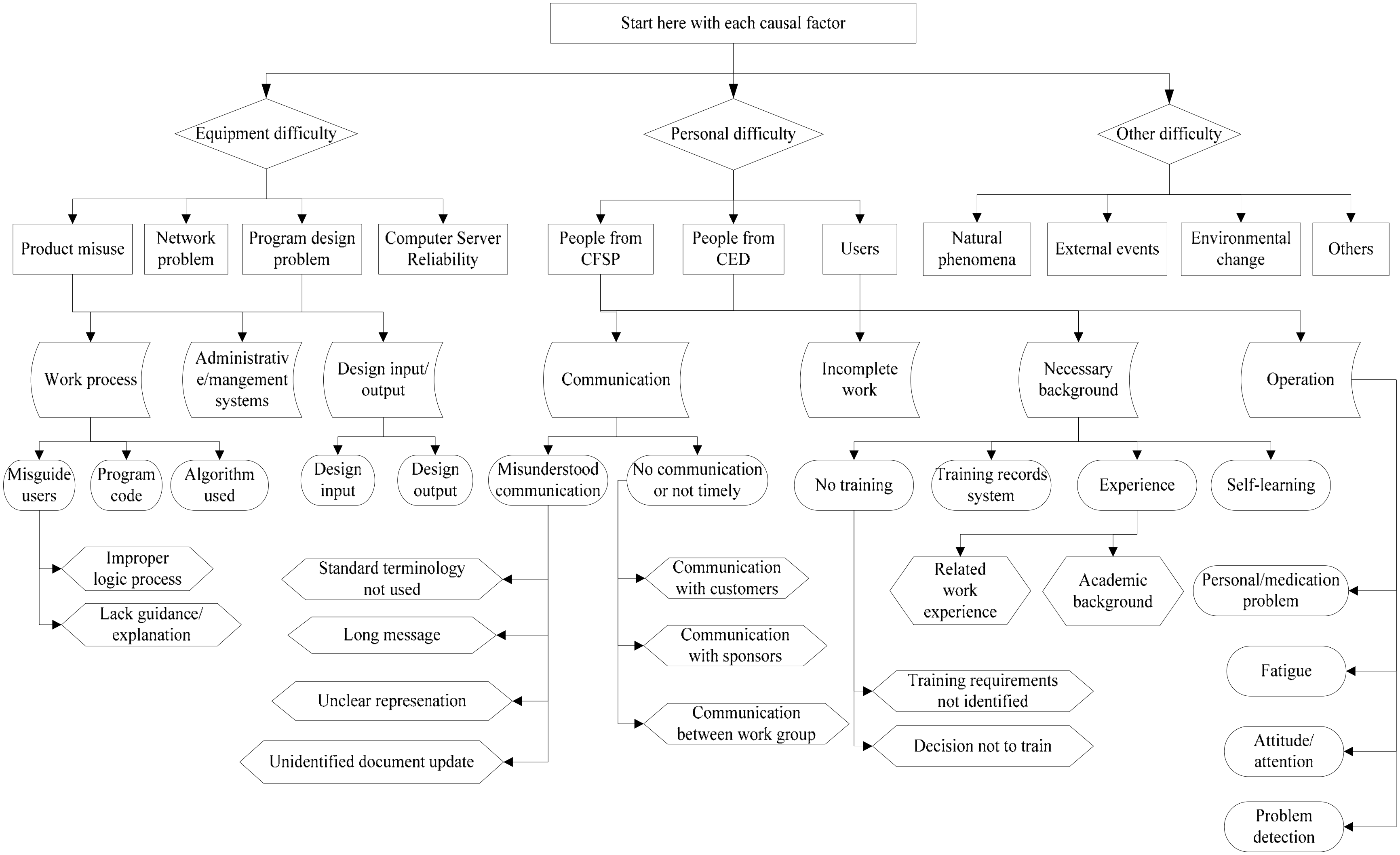


Figure 4-17: Root cause map for RCA in ontology development and utilization

After instances are input, two methods to find root cause and possible solutions can be implemented. The first method is utilizing the SWRL rules and rule engine to infer the possible root causes and solutions for a specific error. This process is based on not only the classes, properties, and instances in the error ontology described above, but also the SWRL rules which are listed below. They are built by using the SWRL Rules Tab in Protégé.

- **Find root cause for an error instance:**

Error_instance(?x) ∧ causal_factor(?x, ?y) ∧ Has_Primary_difficulty_source(?y, ?z) ∧ Has_Problem_Category(?z, ?a) ∧ Has_root_cause_category(?a, ?b) ∧ Has_near_root_cause(?b, ?c) ∧ Has_root_cause(?c, ?d) → possible_root_cause(?x, ?d)

The meaning of this rule is if an error A has a causal factor B, B has a primary difficulty source C, C has a problem category D, D has a root cause category E, E has a near root cause F, F has a root cause G, then G is a possible root cause for error A.

- **Find solutions for an causal factor:**

Causal_Factors(?x) ∧ type_of_error(?x, ?y) ∧ suggested_solution(?y, ?z) → suggested_solution(?x, ?z)

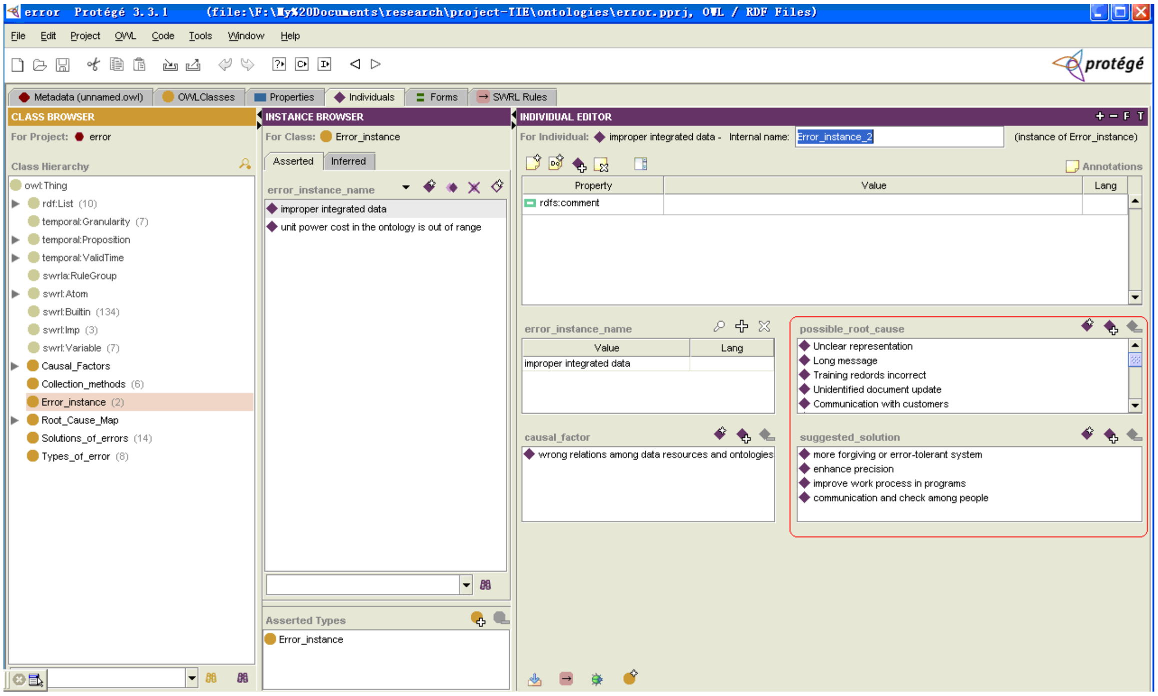
The meaning of this rule is that if the causal factors A belong to the type B, and B has a suggested solution C, then C is a suggested solution of A.

- **Find solutions for an error instance:**

Error_instance(?x) ∧ causal_factor(?x, ?y) ∧ suggested_solution(?y, ?z) → suggested_solution(?x, ?z)

The meaning of this rule is if an error A has a causal factor B, B has a suggested solution C, then the solution C is a suggested solution of the error A.

For example, the improper integrated data is retrieved, and the causal factor is wrong relations among data resources and ontologies. These two data are input into an individual of the error instance class in the error ontology. After running the Jess rule engine, the possible root causes and suggested solutions can be inferred and output in the corresponding two properties of the individual, as shown in Figure 4-18 marked with a box. In this way, more tips can be offered, which facilitates error elimination.



For Project: error

For Class: Error_instance

error_instance_name
- improper integrated data
- unit power cost in the ontology is out of range

Asserted Types: Error_instance

For Individual: improper integrated data - Internal name: Error_instance_2 (instance of Error_instance)

Property	Value	Lang
rdfs:comment		

error_instance_name

Value	Lang
improper integrated data	

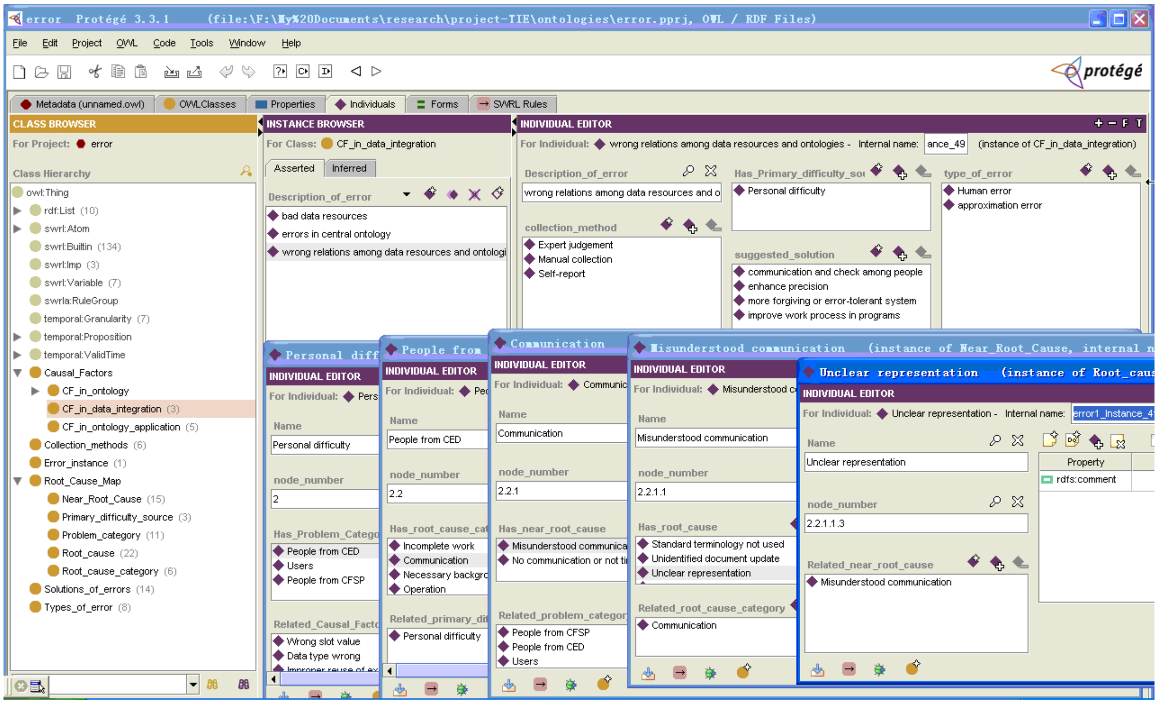
causal_factor
- wrong relations among data resources and ontologies

possible_root_cause
- Unclear representation
- Long message
- Training redords incorrect
- Unidentified document update
- Communication with customers

suggested_solution
- more forgiving or error-tolerant system
- enhance precision
- improve work process in programs
- communication and check among people

Figure 4-18: An example of the reasoning result in EO by SWRL and Jess rule engine

In the second method, by drilling down the path of RCA set in the EO, RCA of an error can be accomplished by going through the instances from causal factor to primary difficulty source, then to problem category, root cause category, near root cause and at last to root cause, as shown in Figure 4-19. For example, suppose a user working with the knowledge source finds the ontology-based integrated data is not proper and finds the causal factor is the improper relation between ontology and a data resource. After locating the related instance in CF_in_data_integration, the ontology can guide users to find the possible primary difficulty source. In this case, as shown in Figure 4-19, personal difficulty is selected, then similarly, the problem category is located at People from CED, root cause category is communication, near root cause is misunderstood communication and the root cause is unclear representation. Although improper relations can be found and corrected by related directly, the ontology supported root cause analysis can help find the real reason, and help the continuous improvement in the working mechanism of developing, reusing and application. Besides, possible solution methods as well as future collection method for this error is also suggested at the same time.



error Protégé 3.3.1 (file:\F:\My%20Documents\research\project-TIE\ontologies\error.pprj, OWL / RDF Files)

CLASS BROWSER
For Project: error
Class Hierarchy
- owl:Thing
  - rdf:List (10)
  - swrl:Atom
  - swrl:Builtin (134)
  - swrl:Imp (3)
  - swrl:Variable (7)
  - swrla:RuleGroup
  - temporal:Granularity (7)
  - temporal:Proposition
  - temporal:ValidTime
  - Causal_Factors
    - CF_in_ontology
    - CF_in_data_integration (3)
    - CF_in_ontology_application (5)
  - Collection_methods (6)
  - Error_instance (1)
  - Root_Cause_Map
    - Near_Root_Cause (15)
    - Primary_difficulty_source (3)
    - Problem_category (11)
    - Root_cause (22)
    - Root_cause_category (6)
  - Solutions_of_errors (14)
  - Types_of_error (8)

INSTANCE BROWSER
For Class: CF_in_data_integration
Description_of_error
- bad data resources
- errors in central ontology
- wrong relations among data resources and ontologi

INDIVIDUAL EDITOR
For Individual: wrong relations among data resources and ontologies - Internal name: ance_49 (instance of CF_in_data_integration)
Description_of_error: wrong relations among data resources and o
Has_Primary_difficulty_sou: Personal difficulty
type_of_error: Human error; approximation error
collection_method: Expert judgement; Manual collection; Self-report
suggested_solution: communication and check among people; enhance precision; more forgiving or error-tolerant system; improve work process in programs

Personal diff
INDIVIDUAL EDITOR
For Individual: Pers
Name: Personal difficulty
node_number: 2
Has_Problem_Catego: People from CED; Users; People from CFSP
Related_Causal_Facto: Wrong slot value; Data type wrong; Improper reuse of ex

People from
INDIVIDUAL EDITOR
For Individual: Peo
Name: People from CED
node_number: 2.2
Has_root_cause_cat: Incomplete work; Communication; Necessary backgro; Operation
Related_primary_dif: Personal difficulty

Communication
INDIVIDUAL EDITOR
For Individual: Communic
Name: Communication
node_number: 2.2.1
Has_near_root_cause: Misunderstood communica; No communication or not ti
Related_problem_categor: People from CFSP; People from CED; Users

Misunderstood communication (instance of Near_Root_Cause, internal n
INDIVIDUAL EDITOR
For Individual: Misunderstood c
Name: Misunderstood communication
node_number: 2.2.1.1
Has_root_cause: Standard terminology not used; Unidentified document update; Unclear representation
Related_root_cause_category: Communication

Unclear representation (instance of Root_caus
INDIVIDUAL EDITOR
For Individual: Unclear representation - Internal name: error1_Instance_4
Name: Unclear representation
node_number: 2.2.1.1.3
Related_near_root_cause: Misunderstood communication
Property: rdfs:comment

Figure 4-19: Screenshot of EO in Protégé to find the root cause

Chapter 5: Ontology-Based Welding Design Examples

This chapter provides examples from industry and the CFSP center that will help to validate and test the approach and methods, including cost analysis of MIG welding (Section 5.2.1), filler material and machine selection using SWRL rules in MIG welding design (Section 5.2.2), pin tool selection using SWRL rules in FSW design (Section 5.2.3) and ontology-based integration and decision support platform for DFM (Section 5.2.4).

5.1 Background of Case Studies

FSW (Friction Stir Welding) technologies hold high promise to significantly reduce production cost and minimize environmental emissions over a wide segment of the industry. To ensure successful implementation of FSW and development of new technologies based on the friction stir concept, the NSF Center for Friction Stir Processing (CFSP) industrial sponsors have requested that e-Design methods be included in the project to assure availability of the latest design approaches to enhance diffusion of friction stir based technologies. The research in the dissertation is funded by an NSF grant, “TIE Research Program on e-Design for Friction Stir Welding and Processing (Terpenny et al., 2006).” This project is a collaborative project between two NSF centers, the Center for e-Design and the Center for Friction Stir Processing. The work of this dissertation contributes to the success of the NSF TIE project. It also provides a more comprehensive approach and methods to support knowledge capture and reuse as well as decision making in the early stages of engineering design.

The intellectual merit of the TIE project is on the development of e-Design tools for concurrent design and manufacturing in the area of FSP to rapidly identify FSP technology insertion opportunities and enhance its implementation in a wide variety of applications including aerospace, aeronautic, and marine applications. The e-Design tools aim at enhancing the Design for Manufacturing (DFM) knowledge management process and facilitate decision making in the area of FSP.

More background information as well as basic design constraints and mathematical models (formulas) used in these welding methods, which are built in the DFM ontology and DFM e-Design ontology, are shown in detail in Appendices B-E. In the next section, four case studies are provided based on this project.

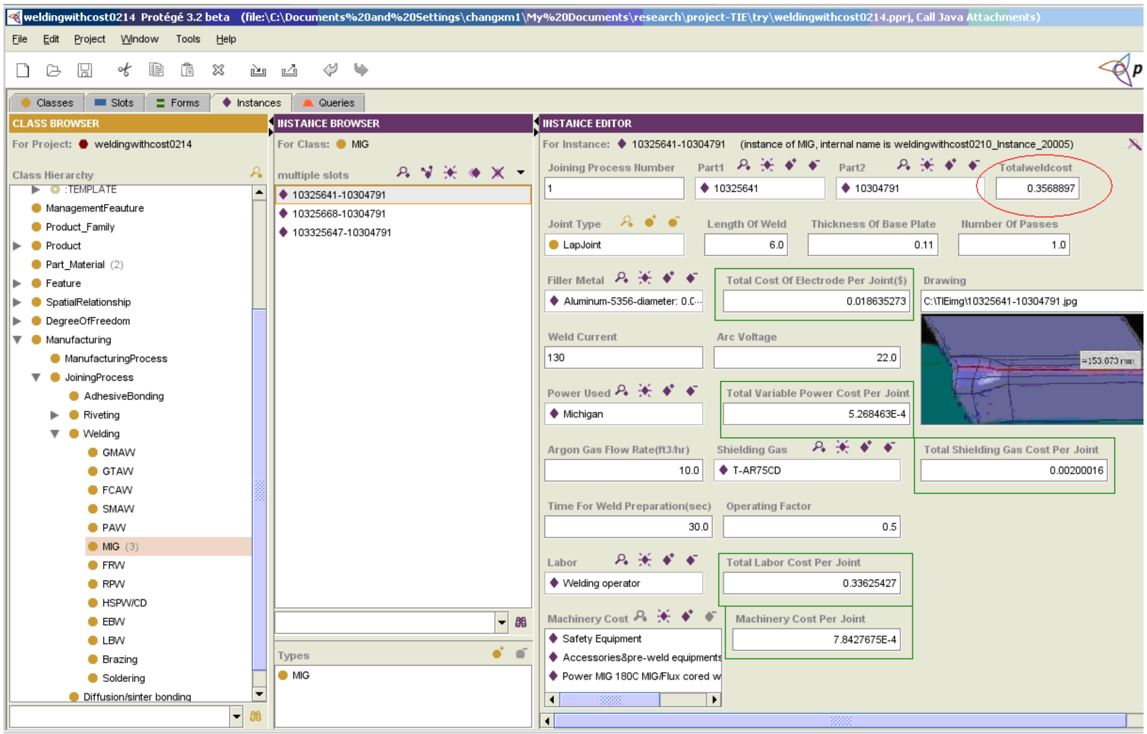
5.2 Case Studies

In this section, four cases are discussed to illustrate ontology-based knowledge management in product design. The first case uses Java Function Calls and realizes the MIG and FSW cost models in DFM ontology; the second and third cases use SWRL rules and an associated rule engine to help consider technical requirements as well as machine availability in MIG and FSW design; and the fourth case illustrates the B/S mode decision support platform based on the ontologies to integrate heterogeneous data resources and support FSW design.

5.2.1 Case Study 1: Cost Analysis of MIG Welding Process

The first case mainly focuses on the decision making based on cost criteria and technical requirements in DFM. For example, the thickness of welded components is a critical criteria to select filler material, as shown in Table D-1 in Appendix D. If the metal thickness is 3/32", for example, the diameter of possible filler material can be 0.03", 0.035" or 3/64". From Table D-2, it can be found that the bigger the diameter, the higher the current required. And Table D-3 shows that the higher the current, the more expensive the welding machine. With the increase of current, the voltage of welding machine also increases, which causes an increase in power costs. So in order to reduce the machinery cost and power cost per joint, smaller filler material diameter is necessary. But as shown in Table D-4, under the same material, smaller filler diameter means higher filler material cost. In order to get the best total weld cost, trade-off between machinery cost, power cost and filler material (electrode) cost is required. Since there are complex technical requirements and economical criteria which need to be considered during parameter selection, the DFM ontology with Java Function Calls programs can be used to represent the relations and help the trade-off.

Mathematical relations built in the Java Function Calls programs are based on the cost models developed by Kumar and Allada (2007). The Java classes have been linked with slots in the DFM ontology, as described in Section 4.2.1.2. After the basic information is input, the advanced information will be calculated automatically according to the Java Function Calls programs, and output directly in the instance of the DFM ontology. For example, after Part 1 and Part 2 are input, thickness of base plate can be obtained, which is the lower thickness of the two parts. The five cost components (marked with rectangular box in Figure 5-1) can also be obtained after input of the related information. With the accomplishment of the cost component calculation in Protégé, the totalweldcost (marked by circle in Figure 5-1) can be calculated.



For Project: weldingwithcost0214

For Class: MIG

10325641-10304791
10325668-10304791
103325647-10304791

For Instance: 10325641-10304791 (instance of MIG, internal name is weldingwithcost0210_Instance_20005)

Joining Process Number: 1
Part1: 10325641
Part2: 10304791
Totalweldcost: 0.3568897
Joint Type: LapJoint
Length Of Weld: 6.0
Thickness Of Base Plate: 0.11
Number Of Passes: 1.0
Filler Metal: Aluminum-5356-diameter: 0.0...
Total Cost Of Electrode Per Joint($): 0.018635273
Drawing: C:\TIEimg\10325641-10304791.jpg
Weld Current: 130
Arc Voltage: 22.0
Power Used: Michigan
Total Variable Power Cost Per Joint: 5.268463E-4
Argon Gas Flow Rate(ft3/hr): 10.0
Shielding Gas: T-AR75CD
Total Shielding Gas Cost Per Joint: 0.00200016
Time For Weld Preparation(sec): 30.0
Operating Factor: 0.5
Labor: Welding operator
Total Labor Cost Per Joint: 0.33625427
Machinery Cost: Safety Equipment; Accessories&pre-weld equipments; Power MIG 180C MIG/Flux cored w
Machinery Cost Per Joint: 7.8427675E-4

Figure 5-1: Design of a new welding alternative using Java Function Calls programs

Designers can consider different cost items and change the basic parameters, such as the filler material used, length of weld, etc., and the resulting change will be automatically displayed. This provides designers an easy and intuitive way to compute the cost of different alternatives, to see the change in cost when different parameters are varied, to get the trade-off between several item costs and to reduce the total weld cost with the consideration of technical requirements. Moreover, through this method, calculation of costs for different alternatives can be largely reduced, and the consistency and accuracy of the data can be enhanced through the use of the DFM ontology.

For example, there are two design alternatives, as shown in Table 5-1. The thickness of the base plate is 0.118", the length of weld is 3", the type of weld is lap joint. The life of the machine is 5 years, and the operation is in Michigan, where the unit power cost is $0.07. In this example, after Java Function Calls programs calculate the result, it is found that the electrode cost is much higher than the machinery and power cost, so the design alternative 2 that uses Aluminum 5356-0.035" as filler material is better.

Table 5-1: Comparison of two design alternatives in MIG welding design

Slot	**Design Alternative 1**	**Design Alternative 2**
Filler diameter	Aluminum 5356-0.03"	Aluminum 5356-0.035"
Allowed current	60-100	100-160
Machine used	Power MIG 140T MIG/Flux-Cored Wire Feeder Welder	Power MIG 180T MIG/Flux-Cored Wire Feeder Welder
Power voltage	19.5	20
Electrode cost	$ 0.011665768	0.010722213
Machinery cost	$0.00002	$0.00002
Power cost	$1.4369356E-4	$2.394756E-4

The design instances are saved in the DFM ontology, and can be used later in the ontology-based web system or directly used in Protégé.

5.2.2 Case Study 2: Filler Material and Machine Selection in MIG design Using SWRL

In this example, the DFM ontology helps the selection of the filler material and welding machine for a MIG welding based joining operation. Given the inputs, parts (Part A and Part B) and their thicknesses (Part A is 0.08" thick, and Part B is 0.09" thick), the DFM ontology can suggest the appropriate filler materials and the welding machine with the help of SWRL rules and the Jess rule engine. The classes and their slots in the DFM ontology, which are used in this case, are shown in Figure 5-2. The representation method of classes and their slots follows the representation method in Section 3.2.2.

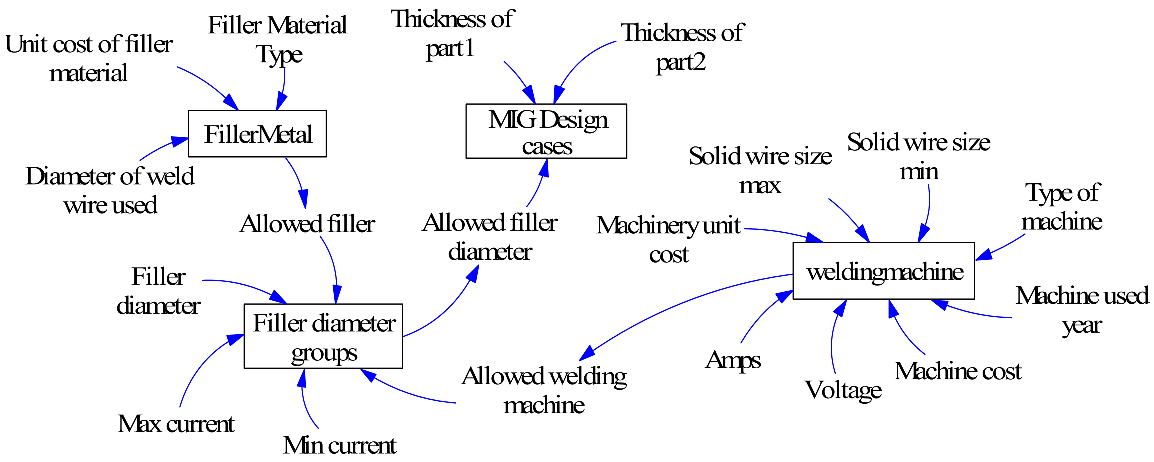


Figure 5-2: Classes and slots used in case study 2

The restrictions among these slots are captured in Figure 5-3, which follows the representation method in Section 3.2.3.1.

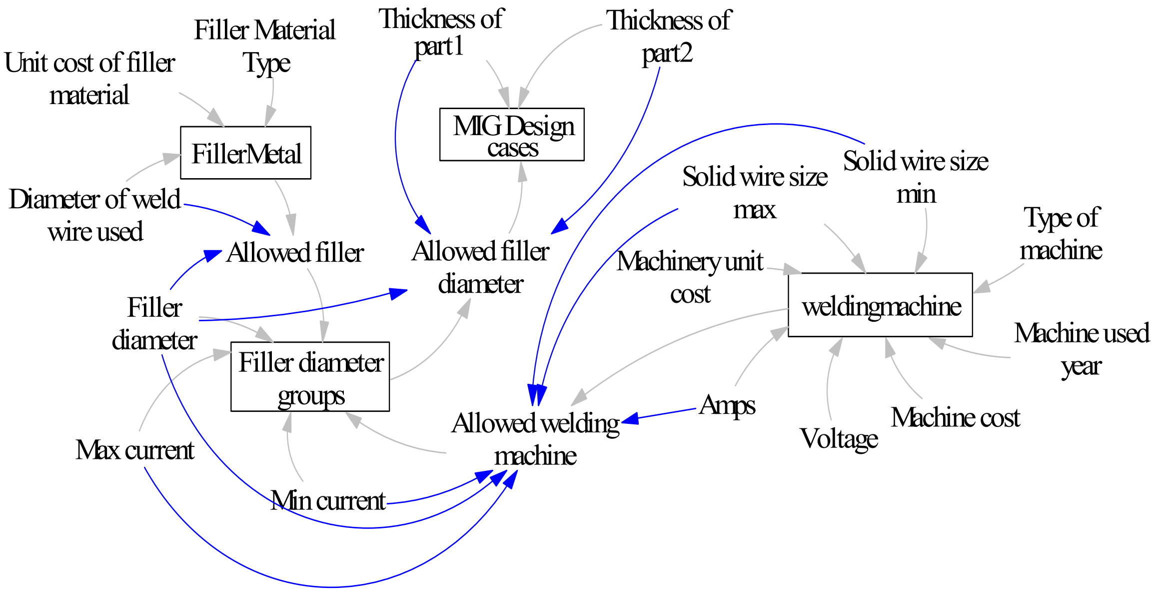


Figure 5-3: Relations among slots used in case study 2

To use the DFM ontology in this way, suitable SWRL rules need to be developed. For example, members in one filler diameter group have the same diameter. Rule 5-1 puts all filler metals that have the same diameter into one group.

Rule 5-1:

FillerMetal(?x) ⋀ Diameter_of_weld_wire_used(?x, ?y) ⋀ Filler_diameter_groups(?b) ⋀ Filler_diameter(?b, ?a) ⋀ swrlb:equal(?y, ?a) → Allowed_filler(?b, ?x)

Relations between metal thickness and filler or wire diameter are shown in Table D-1. Since the thicknesses of two parts in this example are in the interval of 0.0625" to 0.125", the filler material diameter can be between 0.03" and 0.046875". The following SWRL rule (Rule 5-2) supports this rule.

Rule 5-2:

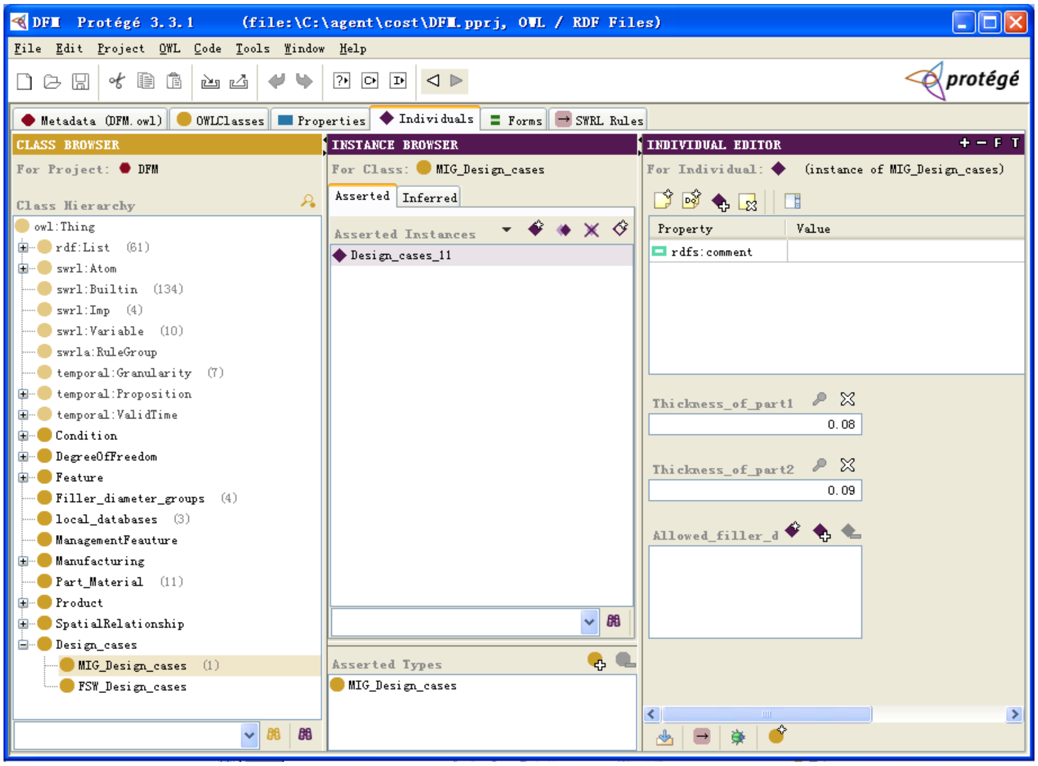
MIG_Design_cases(?x) ⋀ Thickness_of_part1(?x, ?y) ⋀ Thickness_of_part2(?x, ?z) ⋀ swrlb:greaterThan(?y, 0.0625) ⋀ swrlb:greaterThan(?z, 0.0625) ⋀ swrlb:lessThanOrEqual(?y, 0.125) ⋀ swrlb:lessThanOrEqual(?z, 0.125) ⋀ Filler_diameter_groups(?a) ⋀ Filler_diameter(?a, ?b) ⋀ swrlb:lessThanOrEqual(?b, 0.046875) ⋀ swrlb:greaterThanOrEqual(?b, 0.03) → Allowed_filler_diameter(?x, ?a)

The allowed current for different diameter of filler material is shown in Table D-2. The allowed current is recorded in the slot Max_current and Min_current in the class filler_diameter_groups. According to the selected diameter and current allowed, the DFM ontology will provide a list of machines that can be used for this purpose. Table D-3 shows some MIG welding machines with their cost and capacities. If the current of the machine is allowed for the filler diameter group, and the solid wire size range of the machine also meets the group, the welding machine is allowed to be used in this group. The SWRL rule (Rule 5-3) decides the feasible machines that meet the requirements of filler diameter and current range.

Rule 5-3:

Filler_diameter_groups(?x) ∧ Filler_diameter(?x, ?y) ∧ Max_current(?x, ?z) ∧ Min_current(?x, ?a) ∧ weldingmachine(?b) ∧ Solid_wire_size_min(?b, ?c) ∧ Solid_wire_size_max(?b, ?d) ∧ swrlb:lessThanOrEqual(?y, ?d) ∧ swrlb:greaterThanOrEqual(?y, ?c) ∧ Amps(?b, ?e) ∧ swrlb:lessThanOrEqual(?e, ?z) ∧ swrlb:greaterThanOrEqual(?e, ?a) → Allowed_welding_machine(?x, ?b)

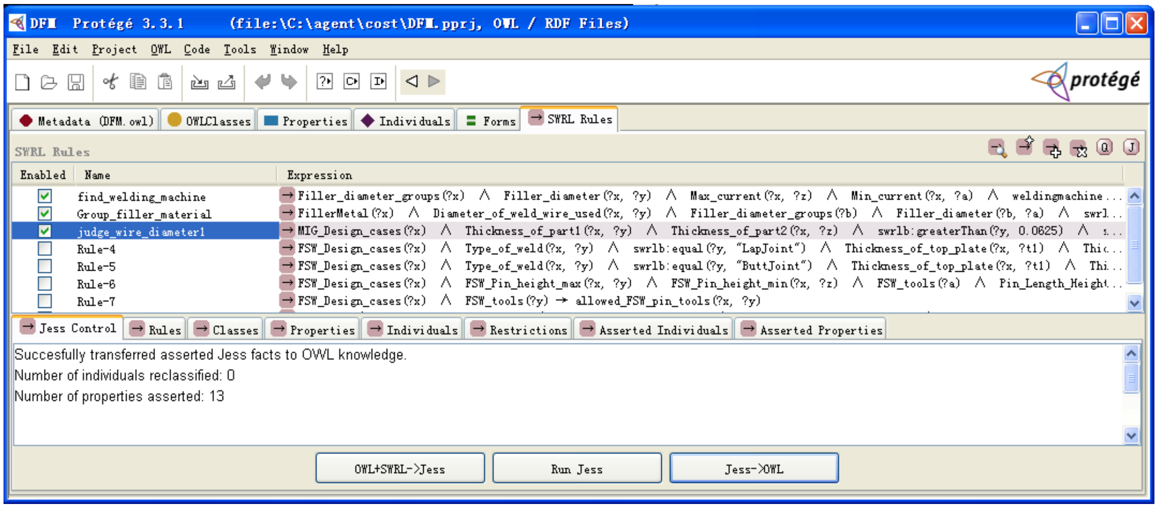
In the DFM ontology, first, the thicknesses of the two parts are input into an instance of the class MIG_Design_cases in Protégé, the ontology editor, as shown in Figure 5-4. Through the SWRLJessTab in Protégé, the new knowledge is inferred by using three SWRL rules, as shown in Figure 5-5. The allowed filler material inferred is shown in Figure 5-6, marked with a circle.



Thickness_of_part1: 0.08

Thickness_of_part2: 0.09

Figure 5-4: A new instance in the MIG_Design_cases class of the DFM ontology

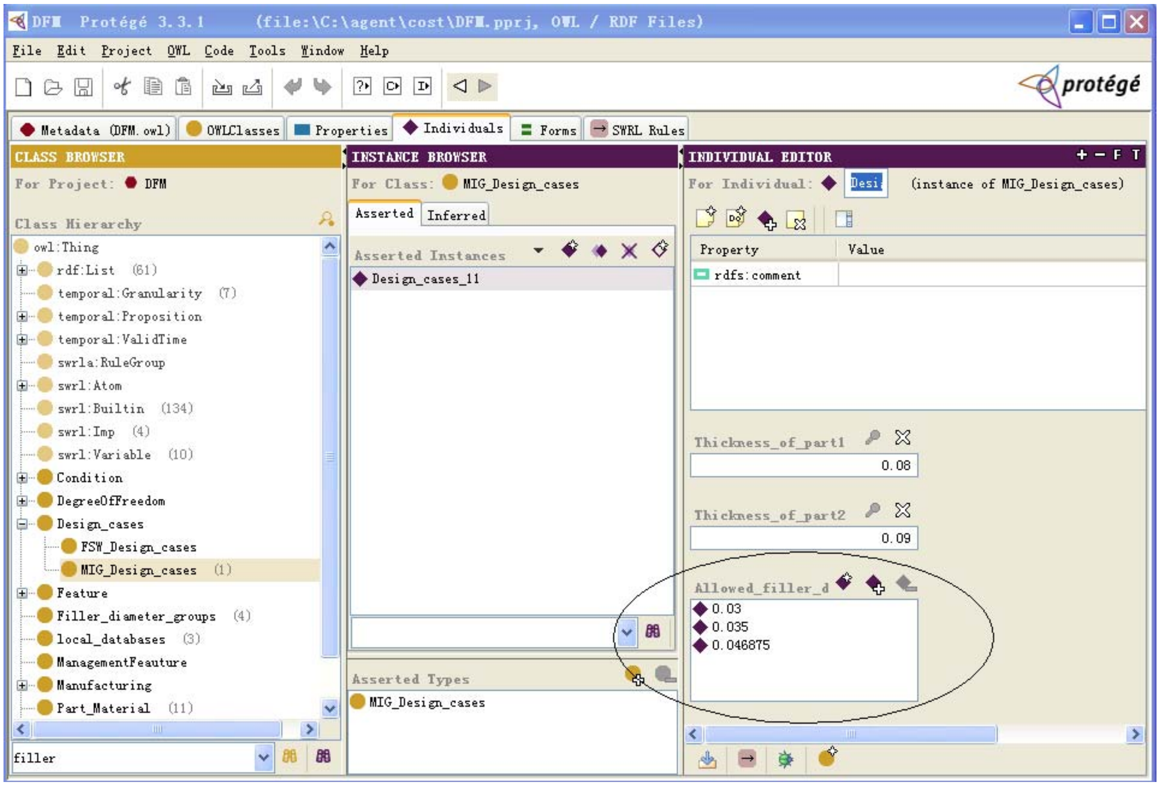


SWRL Rules

Enabled	Name	Expression
☑	find_welding_machine	→ Filler_diameter_groups(?x) ∧ Filler_diameter(?x, ?y) ∧ Max_current(?x, ?z) ∧ Min_current(?x, ?a) ∧ weldingmachine...
☑	Group_filler_material	→ FillerMetal(?x) ∧ Diameter_of_weld_wire_used(?x, ?y) ∧ Filler_diameter_groups(?b) ∧ Filler_diameter(?b, ?a) ∧ swrl...
☑	judge_wire_diameter1	→ MIG_Design_cases(?x) ∧ Thickness_of_part1(?x, ?y) ∧ Thickness_of_part2(?x, ?z) ∧ swrlb:greaterThan(?y, 0.0625) ∧ s...
☐	Rule-4	→ FSW_Design_cases(?x) ∧ Type_of_weld(?x, ?y) ∧ swrlb:equal(?y, "LapJoint") ∧ Thickness_of_top_plate(?x, ?t1) ∧ Thic...
☐	Rule-5	→ FSW_Design_cases(?x) ∧ Type_of_weld(?x, ?y) ∧ swrlb:equal(?y, "ButtJoint") ∧ Thickness_of_top_plate(?x, ?t1) ∧ Thi...
☐	Rule-6	→ FSW_Design_cases(?x) ∧ FSW_Pin_height_max(?x, ?y) ∧ FSW_Pin_height_min(?x, ?z) ∧ FSW_tools(?a) ∧ Pin_Length_Height...
☐	Rule-7	→ FSW_Design_cases(?x) ∧ FSW_tools(?y) → allowed_FSW_pin_tools(?x, ?y)

Succesfully transferred asserted Jess facts to OWL knowledge.
Number of individuals reclassified: 0
Number of properties asserted: 13

Figure 5-5: SWRL rules and Jess rule engine used to infer new knowledge



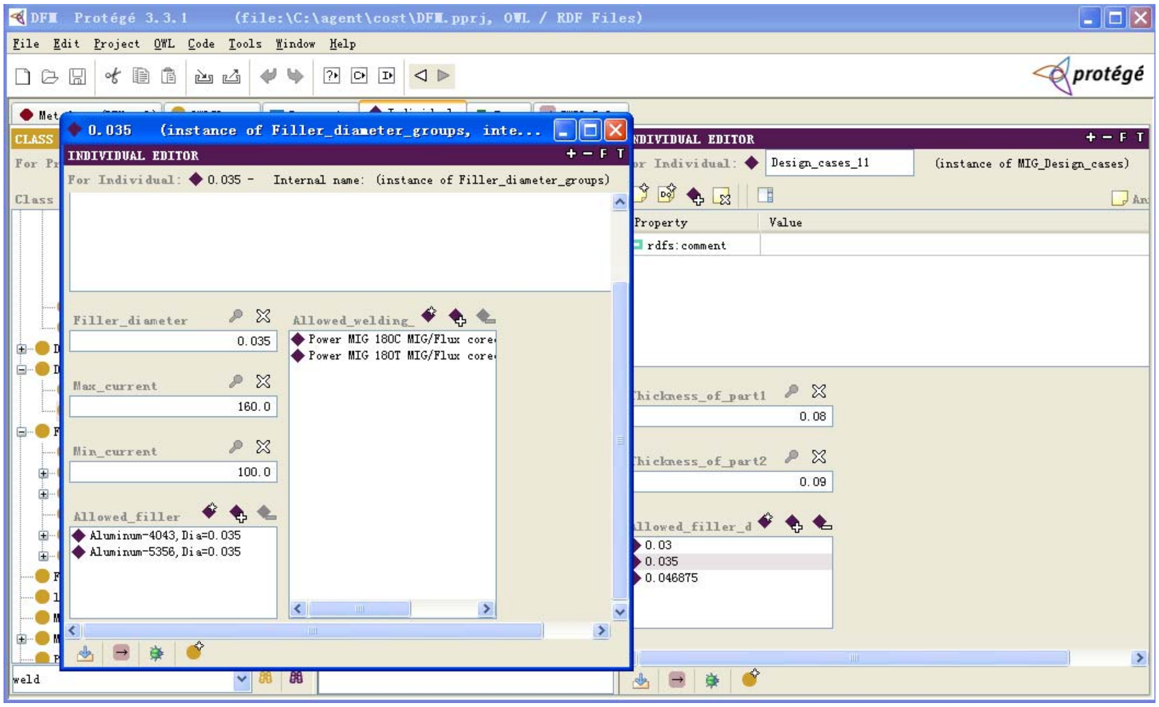
Thickness_of_part1: 0.08

Thickness_of_part2: 0.09

Allowed_filler_d:
- 0.03
- 0.035
- 0.046875

Figure 5-6: Inferred result from SWRL and Jess rule engine for material selection

For instance, if the company does not have the 0.03" filler material, in order to reduce the cost, 0.035" filler material is temporarily chosen, since it is cheaper than the 3/64" one, which is also allowed. After choosing the filler diameter of 0.035", the allowed fillers and machines that are inferred from SWRL rules can be found, as shown in Figure 5-7. The data in the slots “Allowed_filler” and “Allowed welding machine” shown in Figure 5-7 are also inferred from the SWRL rules. The machine that has the cheapest price and is available for the company at that time will be chosen. In this case, K2472-1 (Power MIG 180T MIG/Flux-Cored Wire Feeder Welder) is chosen. Often machines in the same group having the lowest price and meeting all the technical requirements are fully occupied and hence are not available for immediate use. In this case, the second cheapest available machine can be selected.



For Individual: 0.035 - Internal name: (instance of Filler_diameter_groups)

Filler_diameter: 0.035

Max_current: 160.0

Min_current: 100.0

Allowed_filler:
- Aluminum-4043, Dia=0.035
- Aluminum-5356, Dia=0.035

Allowed_welding_:
- Power MIG 180C MIG/Flux core
- Power MIG 180T MIG/Flux core

For Individual: Design_cases_11 (instance of MIG_Design_cases)

Property	Value
rdfs:comment	

Thickness_of_part1: 0.08

Thickness_of_part2: 0.09

Allowed_filler_d:
- 0.03
- 0.035
- 0.046875

Figure 5-7: Allowed fillers and machines inferred in a design case

5.2.3 Case Study 3: Pin Tool Selection in FSW Design Using SWRL

In this case, SWRL rules are used in the DFM ontology to find the allowed pin tool for FSW. Table C-4 in Appendix C describes the criteria for FSW tool selection. Lap joint and Butt joint are available in FSW welding, and the available tools are shown in Table C-5.

In the DFM ontology, the FSW_Design_cases class and the related SWRL rules will facilitate the selection of the tool with the integrated Jess rule engine in Protégé. The related properties in FSW_Design_cases class are shown in Table 5-2.

Table 5-2: Properties in the DFM ontology for the FSW tool selection

Properties	**Range**
Thickness_of_top_plate	float
Thickness_of_bottom_plate	float
Type_of_weld	string
FSW_Pin_height_min	float
FSW_Pin_height_max	float
allowed_FSW_pin_tools	Instance of FSW_tools
Suggested_FSW_pin_tools	Instance of FSW_tools

In the selection of tools, the suggested pin height will first be calculated according to the relations in Table C-4. Two SWRL rules, Rules 5-4 and 5-5, are in charge of this step as shown below. Rule 5-4 represents the requirements in lap joint, while Rule 5-5 represents the requirements in butt joint. According to the requirements in CFSP, the thicknesses of welded components are represented in inch and the pin height is represented in mm, so unit transformation is also complemented in these two rules.

Rule 5-4:

FSW_Design_cases(?x) ∧ Type_of_weld(?x, ?y) ∧ swrlb:equal(?y, "LapJoint") ∧ Thickness_of_top_plate(?x, ?t1) ∧ Thickness_of_bottom_plate(?x, ?t2) ∧ swrlb:multiply(?a, 0.91, ?t1) ∧ swrlb:multiply(?b, 0.185, ?t2) ∧ swrlb:multiply(?c, 0.455, ?t2) ∧ swrlb:add(?min1, ?a, ?b) ∧ swrlb:add(?max1, ?a, ?c) ∧ swrlb:multiply(?min, 25.4, ?min1) ∧ swrlb:multiply(?max, 25.4, ?max1) → FSW_Pin_height_min(?x, ?min) ∧ FSW_Pin_height_max(?x, ?max)

Rule 5-5:

FSW_Design_cases(?x) ∧ Type_of_weld(?x, ?y) ∧ swrlb:equal(?y, "ButtJoint") ∧ Thickness_of_top_plate(?x, ?t1) ∧ Thickness_of_bottom_plate(?x, ?t2) ∧ swrlb:multiply(?a, 0.85, ?t2) ∧ swrlb:multiply(?b, 0.9, ?t2) ∧ swrlb:multiply(?min, ?a, 25.4) ∧ swrlb:multiply(?max, ?b, 25.4) →

FSW_Pin_height_min(?x, ?min) ⋀ *FSW_Pin_height_max(?x, ?max)*

All the available tools that have the pin height in the calculated range will be suggested. Rule 5-6 can be used to find the suggested tools.

Rule 5-6:

FSW_Design_cases(?x) ⋀ *FSW_Pin_height_max(?x, ?y)* ⋀
FSW_Pin_height_min(?x, ?z) ⋀ *FSW_tools(?a)* ⋀
Pin_Length_Height_mm_(?a, ?b) ⋀ *swrlb:lessThanOrEqual(?b, ?y)* ⋀
swrlb:greaterThanOrEqual(?b, ?z) → Suggested_FSW_pin_tools(?x, ?a)

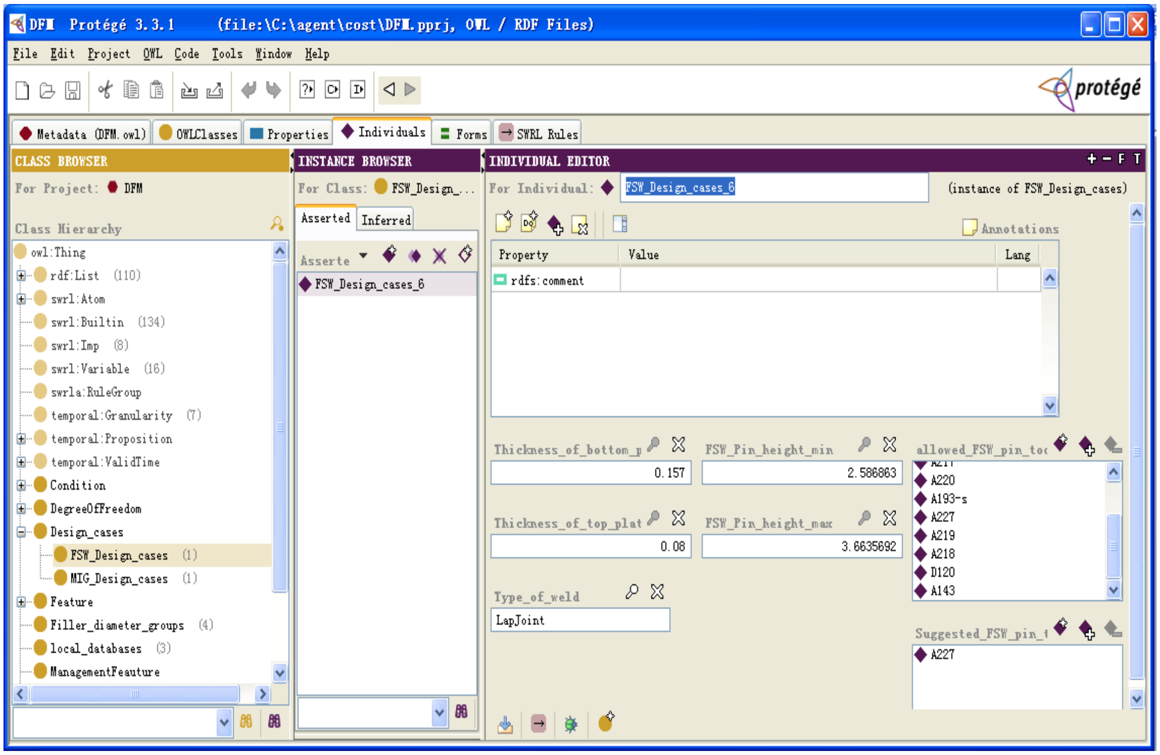
If none of the tool has the pin diameter in the suggested range, the allowed tools will be considered. In this selection, all the tools in Table C-5 are allowed, and the one with the pin height closest to the range will be chosen in common. Rule 5-7 is used to show the allowed tools in the instance.

Rule 5-7:

FSW_Design_cases(?x) ⋀ *FSW_tools(?y) → allowed_FSW_pin_tools(?x, ?y)*

Examples:

In the first case, type of weld is Lap Joint, thickness of the two components are 0.08" (top) and 0.157" (bottom). The suggested pin height is between 2.587 and 3.664mm. Among the available tools, the pin height of A227 is in this range, and can be the suggested pin tool, as shown in Figure 5-8. In the second case, type of weld is Lap Joint, thickness of the two components are 0.118" (top) and 0.157" (bottom). The suggested pin height is between 3.465 and 4.541mm. No pin height of the available tools is in this range with all being less than 3.465mm, so the tool with the biggest pin height is chosen from the allowed_FSW_pin_tools, as shown in Figure 5-9.



For Individual: FSW_Design_cases_6 (instance of FSW_Design_cases)

Thickness_of_bottom_p: 0.157

FSW_Pin_height_min: 2.586863

allowed_FSW_pin_too: A217, A220, A193-s, A227, A219, A218, D120, A143

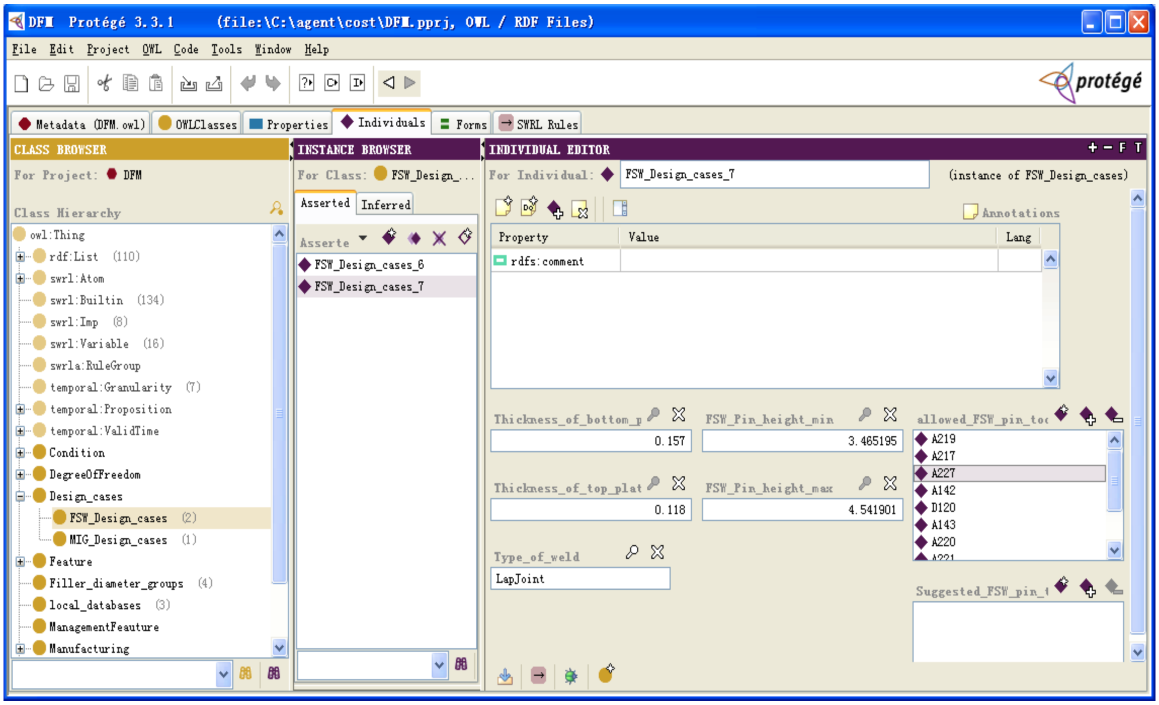
Thickness_of_top_plat: 0.08

FSW_Pin_height_max: 3.6635692

Type_of_weld: LapJoint

Suggested_FSW_pin_t: A227

Figure 5-8: Using the DFM ontology in Protégé to select suggested pin tool



For Individual: FSW_Design_cases_7 (instance of FSW_Design_cases)

Thickness_of_bottom_p: 0.157

FSW_Pin_height_min: 3.465195

allowed_FSW_pin_too: A219, A217, A227, A142, D120, A143, A220, A221

Thickness_of_top_plat: 0.118

FSW_Pin_height_max: 4.541901

Type_of_weld: LapJoint

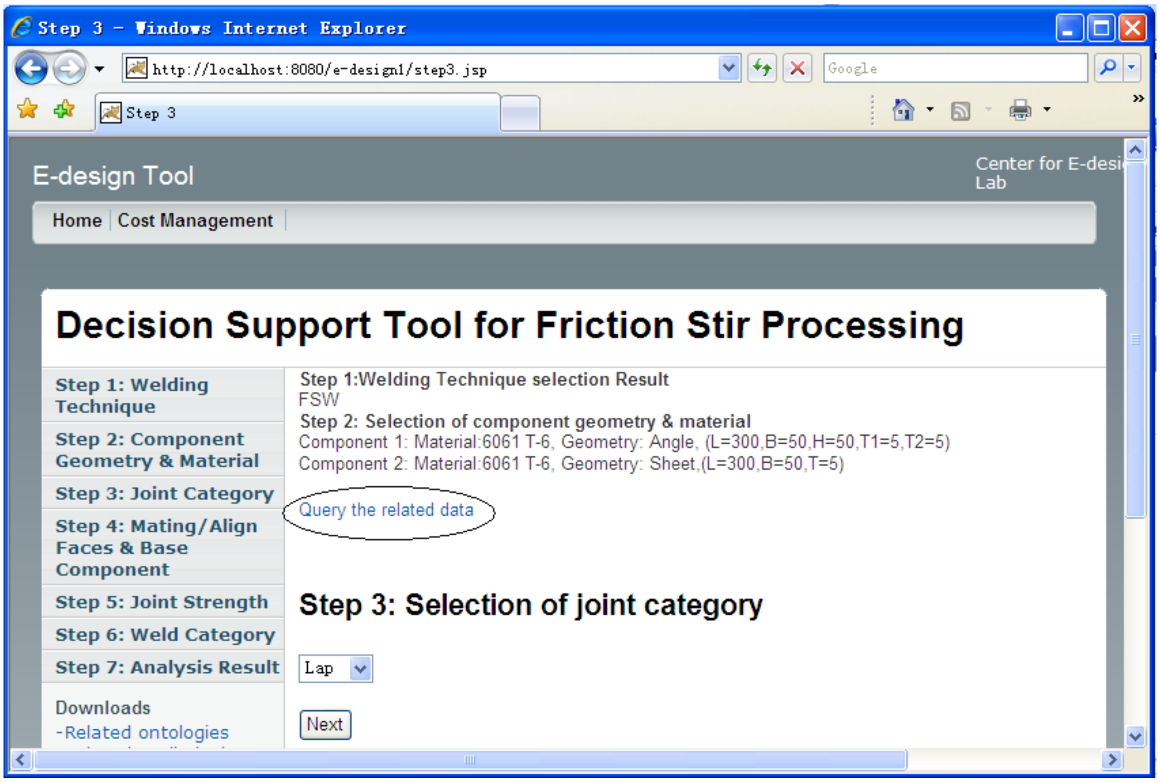
Suggested_FSW_pin_t:

Figure 5-9: Using the DFM ontology in Protégé to select the allowed pin tool

5.2.4 Case Study 4: Ontology-Based Data Integration and Decision Support Platform

This case illustrates the use of ontology-based data integration and the decision support platform that are built using the Apache Tomcat platform. The decision support platform supports the design for Friction Stir Welding (FSW), and has six steps of inputs in the design process. The sequence of these steps are selection of (1) welding technique, (2) component geometry and material, (3) joint category, (4) mating/align faces and the base component, (5) joint strength, and (6) weld category, as described in Section 4.5.

In the input steps, the platform provides the integrated data, which meet the specification of previous inputs, from heterogeneous data resources. For example, in Step 3, designers can get the historical data related to the previous inputs, which are FSW as welding technique and 6061 T-6 as the component material. The platform provides the link to the data integration page, which is circled in Figure 5-10.



Decision Support Tool for Friction Stir Processing

Step 1:Welding Technique selection Result
FSW

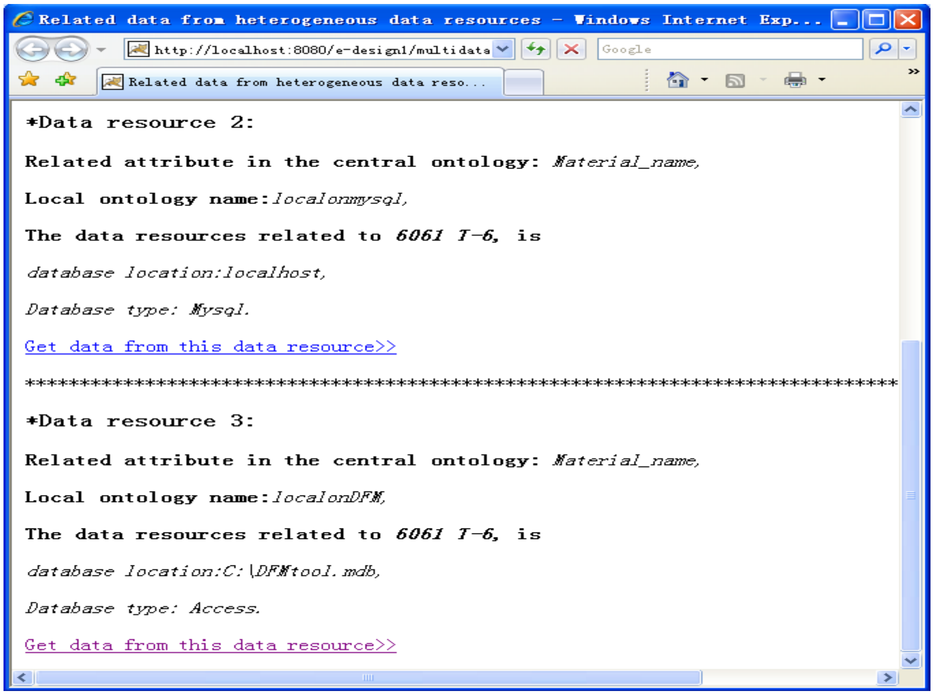
Step 2: Selection of component geometry & material
Component 1: Material:6061 T-6, Geometry: Angle, (L=300,B=50,H=50,T1=5,T2=5)
Component 2: Material:6061 T-6, Geometry: Sheet,(L=300,B=50,T=5)

Step 3: Selection of joint category

Lap

Figure 5-10: Screenshot of Step 3 of the ontology-based web system

The screenshots in Figures 5-11 and 5-12 show the data integration process and its query result. In this example, there are three data resources integrated, as shown in Figure 5-11. After the data resource is selected, the related data in the data resource can be retrieved. The list of possible design alternatives, which are integrated from heterogeneous data resources, can be exported with selected attributes to a text file in the local computer, as shown in Figure 5-12.



*Data resource 2:

Related attribute in the central ontology: *Material_name,*

Local ontology name:*localonmysql,*

The data resources related to *6061 T-6,* is

database location:localhost,

Database type: Mysql.

Get data from this data resource>>

****

*Data resource 3:

Related attribute in the central ontology: *Material_name,*

Local ontology name:*localonDFM,*

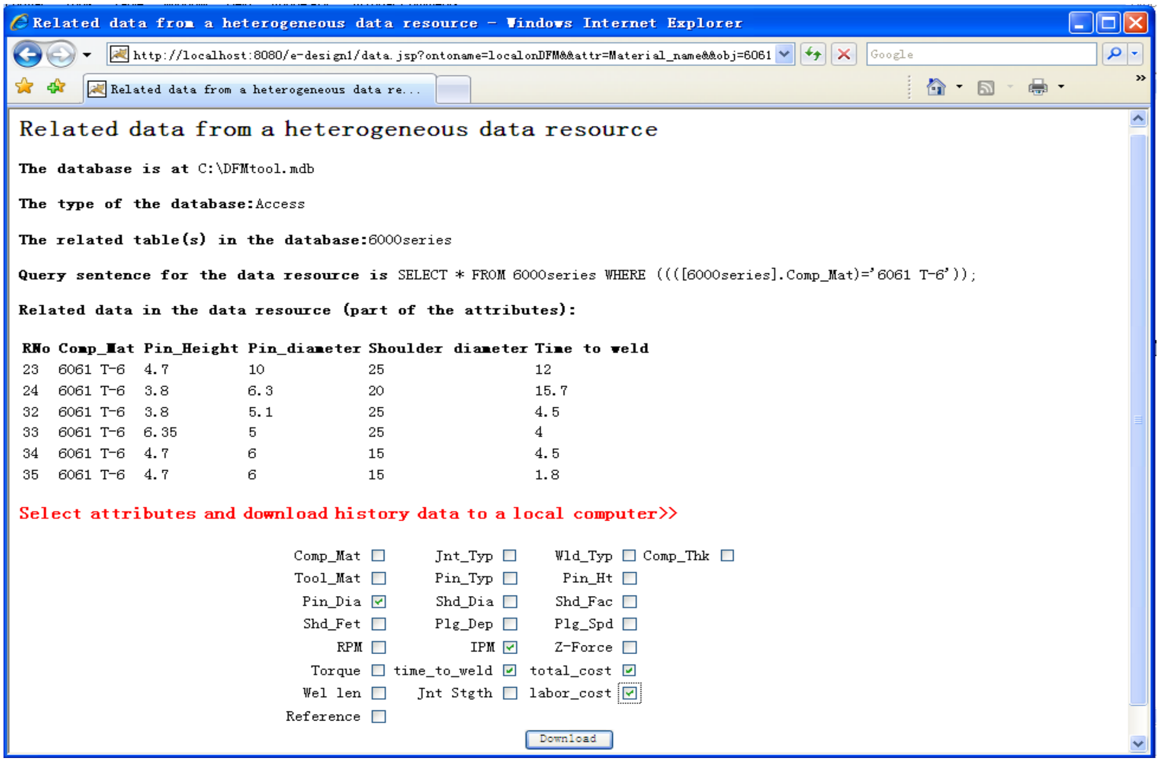
The data resources related to *6061 T-6,* is

database location:C:\DFMtool.mdb,

Database type: Access.

Get data from this data resource>>

Figure 5-11: Screenshot of integrated data resources based on ontology



Related data from a heterogeneous data resource

The database is at C:\DFMtool.mdb

The type of the database: Access

The related table(s) in the database:6000series

Query sentence for the data resource is SELECT * FROM 6000series WHERE ((([6000series].Comp_Mat)='6061 T-6'));

Related data in the data resource (part of the attributes):

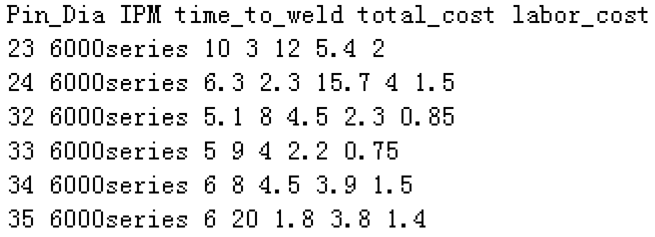
RNo	**Comp_Mat**	**Pin_Height**	**Pin_diameter**	**Shoulder diameter**	**Time to weld**
23	6061 T-6	4.7	10	25	12
24	6061 T-6	3.8	6.3	20	15.7
32	6061 T-6	3.8	5.1	25	4.5
33	6061 T-6	6.35	5	25	4
34	6061 T-6	4.7	6	15	4.5
35	6061 T-6	4.7	6	15	1.8

Select attributes and download history data to a local computer>>

Comp_Mat ☐ Jnt_Typ ☐ Wld_Typ ☐ Comp_Thk ☐
Tool_Mat ☐ Pin_Typ ☐ Pin_Ht ☐
Pin_Dia ☑ Shd_Dia ☐ Shd_Fac ☐
Shd_Fet ☐ Plg_Dep ☐ Plg_Spd ☐
RPM ☐ IPM ☑ Z-Force ☐
Torque ☐ time_to_weld ☑ total_cost ☑
Wel len ☐ Jnt Stgth ☐ labor_cost ☑
Reference ☐

Figure 5-12: Screenshot of integrated data from one data resource

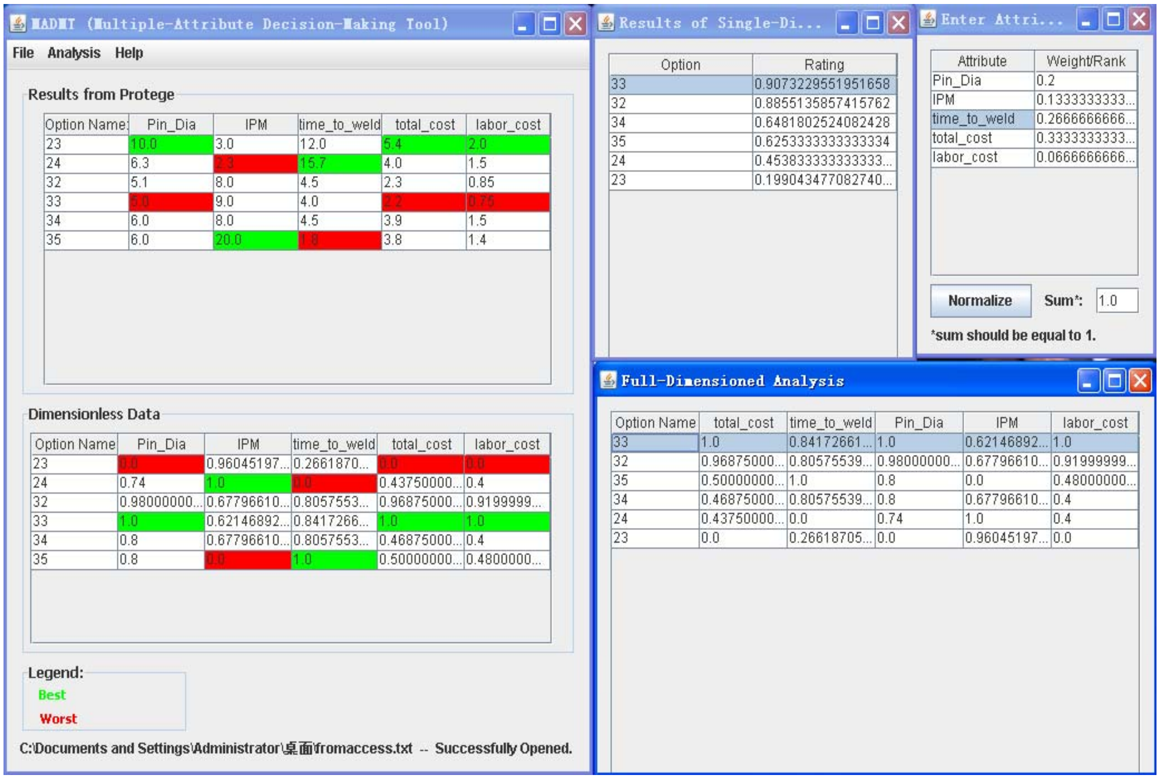
The exported text file is in the form shown in Figure 5-13. Attribute names that exported from the data resource are in the first row. Design alternatives are listed from the second row with id number and table name first and then attribute values in each row. For example, five attributes are exported, which are pin diameter (attribute 1), IPM (attribute 2), time to weld (attribute 3), total cost (attribute 4) and labor cost (attribute 5). All these attributes follow the rule that the smaller the value, the better. In addition, the importance sequence of these five attributes are 4, 3, 1, 2, 5.



```
Pin_Dia IPM time_to_weld total_cost labor_cost
23 6000series 10 3 12 5.4 2
24 6000series 6.3 2.3 15.7 4 1.5
32 6000series 5.1 8 4.5 2.3 0.85
33 6000series 5 9 4 2.2 0.75
34 6000series 6 8 4.5 3.9 1.5
35 6000series 6 20 1.8 3.8 1.4
```

Figure 5-13: Integrated data from the ontology-based web system for DFM

After the integrated data is exported to a local computer, MADMT can be used to compare them and find the best alternative for design reference. MADMT imports the text file, normalizes the value in it, and compares the difference among alternatives. It facilitates the full dimensioned analysis by sequencing the alternatives according to the importance order of attributes, and supports the single dimensioned analysis by calculating weights and sequencing the alternatives, as shown in Figure 5-14. The design of best design alternatives from historical data provides more help in the input phase of FSW design.



Results from Protege

Option Name	Pin_Dia	IPM	time_to_weld	total_cost	labor_cost
23	10.0	3.0	12.0	5.4	2.0
24	6.3	2.3	15.7	4.0	1.5
32	5.1	8.0	4.5	2.3	0.85
33	5.0	9.0	4.0	2.2	0.75
34	6.0	8.0	4.5	3.9	1.5
35	6.0	20.0	1.8	3.8	1.4

Dimensionless Data

Option Name	Pin_Dia	IPM	time_to_weld	total_cost	labor_cost
23	0.0	0.96045197...	0.2661870...	0.0	0.0
24	0.74	1.0	0.0	0.43750000...	0.4
32	0.98000000...	0.67796610...	0.8057553...	0.96875000...	0.9199999...
33	1.0	0.62146892...	0.8417266...	1.0	1.0
34	0.8	0.67796610...	0.8057553...	0.46875000...	0.4
35	0.8	0.0	1.0	0.50000000...	0.4800000...

C:\Documents and Settings\Administrator\桌面\fromaccess.txt -- Successfully Opened.

Option	Rating
33	0.9073229551951658
32	0.8855135857415762
34	0.6481802524082428
35	0.6253333333333334
24	0.453833333333333...
23	0.199043477082740...

Attribute	Weight/Rank
Pin_Dia	0.2
IPM	0.1333333333...
time_to_weld	0.2666666666...
total_cost	0.3333333333...
labor_cost	0.0666666666...

Sum*: 1.0

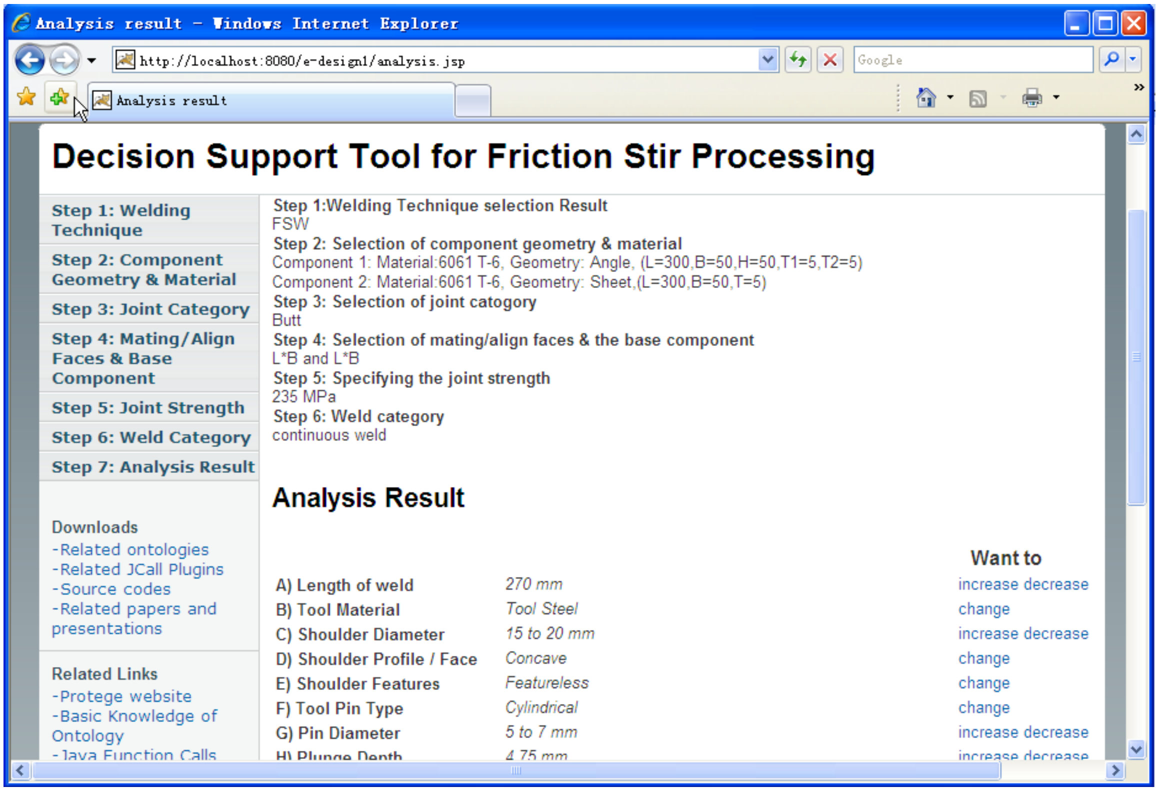
*sum should be equal to 1.

Option Name	total_cost	time_to_weld	Pin_Dia	IPM	labor_cost
33	1.0	0.84172661...	1.0	0.62146892...	1.0
32	0.96875000...	0.80575539...	0.98000000...	0.67796610...	0.91999999...
35	0.50000000...	1.0	0.8	0.0	0.48000000...
34	0.46875000...	0.80575539...	0.8	0.67796610...	0.4
24	0.43750000...	0.0	0.74	1.0	0.4
23	0.0	0.26618705...	0.0	0.96045197...	0.0

Figure 5-14: Screenshot of MADMT on a local computer

In this example, the design alternative 33 is best with the comparison of the other five by using both full dimensioned and single dimensioned analysis. In design alternative 33, joint category is Butt, the joint strength is 235MPa, and weld category is continuous weld. According to the historical data, the following steps of input are more readily specified.

After the inputs in the first six steps are specified, the platform calculates the output according to the technical requirements and formulas in Appendix E, as shown in the output in Figure 5-15.



Decision Support Tool for Friction Stir Processing

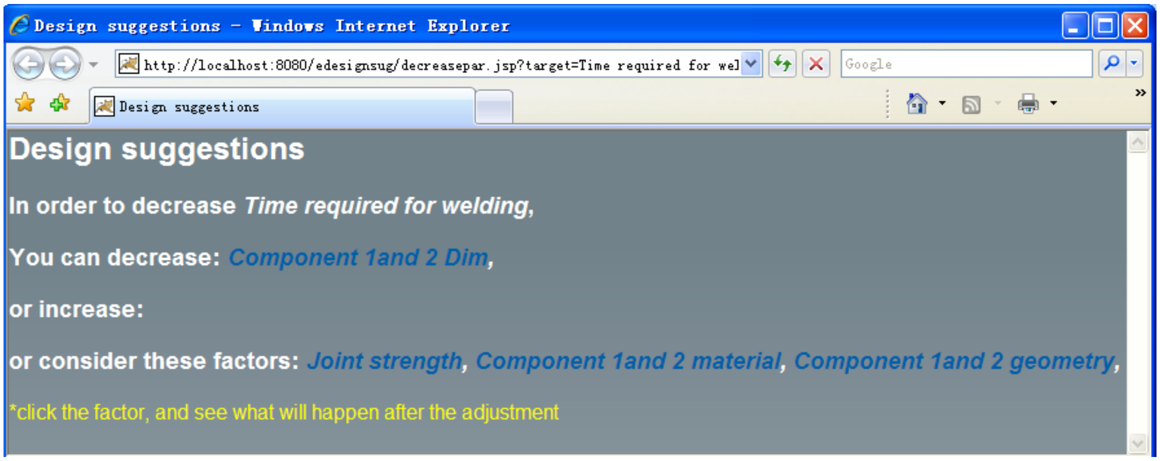
Step 1:Welding Technique selection Result
FSW
Step 2: Selection of component geometry & material
Component 1: Material:6061 T-6, Geometry: Angle, (L=300,B=50,H=50,T1=5,T2=5)
Component 2: Material:6061 T-6, Geometry: Sheet,(L=300,B=50,T=5)
Step 3: Selection of joint catogory
Butt
Step 4: Selection of mating/align faces & the base component
L*B and L*B
Step 5: Specifying the joint strength
235 MPa
Step 6: Weld category
continuous weld

Analysis Result

		Want to
A) Length of weld	*270 mm*	increase decrease
B) Tool Material	*Tool Steel*	change
C) Shoulder Diameter	*15 to 20 mm*	increase decrease
D) Shoulder Profile / Face	*Concave*	change
E) Shoulder Features	*Featureless*	change
F) Tool Pin Type	*Cylindrical*	change
G) Pin Diameter	*5 to 7 mm*	increase decrease
H) Plunge Depth	*4.75 mm*	increase decrease

Figure 5-15: Outputs of the ontology-based web system for DFM

After getting the outputs, some adjustments may be needed. For example, when a user wants to shorten the time required for welding, the tool will query the DFM e-Design ontology, which is described in Section 4.2.3, and find what are the root influence factors of the time required for welding. It reasons out the root influence factors of the time, which can be positive root influence, negative root influence or other root influence factors. After that, the details and the design adjustment hints can be retrieved from the knowledge base, as shown in Figure 5-16. Also, the outputs that will be influenced by the selected adjustment will also be given, as shown in Figure 5-17. Here the adjustment suggestion is to decrease component 1 and 2 dim or adjust the component 1 and 2 geometry, material or joint strength. Decreasing the component 1 and 2 dim can reduce the component 1 and 2 weight, machine table size, etc. and increase the strength to weight ratio. Adjusting the component material and joint strength can change the density, tool pin type, etc. Given these suggestions, the users can go back to the corresponding steps of the first six input nodes, adjust the value of inputs, and let the DFM e-Design tool calculate the output again.



Design suggestions

In order to decrease *Time required for welding*,

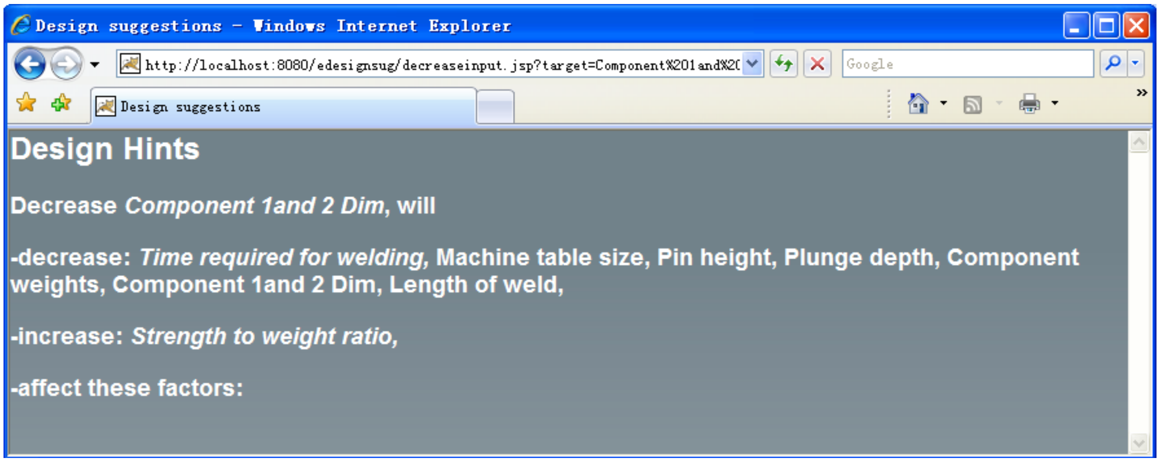
You can decrease: *Component 1and 2 Dim*,

or increase:

or consider these factors: *Joint strength*, *Component 1and 2 material*, *Component 1and 2 geometry*,

*click the factor, and see what will happen after the adjustment

Figure 5-16: Screenshot of parameter adjustment suggestion



Design Hints

Decrease *Component 1and 2 Dim*, will

-decrease: *Time required for welding,* Machine table size, Pin height, Plunge depth, Component weights, Component 1and 2 Dim, Length of weld,

-increase: *Strength to weight ratio,*

-affect these factors:

Figure 5-17: Screenshot of possible result for decreasing the dimension of Component 1 and 2

5.3 Conclusions

In this chapter, four cases are given to illustrate the ontology-based methods for data integration and decision support with the consideration of complex relations in technical requirements and economical objectives. The collaborators in CFSP gave feedback after reviewing the four cases, as quoted below.

"…At the beginning of our project, regarding the FSW design, I learned there was no

standard method to design the FSW process parameters. I also referred some articles (to understand how process parameters for FSW are decided) but they were not of much use. The logic that we developed for our e-Design tool is based upon the discussions with some folks working here on FSW projects and common sense…

…I think the job is almost done. The tool is working fine. Now it's just a matter of populating the FSW database with more data, developing more relations (my part) and then writing code for these relations for the e-Design Tool (your part). In other words, its all about creating more instances of what we have already done. So it is kind of repeating the same task.

These methods would surely allow designers to explore alternatives in a fast way. Previously there were no ways to compare the different methods used by designers.

For the question whether I anticipate the method would lead to a reduction in repeated work, accessing information faster, valuable decision suggestions and design guidance, enhanced design efficiency and quality, etc., my answer is YES.

…Previously, as I known there were no cost models existing for FSW. If they (cost models) were, then they were not put into practice as much. MIG cost models were being used but no such e-Design tool was existing which can compare two joining processes. I developed the cost model for MIG welding process with the case study we received (from GM), and applied the developed cost models to each and every joint. The same process was repeated for developing FSW cost model. We (together) came up with e-Design tool which can incorporate the example joints and also answer the queries for any given joint based on its knowledge and training done through the example joint.

I see that our TIE project as a whole can benefit in the future as it compares two joining process in terms of cost and mechanical design. The e-Design tool we developed is always an added advantage as we can reduced execution times with accurate results and develop complex relationships that are existing in plenty in our project. An example could be a joint. It can explain the way we calculate cost of the joint. A manual comparison can take several hours, and to do the same the e-Design tool can execute *faster than that."*

--Collaborators from CFSP

The feedback, from CFSP collaborators on the research in ontology development and utilization for knowledge management in product design indicates that it is a promising approach to solve the present problems in product design, which are related to data availability, quality, working efficiency, user guidance, design in a systematic way, etc.

Chapter 6: Contributions and Recommendations

In this dissertation, an ontology based method for knowledge management in product design has been proposed, realized and refined. Section 6.1 reviews the research objectives and the proposed research approach, concluding with answers to the research questions posed in Chapter 1. The resulting contributions of the research are summarized in Section 6.2. Limitations of the research and opportunities for future research are given in Section 6.3.

6.1 Research Summary

As stated in Chapter 1, the purpose of this research is to explore the ontology-based method to enhance the consistency and availability of data, increase the efficiency of data utilization, and support the decision making of product design for product life cycle. The field of Design for Manufacturing (DFM) is used in the study, which includes the overall process of manufacturing and focuses on the technical and economical criteria in manufacturing.

For this purpose, the DFM ontology and other assistant ontologies have been developed and used to capture the domain knowledge in DFM, to define a set of data and their structure and relations for product designers who need to share information in a domain of interest, to retrieve the feasible design alternatives from heterogeneous data resources, and to help designers make decisions based on technical and economic characteristics and relations. The ontology-based data integration method has been designed and implemented with a web-based system to guide users through the FSW process, integrate data from heterogeneous data resources, such as Excel worksheets, Access and Mysql databases, and provide references during design. The local decision support tool provides the capability to compare design alternatives through multi-attribute decision making methods. The ontology-based web system gives the analysis results based on the FSW e-Design process and support parameter adjustments by providing adjustment suggestions as well as the identification of possible influences through querying the DFM e-Design ontology.

For continuous improvement of the ontology-based knowledge management method, the possible errors that might occur in its development and utilization have been investigated and the ontology-based root cause analysis method has been proposed and developed. Moreover, for evaluation of the method in the future, an evaluation methodology has also been proposed. It can be used to compare the research with other knowledge management and decision support methods to validate its efficiency, efficacy and quality.

In summary, the resulting research method yields a systematic means to present a common understanding of domain knowledge, integrate multiple heterogeneous data resources from different departments and enterprises, find and utilize relations among concepts and properties, and help designers make decisions with the consideration of complex relations among technical requirements and economical objectives.

The use of this method to support knowledge management and decision making in product design is exploited in the context of three broad categories of research questions

(1) Questions on Ontology Development

Q1. What concepts, properties and constraints related to DFM, such as the product, cost, manufacturing requirements, etc. need to be structured in the ontologies? What ontologies need to be developed and what existing ontologies can be reused to represent these data in DFM?

Q2. Which ontology editor (such as Protégé, Oiled, Onto-Builder, etc.) should be used? How can the ontology be developed based on the chosen ontology editor? How can the concepts and properties in DFM and the complex relations among them be represented and utilized?

Concepts such as products, parts, manufacturing condition and materials are required, and are developed as classes in the DFM ontology in Section 4.2.1.1. Moreover, technical constraints described in Appendices C, D and E are also considered in the ontologies and the ontology-based web system.

In the research, Protégé is used as the ontology editor because of its advantages described in Section 2.3.2. The ontology development and utilization process is described in Section 3.1 and the details of ontology development are explained in Section 3.2.

AsD ontology (Kim et al., 2006) provides assembly information along with the assembly geometry. Based on the AsD ontology (Kim et al., 2006), more classes and slots are added into the DFM ontology as described in Section 4.2.1.1. Java Functions Calls are used to represent cost models and some technical requirements required in Appendix C and D and SWRL rules are used to find possible solutions, as explained in Section 3.2.3, Section 4.2.1.2, and case studies in Section 5.2.

DFM ontology has served as the central ontology during data integration. Other local ontologies are also developed in order to connect with data resources, as explained in Sections 3.3.3 and 4.2.2. In addition, in order to provide parameter adjustment suggestions, the DFM e-Design ontology is also developed in order to manage direct influence factors and infer root influence factors of the design parameters in decision support, as explained in Section 4.2.3. Moreover, in order to support root cause analysis for error control in the research, error ontology is developed with SWRL rules to help find root causes of causal factors for continuous improvement of the work, as explained in Sections 3.5 and 4.6.

(2) Questions on Ontology Utilization

Q3. What mode is suitable for an ontology-based information system capable of integrating and utilizing data on cost, manufacturing process, product, material, etc. for DFM? How can the system guide users to the proper information?

Q4. Which criteria (such as cost, time and quality) and constraints (such as feasibility, flexibility) do product designers consider in DFM? What is the suitable working mechanism for an ontology-based decision support tool that can help designers consider the DFM problem step by step, compare the design alternatives and make decisions based on the criteria?

The overall framework of the ontology-based system is described in Section 4.1. The advantages of using Browser/Server (B/S) mode information system are that it does not require users to install any kind of client programs; information is stored and managed in the server, which can provide consistent and updated data that different users share; and it does not require that users learn any specific tool or programming language, such as Protégé, JTP, and so on. That is why B/S mode is used to develop the ontology-based information system, and the data reasoning method described in Section 3.3.2 is used to connect ontologies with the information system. For the comparison of historical data, the local program is used since it is flexible to compare data not only from web system but also from other sources, such as Protégé, the ontology editor, as explained in Section 4.4.

Criteria and constraints used in DFM are considered during the case studies in Chapter 5. In Case Study 1 in Section 5.2.1, the trade-off between machinery cost, power cost and filler material (electrode) is made in order to get the lowest total weld cost with the consideration of material and machine requirement and availability. Technical requirements (feasibility) and availability analysis of material, machine and tools are often considered during DFM, as illustrated in Case Study 2 of Section 5.2.2 and in Case Study 3 of Section 5.2.3. Attributes related to time and cost are used to compare historical data in Case Study 4 of Section 5.2.4.

The decision making in DFM relates to multiple steps (nodes), and each node has multiple choices. So the decision made here belongs to the structured decision type. In the suitable ontology-based system, users are guided to the proper information according to a tree structure, which can be formed based on the integrated design alternatives. The tree can also be formed according to the specified steps, as explained in Sections 3.4 and 4.5. The tree structure divides the design inputs into several categories, and each category is corresponding to one level of nodes. The later nodes might be affected by the previous nodes. For example, in FSW design, the mating faces of the two components to be welded are influenced by the geometry of the two components, which is specified in a previous step of the process. Designers can consider the DFM problem step by step along the tree. In each node of the tree, designers can retrieve the reference of best historical design alternatives that meet the input in the previous nodes before inputting additional design parameters. Historical data is integrated from heterogeneous data resources based on the DFM ontology (central ontology) and local ontologies, as explained in Sections 3.3 and 4.3. After input, analysis results are also provided by the system according to the models in the system. Parameter adjustment suggestions and the possible influence of the adjustment can also be retrieved by querying the DFM e-Design ontology, as explained in Section 4.5.

(3) Questions on Refinement and Evaluation of the Research Approach

Q5. How can errors in ontology development, maintenance and utilization be reduced? The utilization includes information integration and decision support tools.

Q6. How can the benefits to a DFM ontology and decision support tool be evaluated?

In order to reduce errors in ontology development, maintenance and utilization, possible errors are first categorized in Sections 3.5.1-3.5.3, and the ontology-based root cause analysis for reducing errors and continuous improvement in the research is put forward in Section 3.5.4 and described in detail in Section 4.6.

The evaluation criteria for the research are summarized in Section 3.6 with the independent and dependent variable analysis.

6.2 Contributions of the Research

The research has contributed to the knowledge management and decision support in product design, especially in DFM through a method based on ontologies. Through case studies in Chapter 5, the research contributions are illustrated through the consideration of cost, time and technical requirements, locating the proper information, guiding users and providing design suggestions.

A primary contribution of this research is that it offers a formal specification of knowledge in DFM, which can facilitate the sharing of knowledge and collaboration between designers in a distributed and flexible organization. The DFM ontology in this research can be used to capture various manufacturing and assembly concepts and to share joining information along with the assembly geometry at the same time. This ontology can facilitate collaborative design teams in performing various cost and technical evaluations of different facets in DFM for shortened product development cycles and it also provides the information about local ontologies and the corresponding data resources for data integration.

Secondly, the proposed ontology-based knowledge management method enhances the accuracy and consistency of data used. Ontologies are good at relation representation and utilization. Since relations among data are realized, the inconsistency and inaccuracy can be found when data is input into the ontology. For example, after the basic design parameters of MIG welding are input, such as thickness of components and length of weld, the electrode cost can be calculated automatically through the Java Function Calls programs. If the cost and design parameters do not match with each other according to the cost model, this can be detected.

Thirdly, the research explores a suitable ontology-based information integration system to enhance the availability of data from heterogeneous data resources. Previous research mainly focused on how to translate the query sentences to the proper format for the data resources by utilizing the semantics in ontologies. With the rapid growth of new technologies and data resources, designers find it difficult to identify the proper aspects of information and to locate the data resources. The ontology-based data integration method in this research extends previous ontology-based integration method and guides the users to the proper information. After the proper data resources are located, the detail information can be queried out from different resources. A local Multi-Attribute Decision-Making Tool (MADMT) is also developed and utilized to help analyze the integrated data through multi-attribute analysis. In this way, users avoid abundant and useless information, and utilize only proper detail data which enhances work efficiency and accuracy.

Fourthly, relations, criteria and cost models are realized in the DFM ontology and DFM e-Design ontology. The ontology-based method supports the decision making in product design by considering the technical and economical constraints and relations and making tradeoff among them.

According to Power (2002), decision quality is judged by decision's compatibility with existing constraints, its timeliness, and its incorporation of the optimal amount of information. With the improvement of the aspects above, the method is helpful to enhance the decision quality in product design.

Moreover, the research expedites the growth of ontology-based method by exploring an ontology-based root cause analysis method for errors in the development, maintenance and utilization of ontologies. It also provides evaluation methodology for the ontology-based knowledge management and decision making method.

6.3 Limitations of the Research and Opportunities for Future Research

This topic has been studied in the context of DFM and lays the foundation for a generalized and comprehensive framework for a broadly defined system to support design across lifecycle phases. In the future, the research will be expanded into the entire product design process to manage and reuse a large number of different data related to each phase of the product life cycle, and support decision making in product design from a broad perspective.

In present research, it was found that conflict or repeat in the integrated data is an obstacle for its further use. The format of data is also a problem. Data from heterogeneous data resources are in different units. For example, the same weight can be expressed by ounce, gram, etc. Formulas are also based on different unit systems. In the future research, the integrated data will need to be transformed into a proper format based on ontology, the method to reduce the conflict data and remove the repeat part in data integrated from different places will also need to be investigated. There have been ontologies to transform data in different units (Gruber and Olsen, 1994; Borchert, 2003). Ontology has advantages in the representation and utilization of semantics. In the data transformation method, through querying the proper ontologies and transform the data, the problems in the integrated data will be solved.

CFSP and CED are both formed by multiple universities. The collaboration is dynamic according the requirements of research and education. After the system is developed, it was found that the data resources can not be located sometimes since the sources did not exist or were not in a usable format. The ontology-based method needs to be expanded in the future to automatically find necessary but non-existing data, reserve it from the proper place and later transform and integrate it for further use.

The integration of decision support system (DSS) with other information resources and platforms is one of the major challenges facing DSS (Marakas, 1999). The ontology-based decision support in this research lays the foundation for future work in this area. In order to enrich the function of the decision support tool and provide more flexibility to meet customers' requirements, other tools, such as statistical tools and simulation tools, might be required to support decision making. A port that connects the system in this research with other tools will be studied and developed. Present design input and analysis results will be able to output to the proper tools through the port, and the output results from these tools will also be transformed and sent back through the port.

Finally, the evaluation of this research is at the methodology level. In the future, the ontology development and utilization for knowledge management in product design will be evaluated with the comparison with other methods used in knowledge management in product design.

References

Alexiev V., Breu M., Bruijn J.de, Fensel D., Lara R. and Lausen H., Information integration with ontolgies, John Wiley & Sons Ltd, 2005.

Andersen Bjørn and Fagerhaug Tom, Root cause analysis : simplified tools and techniques. Milwaukee, Wisconsin: ASQ Quality Press, 2000.

Arbegast William J., Friction stir welding after a decade of development It's not just welding anymore, Welding Journal, Vol. 85, no. 3, pp. 28-35, Mar. 2006.

Arens Yigal, Knoblock Craig A. and Shen Wei-Min, Query reformulation for dynamic information integration, International Journal on Intelligent and Cooperative Information Systems, Vol. 6, Nos. 2/3, Jun. 1996.

Aufaure Marie Aude, Grand Benedicte Le, Soto Michel and Bennacer Nacera, Metadata- and ontology-based semantic web mining, Web semantics & ontology, pp.259-295, 2006.

Bagaitkar Harish Sadanand, the Process of the e-Design tool for FSW, working report, UMR, Jun. 2007.

Bandyopadhyay Sanghamitra, Maulik Ujjwal, Holder Lawrence B., Cook Diane J. (Ed.), Advanced methods for knowledge Discovery from complex data, Springer, 2005.

Bennet J.L. (Ed.). Building decision support systems. Reading, Ma: AddisonWesley. 1983.

Bernaras A., Laresgoiti L. and Corera J., Building and reusing ontologies for electrical network applications, European Conference on Artificial Intelligence (ECAI 1996), Budapest, Hungary, John Wiley and Sons, Chichester, United Kingdom, pp. 298-302, 1996.

Berners-Lee, T., Hendler, J. and Lassila, O., The semantic web, Scientific American, May 2001.

Bhavani Thuraisingham, Eric Hughes and Dock Allen, Dependable Semantic Web, Proceedings of the Seventh International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS 2002), 2002.

Boniface Duane, Risk based decision analysis, Vol. 3, Risk Assessment Tools Reference.Marine Safety and Environmental Protection, Coast Guard, United States. 1997.

Borchert Reiner, Measure Units ontology, http://www.oekon.de/protege/jcalls/index.html, 2003.

Box George E.P., Hunter J.Stuart and Hunter William G., Statistics for Experimenters (2nd Edition), 2005.

Brickley, D. and Guha R.V., RDF vocabulary description language 1.0: RDF schema. W3C, http://www.w3.org/TR/rdf-schema, 2003.

Burling W.A., Lax B.J.B., O'Neill, L.A. and Pennino T.P., Product design and introduction support systems, AT&T Technical Journal, Vol. 66, No.5, pp.21-38, 1987.

Canada John R., Sullivan William G., White John and Kulonda Dennis, Capital Investment analysis for engineering and management, Pearson Prentice Hall, 2005.

Ceccaroni Luigi, Cort´es Ulises and Sanchez-Marre Miquel, OntoWEDSS: an ontology-underpinned decision-support system for wastewater management, Proceedings of iEMSs 2002, Lugano, Switzerland, 24-27 June 2002.

Ceccaroni Luigi, ONTOWEDSS-An ontology-based environmental decision-support system for the management of wastewater treatment plants, Ph.D. dissertation, Universitat politecnica de catalunya, Barcelona, Spain, Oct. 2001.

Chang Xiaomeng, Sahin Asli, Terpenny Janis P., Ontology based Graphical Modeling Tool for Product Conceptual Design, 17th International Conference on Flexible Automation and Intelligent Manufacturing (2007 FAIM), Philadelphia, PA, June 18-20, 2007.

Charney, C., Time to Market. Reducing Product Lead Time, Society of Manufacturing Engineers, Dearborn, MI, 1991.

Chen Rung-Ching, Lee Ya-Ching and Pan Ren-Hao, Adding new concepts on the domain ontology based on semantic similarity, Business And Information (BAI), Singapore, Jul. 12-14, 2006.

Creese R.C., Adithan M. and Pabla B.S., Estimating and costing for the metal manufacturing Industries. New York: M. Dekker, 1992.

Crubézy, M., The Protégé Axiom Language and Toolset ("PAL"). http://protege.stanford.edu/plugins/paltabs/pal-documentation/, 2002.

Cruz Isabel F., Xiao Huiyong and Hsu Feihong, An Ontology-based Framework for XML Semantic Integration, Proceedings of International Database Engineering and Applications Symposium (IDEAS'04), pp.217 – 226, Jul. 7-9,2004.

Dellas Nikolaos L., Koutsoloukas Eleftherios A., Lioudakis, Georgios V., Kaklamani, Dimitra I. and Venieris Iakovos S., A Performance Comparison of Ontology Reasoning and Rule Engines. Mobile and Wireless Communications Summit, 16th IST, pp. 1-5, Jul. 2007.

DeLone WH and McLean ER, Information systems success: The quest for the dependent variable. Information Systems Research, Vol.3, Iss. 1, pp.60–95, 1992.

Denny Michael, Ontology editor survey results, http://www.xml.com/2004/07/14/examples/Ontology_Editor_Survey_2004_Table_-_Michael_Denny.pdf, 2004.

Dong Jin Song, Sun Jing and Wang Hai, Checking and Reasoning about Semantic Web through Alloy, Technical report (TRB9/02, Sep. 2002), National University of Singapore, http://nt-appn.comp.nus.edu.sg/fm/alloy/introduction.htm, Sep. 2002.

Dragan Gasevic, Gragan Djuric and Vladan Devedzic, Model driven architecture and ontology development, Springer-Verlag Berlin Heidelberg, 2006.

Drucker P., Entrepreneurship in Business Enterprise, Journal of Business Policy, Vol.1, 1970.

Farbey B, Land F and Targett D, How to assess your IT investment: a study of methods and practices. Butterworth Heinemann, Oxford, 1993.

Fernandes Rui, Grosse Ian and Krishnamurty Sundar, Design and Innovative Methodologies in a Semantic Framework, IDETC/CIE 2007, Las Vegas, Nevada, USA, Sep. 2007.

Fern´andez M, G´omez-P´erez A and Juristo N, METHONTOLOGY: from ontological art towards ontological engineering. In: Proceedings of AAAI97 spring symposium series, workshop on ontological engineering, Stanford, CA, pp. 33–40, 1997.

Fern´andez M, G´omez-P´erez A, Sierra AP, Sierra, JP, Building a chemical ontology using METHONTOLOGY and the ontology design environment, Intelligent Systems and their Applications, IEEE, Vol.14, Iss. 1, pp.37–46,1999.

Frank Gleb, Jenkins Jessica and Fikes Richard, JTP: An Object-Oriented Modular Reasoning System, http://www.ksl.stanford.edu/software/JTP/, 2004.

Freed Michael, Predicting Design-Facilitated Error, Poster presented to the 1996 Conference of the Cognitive Science Society, http://human-factors.arc.nasa.gov/cognition/papers/freed/cogsci/designerr.html, 1996.

Freitas Alex and Carvalho Andre C.P.L.F de, A tutorial on hierarchical classification with application in bioinformatics, Research and Trends in Data Mining Technology and Applications, Idea Group Publishing, 2007.

Gad Shayne Cox, Statistics and Experimental Design for Toxicologists, 3rd edition, CRC Press, 1999.

Gandon F., Distributed artificial Intelligence and knowledge management: ontology and multi-agents systems for a corporate semantic web, Ph.D. thesis, Inria-Sophia, France, 2002.

Gargouri, Y, Lefebvre, B and Meunier, JG, Ontology maintenance using textual analysis, Proceedings of 7TH World Multiconference on Systemics Cybernetics, Cybernetics and Informatics, Vol. I, pp.248-253, 2003.

Gomez-Perez, A., Fernandez-Lopez, M. and Corcho O., Ontological engineering, Springer, 2003.

Graham A., The DAD Reamework Interactive, IEEE Design and Test of Computers, Sep. 1991.

Grosse Ian R., Milton John M. and Wileden Jack C., Ontologies for supporting engineering analysis models, Artificial Intelligence for Engineering Design, Analysis and Manufacturing archive, Vol.19, Iss.1, pp.1-18, Jan. 2005.

Gruber Thomas R., Toward principles for the design of ontologies used for knowledge sharing, International Journal of Human-Computer Studies, Vol.43, Iss.5-6, pp.907-928, Nov.1995.

Gruber Thomas R. and Olsen Gregory R., An Ontology for Engineering Mathematics, Fourth International Conference on Principles of Knowledge Representation and Reasoning, Gustav Stresemann Institute, Bonn, Germany, Morgan Kaufmann, 1994.

Gruninger M, Designing and evaluating generic ontologies, Proceedings of ECAI96's workshop on ontological engineering, Budapest, Hungary, pp 53–64, 1996.

Gruninger M and Fox M, Methodology for the design and evaluation of ontologies, Proceedings of IJCAI95's workshop on basic ontological issues in knowledge sharing, Montreal, Canada, 1995.

Guarino N., Formal Ontology in Information Systems, Proceedings of FIOS'98, Amsterdam, IOS Press, pp.3-15, Trento, Italy, Jun. 6-8, 1998.

Haarslev V. and Moeller R., Racer: A core inference engine for the Semantic Web, In 2nd International Workshop on Evaluation of Ontology-based Tools (EON-2003), Sanibel Island, FL, 2003.

Hamilton Scott and Chervany Norman L., Evaluating Information System Effectiveness - Part I: Comparing Evaluation Approaches, MIS Quarterly, Vol. 5, No. 3, pp. 55-69, Sep. 1981.

Hatamura Yotaro, Decision-making in engineering design, Springer, 2006.

Horridge Matthew, Knublauch Holger, Rector Alan, Steven Robert and Wroe Chris, A practical guide to building OWL ontologies using the Protege-OWL plugin an CO-ODE tools, http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf, Aug. 2004.

Horrocks Ian, DAML+OWL: A reason-able Web ontology language, **,** Proceedings of EDBT 2002, Lecture Notes in Computer Science, Vol.2287, pp.103-116, Springer, 2002.

Horrocks Ian, Patel-Schneider Peter, Boley Harold, Tabet Said, Grosof Benjamin, Dean Mike, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, http://www.daml.org/2003/11/swrl/, Nov.2003.

Huang G.Q., Design for X: Concurrent engineering imperatives, Chapman&Hall, 1996.

Hwang Hyun-Suk, Park Kyoo-Seok, and Kim Chang-Soo, Ontology-Based Information Search in the Real World Using Web Services, Lecture Notes in Computer Science, v 3982 LNCS, Computational Science and Its Applications - ICCSA 2006: International Conference, Proceedings - Part III, pp. 125-133 , 2006.

Hyeon H. Jo, Hamid R. Parsaei, Willam G. Sullivan, Principles of concurrent engineering, Concurrent Engineering contemporary issues and modern design tools, Chapman & Hall, 1993.

Irani Zahir, Information systems evaluation: navigating through the problem domain, Information & Management , Vol.40. Iss.1 , pp.11–24, Oct. 2002.

James J. Rooney and Lee N. Vanden Heuvel, Root cause analysis for beginners, American Society for Quality, Quality Progress, pp. 45-53, Jul. 2004.

Kim K.-Y., Manley D. G., and Yang H., Ontology-based assembly design and information sharing for collaborative product development, CAD Computer Aided Design, Vol. 38, pp. 1233-1250, 2006.

Kim Hyun Hee, ONTOWEB: Implementing an ontology-based web retrieval system, Journal of the American Society for Information Science and Technology, Vol.56, No.11, pp. 1167-1176, Sep. 2005.

Klein Michel, Fensel Dieter, Kiryakov Atanas and Ognyanov Damyan, Ontology Versioning and Change Detection on the Web, Proceedings of 13th International Conference Knowledge Engineering and Knowledge Management (EKAW 02). LNCS 2473. Springer-Verlag. pp. 197-212, 2002.

Klyne Graham and Carroll J. Jeremy, Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C, http://www.w3.org/TR/rdf-concepts/#section-Graph-syntax, 2004.

Knublauch Holger, An AI tool for the real world, Java World, http://www.javaworld.com/javaworld/jw-06-2003/jw-0620-protege.html?page=1, 2003.

Knublauch Holger, Musen Mark A., Rector Alan L., Editing Description Logic Ontologies with the Protégé OWL Plugin, International Workshop on Description Logics, pp.79-87, 2004.

Krishnamurty Sundar, Normative decision analysis in engineering design, chapter 4 of decision making in engineering design, ASME 2006.

Kumar, T. P. and Allada V., a cost model for the metal inert gas (MIG) welding process, DETC 2007, Las Vegas, Sep. 2007.

Kumar, T.P. and Mishra Rajiv, E-Design tools for Friction Stir Welding Processing: Cost Estimation Tool, Working report, University of Missouri Rolla, Apr. 2007

Kumar Kuldeep, Post implementation evaluation of computer-based information systems: current practices, Computeing Practices. ACM, Vol. 33 No. 2, Feb. 1990.

Lambrix Patrick, Habbouche Manal and Perez Marta, Evaluation of ontology development tools for bioinformatics, Bioinformatics, Oxford University Press, Vol. 19, No. 12, pp. 1564–1571, 2003.

Land F, Evaluation of systems goals in determining a design strategy for a computer-based information system. Computer Journal 19(4), pp.290–294, 1976.

Lassila, O. and Swick, R., Resource description framework (RDF) model and syntax specification. http://www.w3.org/TR/REC-rdf-syntax/, 1999.

Le Blanc Louis A. and Rucks Conway T., A Discriminant Analysis of Information Systems Effectiveness:Categorical Dependent Variables of DSS Impact, Proceedings of the 29th Annual Hawaii International Conference on System Sciences, 1996.

Lederer Albert L., Mirani Rajecsh, Neo Boon Siong, Pollard Carol, Prasad Jayesh and Ramamurthy K., Information system cost estimating: A management perspective, MIS Quarterly, Vol.14, Iss.2, pp.159–176, Jun. 1990.

Lee Kang W., Tillman Frank A. and Higgins James J., A Literature Survey of the Human Reliability Component in a Man-Machine System, IEEE Transactions on Reliability, Vol. 37, No. 1, Apr. 1988.

Li Guan-yu, Yu Sui-ming and Dai Sha-sha, Ontology-based query system design and implementation, Network and Parallel Computing Workshops 2007, NPC Workshops. IFIP International Conference, pp.1010 – 1015, Sep. 18-21, 2007.

Li Yu, Huang Biqing, Liu Wenhuang, Gou Hongmei, Wu Cheng, Ontology for goal Management of virtual enterprises, 2001 IEEE International Conference on Systems, Man, and Cybernetics, pp.2034 - 2040 vol.3, Oct. 7-10, 2001a.

Li Yu, Huang Biqing, Liu Wenhuang, Gou Hongmei, Wu Cheng, Ontology based decision support system for partner selection of virtual enterprises, 2001 IEEE International Conference on Systems, Man, and Cybernetics, pp.2041 - 2045 vol.3, Oct. 7-10, 2001b.

Lin Sanbao, Yang Chunli, Li Ming and Wu Lin, Feature-based and object-oriented product information model for welding structure, China-Welding, Vol.8, No.2, Nov. 1999.

Loaiza Francisco L. and Wartik Steven P., A GH-based ontology to support applications for automating decision support, Institute for defense analyses, IDA paper P-3934, March 2005.

Marakas George M., Decision support systems in the twenty-first century, Upper Saddle River, N.J.: Prentice Hall, 1999.

MacGregor, R., Chalupsky, H., Moriarty, D. and Valente, A., Ontology Merging with OntoMorph, http://reliant.teknowledge.com/HPKB/meetings/meet040799/Chalupsky/index.htm, 1999.

Martin Nancy L., Pearson John M. and Furumo Kimberly A, IS Project Management: Size, Complexity, Practices and the Project Management Office, Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.

Mason Richard O., Measuring Information output: A Communication Systems Approach, Information & Management, pp.219-234, Oct. 1978.

McGuinness Deborah L. and van Harmelen Frank, OWL Web Ontology Language Overview, W3C Recommendation, http://www.w3.org/TR/owl-features/, Feb. 2004.

McGuinness D.L., Fikes, R., Rice, J. and Wilder, S., An Environment for Merging and Testing Large Ontologies. Principles of Knowledge Representation and Reasoning: Proceedings of 7th International Conference (KR2000), Morgan Kaufmann, pp.483-493, 2000.

Meister D., the nature of human error, Global Telecommunications Conference and Exhibition, 'Communications Technology for the 1990s and Beyond', GLOBECOM '89., IEEE,1989.

Mishra R.S. and Ma Z.Y., Friction stir welding and processing, Materials Science and Engineering: R: Reports, Vol.50, Iss. 1-2, pp.1–78, Aug.2005.

Mistree Farrokh, Smith Warren and Bras Bert, A decision-based approach to concurrent design, Concurrent Engineering contemporary issues and modern design tools, Chapman & Hall, 1993.

Mitra P., Wiederhold G., and Kersten M., A Graph-Oriented Model for Articulation of Ontology Interdependencies, Proceedings of Conference Extending Database Technology (EDBT 2000), LNCS 1777, Springer-Verlag, pp. 86-100, 2000.

Noy Natalya F. and Klein Michel, Ontology Evolution: Not the Same as Schema Evolution, Knowledge and Information Systems, Springer London, Vol.6, No.4, Jul. 2004.

Noy, N.F. and Musen M.A., Ontology Versioning in an Ontology Management Framework. IEEE Intelligent Systems. Vol. 19, Issue 4, pp.6-13. Jul.-Aug. 2004.

Noy, N.F. and McGuinness, D.L., Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001.

Noy, N. F. and Musen, M. A. , PROMPT: Algorithm and tool for automated ontology merging and alignment. Seventeenth National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, pp. 450-455, 2000.

Noy N. F. and Sintek Michael, Decker Stefan, Crubézy Monica, Fergerson Ray W., Musen Mark A., Creating Semantic Web Contents with Protégé-2000, IEEE Intelligent Systems, 2001.

Object Management Group, UML 2.0 Superstructure Specification. www.uml.org, 2004.

O'Connor MJ, Knublauch H, Tu SW, Grosof B, Dean M, Grosso William and Musen Mark, Supporting Rule System Interoperability on the Semantic Web with SWRL, International Semantic Web Conference (ISWC2005), Galway, pp. 974-986, 2005.

Oden J. Tinsley and Prudhomme Serge, Estimation of Modeling Error in Computational Mechanics, Journal of Computational Physics, Vol.182, Iss. 2, pp.496–515, 2002.

O’Flynn Michael J. and Ahmad M. Munir, Computer-based concurrent engineering systems, Concurrent Engineering contemporary issues and modern design tools, Chapman & Hall, 1993.

O’Guinn Michael C., The complete guide to activity based costing, CCH Tax and Accounting, 1991.

Olson David LeRoy, et al., Welding, Brazing, and Soldering, ASM Handbook, Vol. 6, Materials Park, Ohio, 1993.

Park J. and Simpson T., Development of a Production Cost Estimation Framework to Support Product Family Design, International Journal of Product Research, Vol.43, pp. 731-772, 2005.

Park Jaeil, Terpenny Janis and Chang Xiaomeng, Toward an Activity-based Cost Ontology for Product Family Planning, 2006 IIE Annual Conference & Exposition, May 2006.

Parsaei Hamid R. and Sullivan William G., Concurrent Engineering, Chapman & Hall, 1993.

Pawar Kulwant S., Menon Unny and Riedel Johann C.K.H., Time to Market , Integrated Manufacturing Systems, Vol. 5, Iss. 1, 1994.

Pinto Helena Sofia and Martins João P., Ontologies: How can They be Built?, Knowledge and Information Systems, Springer London, Vol.6, No.4, July 2004.

Power Daniel J., Decision support systems: concepts and resources for manager. Westport, Connecticut, London: Quorum Books, 2002.

Rabinovich Semyon G., Measurement errors and uncertainties, Springer Science and Media, Inc., 2005.

Reiner B., How can a knowledge base run executables on the frame level?, International Protege Workshop Manchester, England, 2003.

Rooney, J.J. and Heuvel, L. N. V., Root cause analysis for beginners, American Society for Quality, Quality Progress, Vol. 37, Iss. 7, pp. 45-53, 2004.

Saaksvuori Antti and Immonen Anselmi, Product Lifecycle Management, Second Edition, Springer, 2005.

Schriven M., The methodology of evaluation: formative and summative evaluation. Evaluating Action Programs, C.H. Weiss, ed., Allyn and Bacon, Boston, Massachusetts, 1972.

Sen Pratyush and Yang Jian-Bo, Multiple Criteria Decision Support in Engineering Design. London; New York: Springer, 1998.

Shannon Claude E. and Warren Weaver, The Mathematical Theory of Communication, University of Illinois Press, Urbana, IL, 1949.

Shen S-H., Smidts C. and Mosleh A., A methodology for collection and analysis of human error data based on a cognitive model: IDA, Nuclear Engineering and Design, Vol.172, No. 1, pp.157-186, Jul. 3, 1997.

Smithson S and Hirschheim R, Analysing information systems evaluation: another look at an old problem, European Journal of Information Systems, Vol.7, No. 3, pp.158-174, Sep. 1998.

Solow, D., How to Read and Do Proofs: An Introduction to Mathematical thought Processes, 4th ed., John Wiley & Sons, Hoboken, N.J, 2005.

Song, Y., Zhang, B., Cai, F. and Meng, Q., Knowledge-based design for manufacture system, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Vol. 2, pp.1220-1224, 1996.

Sprague Ralph H., Jr., A framework for the development of decision support systems, MIS Quarterly, Vol. 4, No. 4., pp. 1-26, Dec. 1980.

Stojanovic, N. and Stojanovic, L., A Logic-based Approach for Query Refinement in

Ontology-based Information Retrieval Systems, Proceedings. 16th IEEE International Conference on Tools with Artificial Intelligence, pp. 450-457, 2004.

Stuckenschmidt Heiner and van Harmelen Frank, Information sharing on the semantic web, Springer-Verlag Berlin Heidelberg, 2005.

Sullivan Willam G., Wicks Elin M. and Luxhoj James T., Engineering Economy, Thirteenth Edition, Pearson Prentice Education, Inc., 2005.

Swartout B., Patil R., Knight K. and Russ T., Toward distributed use of large-scale ontology, AAAI'97 Spring Symposium on Ontological Engineering, Standford University, Californa, pp.138-148, 1997.

Taki Hirokazu, Matsuda Noriyuki, Hiramatsu Michiyasu, Naito Yuki, Nakajima Jiro, Nakamura Tadashi, Imagawa Akihisa, Matsuzawa Yuji, Abe Norihiro and Hori Satoshi, Ontology-Based Medical Information Service System, Lecture Notes in Artificial Intelligence, Vol. 2774, Knowledge-Based Intelligent Information and Engineering Systems, pp. 358-365, 2003.

Terpenny, J.P., Collaborative Research: A TIE Research Program on e-Design for Friction Stir Welding and Processing, National Science Foundation, Industry University Cooperative Research Center (IUCRC), NSF EEC-0632758 to Virginia Tech, 09/01/06-08/31/08, in collaboration with Rajiv S Mishra and Venkat Allada of University of Missouri-Rolla, NSF EEC-0632803, 2006.

Thomas W. M., Nicholas E. D., Needham J. C., Murch M. G., Templesmith P. and Dawes C. J., Friction Stir Butt Welding, G.B. Patent Application No. 9125978.8, Dec. 1991; U.S. Patent No. 5460317, Oct. 1995.

Tjoa1A Min, Andjomshoaa1Amin, Shayeganfar1Ferial and Wagner Roland, Semantic Web Challenges and New Requirements, Proceedings of the 16th International Workshop on Database and Expert Systems Applications,DEXA, 2005.

Tu Samson, A tutorial on PAL (Protégé Axiom Language). Fifth International Protégé User Group Meeting. URL:

http://protege.stanford.edu/plugins/paltabs/pal-quickguide/PALTutorial.pdf., 2001.

Uschold M, Converting an informal ontology into Ontolingua: some experiences, Proceedings of ECAI96's workshop on ontological engineering, Budapest, Hungary, pp. 89–99, 1996.

Uschold M. and King M., Towards a methodology for building ontologies, Proceedings of IJCAI95's workshop on basic ontological issues in knowledge sharing, Montreal, Canada, 1995.

Varga Peter, Meszaros Tamas, Dezsenyi Csaba, and Dobrowiecki Tadeusz P., An Ontology-based Information Retrieval System. Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), Vol. 2718, Developments in Applied Artificial Intelligence, pp. 359-368, 2003.

Vasilecas, O. and Bugaite, D., Ontology-based information systems development: the problem of automation of information processing rules, Advances in Information Systems. 4th International Conference, ADVIS 2006. Proceedings.Lecture Notes in Computer Science, Vol. 4243, pp. 187-196, 2006.

Wache, H., Semantic mediation for heterogeneous information sources, Ph.D. thesis, University of Bremen, In German, 2003.

Witherell Paul, Krishnamurty Sundar and Grosse Ian R., Ontologies for supporting engineering design optimization, Journal of Computing and Information Science in Engineering, Vol. 7, Iss. 2, pp. 141-150, June 2007.

Zhou Jiehan and Dieng-kuntz Rose, Manufacturing ontology analysis and design: toward excellent manufacturing, Industrial Informatics, 2004 2nd IEEE International Conference on, INDIN '04. pp. 39-45, Jun.2004.

Appendix A: Introduction of Ontology

An **ontology** is a formal explicit description of concepts in a domain of discourse (Noy and McGuinness, 2001). It has a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base (Swartout, et al., 1997). It provides the methods to explicitly describe the conceptualization behind the knowledge represented in a knowledge base (Bernaras et al., 1996) and aims to capture the semantics and relations of the notions used, make them explicit and eventually code them in symbolic systems, so that they can be manipulated and exchanged (Gandon, 2002).

Ontology is composed of classes, slots, instances, axioms, etc. According to Noy and McGuinness (2001), **classes** are sometimes called **concepts**. Properties of each concept describe various features and attributes of the concept, which are also called slots or roles. Restrictions on slots are **facets**, which sometimes are called **role restrictions**. Slots can have different facets describing the value type, allowed values, the number of the values (cardinality), and other features of the values the slot can take. An ontology together with a set of individual **instances** of classes constitutes a **knowledge base**.

Classes are the focus of most ontologies. Classes describe concepts in the domain. A class can have **subclasses** that represent concepts more specific than their superclass. These superclasses and subclasses form the class hierarchy.

One of the most important advantages of ontology is that it is good at the representation and utilization of relations. Relations can be the hierarchical relations between classes, subclasses and superclasses, the slots that connect multiple classes, the facets and hierarchy of slots, or the axioms and rules between classes, slots and instances.

The wine agent based on the wine ontology is an example to show the basic principle of ontology and basic working process of the ontology-based method (http://www.ksl.stanford.edu/people/dlm/webont/wineAgent/). The objective in this example is to assist customers in the selection of the appropriate wine that is suitable based on the sort of meal. As shown in Figure A-1, there are two classes in the wine ontology, food and wine. Each class has some subclasses. For example, seafood, pasta, red meat, white meat, fruit and dessert are the subclasses of food. Each class can have some properties, and all its subclasses inherit the properties. These properties are called slots. For example, the class "Wine" and all its subclasses have four slots, which are color, body, flavor and sugar. Moreover, all classes can have instances. For example, Fra diavolo is a kind of pasta with spicy red sauce, so it is an instance in this class. Another example is that Chateau Lafite Rothschild Pauillac belongs to Pauillac, so it is an instance of the class Pauillac.

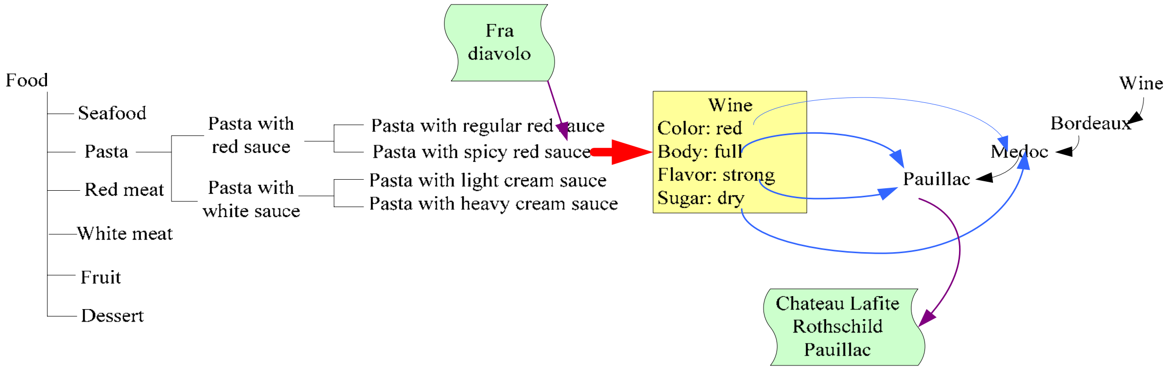


Figure A-1: An example of using ontology in the wine agent

The relations between these data items have been developed in the wine ontology. When a customer orders the food Fra diavolo, the wine agent can find it belongs to the class Pasta with spicy red sauce, and the suitable wine for this class of food should have red color, full body, strong flavor and dry sugar. The class Medoc, which is a subset of Wine, satisfies the color and sugar requirements, and its subsets also satisfy these two requirements. Among these subsets, the class Pauillac satisfies the body and flavor requirements of the wine, so all the instances under the class Pauillac can be selected to match the food Fra diavolo. At the end, the wine agent suggests the customer to use this wine according to the food he/she orders.

Appendix B: Background of FSW and the TIE Project

Friction Stir Welding (FSW) is a relatively new joining process developed initially for aluminum alloys by the Welding Institute (TWI) of UK in 1991 (Thomas et al., 1995). It is a solid-state joining technique that is energy efficient, environmentally friendly, and versatile (Terpenny et al., 2006). It is being touted as the most significant development in metal joining in several decades (Arbegast, 2006). In FSW, a rotating tool with pin and shoulder is inserted in the material to be joined and traversed along the line of the joint. The heating is localized and generated by plastic deformation in the work piece, which results in considerably less energy consumption as compared to conventional welding methods. According to Mishra and Ma (2005), FSW does not use any cover gas or flux, thereby making the process environmentally friendly. Fusion welding involves filler material, while the joining does not involve any use of filler metal and therefore any aluminum alloy can be joined without concern for the compatibility of composition (Mishra and Ma, 2005).

As a replacement for fusion welding and mechanical fastening, FSW (Friction Stir Welding) technologies hold high promise to significantly reduce production cost and minimize environmental emissions over a wide segment of the industry. It is an emerging solid state joining technology that is finding increased interest in the industrial sector. However, much of the knowledge associated with the best situations in which to apply FSW and the specific details of its application is of an ad hoc nature and not explicitly defined. The problem becomes very acute while dealing with new technologies (such as FSP) on where limited information can be obtained. To ensure successful implementation of FSW and development of new technologies based on friction stir concept, the NSF Center for Friction Stir Processing (CFSP) industrial sponsors have requested that e-Design methods be included to assure availability of the latest design approaches to enhance diffusion of friction stir based technologies.

The intellectual merit of the TIE project as well as the research in the dissertation are focused on the development of e-Design tools for concurrent design and manufacturing in the area of FSP to rapidly identify FSP technology insertion opportunities and enhance its implementation in a wide variety of applications including aerospace, aeronautic, and marine applications. The e-Design tools aim at enhancing the Design for Manufacturing (DFM) knowledge management process in the area of FSP. The objective of DFM is to facilitate the sharing of manufacturing knowledge during the product development process and the DFM knowledge management is one of the critical elements of today's enterprises that face intense global competition. The e-Design tools in the TIE project synthesize solutions by bringing in discrete and distributed pieces of knowledge about function, form, shape, size, materials, manufacturing processes, tooling, life cycle costs, testing, quality, etc. in the upfront design process. This involves, for instance, integrating knowledge about function, form, shape, size, materials, manufacturing processes, tooling, costs, testing, quality, etc., early on during the design process.

The following areas need to be addressed and understood to enable future expansion of the technology into more industrial markets:

- Creating foundational framework for e-Design by generating, organizing, and generalizing FSP case knowledge;
- Creating and improving e-Design support tools including cost models and FSP oriented DFM guidelines.

In the TIE project, the activities related to e-Design are listed below.

- Relate the case repository to ontological representations to support search and reasoning capabilities to support the identification and selection of technology insertion opportunities.
- Begin the development of a decision support tool for economic evaluation of FSP insertion opportunities.
- Refine and test the representation scheme and decision support tool with collaborators at CFSP and industrial partners.
- Challenge the fundamental structure and approach of the representation and decision support tool against a wide range of applications.

Appendix C: Cost Model for FSW

According to the Kumar and Mishra (2007), the cost model for the FSW is shown below.

- **Total cost for the FSW:**

$$C_{total} = C_L + C_M + C_P + C_T \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

- **The time to weld W_T:**

$$W_T = \begin{cases} \dfrac{\text{Weld path length (L)}}{\text{Feed rate (F)}} + \dfrac{\text{Depth of plunge (d)}}{\text{Plunge feed rate (F)}} + \text{Dwell time } (\mathrm{T_d}) \\ \qquad\qquad\qquad\qquad \text{(Friction Stir Continuous Welding)} \\ \dfrac{\text{Depth of plunge (d)}}{\text{Feed rate (F)}} + \text{Dwell time } (\mathrm{T_d}) \\ \qquad\qquad\qquad\qquad \text{(Friction Stir Spot Welding)} \end{cases}$$

where,

L= Length of weld (inches)

F = Feed rate (inches/min)

n = Number of weld passes

d = Depth of plunge (inches)

T_d = Dwell time (min)

- **Labor cost,** $C_L = \left(\dfrac{W_T \times n}{OF} + T_{WP} \right) \times \dfrac{C_{LR}}{60}$ ……………….... (2)

Number of labor hours $= \left(\dfrac{W_T \times n}{OF} + T_{WP} \right) \times \dfrac{1}{60}$

where,

L = Length of weld (inches)

F = Feed rate (inches/min)

n = Number of weld passes

OF = Operating factor (0.5, for GMAW)

T_{WP} = Time for weld preparation (min)

C_{LR} = Labor rate (\$/hr)

The assumption here is that all the welds are assumed to be continuous. Labor rate is given by Table C-1.

Table C-1: Average hourly wage of a labor for FSW

Labor Cost	**Avg. hourly wage**
	\$14.91 /hr

SOURCE: http://www.bls.gov/oes/current/oes514121.htm

(U.S. Department of Labor Bureau of Labor Statistics)

- **Machine cost**, $$C_M = \frac{[(W_T \times n) + T_S + T_{Ch}] \times C_{MR}}{MR \times 60} \qquad \text{…………………} (3)$$

where,

T_S = Setup time (min)

T_{Ch} = Tool change-over time (min)

MR = Machine reliability (assuming 95%)

C_{MR}= Machine rate ($/hr)

n = Number of passes

Machine rate (machinery unit cost) is calculated by the years that the machine can be used and the machine cost.

$$\text{machinery unit cost}=\sum_{t=1}^{year} w(t)*P\ (c,t)\ *\ I$$

Where w(t) is the weight given below (specified by experts in the CFSP), P(c,t) is the depreciation parameter given in Table C-2, and I is the total investment on welding machines which can include wire feeder welder, accessories, pre-weld equipments, safety equipment, etc. In MIG welding in Appendix D, it is also used to calculate the machinery unit cost.

3-year class: machinery unit cost=0.0001207*machine cost

w_1=0.28; w_2=0.26; w_3=0.24; w_4=0.22.

5-year class: machinery unit cost=0.0000864*machine cost

w_1 = 0.24; w_2 = 0.21; w_3 = 0.18; w_4 = 0.15; w_5 = 0.12; w_6 = 0.10.

7-year class: machinery unit cost=0.00005925*machine cost

w_1=0.137; w_2=0.133; w_3=0.13; w_4=0.125; w_5=0.123; w_6=0.122; w_7=0.12; w_8=0.11.

10-year class: machinery unit cost=0.0000455*machine cost

w_1=0.115; w_2=0.114; w_3=0.112; w_4=0.11; w_5=0.105; w_6=0.095; w_7=0.09; w_8=0.075; w_9=0.069; w_{10}=0.065; w_{11}=0.05.

Table C-2: MARC's depreciation table

Recovery year	3-year class	5-year class	7-year class	10-year class
1	33.33	20	14.29	10
2	44.45	32	24.49	18
3	14.81	19.2	17.49	14.4
4	7.41	11.52	12.49	11.52
5		11.52	8.93	9.22
6		5.76	8.92	7.37
7			8.93	6.56
8			4.46	6.55
9				6.55
10				6.55
11				3.28

SOURCE: Creese R.C., Adithan M., Pabla B.S., *Estimating and costing for the metal manufacturing Industries*

- **Variable power cost,** $C_P = \dfrac{0.8 \times PR \times C_{PR} \times W_T \times n}{60 \times MR}$ ………………… (4)

where,

PR = Power rating (KVA)

C_{PR} = Power cost ($/kwhr)

W_T = Time to weld (min)

n = number of weld passes

V = Tool traverse speed (inches/min)

MR = Machine reliability (assuming 95%)

The power unit cost is give in Table C-3.

Table C-3: Average electric power cost in different states of USA

State	**Average cost of electric power**	**State**	**Average cost of electric power**
Alaska	\$0.10/kWh	Alabama	\$0.06/kWh
Arkansas	\$0.06/kWh	Arizona	\$0.07/kWh
California	\$0.13/kWh	Colorado	\$0.06/kWh
Connecticut	\$0.10/kWh	Delaware	\$0.07/kWh
Florida	\$0.07/kWh	Georgia	\$0.06/kWh
Hawaii	\$0.13/kWh	Iowa	\$0.06/kWh
Idaho	\$0.06/kWh	Illinois	\$0.07/kWh
Indiana	\$0.05/kWh	Kansas	\$0.06/kWh
Kentucky	\$0.04/kWh	Louisiana	\$0.06/kWh
Massachusetts	\$0.10/kWh	Maryland	\$0.06/kWh
Maine	\$0.11/kWh	Michigan	\$0.07/kWh
Minnesota	\$0.06/kWh	Missouri	\$0.06/kWh
Mississippi	\$0.06/kWh	Montana	\$0.06/kWh
North Carolina	\$0.07/kWh	North Dakota	\$0.05/kWh
Nebraska	\$0.06/kWh	New Hampshire	\$0.10/kWh
New Jersey	\$0.09/kWh	New Mexico	\$0.07/kWh
Nevada	\$0.08/kWh	New York	\$0.11/kWh
Ohio	\$0.07/kWh	Oklahoma	\$0.06/kWh
Oregon	\$0.06/kWh	Pennsylvania	\$0.08/kWh
Rhode Island	\$0.09/kWh	South Carolina	\$0.06/kWh
South Dakota	\$0.06/kWh	Tennessee	\$0.06/kWh
Texas	\$0.07/kWh	Utah	\$0.05/kWh
Virginia	\$0.06/kWh	Vermont	\$0.11/kWh
Washington	\$0.05/kWh	Wisconsin	\$0.06/kWh
West Virginia	\$0.05/kWh	Wyoming	\$0.05/kWh

SOURCE: www.eere.energy.gov/states/state_specific_statistics.cfm/state=MI

Prices are collected from US department of energy efficiency and renewable energy. Average unit prices are based on data collected from 1980 – 2001.

- **Tool cost,** $C_T = \dfrac{C_t \times Q \times W_T \times n}{T} = \dfrac{C_t \times Q \times W_T \times n}{T}$ ………………… (5)

Tool life, T = $Q \times \left[\dfrac{60 \times C_t}{C_{MR}} + t_{ch} \right] \times \left(\dfrac{1-n'}{n'} \right)$

Cutting fraction, Q = $(\theta / 360) \times (L / L_{tot})$

where,

D_p = Diameter of the pin (inches)

L = Length of the weld (inches)

V = Tool travel rate (inches/min)

C_t = Tool cost ($/tool)

C_{MR} = Machine rate ($/hr)

θ = Angle of engagement (degrees)

L_{tot} = Length of weld + Lead + Over travel (inches) = Length of weld + Diameter of the shoulder

T_{Ch} = Time for tool change (min)

n' = Taylor’s tool life exponent (assuming 0.1)

n_t = Tool changes/unit = $\frac{Q \times W_T}{T}$ (interrupted welding)

= $\frac{Q \times W_T}{T}$ (Uninterrupted welding)

The thicknesses of the materials are used to decide the tool to be used. The criterion for selecting a tool is given in Table C-4. The present available tool is shown in Table C-5.

Table C-4: Criteria for FSW tool selection

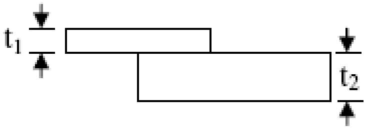
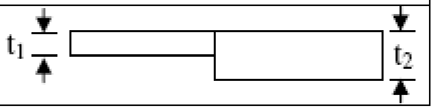
Type of joint	Pin height		Illustration of the joint
	Minimum	**Maximum**	
Lap joint	$(0.91)t_1+(0.185)t_2$	$(0.91)t_1+(0.455)t_2$	
Butt joint	$0.85t_2$	$0.90t_2$	

Table C-5: Currently available tools at UMR

Tool ID	Material	Tool Features	Tool geometry			
			Pin Diameter (mm)	**Pin Length /Height (mm)**	**Shoulder Diameter (mm)**	**Shoulder feature**
A227	densimet	Conical Threaded	2.97	2.98	14.90	concave
D120	densimet	Conical triflute pin	3.88	2.26	12.42	concave
A221	densimet	Conical Threaded	3	1.80	12.00	concave
A216	densimet	Conical Threaded	4.5(root)	0.80	12.00	concave
A217	densimet	Conical Threaded	4.5(root)	1.00	12.00	concave
A218	densimet	Conical Threaded	4.5(root)	1.20	12.00	concave
A219	densimet	Conical Threaded	4.5(root)	1.40	12.00	concave
A220	densimet	Conical Threaded	4.5(root)	1.60	12.00	concave
A142	densimet	Conical Threaded	4.5(root)	1.50	12.00	concave
A143	densimet	Conical Threaded	4.5(root)	2.00	12.00	concave
A193-s	densimet	Conical pin	4.5(root)	1.77	10.00	concave

If none of the tool has the pin diameter in the suggested range calculated by formulas in Table C-4, the one with the pin height closest to the suggested range will be selected in common.

Appendix D: Cost Model for MIG (Metal Inert Gas) Welding

In this appendix, the cost model is first given followed by the values of related parameters.

According to the Kumar and Allada (2007), the cost model for the MIG (Metal Inert Gas) welding is shown below.

- **Total cost**, $C_{MIG} = C_{ec} + C_{pc} + C_{gc} + C_{lc} + C_{mc} + C_{tc}$

where C_{ec} is the filler material (electrode) cost, C_{pc} is the power cost, C_{gc} is the shielding gas cost, C_{lc} is the labor cost, C_{mc} is the machine cost and C_{tc} is the tooling cost.

- **Filler material (electrode) cost,**

$$C_{ec} = \begin{cases} \dfrac{(0.012192862 \times L)}{95\%} \times C_f & \text{Fillet weld} \\ \dfrac{(0.04877 \times z^2 \times L)}{95\%} \times C_f & \text{Lap weld} \\ \dfrac{(0.056315 \times z^2 + 0.00768 \times z^2) \times L}{95\%} \times C_f & \text{Full penetration butt weld} \\ \dfrac{(0.056315 \times z^2 - 0.0011885 \times z - 0.00025557) \times L}{95\%} \times C_f & \text{Partial penetration butt weld} \end{cases}$$

- **Power cost,**

$$C_{pc} = \begin{cases} 3.509 \times 10^{-5} \times n \times V \times C_p \times L & \text{0.03" dia. electrode} \\ 5.7018 \times 10^{-5} \times n \times V \times C_p \times L & \text{0.035" dia. electrode} \\ 7.675 \times 10^{-5} \times n \times V \times C_p \times L & \text{3/64" dia. electrode} \\ 1.0088 \times 10^{-4} \times n \times V \times C_p \times L & \text{1/16" dia. electrode} \end{cases}$$

Power unit cost can be obtained from Table C-3.

- **Shielding gas cost**,

$$C_{gc} = 4.167 \times 10^{-3} \times n \times C_g \times L$$

- **Labor cost**,

$$C_{lc} = \left[4.166 \times 10^{-4} \times n \times L + \left(\frac{T_w}{3600} \right) \right] \times \frac{C_L}{OF}$$

- **Machine cost**,

$$C_{mc} = 8.772 \times 10^{-4} \times n \times C_M \times L$$

The parameters used in the model are shown below.

C_f:	Unit cost of filler material (\$/lb);	C_g:	Unit cost of gas (\$/ft^3);
z:	Thickness of base plate (inches);	T_w:	Time of weld preparation (sec);
L:	Length of weld (inches);	C_L:	Labor rate (\$/hr);
V:	Voltage used (volts);	OF:	Operating factor;
C_p:	Unit cost of power (\$/Kwhr);	C_M:	Unit cost of machine (\$/hr);
n:	Number of weld passes.		

According to Creese et al. (1992), normal weld torch travel speed for MIG welding process is assumed to be 40 inches/min.

Considering the information provided by the Harris Products Group (www.jwharris.com), the required filler material diameter is chosen corresponding to the thickness of the welded components, as shown in Table D-1.

Table D-1: Relation between metal thickness and filler or wire diameter

Metal Thickness t	Filler material / wire diameter
t<=1/16"	0.03"
1/16"<t<=1/8"	0.03", 0.035" or 3/64"
1/8"<t<=3/16"	0.03", 0.035" or 3/64"
3/16" <t<=3/8"	3/64" or 1/16"
t>3/8"	1/16"

Typical operating current can be identified from Table D-2 generated based on the information provided by Airgas Inc.(www.airgas.com).

Table D-2: Relation between metal thickness and current

Filler or wire diameter	0.03"	0.035"	3/64"	1/16"
Current (Amps)	60-100	100-160	150-200	180-280

The handbook for estimating and costing for metal manufacturing industries (Creese et al., 1992) suggests a machine efficiency of 95% and an operating factor of 0.5. According to the current selected, the welding machine can be chosen. If there is no machine that meets the current requirement, the machines whose Amps(current) are higher than the required current can be used.

Table D-3 : Few MIG welding machines costs and capabilities

Item			Price	Rated output			Solid wire size range	Wire feed speed range
				Current (Amps)	Voltage	Duty cycle		
Group 1	K2470-1	Power MIG 140T MIG/Flux-Cored Wire Feeder Welder	$509.95	90	19.5	20%	0.035"	50-500
	K2471-1	Power MIG 140C MIG/Flux-C	$577.00	90	19.5	20%	0.035"	50-500

		ored Wire Feeder Welder						
Group 2	K2472-1	Power MIG 180T MIG/Flux-Cored Wire Feeder Welder	$620.95	130	20	30%	0.035"-0.045"	50-500
	K2473-1	Power MIG 180C MIG/Flux-Cored Wire Feeder Welder	$699.00	130	20	30%	0.035"-0.045"	50-500
	K2326-1	Power MIG - 215	$1279.95	215 190 170	22 23 24	30% 40% 60%	0.035"-0.045"	50-700
Group 3	K2416-1	Power MIG -255C 208/230/1/60	$1785.95	250	26	40%	0.035"-0.045"	50-700
	K2416-2	Power MIG -255C 230/460/575/1/60	$1818.95	250	26	40%	0.035"-0.045"	50-700
	K2417-1	Power MIG -255C One-Pak with Spool Gun	$2524.95	250	26	40%	0.035"-0.045"	50-700
Group 4	K2403-1	Power MIG-350MP	$3295.95	250	26	40%	0.035"-0.045"	50-700
	K2451-1	Power MIG -350MP Push-Pull with Python Plus	$4869.95	300	26	40%	0.035"-0.045"	50-700

SOURCE: www.weldingsupply.com **DATE:** 11/30/2006

Table D-4: Commonly used filler material types and associated price

Diameter of filler material	Type of filler material	Price
0.03"	Aluminum – 4043	$5.29/lb
0.035"	Aluminum – 4043	$4.58/lb
3/64"	Aluminum – 4043	$4.34/lb
1/16"	Aluminum – 4043	$3.52/lb
0.03"	Aluminum – 5356	$5.44/lb
0.035"	Aluminum – 5356	$5.00/lb
3/64"	Aluminum – 5356	$4.59/lb
1/16"	Aluminum – 5356	$3.72/lb

SOURCE: www.weldingsupply.com **DATE:** 11/30/2006

Table D-5: Commonly used shielding gas proportion, quantities and associated price

Item		Price
20-AR75CD	75% Argon 25% Carbon Dioxide (20 CF)	$ 18
40-AR75CD	75% Argon 25% Carbon Dioxide (40 CF)	$ 20
Q-AR75CD	75% Argon 25% Carbon Dioxide (80 CF)	$ 25
S-AR75CD	75% Argon 25% Carbon Dioxide (130 CF)	$ 27
T-AR75CD	75% Argon 25% Carbon Dioxide (375 CF)	$ 30
PB-AR75CD	75% Argon 25% Carbon Dioxide (4,032 CF)	$ 225
T-AR80CD	80% Argon 20% Carbon Dioxide (375 CF)	$ 75

SOURCE: www.weldingsupply.com **DATE:** 11/30/2006

Table D-6: Average hourly wage of a labor for MIG welding

Labor Cost	Avg. hourly wage
	$15.52 /hr

SOURCE: http://www.bls.gov/oes/current/oes514121.htm

(U.S. Department of Labor Bureau of Labor Statistics)

Time for weld preparation is assumed to be 30 seconds.

Table D-7: Price list of generally used tools and weld preparation equipment

Tools	**Price**
3/32" collets, collet bodies, #7 alumina nozzles, 3 of each for type, 2 series gmaw torch	$ 18.21
Tool box	$ 10.00
Tip cleaner	$ 3.35
Slag hammer	$ 8.00
Victor #00 weld tip for 100FC torch body	$ 25.00
12' Tape measure	$ 4.00
Wire brush (for arc)	$ 2.86
Wire brush (2 stainless Steel for GMAW-AL)	$ 3.00
90 Degree magnet (small) (need 2)	$ 12.50
6" Locking pliers	$ 1.88
Wire cutters	$ 1.88
10" Adjustable wrench	$ 5.00
Metal marking pencil (silver)	$ 1.18
Welding gas hose reel	$ 460.99
Striker	$ 3.10

Table D-8: Price list of generally used safety equipment

Safety Equipment	**Price**
Leather Jacket	$ 50.00
Leather Chaps	$ 25.00
Leather Gloves (Heavy, for arc, 2 pairs)	$ 16.00
Leather Gloves (Light, for gas, gtaw)	$ 12.00
Leather Boots	$ 45.00
Welding Cap (2)	$ 10.00
Welding hood, HornellSpeedglass 9000V	$ 270.00
Pkg of 10 replacement lens for 9000V	$ 25.00
Gas Welding Goggles	$ 10.00
Safety glasses	$ 3.46

SOURCE:

http://www.olympic.edu/Students/AcadDivDept/BusinessAndTechnology/WeldingTechnology/

For average electric power cost in different states of USA and MARC’s depreciation table, please refer Tables C-2 and C-3.

Appendix E: FSW e-Design Process

User Inputs

1) Selection of welding technique

Select from the available list of welding techniques, such as MIG or FSW.

2) Selection of component geometry and material

Component geometry options are shown in Figure E-1.

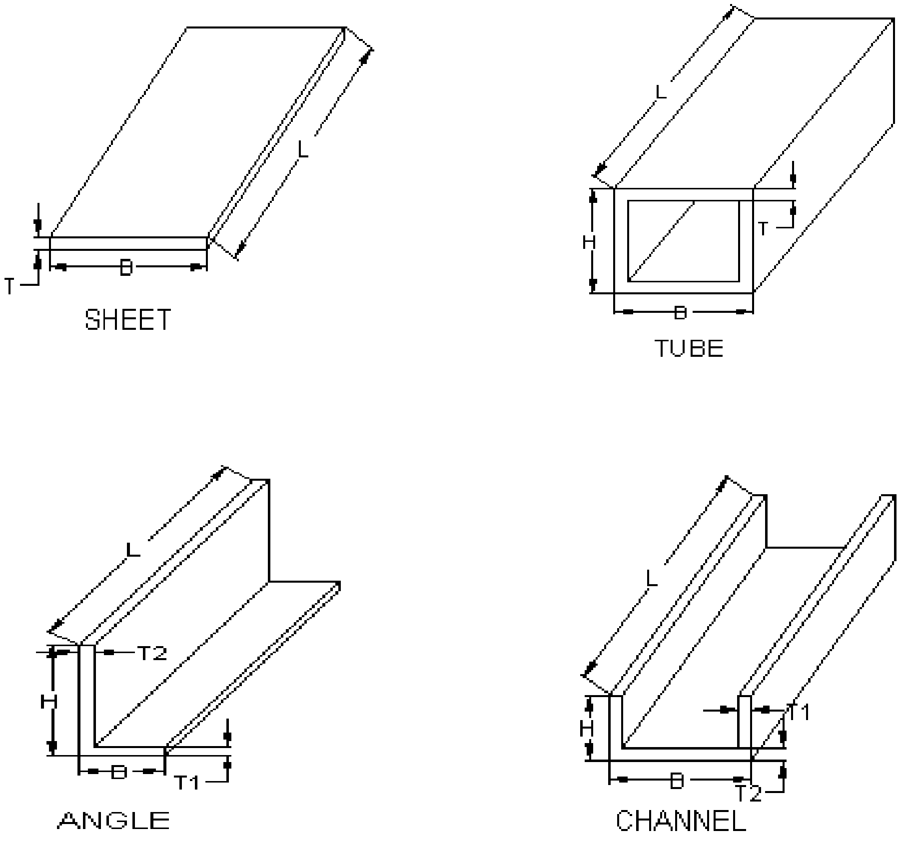


Figure E-1: Component geometry options in FSW

Two components are welded here.

Component-1 Geometry options: Sheet / Tube / Angle / Channel

	Material options:	List of all materials
Component-2	Geometry options	Sheet / Tube / Angle / Channel
	Material options:	List of all materials

** Density and UTS of each associated material can be specified.*

** Pictorial representation of each geometry type is available during selection.*

** User has to input dimensions for the selected geometry.*

3) Selection of joint category

Joint categories: Lap / Butt, as shown in Figure E-2.

Figure E-2: Joint category in FSW

** Pictorial representation for all joint categories is available while selection.*

4) Selection of mating / aligning faces and the base component

Different geometry selections in Step 2 bring different options that can be chosen here, as listed below.

1) Sheet : L X B

2) Tube : L X H

3) Angle : L X H and L X B

4) Channel : L X H and L X B

5) Specifying the joint strength

On confirming the mating / aligning faces, a window for inputting the required joint strength will be displayed. The user has to input the value for the required joint strength. A message should be displayed to limit the value of joint strength entered by the user.

6) Specifying the weld category

The options available of the required weld category is continuous weld, stitch weld and spot weld.

<u>Outputs</u>

After obtaining all inputs from the user, the output window will be displayed. This window displays the following information:

- Weld geometry parameters: Number of spots
  - Number of stitches and stitch length
  - Total weld length
- Tool details: Tool Material
  - Tool pin type
  - Shoulder profile
  - Shoulder features
  - Shoulder diameter
  - Pin diameter / s
  - Pin height
- Process parameters: Tool tilt angle

Plunge depth

Plunge speed

Tool rpm

Tool travel rate

Plunge force

Dwell time

- Machine parameters: Machine degrees of freedom

  Machine table size

- Fixturing details: Fixturing elements
- Time required for welding (exclusive of setup time)
- Ratio of joint strength to joint weight

Assumptions:

1. The e-Design tool deals only with joining of similar materials.
2. The e-Design tool deals only with lap and butt joints.
3. The values of density and UTS for all aluminum varieties are average figures calculated from the ASM Handbook database.
4. In the cases of spot welds and stitch welds, the spots / stitches are uniformly spaced and equidistant from the component geometry.
5. The e-Design tool deals only with simple joints which required linear welds. These joints can be welded only by positioning the FSW tool vertically downwards. Hence FSW machine with DOF 3 is selected for all inputs.

6. 500 mm are added while calculating the machine table size to accommodate fixture elements.

Note:

Before calculating the outputs, the joint is categorized as high strength joints (HSJ), medium strength joints (MSJ) and low strength joints (LSJ) based on fraction of UTS of the parent material.

HSJ: Joint strength >= 75% of parent material UTS

MSJ: 75 % of parent material UTS > Joint strength >= 50% of parent material UTS

LSJ: Joint strength < 50% of parent material UTS)

Other output calculation methods are given with the example below.

Example:

<u>Inputs</u>

1) Selection of Welding Technique

User input – Friction Stir Welding

2) Selection of component geometry

User input	Component-1	Angle		
		Material-1	6061-T6	
		Component dims	L	300 mm
			B	50 mm
			H	50 mm
			T1	5 mm

		T2	5 mm
Component-2	Sheet		
	Material-2	6061-T6	
	Component dims	L	300 mm
		B	50 mm
		T	5 mm

For 6061 T-6 UTS = 310 MPa

Density = 2.7 gm/cc

Note: For selecting the component dimensions, the user should refer to the catalogue available in the e-Design tool.

3) Selection of joint category

User input Joint Category - Butt Joint

4) Selection of mating / aligning faces and the base component

User Input For component-1 face B x L

For component-2 face B x L

5) Specifying the joint strength

User Input 235 MPa (tensile strength)

6) Specifying the weld category

User Input Continuous weld

Output

235 MPa (user input) / 310 MPa (parent material UTS) = 75.8 % => HIGH STRENGTH JOINT

A) Length of weld 270 mm

GL: length of weld is approximately 90% of the total length available for welding

B) Tool Material Tool Steel

GL: recommended for 6061-T6 aluminum

Butt joint

Continuous weld

C) Shoulder Diameter 15-20 mm

GL: recommended for Butt joint

Continuous weld

High strength joint

D) Shoulder Profile / Face- Concave

GL: recommended for 6061-T6 aluminum

Butt joint

Continuous weld

E) Shoulder Features Featureless

GL: recommended for Butt joint

Continuous weld

Concave shoulder

F) Tool Pin Type Cylindrical

GL: recommended for 6061-T6 aluminum

Butt joint

Continuous weld

G) Pin Diameter 5 mm-7 mm

GL: ratio of shoulder-pin diameter ≈ 3 for Butt joint

Continuous weld

H) Plunge Depth 4.75 mm

GL: plunge depth = 0.95 x component thickness

recommended for High strength joints

Butt joint

Same thickness components

I) Pin height 4.7 mm

GL: pin height ≈ 0.98 x plunge depth

recommended for Butt joint

Cylindrical pin

J) Tool Tilt Angle 2-3 degrees

GL: recommended for Butt joint

Continuous weld

K) Plunge Speed 0.5-2 mm/sec

GL: recommended for Butt joint

Continuous weld

L) Tool RPM 800-1200

GL: recommended for Butt joint

Continuous weld

High strength joint

M) Tool IPM 5-10

GL: recommended for 6061-T6 aluminum

High strength joint

N) Plunge Force 5 kN-10 kN

GL: recommended for 6061-T6 aluminum

O) Dwell Time 0 sec

GL: recommended for Continuous weld

P) Machine degrees of freedom 3

GL: This value of DOF will remain same irrespective of the inputs entered by the user

Q) Machine Table Size 1st horz. Dim. = 300 + 300 + 500 = 1100 mm

2nd horz. Dim. = 50 + 50 + 500 = 600 mm

GL: 1st horz dim = 1st horz dim of the total of the 2 mating/aligning faces + 500 mm

2nd horz dim = 2nd horz dim of the total of the 2 mating/aligning faces + 500 mm

R) Component Weights

GL: Component-1 Angle

Weight = (BT1 + $HT_2 – T_1T_2$) x L x Material density = 384.75 grams

Component-2 Sheet

Weight = (BT) x L x Material density = 202.5 grams

S) Time required for welding 1.41 minutes

GL: Continuous weld

Weld time = (total weld length / tool travel rate) + dwell time

Tool travel rate is taken as 7.5 IPM for calculation (mean of 5 and 10)

T) Strength-weight ratio 0.4

GL: Ratio = Joint strength / Sum of the weights of the 2 components

(Joint strength in MPa, Components weights in grams)