

Holger Kampffmeyer

Formalization of Design Patterns by Means of Ontologies

Diploma Thesis

YOUR KNOWLEDGE HAS VALUE



- We will publish your bachelor's and master's thesis, essays and papers
- Your own eBook and book - sold worldwide in all relevant shops
- Earn money with each sale

Upload your text at www.GRIN.com
and publish for free



Diploma Thesis

Formalization of Design Patterns by Means
of Ontologies

submitted by

Holger Kampffmeyer

Technische Universität Dresden

Fakultät Informatik
Institut für Software- und Multimediatechnik
Lehrstuhl Softwaretechnologie

Submitted February 26, 2007

Abstract

Design patterns have proven to be important building blocks and means of reuse in software design. However, the mere number of available design patterns complicates the decision-making which design pattern to choose and demands tools assisting in this process. We hence propose a knowledge-based formal representation of design patterns, a representation that is accessible by tools. Existing approaches to formalizing design patterns generally cover solely the formal description of the *structure* of design patterns. However, an important part of a design pattern description is the *intent* section, because the intent describes what the design pattern does and which design problems a pattern addresses. In this work, we develop a novel approach of formalizing design patterns by their intent. The formal representation is based on OWL, the web ontology language. The developed ontology can serve as support for the decision-making of choosing the right design pattern. We furthermore develop a tool that uses the ontology as a knowledge-base. The tool allows the user to visually describe design problems and gives suggestions of design patterns that solve a given design problem.

Acknowledgements

First of all I would like to thank my supervisor Steffen Zschaler for his valuable advice, his continues encouragement and the many hints that have improved this text. I would also like to thank Sebastian for reading this work and providing valuable suggestions. Finally, I would like to thank my friends, my whole family and–most important of all–my girlfriend Tatjana for their support. I would not have been able to complete this work without you.

Contents

1	Introduction	1
2	Related Work	3
2.1	Classification of Design Patterns	3
2.2	Ontological and Other Formal Approaches	7
3	Design Patterns	9
3.1	The GOF Pattern Catalogue	9
3.1.1	Abstract Factory	10
3.1.2	Builder	10
3.1.3	Factory Method	10
3.1.4	Prototype	11
3.1.5	Singleton	11
3.1.6	Adapter	11
3.1.7	Bridge	11
3.1.8	Composite	12
3.1.9	Decorator	12
3.1.10	Facade	12
3.1.11	Flyweight	13
3.1.12	Proxy	13
3.1.13	Chain of Responsibility	13
3.1.14	Command	14
3.1.15	Interpreter	14
3.1.16	Iterator	14
3.1.17	Mediator	15
3.1.18	Memento	15
3.1.19	Observer	15
3.1.20	State	16
3.1.21	Strategy	16
3.1.22	Template Method	16
3.1.23	Visitor	17

4	Ontologies	19
4.1	The Web Ontology Language OWL	21
4.1.1	Classes	21
4.1.2	Properties	22
4.1.3	Individuals	23
4.1.4	The Manchester OWL Syntax	24
4.2	Other Ontology Languages	25
4.2.1	RDF and XML	25
4.2.2	RDFS	26
4.2.3	DAML+OIL	26
4.3	Conclusions	26
5	Ontology Engineering Tools	29
5.1	Protégé	30
5.2	Querying and Inference Approaches	31
5.2.1	SPARQL and the Jena Framework	31
5.2.2	The Pellet DIG Reasoner	33
5.2.3	RacerPro	33
5.2.4	The nRQL Query Language	34
5.3	Conclusions	36
6	Developing the DPIO	39
6.1	Knowledge-Engineering Methodology	39
6.2	Domain and Scope of the Ontology	40
6.3	Competency Questions	41
6.3.1	Meta Questions	41
6.3.2	Questions in the Domain of Design Patterns	41
6.4	Initial Vocabulary	43
6.4.1	Intent Classification	43
6.5	Modeling the Ontology	43
6.5.1	Classes	44
6.5.2	Properties	46
6.5.3	Instances	46
6.5.4	Examples of Modeled Design Problems	46
6.6	Completeness of the Ontology	51
6.7	Conclusions	52
7	Evaluation	61
7.1	Formalization of Competency Questions	61
7.1.1	Meta Questions	61
7.1.2	Questions in the Domain of Design Patterns	63
7.2	Test of the Ontology	66
7.3	Conclusions	67

<i>CONTENTS</i>	vii
8 The Design Pattern Wizard	69
8.1 Requirements	69
8.2 Architecture	70
8.3 The Wizard User Interface	76
8.4 Conclusions	79
9 Conclusions and Future Work	81
A The DPIO Project-CD	i
A.1 The Eclipseproject	i
A.2 Configuration of the Design Pattern Wizard	i
A.3 Starting up the Design Pattern Wizard	ii
A.4 The Protégé Project	iii
A.5 The Report	iii
Bibliography	ix

List of Figures

2.1	Zimmer's arrangement of design patterns in layers (from [57]).	5
4.1	The semiotic triangle (inspired by [52, 1])	20
5.1	Screenshot of the Protégé ontology editor	30
6.1	The parent classes of the three hierarchies <i>Design Patterns</i> , <i>Design Problems</i> and <i>Problem Concepts</i>	44
8.1	Use case diagram of the design pattern wizard	70
8.2	UML component diagram of the design pattern wizard	71
8.3	UML class diagram of the model of the design pattern wizard	74
8.4	UML class diagram of the tree model of the design pattern wizard	75
8.5	UML communication diagram of the design pattern wizard .	76
8.6	Screenshot of the Design Pattern Wizard	77
8.7	Screenshot of the dialog constraining a problem predicate . .	77
8.8	Screenshot of the dialog constraining a problem concept . . .	78
8.9	Screenshot of the result window suggesting suitable design patterns	78