

# Requirements-Oriented Methodology for Evaluating Ontologies

A thesis submitted for the degree of  
Doctor of Philosophy

Jonathan Yu  
B.App.Sc.(CS)(Hons.), Dip.Bus.(Mgt.)

School of Computer Science and Information Technology,  
Science, Engineering, and Technology Portfolio,  
RMIT University,  
Melbourne, Victoria, Australia.

July, 2008

## **Declaration**

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Jonathan Yu

School of Computer Science and Information Technology

RMIT University

April 17, 2009

## Acknowledgments

First of all, I would like to thank my primary supervisor James Thom for introducing me to the world of academia. It is his rigour and discipline that has guided me through the last few years in preparing this thesis. I also thank my secondary supervisor Audrey Tam for her support along the way with my research, and her colourful whiteboard markers.

I thank my family – my father, George Yu, my mother, Ooi Siew Imm, my sister, Jessica Yu, and my grandmother, Tan Thye Neo – for their encouragement to further studies and learning, and for their support in every sense of the word. Also, thanks to my fiancé Debbie Low, for her support throughout my PhD candidature – for sticking by me and helping me every step of the way. I thank Roy Pang for his expertise with the diagrams in this thesis.

I also thank the many friends at RMIT University whom I have had collaboration with, who have supported me, contributed to the completion of this thesis and made my PhD that much more enjoyable: Halil Ali, Nikolas Askitis, Jelita Asian, Yaniv Bernstein, Bodo von Billerbeck, Steven Burrows, Michael Cameron, Abhijit Chataraj, Pauline Chou, Jun Jie Foo, Steve Garcia, Nick Lester, Jovan Pehcevski, Simon Puglisi, Falk Scholer, Iman Suyoto, Yohannes Tsegay, Mingfang Wu, and Ying Zhao.

I thank Ron Gallagher, Bruce Melendy and colleagues from Lonely Planet for their time and input in this thesis.

## Credits

Portions of the material in this thesis have previously appeared in the following publications:

- J. Yu, J. A. Thom, and A. Tam. Evaluating ontology criteria for requirements in a geographic travel domain. In *Proceedings of 4th International Conference on Ontologies, Databases and Applications of Semantics (ODBASE)*, volume 3761 of *Lecture Notes in Computer Science*, pages 1517–1534, Ayia Napa, Cyprus, 2005. Springer-Verlag. (Chapters 4 and 6)
- J. Yu, J. Thom, and A. Tam. Ontology evaluation: Using Wikipedia categories for browsing. In *Proceedings of 16th Conference on Information and Knowledge Management (CIKM)*, pages 223–232. ACM Press, 2007. (Chapters 5 and 6)

This research was conducted with the support of an Australian Postgraduate Award with stipend. Thanks to the School of Computer Science and Information Technology for their conference travel support. The thesis was written using the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> document preparation system. All trademarks are the property of their respective owners.

## Note

Unless otherwise stated, all fractional results have been rounded to the displayed number of decimal figures.

# Contents

<b>Abstract</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Ontology evaluation . . . . .	4
1.2 Challenges in ontology evaluation . . . . .	5
1.3 Thesis overview . . . . .	8
<b>2 Ontologies and their evaluation</b>	<b>10</b>
2.1 Ontologies . . . . .	11
2.1.1 Simple and structured ontologies . . . . .	12
2.1.2 Ontology specification languages and OWL . . . . .	15
2.1.3 Ontology granularity . . . . .	17
2.1.4 Ontologies used in applications . . . . .	21
Data integration and interoperability . . . . .	22
Navigation systems and web applications . . . . .	24
Information and multimedia retrieval systems . . . . .	24
Knowledge management, organisational memory and group memory .	25
Software specification and development . . . . .	25
Teaching systems and eLearning . . . . .	26
2.2 Ontology engineering methodologies . . . . .	26
2.2.1 Formal method for ontology engineering . . . . .	27
2.2.2 Methontology . . . . .	29
2.2.3 On-To-Knowledge . . . . .	30

2.2.4	SENSUS-based ontology methodology . . . . .	32
2.2.5	Discussion . . . . .	32
2.3	Ontology evaluation methodologies . . . . .	33
2.3.1	OntoClean . . . . .	34
	Meta-properties . . . . .	34
2.3.2	OntoMetric . . . . .	36
2.3.3	Software evaluation methodologies . . . . .	41
	Factors-Criteria-Metric framework (FCM) . . . . .	41
	The Goal Question Metric methodology (GQM) . . . . .	42
2.4	Measures . . . . .	44
2.4.1	Ontology evaluation criteria . . . . .	45
2.4.2	Ontology evaluation measures . . . . .	48
	Detailed descriptions of selected ontology measures . . . . .	50
	Discussion . . . . .	60
2.4.3	Validating measures . . . . .	62
2.5	Discussion . . . . .	64
2.6	Summary . . . . .	65
<b>3</b>	<b>The ROMEO methodology</b>	<b>67</b>
3.1	The ROMEO methodology . . . . .	68
3.2	Ontology requirements . . . . .	68
3.2.1	Establishing the roles of the ontology . . . . .	69
3.2.2	Obtaining a set of ontology requirements. . . . .	72
3.3	Questions . . . . .	74
3.3.1	Criteria-questions . . . . .	76
3.4	Measures . . . . .	79
3.4.1	Suggested mappings between criteria-questions and existing measures	80
3.5	Discussion . . . . .	85
3.6	Summary . . . . .	89

<b>4</b>	<b>Lonely Planet</b>	<b>91</b>
4.1	Content management for Lonely Planet . . . . .	92
4.1.1	Travel guidebooks and their issues . . . . .	93
	Achieving consistent vocabulary . . . . .	94
	Achieving consistent book structure . . . . .	94
	Achieving consistent content across guidebooks . . . . .	94
4.1.2	Digital content and its issues . . . . .	95
4.1.3	Previous experience of Lonely Planet in reusing ontologies . . . . .	96
	Appropriate representation of geographic places . . . . .	97
	Right level of content granularity . . . . .	100
4.1.4	Roles of the suitable ontology . . . . .	100
4.2	ROMEO ontology requirements for Lonely Planet . . . . .	102
4.2.1	Identifying ontology requirements . . . . .	102
4.2.2	Ontology requirement 1: Controlled vocabulary of names, places and terms . . . . .	104
4.2.3	Ontology requirement 2: Flexible classification of geographic items . . . . .	105
4.2.4	Ontology requirement 3: Appropriate granularity . . . . .	105
4.3	ROMEO questions for Lonely Planet . . . . .	106
4.3.1	Questions for ‘Controlled vocabulary of names, places and terms’ . . . . .	106
4.3.2	Questions for ‘Flexible classification of geographic items’ . . . . .	107
4.3.3	Questions for ‘Appropriate granularity’ . . . . .	108
4.4	ROMEO measures for Lonely Planet . . . . .	108
4.4.1	Measures for ‘How many identical concepts are modelled using different names?’ . . . . .	109
4.4.2	Measures for ‘How many identical instances are modelled using different names?’ . . . . .	110
4.4.3	Measures for ‘Do the relationships between concepts in the ontology adequately cover the relationships between concepts in the domain?’ . . . . .	110

4.4.4	Measures for ‘Does the ontology have an appropriate level of granularity with regard to its concepts compared with the domain being modelled?’ . . . . .	112
4.4.5	Measures for ‘Does the ontology have an appropriate level of granularity with regard to its instances compared with the domain being modelled?’ . . . . .	113
4.5	Summary . . . . .	114
<b>5</b>	<b>Wikipedia</b>	<b>117</b>
5.1	Wikipedia and its categories . . . . .	118
5.1.1	Wikipedia content, policies and guidelines . . . . .	118
	Wikipedia policies and guidelines . . . . .	119
5.1.2	Navigating and exploring articles . . . . .	121
5.1.3	Wikipedia categories . . . . .	122
	Design of category structure . . . . .	122
	Wikipedia category structure as an ontology . . . . .	123
5.2	ROMEIO ontology requirements for Wikipedia . . . . .	124
5.2.1	Identifying ontology requirements . . . . .	125
5.2.2	Ontology requirement 1: Adequate level of category intersection . . .	129
5.2.3	Ontology requirement 2: Categories should be appropriately grouped .	129
5.2.4	Ontology requirement 3: Avoiding cycles in the category structure . .	130
5.2.5	Ontology requirement 4: Ensure the set of categories is complete . . .	131
5.2.6	Ontology requirement 5: Ensure categories associated in articles are correct . . . . .	131
5.3	Questions . . . . .	131
5.3.1	Questions for ‘Adequate level of category intersection’ . . . . .	132
5.3.2	Questions for ‘Categories should be appropriately grouped’ . . . . .	132
5.3.3	Questions for ‘Avoiding cycles in the category structure’ . . . . .	133
5.3.4	Questions for ‘Ensure a complete set of categories’ . . . . .	133
5.3.5	Questions for ‘Ensure categories associated in articles are correct’ . . .	135



5.4	Measures . . . . .	136
5.4.1	Measures for ‘Does the category structure have an adequate intersection of categories?’ . . . . .	136
5.4.2	Measures for ‘Does the ontology capture concepts of the domain correctly?’ . . . . .	138
5.4.3	Measures for ‘How many cycles are found in the ontology?’ . . . . .	139
5.4.4	Measures for ‘Does the ontology have concepts missing with regard to the relevant frames of reference?’ . . . . .	139
5.4.5	Measures for ‘Is the set of categories correctly associated with a given article?’ . . . . .	143
5.5	Summary . . . . .	144
<b>6</b>	<b>Empirical validation</b>	<b>146</b>
6.1	The validation process . . . . .	147
6.1.1	The validation environment . . . . .	147
6.1.2	Obtaining comparable ontologies . . . . .	149
6.1.3	Select appropriate tasks and benchmarking standards . . . . .	150
6.2	Validating granularity mapping . . . . .	150
6.2.1	Experimental setup . . . . .	151
	Ontologies used . . . . .	152
	Tasks . . . . .	153
6.2.2	Results . . . . .	153
	Outcomes of validation experiment . . . . .	155
6.3	Validating intersectedness mapping . . . . .	157
6.3.1	Experimental setup . . . . .	158
	Ontologies used . . . . .	159
	Tasks and domains . . . . .	162
6.3.2	Analysis of varied ontologies . . . . .	164
6.3.3	Benchmarking . . . . .	167
	Significance testing . . . . .	168

6.3.4	Results	168
	Best method for obtaining untangled ontology	169
	Comparing Subtree <i>a</i> (base) and Subtree <i>b</i> (untangled)	171
6.3.5	Outcome of validation experiment	176
6.4	Summary	177
<b>7</b>	<b>Conclusions and future work</b>	<b>179</b>
7.1	The ROMEO methodology for ontology evaluation	180
7.2	Empirical validation of ontology evaluation methods	184
7.3	Summary	187
	<b>Glossary</b>	<b>191</b>
<b>A</b>	<b>ROMEO templates and suggested mappings</b>	<b>192</b>
A.1	ROMEO template	192
A.2	Suggested mappings between questions and measures	194
<b>B</b>	<b>ROMEO analysis: Lonely Planet</b>	<b>198</b>
B.1	Role of the ontology	198
B.2	Requirements	199
B.3	Questions	201
B.4	Measures	203
<b>C</b>	<b>ROMEO analysis: Wikipedia</b>	<b>206</b>
C.1	Role of the ontology	206
C.2	Requirements	207
C.3	Questions	209
C.4	Measures	212
<b>D</b>	<b>Requirements gathering: Wikipedia</b>	<b>216</b>
D.1	Excerpts from Meta:Categorization requirements	216
D.1.1	Goals (or, “Why implement categories?”)	216

D.1.2	Theoretical basis . . . . .	218
D.2	Excerpts from <i>Wikipedia: Categorization</i> . . . . .	220
D.2.1	When to use categories . . . . .	220
D.2.2	Guidelines . . . . .	221
Some general guidelines	. . . . .	221
Categories vs. lists vs. info boxes	. . . . .	222
Categories applied to articles on people	. . . . .	223
Categories do not form a tree	. . . . .	223
Cycles should usually be avoided	. . . . .	223
D.2.3	Grouping categories . . . . .	224
<b>E</b>	<b>Wikipedia Browsing Experiment</b>	<b>225</b>
E.1	User handouts . . . . .	225
	<b>Bibliography</b>	<b>236</b>

# List of Figures

2.1	A simple ontology . . . . .	13
2.2	A structured ontology . . . . .	14
2.3	Model of Paris and France in OWL: a) Schema; and b) Instances . . . . .	16
2.4	Levels of granularity of ontologies . . . . .	18
2.5	Example of the granularity of ontologies . . . . .	20
2.6	Methodology presented by Grüninger and Fox [1995] . . . . .	28
2.7	The Multilevel Tree of Characteristics (MTC) in OntoMetric [Lozano-Tello and Gómez-Pérez, 2004] . . . . .	37
2.8	Obtaining a customised MTC $t$ from the original MTC . . . . .	39
2.9	Linguistic scores of characteristics for candidate ontologies . . . . .	40
2.10	The Factors, Criteria & Metrics methodology [Cavano and McCall, 1978] . . . . .	42
2.11	The Goal-Question-Metric paradigm . . . . .	43
2.12	Examples of circularity error at a distance of 0, 1, and $n = 3$ . . . . .	56
2.13	Steps for metric definition adapted from Piattini et al. [2001] . . . . .	63
3.1	Overview of the ROMEIO methodology . . . . .	69
3.2	Ontology driven search engine . . . . .	71
3.3	GQM and ROMEIO . . . . .	88
4.1	Map of the districts in Paris and La Défense . . . . .	98
4.2	Map of Lake Tahoe in the United States . . . . .	99
5.1	Example of a cycle between categories of <i>Reality</i> and <i>Ontology</i> . . . . .	127
6.1	White-box validation process . . . . .	148

6.2	Black-box validation process . . . . .	148
6.3	White-box validation process for the F-measure using Paris and France ontologies	152
6.4	(a) Results showing granularity using the F-measure . . . . .	155
6.5	(b) Recall and (c) Precision . . . . .	156
6.6	Black-box validation process for the tangledness measure on Wikipedia category structure . . . . .	158
6.7	Racing Sports subtrees . . . . .	166
6.8	Foods subtrees . . . . .	166
6.9	Relevant articles found (definite) vs. Backtracking clicks for Subtrees <i>a</i> (base), <i>b</i> (untangled), and <i>c</i> (generated) . . . . .	169
6.10	Average number of clicks by users for Subtrees <i>a</i> (base), <i>b</i> (untangled), and <i>c</i> (generated) . . . . .	170
6.11	Number of relevant articles retrieved for Subtrees <i>a</i> (base), <i>b</i> (untangled), and <i>c</i> (generated) . . . . .	170
6.12	Visualisation of an excerpt from Subtree <i>a</i> for Racing Sports . . . . .	175
6.13	Visualisation of an excerpt from Subtree <i>b</i> for Racing Sports . . . . .	176
E.1	Pre-study questionnaire . . . . .	225
E.2	Interface description . . . . .	226
E.3	Training task . . . . .	227
E.4	Post-task questionnaire . . . . .	228
E.5	Task 1 description . . . . .	229
E.6	Task 2 description . . . . .	230
E.7	Task 3 description . . . . .	231
E.8	Task 4 description . . . . .	232
E.9	Task 5 description . . . . .	233
E.10	Task 6 description . . . . .	234
E.11	Post-study questionnaire . . . . .	235

# List of Tables

2.1	OntoClean meta-property constraints . . . . .	36
2.2	Characteristics listed for the Content dimension in OntoMetric [Lozano-Tello and Gómez-Pérez, 2004] . . . . .	38
2.3	Example of pairwise weights in a comparison matrix for MTC $t$ . . . . .	39
2.4	Measures for errors in ontology for consistency, completeness and conciseness proposed by Gómez-Pérez [2001] . . . . .	49
2.5	Measures for coverage of an ontology proposed by Brewster et al. [2004] . . . . .	49
2.6	Structural measures proposed by Gangemi et al. [2005] . . . . .	51
2.7	Measures proposed by Tartir et al. [2005] . . . . .	52
2.8	Measures proposed by Orme et al. [2006] . . . . .	52
2.9	Equivalent measures proposed in different work found in literature (rows indicate the equivalent measures). . . . .	62
3.1	Requirements description template . . . . .	70
3.2	Ontology role template . . . . .	70
3.3	Example application context and role of ontology . . . . .	72
3.4	Example: Ontology requirement 1 . . . . .	75
3.5	Example: Ontology requirement 2 . . . . .	75
3.6	Question template . . . . .	76
3.7	Criteria-questions: clarity, consistency, and conciseness. . . . .	77
3.8	Mappings between criteria-question to measures: expandability and correctness . . . . .	78
3.9	Criteria-questions: completeness and coverage . . . . .	78
3.10	Criteria-questions: minimal ontological commitment . . . . .	78

3.11	Questions for OR1 in the example application . . . . .	79
3.12	Questions for OR2 in the example application . . . . .	79
3.13	Measures template . . . . .	80
3.14	Mappings between criteria-question to measures: clarity, consistency, and consistency. . . . .	81
3.15	Mappings between criteria-question to measures: expandability and correctness	82
3.16	Mappings between criteria-question to measures: completeness, coverage and minimal ontological commitment . . . . .	84
3.17	Measures of Q1 . . . . .	85
3.18	Measures of Q2 . . . . .	86
3.19	Comparison between evaluation methodologies . . . . .	90
4.1	Lonely Planet guidebook publications . . . . .	93
4.2	Lonely Planet digital travel content . . . . .	96
4.3	Lonely Planet application context and role of an ontology . . . . .	102
4.4	LP ontology requirement: Controlled vocabulary of names, places and terms .	104
4.5	LP ontology requirement: Flexible classification of geographic items . . . . .	105
4.6	LP ontology requirement: Appropriate granularity . . . . .	106
4.7	Questions for OR1 . . . . .	107
4.8	Questions for OR2 . . . . .	108
4.9	Questions for OR3 . . . . .	109
4.10	Measures for Q1 . . . . .	110
4.11	Measures for Q2 . . . . .	111
4.12	Measures for Q3 . . . . .	111
4.13	Measures for Q4 . . . . .	113
4.14	Measures for Q5 . . . . .	114
4.15	Summary of the ROMEO analysis of ontology requirements for Lonely Planet	116
5.1	Size of the Wikipedia category structure . . . . .	122
5.2	Application context and purpose of ontology for Wikipedia . . . . .	124

5.3	Ontology requirement 1 . . . . .	129
5.4	Ontology requirement 2 . . . . .	130
5.5	Ontology requirement 3 . . . . .	130
5.6	Ontology requirement 4 . . . . .	131
5.7	Ontology requirement 5 . . . . .	132
5.8	Questions for OR1 . . . . .	133
5.9	Questions for OR2 . . . . .	134
5.10	Questions for OR3 . . . . .	134
5.11	Questions for OR4 . . . . .	135
5.12	Questions for OR5 . . . . .	135
5.13	Some measures for the Wikipedia category structure . . . . .	136
5.14	Measures for Q1 . . . . .	137
5.15	Measures for Q2 . . . . .	138
5.16	Measures for Question 3 . . . . .	140
5.17	Measures for Question 4 . . . . .	142
5.18	Measures for Question 5 . . . . .	143
5.19	Summary of the ROMEO analysis of ontology requirements for Wikipedia . .	145
6.1	Mapping of Q5 to M5 from the Lonely Planet case study in Chapter 4 . . . .	151
6.2	Results for ontology and corpus size, number of overlapping terms, precision, recall and the F-measure . . . . .	154
6.3	Mapping of Q1 to M1 from the Wikipedia case study in Chapter 5 . . . . .	157
6.4	Measures for the Wikipedia category structure . . . . .	160
6.5	Subtrees used in validation experiments . . . . .	162
6.6	Tasks used for comparing ontologies for browsing articles . . . . .	164
6.7	Task sequence of Tasks 1 to 6 (t1-t6) for each user in validation user experiment comparing Subtree $a$ (base), $b$ (untangled) and $c$ (generated) . . . . .	165
6.8	Characteristics of the set of subtrees used . . . . .	167
6.9	Results and significance tests for Subtrees $a$ (base) and $b$ (untangled) . . . .	172



6.10 Results and significance tests combining tasks for Subtrees $a$ (base) and $b$ (untangled) . . . . .	173
6.11 Results for Subtree $a$ vs. $b$ . . . . .	174
A.1 Ontology role template . . . . .	192
A.2 Requirements template . . . . .	193
A.3 Question template . . . . .	193
A.4 Measures template . . . . .	193
A.5 Clarity, consistency, and conciseness mappings . . . . .	194
A.6 Expandability and correctness mappings . . . . .	195
A.7 Completeness, coverage and minimal ontological commitment mappings . . .	196
A.8 Intersectedness and appropriate granularity mappings . . . . .	197
B.1 Lonely Planet application context and role of an ontology . . . . .	198
B.2 LP Ontology Requirement: Controlled vocabulary of names, places and terms	199
B.3 LP Ontology Requirement: Flexible classification of geographic items . . . . .	199
B.4 LP Ontology Requirement: Appropriate granularity . . . . .	200
B.5 Questions for OR1 . . . . .	201
B.6 Questions for OR2 . . . . .	201
B.7 Questions for OR3 . . . . .	202
B.8 Measures for Q1 . . . . .	203
B.9 Measures for Q2 . . . . .	204
B.10 Measures for Q3 . . . . .	204
B.11 Measures for Q4 . . . . .	205
C.1 Application context and purpose of ontology for Wikipedia . . . . .	206
C.2 Ontology Requirement 1 . . . . .	207
C.3 Ontology Requirement 2 . . . . .	207
C.4 Ontology Requirement 3 . . . . .	208
C.5 Ontology Requirement 4 . . . . .	208
C.6 Ontology Requirement 5 . . . . .	209

C.7 Questions for Ontology Requirement 1 . . . . .	209
C.8 Questions for Ontology Requirement 2 . . . . .	210
C.9 Questions for Ontology Requirement 3 . . . . .	210
C.10 Questions for Ontology Requirement 4 . . . . .	210
C.11 Questions for Ontology Requirement 5 . . . . .	211
C.12 Measures for Question 1 . . . . .	212
C.13 Measures for Question 2 . . . . .	213
C.14 Measures for Question 3 . . . . .	213
C.15 Measures for Question 4 . . . . .	214
C.16 Measures for Question 5 . . . . .	215

# List of Algorithms

1	The Floyd cycle-finding algorithm . . . . .	141
2	Varying a subtree . . . . .	161



# Abstract

Ontologies play key roles in many applications today. Therefore, whether using an ontology that is specified from scratch or selected from a collection of ontologies for use in its target application, it is important to determine the suitability of an ontology to the application. This need is addressed by carrying out *ontology evaluation*, which determines qualities of an ontology using methodologies, criteria or measures. However, for addressing the ontology requirements from a given application, it is necessary to determine what the appropriate set of criteria and measures are.

In this thesis, we propose a *Requirements-Oriented Methodology for Evaluating Ontologies* (ROMEEO). ROMEEO outlines a methodology for determining appropriate methods for ontology evaluation that incorporates a suite of existing ontology evaluation criteria and measures. The ROMEEO methodology helps ontology engineers to determine relevant ontology evaluation measures for a given set of ontology requirements. ROMEEO links these requirements to existing ontology evaluation measures through a set of questions. There are three main parts to ROMEEO. First, *ontology requirements* are elicited from a given application and form the basis for an appropriate evaluation of ontologies. Second, appropriate *questions* are mapped to each ontology requirement. Third, relevant ontology evaluation *measures* are mapped to each of those questions.

From the ontology requirements of an application, the ROMEEO methodology is used in this thesis to determine appropriate methods for ontology evaluation by mapping applicable questions to the requirements and mapping those questions to appropriate measures. We perform the ROMEEO methodology to obtain appropriate ontology evaluation methods for ontology-driven applications, and in particular, in case studies of Lonely Planet and

Wikipedia.

Since the mappings determined by ROMEEO are dependent on the analysis of the ontology engineer, a method for validating these mappings is also needed. As such, in addition to proposing the ROMEEO methodology, a method for the *empirical* validation of ROMEEO mappings is also proposed in this thesis. We report on two empirical validation experiments that are carried out in a controlled environment to examine the performance of the ontologies over a set of tasks. These tasks vary and are carried out to compare the performance of a set of ontologies in the experimental environment. The ontologies are varied based on a specific ontology quality or measure being examined.

Empirical validation experiments are conducted for two mappings between *questions* and their associated *measures*, which are drawn from case studies of Lonely Planet and Wikipedia. These validation experiments focus on mappings between questions and their measures. Furthermore, as these mappings are application-independent, they may be reusable in subsequent applications of the ROMEEO methodology. Using a ROMEEO mapping from the Lonely Planet case study, we validate a mapping of a *coverage question* to the *F-measure*. The validation experiment carried out for this mapping was inconclusive, thus requiring further analysis. Using a ROMEEO mapping from the Wikipedia case study, we carry out a separate validation experiment examining a mapping between an *intersectedness question* and the *tangledness measure*. The results from this experiment showed the mapping to be valid. For future work, we propose additional validation experiments for mappings that have been identified between questions and measures.

# Chapter 1

## Introduction

*A science is as mature as its measurement tools.*

— Louis Pasteur

Ontologies are used in many fields today to provide explicit, consensual, and controlled vocabularies for various domain and application needs. They are used in many geographic, scientific and engineering domains to support knowledge-intensive applications requiring information interoperability and integration. However, with the growing use of ontologies in these domains, their evaluation is important in helping to determine the suitability of an ontology for the requirements of an application or domain.

The term *ontology* in the computing sense, that we use in this thesis, is derived from the ontological domain in philosophy. *Ontology* in philosophy is the study of existence, specifically, the existence of entities, their features and how they relate to other entities in a metaphysical sense. In computer science, an ontology refers to an “explicit specification of a conceptualisation” [Gruber, 1995]. More specifically, an ontology in computing is a specification of concepts, instances and their relationships for a given domain; it includes attributes, assertions and logical constraints of those concepts, instances and relationships.

Whether building a new ontology or reusing an existing ontology, a common problem is knowing whether a given ontology is suitable for the application at hand. In the case of building a new ontology, it is important to ascertain whether the ontology is adequately

specified. In the case of selecting an ontology, it is important to distinguish the qualities of a suitable ontology, as there may already be very many ontologies available. Hence, *ontology evaluation* is essential for designing, building, testing and maintaining ontologies, that is, *ontology engineering*, as well as the selection of an appropriate ontology from a pool of already defined ontologies, that is, *ontology selection*.

Although ontology content is application independent, its evaluation may be influenced by the set of requirements. Pfleeger and Atlee [2005] define a *requirement* as “an expression of desired behaviour” and based on this definition, we define an *ontology requirement* as the expression of desired quality or competency of an ontology. Ontology requirements may vary between applications. As such, an examination of the relevant set of ontology requirements for a specific application is necessary for the appropriate evaluation of ontologies.

In this thesis, we address some fundamental questions on ontology evaluation. In particular, we consider how ontologies are evaluated according to a set of ontology requirements for a given application; and how aspects of an ontology evaluation method may be validated. We propose a requirements-oriented methodology for evaluating ontologies and perform an analysis of appropriate ontology evaluations using this methodology across two case studies. We contend that this methodology allows for ontology evaluation to be carried out on different ontology-driven applications, and is flexible enough to accommodate both existing ontology evaluation criteria and measures as well as those that may be proposed in the future.

## 1.1 Ontology evaluation

*Ontology evaluation* is carried out on ontologies using methodologies, methods and measures, to examine a set of ontology criteria. A particular *ontology evaluation methodology* may include the use of methods, ontology evaluation criteria and ontology evaluation measures. A *methodology* is a collection of procedures, phases or processes, metrics, and measures with an underlying rationale or philosophy associated with it [Avison and Fitzgerald, 2002] and a *method* is “a formal procedure for producing some result” [Pfleeger and Atlee, 2005]. A *methodology* differs from a *method* in that the application of a methodology produces one or more methods. A *methodology* provides a framework for using procedures and components,



according to an underlying rationale, to produce a specific *method*. For example, in the Goal-Question-Metric methodology (GQM) for software measurement [Basili and Weiss, 1984], a *methodology* is described to elicit goals, associate questions to discern the goal elicited, and apply relevant metrics to answer the associated questions. The use of GQM may produce many different *methods* of software evaluation, for example in one instance, the goal may be to improve the quality of software components, and in another instance, the goal may be to increase the efficiency of software specification.

*Ontology evaluation criteria* are general qualities an ontology ought to possess. A *criterion* is defined as a quality used for making judgement or comparison. Examples of criteria described by Gómez-Pérez [1996] include: *consistency*, which evaluates the logical consistency of an ontology; *correctness*, which refers to whether entities in the ontology and their properties, are modelled according to the entities in the world being modelled; and *conciseness*, which determines whether an ontology is without redundant definitions.

*Ontology evaluation measures* help to quantify some aspect of an ontology, for example, the depth measure quantifies an aspect of an ontology's structure. We adopt the definition of a *measure* given by Pressman [2005] as “a quantitative indication of extent, amount, dimensions, capacity, size of some attribute of a product or process”, although the term *measure* is defined differently elsewhere.<sup>1</sup> An example of an existing ontology evaluation measure is the *number of circularity errors* by Gómez-Pérez [2001], which is used to measure cycles in an ontology.

## 1.2 Challenges in ontology evaluation

One of the challenges in ontology evaluation is determining an appropriate method of evaluation. According to Gómez-Pérez [2004], ontology evaluation is imperative for ontologies

---

<sup>1</sup>A *measure* is often used interchangeably with a *metric* in referring to the qualitative assessment of a product or system attribute. This same occurrence appears in a discussion by Fenton [1991] and definitions given by Pressman [2005] and the IEEE [1990]. However, the definition of a *metric* given by the IEEE [1990] is often extended to refer to a well defined *measure* with a specific standard unit of *measurement* associated, for example, length in centimetres. The definition of a *measure* is broader and need not have units of *measurements* associated with it.

that are used in software applications and in the quote below, she suggests that publishing an ontology without prior evaluation is “unwise”. Gómez-Pérez implies that an appropriate method of ontology evaluation is by using general principles of consistency, correctness and conciseness.

“Evaluating content is a must for preventing applications from using inconsistent, incorrect, or redundant ontologies. It is unwise to publish an ontology that one or more software applications will use without first evaluating it.”

There are cases, however, in which ontology evaluation should go beyond these general principles, specifically, the evaluation of ontologies for a given application. Ontologies that are used in applications should be evaluated based on the ontology requirements of the applications that utilise them. Furthermore, an aspect of this challenge is determining appropriate measures. There have been numerous measures proposed. However, there is a need for a means of determining a set of measures that addresses specific ontology requirements of an application, rather than using ontology evaluation measures arbitrarily. The challenge here is determining an appropriate method for evaluating ontologies for a given application, which includes associating the relevant set of measures, based on the ontology requirements. A suitable ontology evaluation methodology may help to address this challenge. However, current methodologies do not determine appropriate methods for building and selecting ontologies based on ontology requirements, and thus may be unsuitable.

Two existing ontology evaluation methodologies are: *OntoClean* [Welty and Guarino, 2001; Guarino and Welty, 2002] and *OntoMetric* [Lozano-Tello and Gómez-Pérez, 2004]. *OntoClean* evaluates the correctness of the subclass relationships of an ontology. The underlying rationale of this methodology is that notions adapted from philosophy, such as essence and identity, help with correcting the set of subclass relationships modelled in an ontology. *OntoClean* only evaluates this set of relationships, which is only one aspect of an ontology. Furthermore, it does not help to determine whether an ontology satisfies the ontology requirements of an application.

*OntoMetric* is used for ontology selection. Its rationale is that suitable ontologies are determined by an analysis of desirable ontology characteristics based on application re-

quirements. Each candidate ontology is then compared against this set of characteristics. Although, OntoMetric is able to associate ontology characteristics with requirements in selecting ontologies, a major limitation with this methodology is that it does not help evaluate ontologies for the case of building new ones. In order for this challenge to be met, a new ontology evaluation methodology is needed – one that specifically uses ontology requirements from applications as a basis for evaluation activities for both ontology selection and ontology engineering.

Another challenge of ontology evaluation is validation. Fenton [1991] defines *validation* as “the process of ensuring that the measure is a proper numerical characterization of the claimed attribute”. The *validation* of each aspect of an ontology evaluation method is important to ensure that what is measured, corresponds to the relevant ontology requirements.

One way of validation is through empirical experiments to examine attributes of ontologies or the performance of tasks using ontologies. Sure [2004] notes the benefits that experimentation can bring to the field of ontologies as it has with other fields and cites the Text REtrieval Conference (TREC) being a platform for evaluation in information retrieval as an example. Thus, a validation process involving experiments may help to address this challenge of validating aspects of an ontology evaluation method.

In this thesis, we focus on addressing these two challenges. We specifically seek to answer the following research questions:

1. What is an appropriate methodology to determine the method for evaluating ontologies for a given set of ontology requirements of an application?
2. How are aspects of an ontology evaluation method validated?

This thesis addresses the first question by proposing a *Requirements Oriented Methodology for the Evaluation of Ontologies* (ROMEEO). ROMEEO outlines a process which helps the ontology engineer to relate the appropriate aspects of existing ontology evaluation criteria and measures to the ontology requirements for an application using questions. We apply ROMEEO to two case studies of ontology-driven applications, specifically, from Lonely Planet and Wikipedia, to explore whether this methodology is suitable for ontology evaluation. In particular, we examine whether ROMEEO helps to identify an appropriate set of measures to

address ontology requirements of an application.

We address the second question by proposing a validation process. In particular, this validation process uses experiments to validate aspects of specific ontology evaluation methods that may be produced using ROMEO. Using this validation process, we then carry out two experiments to examine aspects of ontology evaluation methods obtained by using ROMEO.

### 1.3 Thesis overview

To provide a background to the research questions, we discuss existing aspects of ontology evaluation – proposed methodologies, criteria, and measures – and present these in Chapter 2. We also survey methodologies in software engineering in that chapter, and draw on existing software engineering methodologies to incorporate relevant ideas for a methodology for ontology evaluation.

In Chapter 3, the ROMEO methodology is introduced. ROMEO provides a framework to apply the existing evaluation criteria and measures for addressing ontology requirements from an application. The underlying rationale for the ROMEO methodology is that ontology evaluation depends on examining the ontology-driven application and its ontology requirements. We outline the specific components in ROMEO — *requirements*, *questions* and *measures* — which are adapted from the GQM methodology. We specify how these components are used to obtain suitable methods for ontology evaluation.

We then apply the ROMEO methodology to obtain appropriate ontology evaluation methods for two case studies. In Chapter 4, we examine a case study involving Lonely Planet where we apply ROMEO for evaluating ontologies used in the travel domain. We elicit a set of ontology requirements from an analysis of the Lonely Planet application for content management of travel media. ROMEO is used to determine an appropriate method from those requirements. In Chapter 5, we present a case study of Wikipedia, and in particular, its category structure which is used for browsing Wikipedia articles. Ontology requirements are gathered from an analysis of its category structure with regards to its role in the navigation and browsing of Wikipedia articles. These requirements are then used to determine an appropriate ontology evaluation method for this application using ROMEO. In both cases,

mappings between ontology requirements to questions, and questions to appropriate measures are established using ROMEO.

The mappings established from a given ROMEO ontology evaluation analysis require validation to determine whether they are correct. In Chapter 6, a process is presented for empirical validation. This involves tasks to be performed using a set of varied ontologies in controlled experimental environments. The ontologies used in these experiments are taken from a base ontology and vary on aspects determined by the mapping being considered. If the results from each experiment vary according to the properties of the ontologies used, then the mapping being examined is validated. This validation process can be used to validate any aspect of a given ontology evaluation method. However, validated mappings between questions and measures are more reusable, as they are not necessarily dependent on any application. Thus, we use this validation process in this thesis for some mappings between questions and their measures from the case studies of Lonely Planet and Wikipedia as presented in Chapters 4 and 5 respectively. In particular, we validate new mappings that we discover using ROMEO from the two case studies in this thesis.

Finally, in Chapter 7, we conclude and present implications of ROMEO for ontology evaluation, and summarise our findings from applying ROMEO to the two case studies. We also summarise the two experiments carried out using the validation process we propose and present avenues for future work.

## Chapter 2

# Ontologies and their evaluation

*The only good is knowledge and the only evil is ignorance.*

— Socrates

Ontology evaluation is as important for the quality of ontologies as software measurement is to software quality. *Measurement*, in a broad sense, helps to examine the level of quality of a product or a given process. For ontologies, *evaluation* helps to examine whether requirements for a given ontology have been met, for example, the requirement of a consistent ontology. However, ontology requirements differ depending on the application and domain they apply to.

This chapter provides a background to ontology evaluation, specifically, we present current methods, measures, and methodologies for ontologies and their evaluation. We also survey some literature regarding software measurement and consider how relevant aspects of it may be applied to the evaluation of ontologies.

Before examining specific ontology evaluation methods, an understanding is needed of the various roles and application areas where ontologies are used and present these in Section 2.1 of this chapter. In that section, we motivate the need for ontology evaluation based on requirements of the ontology for a given application and role it plays.

In the remaining sections of this chapter, we review aspects of ontology evaluation in the literature and we seek to identify any gaps that may exist. We recognise similarities between

software engineering and ontology engineering which leads us to examine how methodologies for software measurement can be used in ontology evaluation. Section 2.2 presents some methodologies for ontology engineering and examines the degree of ontology evaluation in them. We examine existing work on evaluation methodologies and compare existing ontology evaluation methodologies with software evaluation methodologies in Section 2.3. Last, in Section 2.4, we survey specific measures for ontology evaluation. In comparing ontology evaluation with software evaluation, we aim to reflect on lessons in literature about software engineering methodologies and how they may be applied to ontology evaluation.

## 2.1 Ontologies

The term *Ontology* dates back to the ancient Greek philosophers who sought to account for existence in the world through reasoning and logic, by using philosophical principles such as *essence*. The artificial intelligence community adapted the term ontology to refer to an explicit model of a domain of knowledge consisting of a set of concepts, their definitions and inter-relationships [Uschold and Grüninger, 1996; Gruber, 1995]. Since then, this kind of ontology has been widely used in various applications and computing disciplines and is increasingly becoming a key component in a variety of applications. A widely accepted definition of this kind of ontology is that of “an explicit specification of a conceptualisation” [Gruber, 1995].

The content, structure and granularity of ontologies may differ from one another. Ontology content differs according to the domain it models, and the purpose for its specification. A domain here refers to a domain of discourse, which encompasses knowledge and rules in a given discipline, a field of study or subject area. Depending on the modelling choices made by the ontology engineer and ontology specification language, the structure of an ontology may be different from another. Regarding granularity, ontologies may vary according to its scope; there may be general ontologies, which encompass a broad spectrum of domains, or very specific domain or task ontologies. Ontologies available online range from the more generic upper ontologies, such as Suggested Upper Merged Ontology <sup>1</sup>, to specialised ones for geo-

---

<sup>1</sup><http://ontology.teknowledge.com>

graphical needs, such as the Geography Ontology in the DARPA Agent Markup Language Library <sup>2</sup>.

Additionally, there are different types of ontologies due to the various roles an ontology may take depending on the application it is used in. Some roles that ontologies play in applications are: supplying a controlled vocabulary, maintaining consistency, specifying conceptualisations and knowledge in a given domain, organising information for navigation, enabling dynamic data-driven presentation, and supporting interoperability. Also, ontologies are used in domains such as: medicine, biology, commerce, web applications, information retrieval and navigation, and knowledge management.

With such a variety of ontologies, there is an implication for their evaluation. Since ontology-driven applications differ in purpose, domain, and requirements, not all ontologies may be relevant for a given application. Consequently, a method of evaluating ontologies based on the specific requirements of a given application is important.

In this section, we examine the different types of ontologies, their specification, how ontologies are currently used and the roles they play in various applications. In Section 2.1.1, we distinguish between simple and structured ontologies and discuss the expressive capability of each. We present a brief survey of ontology specification language in Section 2.1.2, in particular, we describe how ontologies are specified using OWL. Section 2.1.3 outlines the different types of ontologies with regard to their granularity. We also consider the roles of ontologies and examples of ontology usage in applications in Section 2.1.4.

### 2.1.1 Simple and structured ontologies

Broadly, McGuinness [2002] classifies ontologies as either simple or structured ontologies.

*Simple ontologies* possess, at the least, the following features:

- a finite controlled vocabulary;
- unambiguous interpretation of classes and term relationships; and
- strict hierarchical subclass relationships between classes.

---

<sup>2</sup><http://www.daml.org/ontologies/412>



Simple ontologies are less expressive than structured ones and are known as *terminological ontologies* as they often do not have axioms and definition constraints [Sowa, 2000]. They are often easier to specify and many are available today. Figure 2.1 gives an example of a geography ontology with three defined concepts or classes – *Region*, *Country*, *City*. Additionally, these concepts have simple relationships defined – *Country* and *City* are both a type of *Region*, and a given *Country* can have a *City* within it. Uses for simple ontologies include:

- *umbrella structures* that may be extended for specialised applications;
- standardised ontology for sharing knowledge [Noy and McGuinness, 2001], for example, SNOMED & UMLS for medical informatics; and
- agreement of naming conventions, for example, set of terms used by different authors.

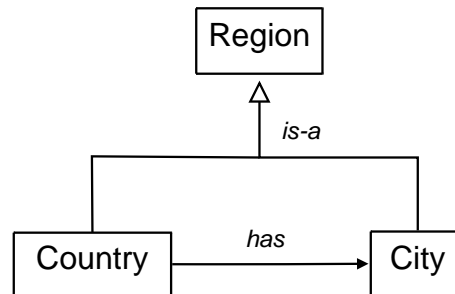


Figure 2.1: A simple ontology

*Structured ontologies* capture descriptions of a given set of concepts and relationships between them with greater expressivity than simple ontologies. Generally, ontologies with richer descriptions and more constraints placed on them that go beyond simple generalization and specialization relations of taxonomies are called structured ontologies [McGuinness, 2002]. Structured ontologies also allow the definition of equality axioms and mappings that support interoperability between two data models. Last, terms can be expressed precisely with regard to one another [McGuinness, 2002]. As such, structured ontologies approach the expressiveness and property constraints of first order logic. Such constraints and ex-

pressivity allow for expected values and type to be checked for each concept that is defined. They can also be referred to as *formal ontologies*, if they include axioms and definitions of constraints [Sowa, 2000]. An example of a constraint described in a structured ontology and not in a simple ontology is the cardinality restrictions placed on concepts in an ontology.

Figure 2.2 gives an example of a more complex ontology from before. Here we have the same concepts defined but in this ontology, the descriptions of relationships are richer. We have defined a *Region* to have sub-regions within it with transitive properties, that is, for all sub-regions of a region, it is part of the ancestor region. Also, the relationship between *Country* and *City* is revised to include cardinality constraints, in particular, a *Country* is assumed to have at least one or more *Cities*. Also, an assumption in this model is that each *Country* has a capital city. We model this relationship with an inverse functional property, thus modelling that each *Country* has at most one capital *City*. Lastly, we model the *City* concept to be part of at most one *Country*.

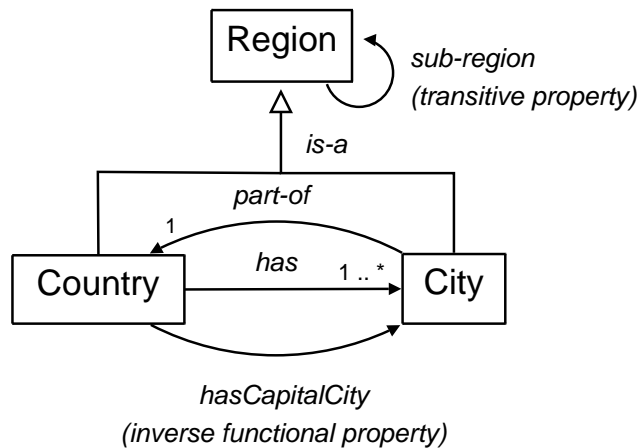


Figure 2.2: A structured ontology

From the examples given in Figures 2.1 and 2.2, we can see that structured ontologies can provide richer descriptions for an ontology. There are cases where simple ontologies lack expressiveness required in some applications, for example, if cardinality constraints were needed between classes or properties (for example, each country has only one capital city).

Such constraints ensure that the term being modelled is consistent with the world model. In such cases, structured ontologies should be used. The drawback of a structured ontology is that its specification may be more time consuming and have increased complexity. As such, the expressivity of simple ontologies may be adequate for some applications. It is left up to the knowledge engineer to weigh the trade-offs and decide between specifying simple and structured ontologies based on a given ontology's expressive needs.

### 2.1.2 Ontology specification languages and OWL

Ontology specification languages are the means for ontologies to be expressed in an explicit specification. There have been many ontology languages proposed. Early ontology languages include KIF<sup>3</sup>, LOOM<sup>4</sup> and F-Logic [Kifer et al., 1995]. There are also ontology languages for the web such as SHOE<sup>5</sup>, XOL<sup>6</sup>, OML<sup>7</sup>, DAML<sup>8</sup> and OIL<sup>9</sup>. However, these ontology languages were proposed for specific user groups, each having their own syntax. They also vary in expressiveness and computational properties, that is, some are more expressive and complex to compute, while others are simple and easily computed.

**Web Ontology Language (OWL)** OWL is an ontology specification language proposed in 2004 by the W3C as part of the Semantic Web initiative [Herman, 2004]. OWL is founded on description logic and has been designed for the web and incorporates existing web standards like XML and RDF. It allows for web documents to be formally described for machine processability and reasoning tasks [McGuinness and van Harmelen, 2004]. This recent specification language was based on past languages such as DAML, OIL and DAML+OIL.

OWL differs from other proposed ontology specification languages, due to its integration with web standards but also by allowing for more expressivity and considerations of scalability [Herman, 2004]. That is, it has the flexibility to fully describe an ontology in a way

---

<sup>3</sup><http://www-ksl.stanford.edu/knowledge-sharing/kif/>

<sup>4</sup><http://www.isi.edu/isd/LOOM/LOOM-HOME.html>

<sup>5</sup><http://www.cs.umd.edu/projects/plus/SHOE/>

<sup>6</sup><http://www.ai.sri.com/pkarp/xol>

<sup>7</sup><http://www.ontologos.org/OML/..%5COML%5COML.html>

<sup>8</sup><http://www.daml.org>

<sup>9</sup><http://www.ontoknowledge.org/oil>

that is computationally decidable.

The full expressive capability that can be achieved with ontology languages like OWL may suit some applications which require complex semantics of concepts and relations to be captured. However, this may not be applicable in all cases, for example, some applications define a very small ontology with simple specifications that fit with the expressivity of OWL Lite. Thus, there is a place for both levels of specification depending on the application needs.

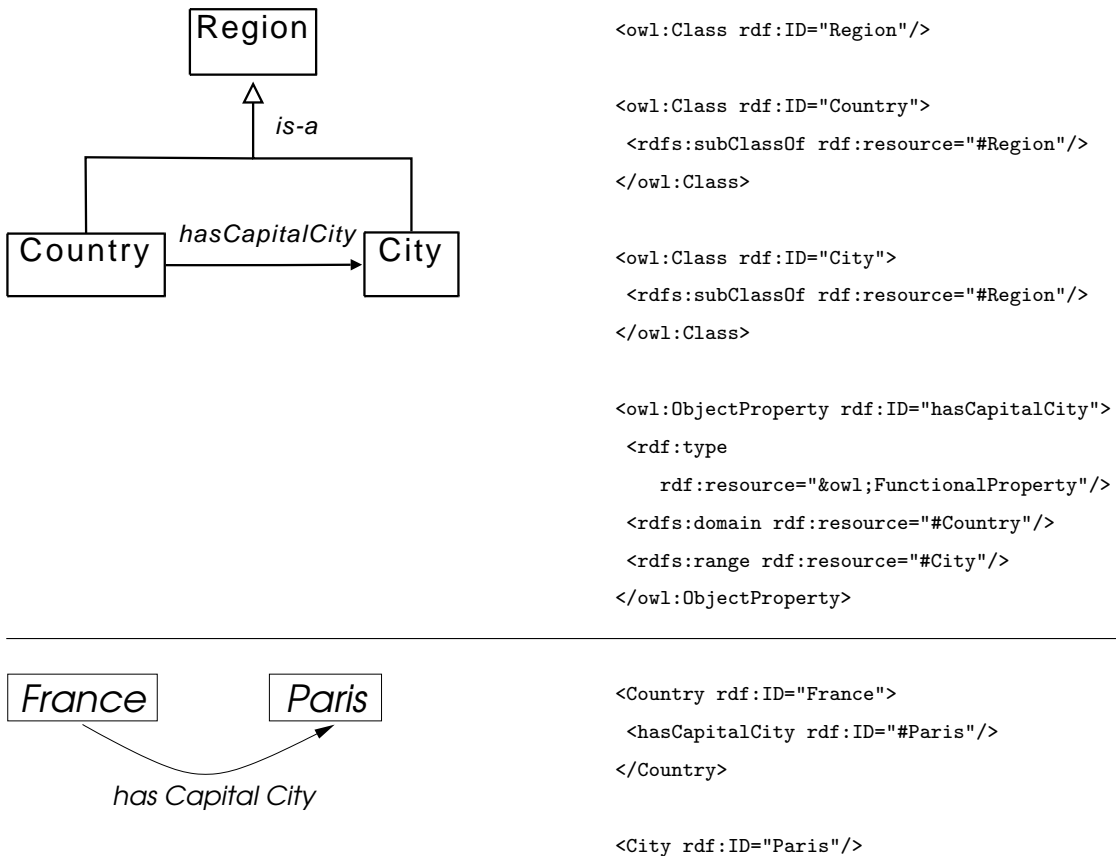


Figure 2.3: Model of Paris and France in OWL: a) Schema; and b) Instances

Figure 2.3 shows how a simple and course grained conceptual model of *Paris* in *France* is specified in OWL. In the example, three classes are defined – *Region*, *Country* and *City*. Constraints are put on these classes such that *City* and *Country* are both subclasses of

*Region*. Additionally, a functional property is specified for the class *Country* of a capital city called ‘*hasCapitalCity*’. We then specify instances of these classes – a country, *France* and a city, *Paris*. Lastly we specify that *France* has *Paris* as its capital city with a ‘*hasCapitalCity*’ relationship.

### 2.1.3 Ontology granularity

Some concepts in an ontology have higher abstraction or granularity than others. For instance, the *Region* concept, that is defined in Figures 2.1, 2.2 and 2.3, is more abstract than the *Country* concept. [Keet, 2008] defines granularity in a broad sense as “the ability to represent and operate on different levels of detail in data, information and knowledge.” In ontologies, it is how we represent its levels of detail and delimit the scope of a given ontology. For the example above, if we use the *Region* concepts alone, we operate on a coarser level of granularity than if we include the concepts of *Country* and *City*, which have a finer level of granularity.

Ontologies may define a set of concepts that fall within particular levels of granularity. However, in literature, the definition of the general levels of granularity is not consistent. Gandon [2001] presents a model with 3 layers of abstraction for different ontologies: a top layer consists of an ontology with abstract concepts, a middle layer that defines concepts for an application and a bottom layer that has concepts with notions related to a specific task. Fonseca et al. [2002b] outline 4 levels of ontologies: top-level ontologies describing general concepts, domain ontologies describing domain related concepts, task ontologies describing task or activity concepts, and application ontologies describing concepts that are a combination of domain and task concepts. Alexakos et al. [2006] defines 3 layers for searching within distributed hypermedia: an upper search ontology layer consisting of basic concepts, a domain description ontology layer consisting of domain concepts, and a semantic metadata layer that handles heterogeneous hypermedia server data.

Hence, as the aforementioned models for levels of granularity differ, we gather these ideas and classify the levels. Incorporating the above ideas, we define the levels of ontology granularity to be upper ontologies, middle ontologies, domain ontologies and application

ontologies (although there are ontologies which feature all levels of granularity from upper foundational concepts, to application specific concepts, such as Wordnet and Cyc [Lenat and Guha, 1989; Lenat, 1995]).

Figure 2.4 illustrates the levels we define. We discuss each level and give an example of how some existing ontologies fit with this model below.

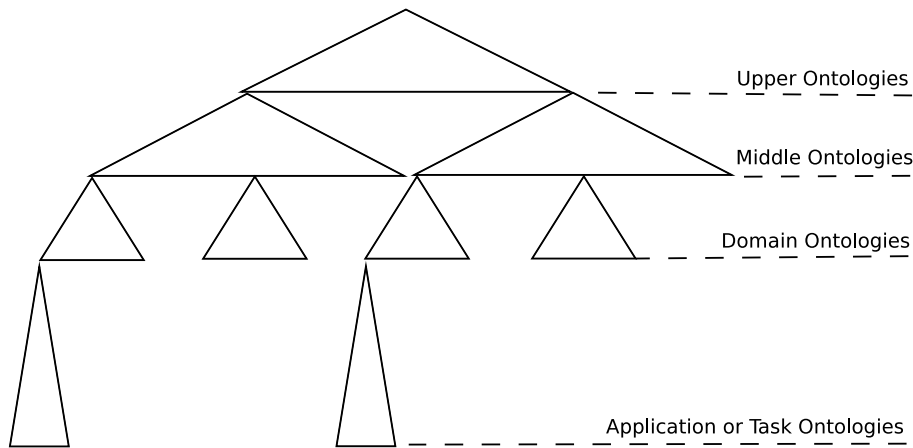


Figure 2.4: Levels of granularity of ontologies

**Upper ontologies.** These ontologies seek to generalize how abstract concepts relate to each other. In doing so, they may be used as umbrella structures to organise concepts in lower levels of an ontology. While, it is possible for a universal ontology to exist, this is not the case in practice due to various reasons. For example, there may be many disagreements about how a particular concept is represented. Therefore, it is possible for even more disagreements to occur in an ontology, where many concepts and their relationships may be defined. Poli [1996] points to a more viable scenario – having several general ontologies made available for parties to commit to one over another. Examples of upper ontologies include SUMO [Niles and Pease, 2001], Sowa upper ontology [Sowa, 2000], Dolce [Gangemi et al., 2002], and CliP [Batres et al., 2007].

**Middle ontologies.** There are also ontologies that serve to provide intermediate content be-

tween abstract concepts of the upper ontologies such as SUMO and domain ontologies. An example is Middle Level Ontology (MILO)<sup>10</sup>.

**Domain ontologies.** A more general type of ontology than the task ontology is the domain ontology. Guarino [1997] argues that “reusability across multiple tasks or methods should be systematically pursued even when modeling knowledge related to a single task” for the aim of obtaining “task-independent” aspects of a conceptualization. The result of this is a domain ontology. Domain ontologies are generally more reusable than an application ontology and possess a higher shared value. Examples of domain ontologies are:

**Earth sciences:** Semantic Web for Earth and Environmental Terminology (SWEET)

**Medicine and Life sciences:** SNOMED, GALEN, MEDLINE, UMLS, Gene Ontology, Protein Ontology

**Agriculture and Food:** United Nations Standard Products and Services Code (UNSPSC), AGROVOC, Plant ontology

**Geography:** Ethanalogue, Getty Thesaurus of Geographic Names

Additionally, there are domain ontologies available in conjunction with SUMO and MILO<sup>11</sup>. This includes domain ontologies of Communications, Countries and Regions, Government, Geography, People, Finance, and Economics.

**Application ontologies** These ontologies are very specific in nature and relate directly to details of a given application or task, for example, in web services, ontologies can describe specific mappings for a given application. We include tasks ontologies in this level of granularity. Requirements of application ontologies tend to be narrower as they are designed for various applications.

Upper levels of the above classification, allow for umbrella-like structures to enable disambiguation, integration and grounding of concepts in lower levels of granularity. Upper

---

<sup>10</sup><http://suo.ieee.org/SUO/SUMO/index.html>

<sup>11</sup><http://www.ontologyportal.org>

ontologies define concepts that have higher abstraction, thus allowing more granules to be defined to help partition concepts in domain and application ontologies. The example below shows how upper, middle, domain and application ontologies may be used with each other for the purpose of describing a restaurant in Paris, France using some existing ontologies.

**Example.** In this example, we use the following ontologies: *SUMO*, *MILO*, the *Countries and Regions* ontology and a separate *task ontology for describing restaurants in France*. *SUMO* and *MILO* provide the concepts to support the domain ontology of *Countries and Regions*. The *application* ontology, which requires the specification of instances of restaurants, uses the concepts, relationships and instances provided by the *Countries and Regions* domain ontology. Figure 2.5 illustrates this example.

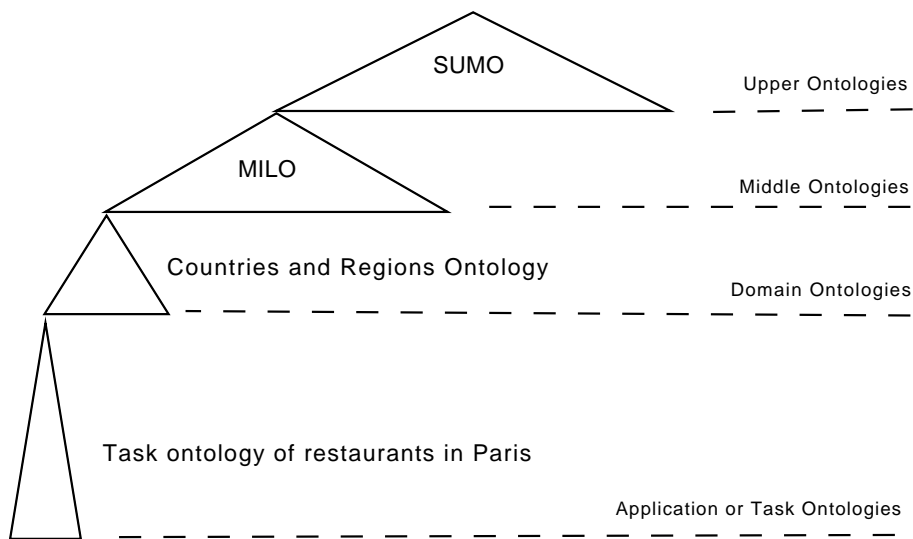


Figure 2.5: Example of the granularity of ontologies

Additionally, there may be further distinctions for concepts in a particular level of granularity. For example, let us consider the *Region* concept for the geographic domain. There may be additional levels of granularity to model the subclasses of a *Region*, like the concepts of *Political region*, *Natural region*, *Tourism region* and *Religious region*. Further levels of



granularity may be represented within each of these concepts. For example, within the concept of a *Political region*, we may model concepts of *Country*, *State*, *City*, *Province*, *County*, *Township*, *Territory*, and *Municipality*. Last, the level of detail of instances may also be defined, such the city of *Paris* and the town of *Clamecy* in the country *France*. The example given above is one way of partitioning granularity, however, there have been others proposed for representing granularity for geography and other domains in literature [Keet, 2008; Kumar et al., 2004; Bittner and Smith, 2003; 2001].

Despite the ability to define concepts to a very fine level of detail, certain levels of granularity may be appropriate for a given application. In the geographic domain, concepts may be defined to a fine level of granularity such that it includes Global Positioning System (GPS) co-ordinates, but this level of granularity may only be appropriate for applications which plot maps to those fine levels of granularity. Other applications may require coarser levels of granularity, such as travel-related applications.

An approach to get around the problem of determining appropriate granularity is to leave it up to the application. This approach is taken by Fonseca et al. [2002b] with geographic ontologies. However, in other cases, applications may require ontologies with the right level of granularity modelled to begin with.

#### 2.1.4 Ontologies used in applications

Ontologies are used in a variety of applications to bring a range of benefits and to meet specific application needs from areas such as Knowledge Management, e-Science, the Semantic Web [Berners-Lee et al., 2001] and E-Commerce. Ontologies facilitate communication, sharing and reuse of knowledge that is understood between applications.

The key benefits for applications using ontologies are interoperability [Sure, 2004], data integration of heterogeneous information repositories, and a shared representation of knowledge. Regarding interoperability, ontologies offer a way to handle heterogeneous representations of information [Davies et al., 2003]. They enable parties to agree on a conceptualisation for use in a given domain or across domains.

Regarding data integration, ontologies enable applications such as the Semantic Web, for

bringing heterogeneous data together and integrating various data stores and information sources and providing an ability to inference over them using shared semantics. An example of this is found in the life science discipline, that has adopted the use ontologies to maintain and integrate data repositories for biology, chemistry and medicine for sharing and maintaining consistent vocabularies as well as advancing research in their respective areas. Below, we elaborate further on each benefit described and give examples of how ontologies are able to benefit applications.

There are also specific roles that ontologies play in various applications, for example, ontologies can be used as a medium to organise and access information. It has been used in search tasks and also in information navigation systems. Other applications use ontologies to ensure consistency in the use of terms, definitions and concepts within a domain, thus reducing ambiguity and allowing common understanding. This may involve groups of people rather than automated programs and software, for example, common understanding in given domain like bio-medicine. Thus, ontologies have a variety of application areas. We also discuss and present specific examples of these applications below.

### **Data integration and interoperability**

Ontologies may be able to resolve issues with regard to the integration and sharing of the “overwhelming amount of information stored in various repositories” [Pisanelli et al., 2002]. Often each repository has its own format. Sure [2004] notes that a key benefit of ontologies is interoperability.

Data integration is related to interoperability. Interoperability arises as collaboration between groups and parties increase. Thus requiring the sharing through standardised communication both between people and machines, for example, in business applications, communication and exchange of information is a key requirement.

Ontologies can do so by facilitating communication, sharing and the reuse of knowledge. Application areas such as E-commerce, B2B transactions and web services use ontologies for interoperability support. Disciplines, such as medicine, computing and science, have developed standardised ontologies for sharing [Noy and McGuinness, 2001].

Ontologies, such as domain specific ontologies in the areas from biology to geography to upper ontology mappings, allow for data integration. Hovy [2003] integrates data sources via ontology using logical mappings and SENSUS – an upper ontology. Maedche et al. [2003] mapped and integrated ontologies using a mediator. Data repositories are integrated with ontologies in the work of Maedche et al. [2003]. Hakimpour and Geppert [2002] integrate database schemas using ontologies – one for each database and another for integrating the ontologies.

Many ontologies have been made available in the medical and bio-informatics domain. In the biology domain, Bechhofer et al. [2005] propose a system to organise hyperlinks and utilise the Gene ontology to resolve concepts. The system maps synonyms of labels given to particular biology terms as well as presents related links. They show an improvement in being able to browse the Gene ontology. Noy et al. [2004] propose a unified view of ontologies from the different medical informatics data sources (such as UMLS<sup>12</sup>, GALEN<sup>13</sup>, SNOMED-CT<sup>14</sup>, GenBank<sup>15</sup>, MEDLINE<sup>16</sup> and the Gene ontology<sup>17</sup>). They go on to propose features an ontology repository should have or support, for example, ontology summarisation, rating, browsing and search utilities.

In the geographic domain, Fonseca et al. [2002a] present an implementation of their ontology-driven geographic information system (ODGIS). In this system, geographic heterogeneous data are integrated using ‘mediators’. Ontologies are used to integrate data models of these heterogeneous information sources, to link with mediator, and to browse information.

Obrst et al. [2003] describe scenarios where ontologies would be suited for business applications (B2B and E-Commerce). In e-commerce, ontologies could be used to link buyer and seller product information. Ontologies could also facilitate a common interface and an integrated conceptual model. They provide attributes such as semantic soundness, controlled vocabulary as well as consistency which is desired in business applications. However, ontolo-

---

<sup>12</sup><http://www.nlm.nih.gov/research/umls>

<sup>13</sup><http://www.opengalen.org>

<sup>14</sup><http://www.snomed.org>

<sup>15</sup><http://www.ncbi.nlm.nih.gov/Genbank>

<sup>16</sup><http://medline.cos.com>

<sup>17</sup><http://www.geneontology.org>

gies are more commonly used for representing business logic and data as opposed to aspects of presentation to the user. That is, the application layer would implement presentation components that utilises an ontology or a set of ontologies to display different views dependant on the user's context.

### **Navigation systems and web applications**

Ontologies are used as conceptual structures abstracted over data for navigation. Domingo and Eklund [2005] apply Formal Concept Analysis (FCA) to email management. Although ontologies are not used, the method of applying an ontology to navigate or manage email in place of FCA could be done. Tho et al. [2006] use FCA techniques to generate an ontology, as well as attributes from a database, for use in a Semantic Web application in the academic domain. Similarly, Burrow [2004] applies lattices to determine access rules for publishing, viewing and collaboration in a wiki. Ontologies are also used in online portal systems [Lara et al., 2004; Oberle and Spyns, 2002].

### **Information and multimedia retrieval systems**

Ontologies have been used to enhance and complement information retrieval. In Hyvönen et al. [2003], their system, *Ontogator*, makes use of ontologies to aid in semantic browsing and multi-faceted search for image retrieval. In similar work, Schreiber et al. [2001] present a content-based approach to image annotation and search using ontologies. The ontology used by Schreiber et al. [2001] serves as a controlled vocabulary (which helps search), it captures relations and provides a means to widen or narrow the query, and is able to link and comply with other sources of metadata standards such as Dublin Core.

Andreou [2005] uses an ontology to improve results from a text retrieval system. The Wordnet ontology was used to boost candidate query expansion terms based on the semantic similarity of queries to candidate terms. This was performed on a subset of the TREC HARD track for 2003. Results from this work show that by selecting terms with high entropy, the ontology was able to help boost the performance of selected search tasks.

In the area of spatial information retrieval, Jones et al. [2004] use an ontology to define

the semantics in their spatial retrieval engine, SPIRIT. The ontology is a simple ontology but possesses a large set of instances of places and other related spatial information.

With regard to clustering, Hotho et al. [2003] present a clustering scheme over news articles and web documents using the Wordnet ontology as background knowledge. An ontology was used to resolve synonyms in an attempt to disambiguate text so that matching can be done conceptually. Breaux and Reed [2005] describe a system for hierarchical clustering and evaluations on a small topic-centric text. These text collections consisted of 29 articles with document size of less than 200 words. The clustering scheme uses term expansion and results show that it is “as good as a meta-word search”.

### **Knowledge management, organisational memory and group memory**

Ontologies have been used to organise and manage knowledge and information both on the individual and corporate level. Haase et al. [2005] explore the evolution of a user’s personal ontology based on collaborative filtering systems for use in Bibster, a bibliography management tool. Yu et al. [2004] consider the effectiveness of navigating group memory for a project, in particular, emails exchanged between members of a project team, using an ontology that represented task related concepts. Vasconcelos et al. [2002] propose a framework of a organizational memory system using ontologies. Here ontologies are used for different sections of the conceptual layer – organisation ontology, information ontology, domain ontologies, formal ontologies. They are used in the application layer to support “organisational knowledge-intensive tasks”.

### **Software specification and development**

Another area ontologies have been used is software development. Knublauch [2004] propose an architecture for developing a semantic web application using ontologies. Macias and Castells [2003] present dynamic web page authoring using domain information captured in an ontology.

## Teaching systems and eLearning

Ontologies can be used for the purposes of delivering and supporting teaching and eLearning. Ontologies are used here as a point of truth to abstract teaching material and the applications driving the teaching activities. These kind of applications use ontologies to describe semantics and the domain, the context, and provides structure to the materials in the learning course [Stojanovic et al., 2001], although sometimes, ontologies may not be used in all three aspects. Yang et al. [2004] present a teaching system in a virtual community that uses ontologies for knowledge management of e-documents and for driving semantic applications using those ontologies. Ontologies are used to annotate documents to manage aspects of the repository and further leverage on those ontologies for semantic content retrieval. Gascuea et al. [2006] propose a domain ontology for adaptive learning materials that takes into consideration context of the user's learning style and delivery modes. The domain ontology that is used here allows a separation from the delivery device and the learning content, thus allowing content to be delivered to a wide range of devices. Ontologies are also used here to annotate descriptions of the learning objects with different learning styles in mind.

## 2.2 Ontology engineering methodologies

Ontology engineering involves the construction, refinement (or evolution) and usage of a given ontology for an application. An ontology engineering methodology outlines a process of building ontologies involving requirements gathering, development, refinement, repository management, and evaluation.

Most ontology engineering methodologies follow the model that Uschold and King [1995] outline as a skeletal methodology for specifying ontologies. The steps for this are presented below:

1. Identify purpose and scope
2. Building the ontology
  - (a) ontology capture
  - (b) ontology coding

(c) integrating existing ontologies

3. Evaluation

4. Documentation

Uschold and King [1995] also include ‘Guidelines for each phase’ in this skeletal methodology. They describe both informal and formal ontology engineering methodologies.

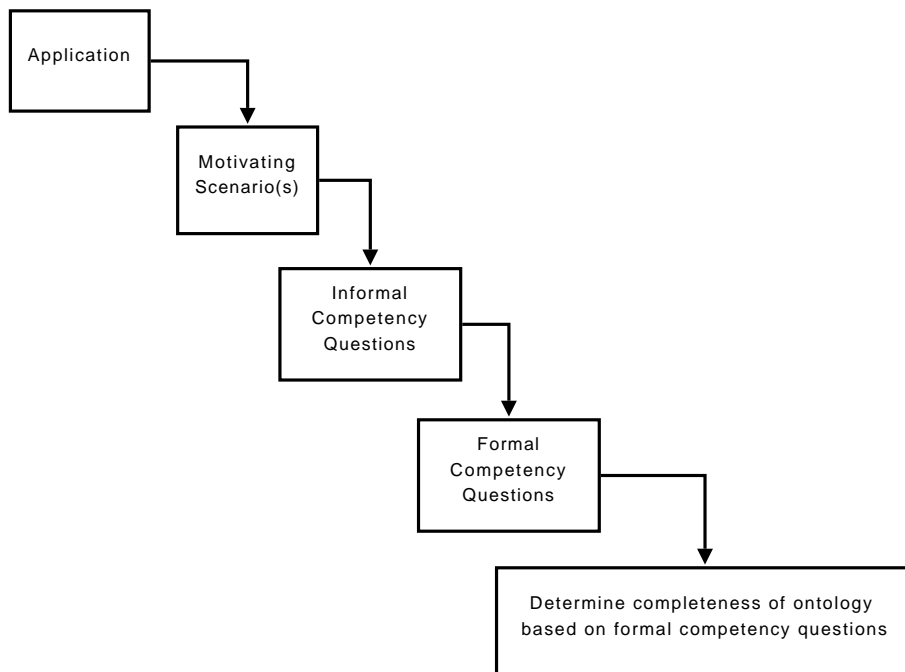
In this section, we present a selection of ontology engineering methodologies and how ontologies are developed using them. We also seek to examine the degree to which ontologies are evaluated using these methodologies. For ontology engineering methodologies, evaluation is only one aspect and features a limited scope for ontology evaluation. In Section 2.2.1, we present a formal method for ontology engineering that Grüninger and Fox [1995]; Uschold and Grüninger [1996] propose. We introduce Methontology [López et al., 1999] in Section 2.2.2, which applies ontology evaluation based on domain expert verification and the satisfaction on competency questions. *Competency questions* outline a list of questions a particular ontology should be able to answer. They can be taken as a *frame of reference* in the evaluation of ontologies. However, there have been some methodologies focused solely on ontology evaluation, specifically OntoClean and OntoMetric, and we present these as part of discussions of specific ontology evaluation methodologies in Section 2.3. The On-To-Knowledge methodology, presented in Section 2.2.3, also relies on competency questions being satisfied as a means of ontology evaluation. Last, we present a methodology for ontology engineering using the SENSUS ontology in Section 2.2.4.

### 2.2.1 Formal method for ontology engineering

Grüninger and Fox [1995]; Uschold and Grüninger [1996] propose a formal methodology for ontology design and evaluation. This methodology is described by Grüninger and Fox [1995] in building an enterprise model called a Common Sense Enterprise model for the TOVE enterprise modelling project. This methodology proposes a process of ontology engineering outlined and is described below. Figure 2.6 illustrates the process.

1. Motivating scenarios – requirements are gathered from analysis of the application scenarios.

2. Competency questions – a set of competency questions arise from the motivating scenarios. These questions do not have to be exhaustive but give an indication of what the scope of the ontology should be.
3. Formalise ontology – formalisation using first order logic of the terminology of the ontology (that is, objects in the domain of discourse and their relationships), and axioms (that is, constraints upon the domain objects and their relationships).
4. Formalise competency questions
5. Verify ontology completeness – a set of instances is used as ground literals to determine whether the ontology satisfies a given competency question.



*Figure 2.6: Methodology presented by Grüninger and Fox [1995]*

This methodology guides the design of ontologies and provides a starting point for ontology evaluation using competencies identified from requirements of a given application. The use of formal logic is emphasised in this methodology. However, there are limitations with this formal approach. They are:



- The formal approach requires expertise with an ontology specification language that allows for formal logic expressions. These languages are used to specify the set of competency questions and to represent ontologies.
- This methodology relies on the existence of a given set of instances to determine whether an ontology satisfies a given competency question, which may not necessarily be available.
- The method of evaluation proposed in this methodology is only feasible for ontologies that have been specified with a formal logic language, such as first order logic; not all ontologies today are specified with formal logic language. Thus, to use this evaluation method, ontologies may require a conversion of its specification language to one supporting formal logic. This is a reasonable task for small ontologies, however, it may not be feasible for large ontologies.
- Ontology reuse using this methodology relies on a library of existing ontologies specified using a formal language. Also, selecting ontologies for reuse using this methodology depends on formally specifying a set of competency questions.

Despite these limitations, the use of competency questions reflects the importance of requirements in performing ontology evaluation and they have been used in methodologies such Ontology-101 [Noy and McGuinness, 2001], and has influenced the OBI ontology [OBI], and Methontology, which we present next.

### 2.2.2 Methontology

*Methontology* is an ontology development methodology [López et al., 1999; Corcho et al., 2002; 2003] and is based on the IEEE standard for software development [Breitman et al., 2007]. This methodology is used in conjunction with the WebODE ontology engineering tool developed for browsing and editing ontologies, to support ontology development.

The methodology aims to identify the required knowledge for conceptualising an ontology using the following series of steps:

1. Identify main concepts and build concept classification tree

2. Create ad-hoc binary relations between concepts
3. Define class attributes and instance attributes of concepts
4. Create axioms and rules

Methontology's emphasis in the specification of an ontology is maintaining an ontology's consistency and completeness throughout its development [López et al., 1999; Lozano-Tello and Gómez-Pérez, 2004]. The philosophy or underlying rationale of this methodology is that "you evaluate ontology content throughout the entire lifetime of the development process". Methontology seeks to "carry out most of the evaluation (mainly consistency checking in concept taxonomies) during the conceptualization activity to prevent errors and their propagation in the implementation" [Lozano-Tello and Gómez-Pérez, 2004]. This is seen in the recommendation to recognise disjoint and exhaustive partitions wherever possible in identifying the set of concepts in Step 1. However, it does not discuss documentation explicitly, which is one of the steps described in the skeletal methodology by Uschold and King [1995].

A limitation with this methodology is that it only evaluates ontologies with evaluation criteria of consistency and completeness. However, for some applications, ontologies may be required to be evaluated according to other ontology evaluation criteria.

### 2.2.3 On-To-Knowledge

Staab et al. [2001] describe a methodology for developing ontologies for knowledge management (KM) applications called On-To-Knowledge. The On-To-Knowledge methodology begins at project setup and follows through to the roll-out of the application. This methodology features 5 steps and they are presented below:

1. Feasibility study – this is carried out before ontology development begins and identifies foreseeable problem areas and their possible solutions.
2. Kickoff – this step describes ontology requirements in an ontology requirement specification document. This document outlines the goals, scope and domain of the ontology, the users and their use cases, the competency questions and possible existing ontologies available for reuse.

3. Refinement – this step aims to develop an ontology according to the specification document. Also, the target ontology is refined to the point where it is usable in the KM application.
4. Evaluation – the ontology engineer checks the target ontology against the competency questions identified in the specification document. This step also involves testing the ontology in the application environment and getting user feedback on it. Evaluating the ontology may involve cycling back to the refinement step until the ontology reaches a level of satisfaction.
5. Maintenance – after the ontology has been adopted and is in use for a period of time, maintenance may be required to reflect changes and updates to accommodate changes in the state of the application environment.

In contrast to the skeletal methodology outlined by Uschold and King [1995], the step of documentation occurs early in this methodology rather than near the end as the skeletal methodology suggests. It is then used as a reference throughout the ontology engineering life cycle. The main emphasis of this methodology is in cycling back and forth with adjacent steps until a level of maturity. Only when this level of maturity is reached that subsequent steps are taken, for example, an ontology is refined and evaluated until the ontology is at a satisfactory state before moving on.

Ontologies are evaluated by the ontology engineer to determine whether the ontology being developed satisfies competency questions and ontology requirements of an application outlined in the specification document [Staab et al., 2001]. This step also involves testing the usefulness of the ontology in the application. Thus, this approach relies on evaluating the performance of the ontology in an application. The merit of this approach is that ontology evaluation activities carried out are specific to the application where the ontology is used. However, the limitation of this approach is determining the appropriate measures or experiments to use to evaluate ontologies according to the requirements of an application. In On-To-Knowledge, this is left up to the ontology engineer, as there is no systematic way of determining appropriate measures to evaluate ontologies.

### 2.2.4 SENSUS-based ontology methodology

Hovy [1998; 2003] proposes a methodology for ontology engineering of domain ontologies that is based on the SENSUS ontology. The proposed methodology involves pruning the SENSUS ontology, which is an ontology comprising of around 70,000 concepts, to exclude irrelevant concepts outside of a given domain. The resulting ontology can be further extended to model additional knowledge for the respective domain. The steps of this methodology are presented as follows:

1. Identify seed terms – seed terms are identified, that is the key domain terms.
2. Manually link seed terms to SENSUS ontology – this is then linked to the SENSUS ontology.
3. Add paths to the root – paths to the root of the ontology is then identified.
4. Add new domain terms – relevant terms are added manually to the ontology that are not in the ontology already.
5. Add complete subtree – decisions need to be made as to which nodes to add within the subtree.

The emphasis of this methodology is grounding domain ontologies with the SENSUS ontology. However, this methodology does not discuss any steps for evaluation or documentation.

### 2.2.5 Discussion

The methodologies we presented in this section adhere loosely to the skeletal methodology proposed by Uschold and King [1995]. Each methodology deviates depending on the emphasis it places.

Where a methodology included a step of evaluation, it is only part of a much bigger overall process of ontology development and has limitations. The formal method for developing ontologies proposed by Grüninger and Fox [1995]; Uschold and Grüninger [1996], depends on the ability to express all components involved using formal logic, such as, the requirements, ontologies and competency questions. They are used to determine any deficiencies in the

ontology. There is a drawback with this, that is, the expertise needed to manually specify each component using formal logic languages.

Methontology relies on competency questions for evaluating ontologies as well as ensuring evaluation criteria of consistency and completeness of an ontology throughout the ontology development cycle. However, ontologies may be required to be evaluated according to other ontology evaluation criteria rather than those considered in Methontology.

On-To-Knowledge also relies on competency questions for evaluating ontologies. In its evaluation phase, it includes the testing of ontologies in the application according to these questions to determine the suitability of the ontology. The ontology engineer continues to develop the ontology until its use in an application is deemed adequate, assuming that the ontology engineer knows what to modify the ontology to include. The limitation with this evaluation approach is that ontology engineers must have a sense of what constitutes an appropriate ontology. The use of measures may help with this. However, this methodology does not discuss any specific methods or techniques for the measurement of the ontology itself to help ontology engineers determine appropriate measures to base their evaluation.

### **2.3 Ontology evaluation methodologies**

Broadly speaking, evaluation provides relevant feedback for the purposes of improving, assessing the quality of something or process. “Professional evaluation is defined as the systematic determination of the quality or value of something” [Scriven, 1991]. “Evaluations are generally conducted for one or two main reasons: to find areas for improvement and/or to generate an assessment of overall quality or value (usually for reporting or decision-making purposes)” [Davidson, 2005]. Evaluation also helps us determine when the desired level of quality has been attained. Moeller and Paulish [1993] state that “you cannot easily manage what you cannot measure”. Another motivation for evaluation is to challenge assumptions and accepted practices. In the case of ontologies, evaluation is carried out for: ontology selection, tracking progress in ontology development, and determining qualities such as completeness for ontologies.

In this section, we present two ontology evaluation methodologies. Ontology evaluation

methodologies are distinguished from ontology engineering methodologies as they provide a framework for defining appropriate methods for evaluating ontologies. The two ontology evaluation methodologies that we present in this section have been influential in ontology evaluation to date. They are OntoClean and OntoMetric, which are presented in Sections 2.3.1 and 2.3.2 respectively.

In this section, we also consider how aspects in software evaluation methodologies can be used for improving ontology evaluation methodologies. In Section 2.3.3, the specific software engineering methodologies we present are the Factor-Criteria-Measure methodology (FCM) and the Goal-Question-Metric methodology (GQM). FCM and GQM demonstrate that software measures are driven by a set of software qualities or goals.

### 2.3.1 OntoClean

The OntoClean methodology evaluates ontologies using formal notions from philosophy. The OntoClean goal is used to “make modelling assumptions clear” [Welty and Guarino, 2001]. Applying OntoClean may help an ontology meet the evaluation criterion of correctness. Correctness refers to whether the modelled entities and properties in an ontology correctly represents entities in the world being modelled. OntoClean addresses this by introducing meta-properties to capture various characteristics of classes, and constraints upon those meta-properties, which help to assess the correct usage of the subsumption relation between classes in an ontology [Welty and Guarino, 2001; Guarino and Welty, 2002]. “OntoClean is a method to clean concepts taxonomies according to meta-properties such as rigidity, identity and unity” [Lozano-Tello and Gómez-Pérez, 2004].

#### Meta-properties

The OntoClean methodology seeks to correct classes in an ontology that are modelled as subclasses when they should be consistently modelled as properties, as a subclass of another concept or even a separate class on its own. The principles of rigidity, identity, unity and dependence are used to determine this. We will briefly describe these notions below:

**Rigidity.** There are 3 types of properties under rigidity: rigid, non-rigid and anti-rigid. A

rigid property is one essential to the object and to all of its instances (+R). A non-rigid property is where only some of the object's instances are rigid (-R). Otherwise, where a property is not essential to any of an object's instances, it is anti-rigid ( $\sim$ R).

**Identity** “is related to the problem of distinguishing a specific instance of a certain class from other instances of this class by means of a characteristic property which is unique for it” [Guarino, 1998]. An identity criteria (IC) identifies an object via a sameness relation. That is, they are “used to determine equality (sufficient conditions) and that are entailed by equality (necessary conditions)” [Guarino and Welty, 2002]. Hence, a property carries identity (+I) via an IC. A property supplies identity, if it imparts its own IC to its instances (+O). Otherwise a property neither supplies, nor carries an identity (-I)

**Unity** “is related to the problem of distinguishing the parts of an instance from the rest of the world by a unifying relation that binds the parts, and only the parts together” [Guarino, 1998]. There are 3 types of properties here: unity, non-unity and anti-unity. A property has unity if there is a common relationship such that all instances are wholes under this relation (+U). A property has non-unity if only some of the object's instances are whole under the common relation (-U). A property has anti-unity if there are no common relation between all instances of the object that make it whole ( $\sim$ U).

**Dependence.** A property is dependent when it is necessary that another class must exist for it to exist. For example, for a child to exist, it is dependent on there being a parent. A property that is dependant is denoted by +D and -D if it is not.

The four meta-properties above are used as ontological tools to analyse subclasses in an ontology. Looking at the definitions of the four types of meta-properties, it can be seen that some classes with a given meta-property may only subsume a set of classes that possess specific meta-properties. For example, an anti-rigid class will naturally subsume another anti-rigid class; it is counter-intuitive for it to subsume a rigid class. Hence, after tagging meta-properties to classes in an ontology, these constraints are used to detect whether there are conflicting subsumption relations. Guarino [1998] includes all possible constraints and we present them in Table 2.1.

Table 2.1: *OntoClean meta-property constraints*

Superclass	Must Subsume
$\sim R$	$\sim R$
+I	+I
+U	+U
$\sim U$	$\sim U$

Yu et al. [2005a] carry out an application of OntoClean and found that it is able to clarify modelling assumptions made in an ontology. The most crucial part of this ontology methodology is that each class in the ontology is required to be tagged with these formal meta-properties. Thus, a limitation with this is it requires the set of concepts be manually classified using the proposed meta-properties. This activity is very time-consuming and tedious. Also, often these classifications may be subjective as it depends on the interpretation of the concept by the knowledge engineer. There are tools that aim to help in the application of this methodology, for example, AEON [Völker et al., 2005], which processes documents found online using natural language processing tools to determine various types of relations and automates the assignment of OntoClean meta-properties to concepts. However, they do not completely remove subjectivity and manual classification.

Also, another limitation of OntoClean is that it only evaluates the correctness of the subsumption relationships of an ontology. There may be other aspects of the ontology that require evaluation, for example, the completeness of an ontology.

### 2.3.2 OntoMetric

Lozano-Tello and Gómez-Pérez [2004] propose the OntoMetric methodology which uses application constraints as a basis for ontology selection. Ontology selection using OntoMetric is achieved using an adapted version of the Analytic Hierarchy Process (AHP) method proposed by Saaty [1994], which aids decision-making based on multiple criteria. Using AHP, OntoMetric differentiates ontologies based on a set of ontology characteristics.



In OntoMetric, a key component is the multilevel tree of characteristics (MTC), which is a taxonomy of characteristics (shown in Figure 2.7). The top level of this taxonomy has five *dimensions* of an ontology and are: content, language, methodology used to build the ontology, tools used to build the ontology, and the cost to utilise the ontology.

A set of factors is associated with each dimension and for each factor, there are characteristics associated also. These characteristics are taken from existing work and include design qualities, ontology evaluation criteria, cost, language characteristics. The content dimension is presented below in Table 2.2.

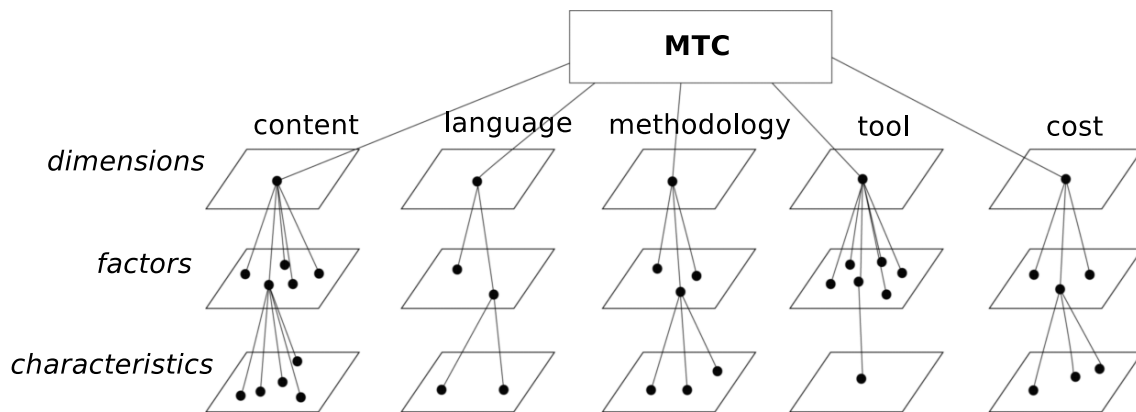


Figure 2.7: The Multilevel Tree of Characteristics (MTC) in OntoMetric [Lozano-Tello and Gómez-Pérez, 2004]

The process in utilising this methodology is:

**Step 1. Analyse project aims.** From the analysis of an ontology engineer, a set of objectives are specified according to guidelines for a suitable ontology given by the respective organisation seeking to adopt an ontology. “They must decide on the importance of the terms of the ontology, the precision of definitions, the suitability of relations between concepts, the reliability of the methodology used to build the ontology, etc.” [Lozano-Tello and Gómez-Pérez, 2004].

**Step 2. Obtain a customised MTC.** Based on the set of objectives from the project aims, the MTC describe above is modified to include a set of desired characteristics.

Table 2.2: Characteristics listed for the Content dimension in *OntoMetric* [Lozano-Tello and Gómez-Pérez, 2004]

Characteristic	Linguistic scale
Essential concepts	very low, low, medium, high, very high
Essential concepts in superior levels	very low, low, medium, high, very high
Concepts properly described in NL	very low, low, medium, high, very high
Formal specification of concepts coincides with NL	very low, low, medium, high, very high
Attributes describe concepts	very low, low, medium, high, very high
Number of concepts	very low, low, medium, high, very high
Essential relations	very low, low, medium, high, very high
Relations relate appropriate concepts	very low, low, medium, high, very high
Formal specification of relations coincides with NL	very low, low, medium, high, very high
Arity specified	very low, low, medium, high, very high
Formal properties of relations	very low, low, medium, high, very high
Number of relations	very low, low, medium, high, very high
Several perspectives	very low, low, medium, high, very high
Appropriate not-subclass-of	very low, low, medium, high, very high
Appropriate exhaustive partitions	very low, low, medium, high, very high
Appropriate disjoint partitions	very low, low, medium, high, very high
Maximal depth	very low, low, medium, high, very high
Average of subclasses	very low, low, medium, high, very high
Axioms solve queries	very low, low, medium, high, very high
Axioms infer knowledge	very low, low, medium, high, very high
Axioms verify consistency	very low, low, medium, high, very high
Axioms not linked to concepts	very low, low, medium, high, very high
Number of axioms	very low, low, medium, high, very high

This results in a customised MTC  $t$ , as illustrated in Figure 2.8.

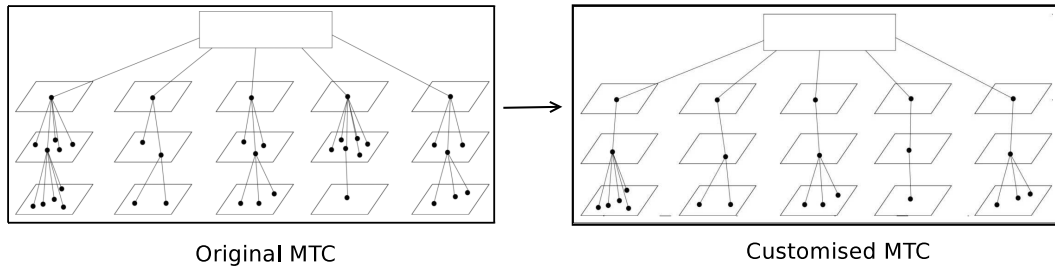


Figure 2.8: Obtaining a customised MTC  $t$  from the original MTC

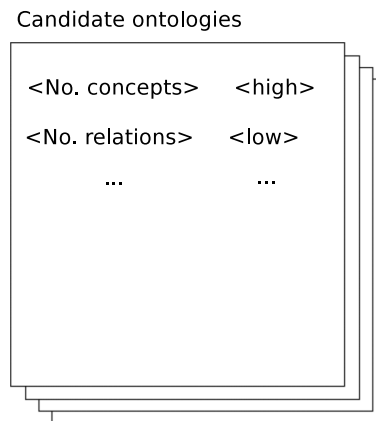
**Step 3. Weigh up each characteristic against each other.** Pairs of characteristics are then weighted against each other to indicate the importance of one characteristic over another. For each characteristic, a weight  $w_t$  is assigned against each other. This pairwise comparison forms a comparison matrix and eigenvectors are calculated from this matrix. This is used to determine the suitable ontologies in Step 5. An example of a comparison matrix is given in Table 2.3 below.

Table 2.3: Example of pairwise weights in a comparison matrix for MTC  $t$

Characteristic	Number of concepts	Number of relations
Number of concepts	1	2
Number of relations	0.5	1

**Step 4. Assign linguistic score for each characteristic of a candidate ontology.** For each candidate ontology  $c$ , assess their characteristics and assign score  $w_c$  along the linguistic scale for the given characteristic. The linguistic scale is applied for each characteristic and varies according to the characteristic. A typical set of linguistic values may be  $\langle \text{very low}, \text{low}, \text{medium}, \text{high}, \text{very high} \rangle$  or  $\langle \text{non-supported}, \text{supported} \rangle$ . Figure 2.9 illustrates the assignment of linguistic scores for a set of candidate ontologies.

**Step 5. Select the most suitable ontology.** The similarity between the characteristics of each candidate ontology is evaluated. This is achieved by comparing vectors of the weights  $w_t$  and  $w_c$  of the candidate ontology  $c$  and the modified taxonomy of



*Figure 2.9: Linguistic scores of characteristics for candidate ontologies*

characteristics  $t$ .

OntoMetric is a criteria-based ontology evaluation methodology for choosing an ontology based on a set of ontology characteristics. OntoMetric provides a way to compare ontologies based on various objectives. The methodology bases its evaluation on multiple criteria which link directly to the objectives. However, there are some limitations with this methodology and we outline them below:

- Determining the customized MTC for ontology selection depends on the user's manual specification, which may be subjective and inconsistent. For example, in Step 1 of OntoMetric, the methodology instructs ontology engineers to determine the importance of aspects of an ontology such as the set of terms and relationships. However, OntoMetric does not help guide ontology engineers with the process of mapping these objectives to specific aspects of ontologies, especially evaluating content of an ontology. A crucial part of this methodology is that users need to be familiar with the set of ontology characteristics available.
- The list of characteristics for evaluating content is limited. There are other existing measures proposed in literature which we present in Section 2.4.
- The linguistic scale does not use specific measurements of an ontology characteristic. It is up to the user to assign values of an ontology characteristic for a candidate on-

tology according to the linguistic scale. There are no quantifiable indicators for a given association of a value on the scale to the criteria, which may limit how meaningful each comparison is. For example, a value of *high* given for the number of concepts in a candidate ontology may refer to 100 concepts or 1000 concepts.

- OntoMetric does not help ontology evaluation for the case of ontology development. Rather, OntoMetric can be used to decide which ontology is the most suitable from a set of candidate ontologies. In the case that a new ontology is being developed, this methodology does not help to determine a suitable method for evaluating that ontology.

### 2.3.3 Software evaluation methodologies

We now examine two methodologies used for evaluating software below. From these software evaluation methodologies, the way in which goals and abstract factors may be useful for determining a specific and focused set of ontology measures, which we consider in Chapter 3.

#### Factors-Criteria-Metric framework (FCM)

Cavano and McCall [1978] discuss the importance of the relationships between software characteristics and requirements. They propose a framework where factors are identified from the requirements, relate them to criteria and seek to quantify them using metrics with the aim of determining whether requirements are met. Figure 2.10 shows how the factors, criteria and metrics relate to each other.

Regarding software factors, there are many factors proposed in software engineering [Pressman, 2005], but we discuss the ISO9126 Quality Factors [ISO9126] as this has become the international standard reference for quality factors used for the evaluation of software. We give a brief description of each quality factor below. For each of the factors described, a set of sub-characteristics is further defined in the ISO 9126 standard. These elaborate the quality factor in greater depth. However, we will consider the 6 main factors and they are presented below.

**Functionality** The degree that the software developed satisfies stated needs.

**Reliability** The availability of the software.

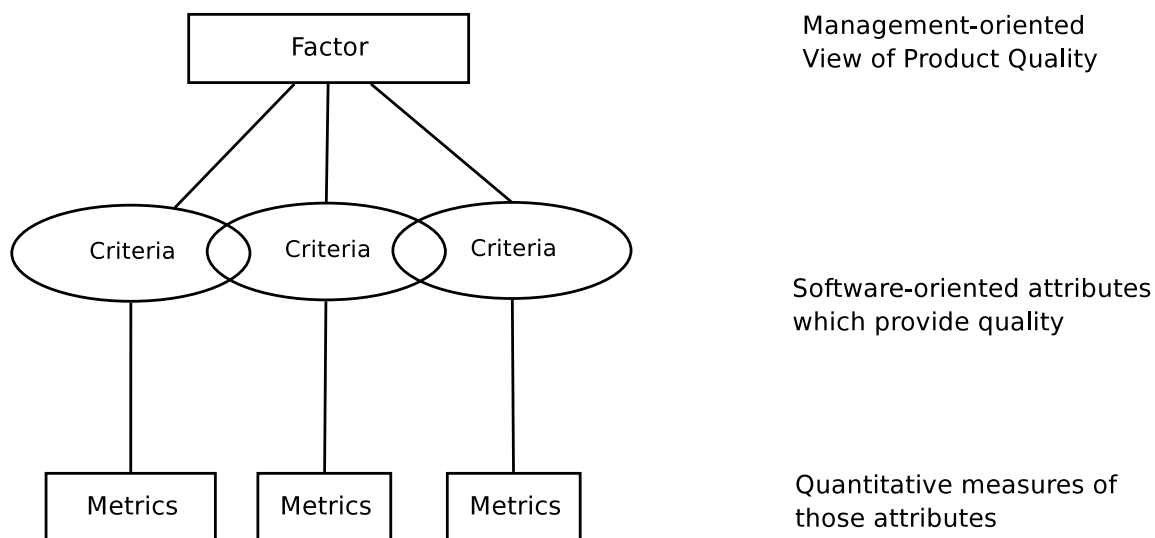


Figure 2.10: The Factors, Criteria & Metrics methodology [Cavano and McCall, 1978]

**Usability** Ease of use of the software.

**Efficiency** The degree that the software uses resources in an optimal manner.

**Maintainability** How easy a given software is able to be repaired.

**Portability** How easy a given software is able to be brought into different environments

Ontology evaluation often follows this model where an evaluation criterion is associated with measures and subsequently ontologies are evaluated from examining these measures for the criterion. These criteria for ontology evaluation are discussed in Section 2.4.1.

### The Goal Question Metric methodology (GQM)

Motivated by the lack of experimental validation in the field of software engineering, the Goal Question Metric methodology (GQM) [Basili and Weiss, 1984; Basili et al., 1994] was proposed. Basili and Weiss [1984] propose causal analysis to confirm or deny claims of various software development methodologies and improve software quality. This analysis requires useful data, which is obtained through appropriate data collection and emphasises the importance of the careful selection of collected data. Referring to the GQM methodology,

Piattini et al. [2001] “emphasize that the measurement goal should be clearly connected with an industrial goal, so the measurement program corresponds with a software organization’s needs”.

The steps outlined by Basili and Weiss [1984] for GQM are:

1. Establish *goals* of data collection
2. Develop a list of *questions* of interest
3. Establish data categories (that is, *metrics*)
4. Design and test collection form
5. Collect and validate data
6. Analyse data

Figure 2.11 shows a diagram of the GQM methodology.

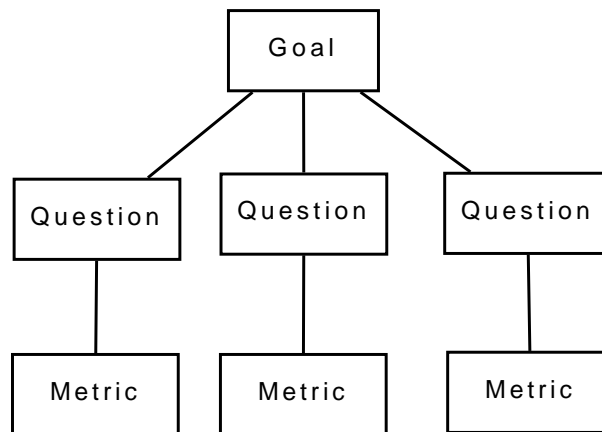


Figure 2.11: The Goal-Question-Metric paradigm

The benefits of this methodology are its flexibility, simplicity and ease in incorporating the methodology into an existing software measurement program with relatively little disruption. Questions play a key role in establishing relevant measures to aspects of a software quality factor. In some cases, this may be a limitation, as what is analysed and measured does not necessarily correspond fully to a factor. However, in other cases, this may be desirable as only certain aspects of a factor apply.

GQM has been used in other areas apart from software engineering, such as the evaluation of knowledge bases, which is proposed by Lethbridge [1998]. The GQM methodology is adapted there to determine appropriate metrics for knowledge bases. Lethbridge [1998] uses GQM to define general goals for knowledge engineers aiming to measure size and complexity of knowledge bases for knowledge management applications. These high level goals include: monitoring a knowledge base, and gaining greater understanding of the system, its users and the domain of the knowledge base. Specific tasks are associated with these goals and these tasks include determining the present state of the knowledge base, development time and effort tasks, and comparison tasks. Lastly for each of those tasks, specific metrics for knowledge bases are proposed.

Instead of linking questions to each goal, Lethbridge [1998] adapts the question component of the GQM methodology, and proposes to use tasks instead. This assumes that goals for knowledge engineering are static and Lethbridge [1998] only includes a limited number of goals in his discussion.

However, there has not been work conducted to apply GQM to ontology evaluation. The GQM methodology may be used for ontology evaluation, specifically, to help determine appropriate measures to use for an ontology evaluation method. Questions may be specified to explore aspects of an ontology evaluation criteria and relate to specific ontology requirements from an application. We present existing ontology criteria and measures in the next section before discussing the implications of the GQM methodology for ontology evaluation.

## 2.4 Measures

In this section, we consider existing ontology criteria and measures from the ontology evaluation literature. A previous survey in the area of ontology evaluation has been conducted by Brank et al. [2005], however, the evaluation methods considered were categorised by the type of evaluation, that is, data-driven, criteria based, or application based. We present a survey of ontology evaluation measures differently. Section 2.4.1 surveys existing ontology evaluation criteria which help to examine qualities of a given ontology. Section 2.4.2 presents existing ontology evaluation measures and associates them with the existing criterion where



possible. We also present a more detailed discussion of selected measures that are used in this thesis and adopt a consistent set of notations for them.

This section also examines software engineering literature and considers how measures may be defined and validated. We aim to draw on some of the foundations for software measurement and use them for ontology evaluation. In Section 2.4.3, we present a method for defining and validating metrics used in software engineering.

### 2.4.1 Ontology evaluation criteria

Various criteria have been proposed for the evaluation of ontologies. Ontology evaluation criteria are similar to Software Factors described for the FCM methodology presented in Section 2.3.1, in that they are used to examine software quality. In this case, ontology evaluation criteria examine the quality of ontologies.

Upon analysis, some of the criteria proposed by the different researchers address similar aspects when evaluating ontologies and overlap. We collate and describe them below as nine distinct criteria.

**Clarity** refers to a criterion of having definitions in an ontology defined unambiguously. A clear ontology should “effectively communicate the intended meaning of defined terms” and where possible the definition should be stated formally [Gruber, 1995]. Definitions in an ontology should be clearly specified such that there is no doubt or ambiguity.

**Consistency** describes the logical consistency of an ontology [Gómez-Pérez, 1996]. For a given ontology’s set of definitions and axioms (explicit or inferred), there should be no contradictions. This can be also referred to as *coherence* [Gruber, 1995],

**Conciseness** indicates a criterion of an ontology without redundant definitions. According to Gómez-Pérez an ontology is concise if:

1. It does not store any unnecessary or useless definitions
2. Explicit redundancies do not exist between definitions
3. Redundancies cannot be inferred using axioms attached to other definitions
4. The definition itself is not redundant

That is, an ontology is concise if redundancies do not exist and cannot be inferred from its definitions and axioms [Gómez-Pérez, 1996].

**Expandability** relates to the ability of an ontology to be extended further to describe specific application domains in a way that does not change the current definitions within the ontology [Gómez-Pérez, 1996]. This is also referred to as extendability by Gruber [1995].

**Correctness** refers to whether the concepts, instances, relationships and properties that are modelled, correlate with those in the world being modelled. Correctness depends on the frame of reference that the ontology is based on. Gómez-Pérez [1996] includes the metaphysical aspect as part of her description of consistency and we include this in our definition of correctness also.

One approach to measure correctness is to verify the correct modelling of concepts from the real world with interviews with domain experts and literature regarding the domain. This is found in the Methontology design process [López et al., 1999]. There, the ontology developers verified the correctness using a frame of reference from interviews with domain experts and other information sources. However this approach is time consuming and involves manual inspection.

**Completeness** refers to an ontology with absolute sufficiency in its definitions to all possible domains. Whether an ontology or its individual definitions are complete cannot be proven, however we can deduce an incomplete ontology by detecting individual definitions that may be incomplete or that at least one definition is missing from an ontology [Gómez-Pérez, 1996]. More specifically, we use as many relevant frames of reference of the world being modelled to help determine whether an ontology is incomplete.

**Coverage** captures the sufficiency of the ontology to represent the domain. Hovy [2002] refers to coverage in two distinct ways: *coverage of terms over the domain from concepts identified in the domain*; and *coverage or completeness of instances*. Brewster et al. [2004] considers the set of concepts and instances in the domain to be represented by

a corpus of information. They both refer to coverage as the *congruence* or *fit* of an ontology with the domain represented by the corpus of information. This is different from completeness, where the completeness criterion is having the ontology sufficient for all frames of reference. Coverage only considers whether the ontology has sufficiently modelled concepts and instances for one such *frame of reference*, which may be a set of concepts identified in a domain, or what is represented by the relevant corpus of information.

**Minimal ontological commitment** refers to minimising the ontological commitment of an ontology to allow more freedom in an ontology's usage. Ontological commitment refers to an ontology being able to be agreed upon by users or ontology adopters. Studer et al. [1998] refer to adopters of an ontology as having agreed to *commit* to that ontology which describes some domain knowledge. Gruber [1993] defines ontological commitment as "*an agreement to use a vocabulary (i.e. ask queries and make assertions) in a way that is consistent (but not complete) with respect to the theory specified by an ontology*". Minimal ontological commitment is about not over-defining terms that may impede some potential users of the ontology. One way to minimise ontological commitment is to make "as few claims as possible about the world being modelled" [Gruber, 1995]. Another way is by breaking an ontology into separate ontologies [Gruber, 1995].

**Minimal encoding bias** An encoding bias occurs when "representational choices are made purely for the convenience of notation or implementation" [Gruber, 1995]. This should be minimised to prevent an ontology to be specific only to a particular ontology notation, language specification or implementation.

Some of these criteria can be successfully determined using ontology tools, such as reasoners like FaCT<sup>18</sup> and RACER<sup>19</sup>. Reasoners provide the means to verify the consistency of the set of defined concepts, relationships, and inferred relationships in ontologies, such as redundant terms, inconsistencies between definitions and missing definitions. Additionally, to examine the consistency in ontologies in the military domain, Dong et al. [2004] used

---

<sup>18</sup><http://www.cs.man.ac.uk/horrocks/FaCT>

<sup>19</sup><http://www.racer-systems.com>

existing software engineering modelling languages and tools. In particular, Dong et al. [2004] model ontologies written in DAML+OIL using the software engineering modelling language Z, and use the proof tool Z/EVES to check for consistency errors in those ontologies.

Some criteria are more challenging to evaluate as they may be hard to quantify. That is, they require manually inspecting the ontology, for example, correctness requires a domain expert or ontology engineer to manually verify that the definitions are correct with reference to the real world. This may not be a feasible task for a repository of ontologies. Other criteria, such as clarity and expandability, can be difficult to evaluate as there are no measures for them. Also, while the completeness of an ontology can be demonstrated, it cannot be proven.

#### 2.4.2 Ontology evaluation measures

*Ontology evaluation measures* are a quantitative means for assessing various aspects of an ontology. Gómez-Pérez [2001] outlines a list of measures looking at possible errors that could manifest with regard to ontology consistency, completeness and conciseness and we present these in Table 2.4.

Given an application and a text corpus that represents the knowledge in that domain, Brewster et al. [2004] presents some approaches to identify a suitable ontology from a given set of ontologies. They are: *counting the number of overlapping terms*, *vector space similarity measure*, *structural fit by clustering and mapping terms*, and using *conditional probability* to evaluate the ‘best fit’ of an ontology. Two of the above approaches involve ontology evaluation measures for analysing coverage over a given domain are presented in Table 2.5.

Guarino [2004] proposes measures of precision and coverage. Precision is a measure of correctness as it examines the overlap between what is modelled in the ontology and the intended domain being modelled as a proportion of what is modelled in the ontology itself. The measure defined as coverage by Guarino [2004] is used for Information Retrieval [Baeza-Yates and Ribeiro-Neto, 1999], but is referred to instead as recall. We adopt the term recall for this measure, as we refer to coverage as an ontology evaluation criterion. It can be used to examine which definitions in the ontology are not in the domain being modelled.

Recall is a measure of completeness as it examines the overlap between what is modelled

<b>Criteria</b>	<b>Measure(s)</b>	<b>Parameters</b>
<b>Consistency</b>	Circularity error	At distance 0 At distance 1 At distance N
	Partition error	Common instances in disjoint decompositions and partitions External classes in exhaustive decompositions and partitions External instances in exhaustive decompositions and partitions
	Semantic inconsistency error	
<b>Completeness</b>	Incomplete concept classification	
	Partition errors	Disjoint knowledge omission
	Partition errors	Exhaustive knowledge omission
<b>Conciseness</b>	Grammatical redundancy error	Redundancies of ‘subclass-of’ Redundancies of ‘instance-of’
	Identical formal definition	Classes
	Identical formal definition	Instances

Table 2.4: Measures for errors in ontology for consistency, completeness and conciseness proposed by Gómez-Pérez [2001]

<b>Criteria</b>	<b>Measure(s)</b>
<b>Coverage</b>	No. overlapping concepts
	Vector space similarity measure

Table 2.5: Measures for coverage of an ontology proposed by Brewster et al. [2004]

in the ontology and the intended domain being modelled as a proportion of what is modelled in the intended domain. Thus, it can be used to examine which definitions are deficient in the ontology.

Measures focusing on *structural aspects* of an ontology have also been proposed, in particular, the quality of its instantiation and how classes interact with its instances in the knowledge base. Gangemi et al. [2005] present a suite of measures focusing on the structure of an ontology and we present these measures in Table 2.6. Tartir et al. [2005] propose measures to evaluate an ontology’s capacity or “potential for knowledge representation” and we present these in Table 2.7.

### Detailed descriptions of selected ontology measures

Below we outline specific details for a selected set of ontology measures that will be used in this thesis.

There are differing formal definitions of ontologies proposed in literature [Haase et al., 2005; Guarino, 2004; Maedche and Staab, 2002; Sowa, 2000]. However, we adopt a simplified definition of an ontology with the following definitions to help describe the evaluation measures proposed in literature consistently:

- an ontology  $\mathcal{O} = \langle \mathcal{O}_c, \mathcal{O}_i, \mathcal{O}_r \rangle$  where  $\mathcal{O}_c$  is the set of concepts in an ontology,  $\mathcal{O}_i$  is the set of instances in an ontology,  $\mathcal{O}_r$  is the set of relationships in an ontology which is also the union of the set of relationships between concepts  $\mathcal{O}_{cr}$  and the set of relationships between instances  $\mathcal{O}_{ir}$ .
- frame of reference  $\mathcal{F} \in \mathcal{F}^*$  and  $\mathcal{F} = \langle \mathcal{F}_c, \mathcal{F}_i, \mathcal{F}_r \rangle$ , where  $\mathcal{F}^*$  is the set of all frames of reference,  $\mathcal{F}_c$  is the set of concepts in a frame of reference,  $\mathcal{F}_i$  is the set of instances in a frame of reference, and  $\mathcal{F}_r$  the set of relationships in a frame of reference which is the union between the set of relationships between concepts  $\mathcal{F}_{cr}$  and the set of relationships instances  $\mathcal{F}_{ir}$ . Examples of frames of reference as described by Gómez-Pérez [2001] are: the real world, a set of requirements, or a set of competency questions. A frame of reference can also be the set of concepts, instances and relationships of a particular domain.

Measure types	Measure(s)	Implementation
Structural measures	Depth	Absolute Average Maximal
	Breadth	Absolute Average Maximal
	Tangledness	
	Fanoutness	Absolute leaf cardinality Ratio of leaf fanoutness Weighted ratio of leaf fanoutness Maximal leaf fanoutness Absolute sibling cardinality Ratio of sibling fanoutness Weighted ratio of sibling fanoutness Average sibling fanoutness Maximal sibling fanoutness Average sibling fanoutness without metric space Average sibling fanoutness without list of values
	Differentia specifica	Ratio of sibling nodes featuring a shared differentia specifica Ratio of sibling nodes featuring a shared differentia specifica among elements
	Density	
	Modularity	Modularity rate Module overlapping rate
	Logical adequacy	Consistency ratio Generic complexity Anonymous classes ratio Cycle ratio Inverse relations ratio Individual/class ratio
	Meta-logical adequacy	Meta-consistency ratio
	Degree distribution	

Table 2.6: Structural measures proposed by Gangemi et al. [2005]

Measure Types	Measure(s)
<b>Schema measures</b>	Relationship richness Attribute richness Inheritance richness
<b>Instance measures</b>	Class richness Average population Cohesion Importance Fullness Inheritance richness Relationship richness Connectivity Readability

*Table 2.7: Measures proposed by Tartir et al. [2005]*

Measure Types	Measure(s)
<b>Coupling measures</b>	Number of external classes (NEC) Reference to external classes (REC) Referenced includes (RI)

*Table 2.8: Measures proposed by Orme et al. [2006]*



**Number of overlapping terms** This measure that Brewster et al. [2004] proposes identifies a count of the number of terms which overlap between a set of concepts in an ontology  $\mathcal{O}_c$ , and set of terms extracted from terms in a frame of reference  $\mathcal{F}_c$ , such as a text corpus in a given domain. The text corpus used is taken to represent a given domain that the ontology is being compared to.

$$\text{overlapping terms}(\mathcal{O}_c, \mathcal{F}_c) = | \text{terms}_{\mathcal{O}_c} \cap \text{terms}_{\mathcal{F}_c} | \quad (2.1)$$

where  $\text{terms}_{\mathcal{O}_c}$  is the set of terms in  $\mathcal{O}_c$  and  $\text{terms}_{\mathcal{F}_c}$  is the set of terms in  $\mathcal{F}_c$ .

This measure may also use parameters of  $\mathcal{O}_i$  for the set of instances for a given ontology. We present an additional equation for this measure below.

$$\text{overlapping terms}(\mathcal{O}_i, \mathcal{F}_i) = | \text{terms}_{\mathcal{O}_i} \cap \text{terms}_{\mathcal{F}_i} | \quad (2.2)$$

where  $\text{terms}_{\mathcal{O}_i}$  is the set of terms in  $\mathcal{O}_i$  and  $\text{terms}_{\mathcal{F}_i}$  is the set of terms in  $\mathcal{F}_i$ .

**Precision** Guarino [2004] proposes the precision measure, which is the percentage of a set of concepts in an ontology  $\mathcal{O}_c$  that overlaps with the intended model, that is a set of terms from a frame of reference  $\mathcal{F}_c$ . This measure is also used for information retrieval [Baeza-Yates and Ribeiro-Neto, 1999]. This is given by:

$$\text{precision}(\mathcal{O}_c, \mathcal{F}_c) = \frac{| \mathcal{O}_c \cap \mathcal{F}_c |}{| \mathcal{O}_c |} \quad (2.3)$$

This measure may also use parameters of  $\mathcal{O}_i$  and  $\mathcal{O}_r$  for the set of instances, and relationships between instances and concepts for a given ontology, respectively. We present equations for these measures below.

$$\text{precision}(\mathcal{O}_i, \mathcal{F}_i) = \frac{| \mathcal{O}_i \cap \mathcal{F}_i |}{| \mathcal{O}_i |} \quad (2.4)$$

$$\text{precision}(\mathcal{O}_r, \mathcal{F}_r) = \frac{| \mathcal{O}_r \cap \mathcal{F}_r |}{| \mathcal{O}_r |} \quad (2.5)$$

$$\text{precision}(\mathcal{O}_{cr}, \mathcal{F}_{cr}) = \frac{| \mathcal{O}_{cr} \cap \mathcal{F}_{cr} |}{| \mathcal{O}_{cr} |} \quad (2.6)$$

$$\textit{precision} (\mathcal{O}_{ir}, \mathcal{F}_{ir}) = \frac{|\mathcal{O}_{ir} \cap \mathcal{F}_{ir}|}{|\mathcal{O}_{ir}|} \quad (2.7)$$

**Recall** Guarino [2004] proposes the coverage measure, which we refer to in this thesis as recall. Recall is the percentage of the overlap between a set of terms from the domain  $\mathcal{D}$ , and a set of concepts in an ontology  $\mathcal{O}$ . This is given by:

$$\textit{recall} (\mathcal{O}_c, \mathcal{F}_c) = \frac{|\mathcal{O}_c \cap \mathcal{F}_c|}{|\mathcal{F}_c|} \quad (2.8)$$

This measure may also use parameters of  $\mathcal{O}_i$  and  $\mathcal{O}_r$  for the set of instances, and relationships between instances and concepts for a given ontology, respectively. We present equations for these measures below.

$$\textit{recall} (\mathcal{O}_i, \mathcal{F}_i) = \frac{|\mathcal{O}_i \cap \mathcal{F}_i|}{|\mathcal{F}_i|} \quad (2.9)$$

$$\textit{recall} (\mathcal{O}_r, \mathcal{F}_r) = \frac{|\mathcal{O}_r \cap \mathcal{F}_r|}{|\mathcal{F}_r|} \quad (2.10)$$

$$\textit{recall} (\mathcal{O}_{cr}, \mathcal{F}_{cr}) = \frac{|\mathcal{O}_{cr} \cap \mathcal{F}_{cr}|}{|\mathcal{F}_{cr}|} \quad (2.11)$$

$$\textit{recall} (\mathcal{O}_{ir}, \mathcal{F}_{ir}) = \frac{|\mathcal{O}_{ir} \cap \mathcal{F}_{ir}|}{|\mathcal{F}_{ir}|} \quad (2.12)$$

**Meta-consistency ratio** Gangemi et al. [2005] propose the meta-consistency ratio, which relates to both the consistency and correctness ontology evaluation criteria. The measure examines the consistency of concepts in the ontology by applying principles of formal semantics such as the meta-properties proposed in the OntoClean methodology [Guarino and Welty, 2002], which we discussed in Section 2.3.1. For example, a concept may be examined with regards to its rigidity metaproperty, to determine whether it is consistent with the rigidity metaproperty of the other concepts it is related to. A concept that has the rigidity metaproperty and is subsumed by another concept that has the rigidity metaproperty upholds the meta-consistency constraint. However, a rigid concept that is subsumed by

another concept that has an anti-rigid metaproperty would not uphold the meta-consistency constraint. In other words, meta-consistency is a semantic validation examining whether each concept is defined with the appropriate meta-properties in relation to other concepts within the ontology. This measure relates to correctness because the assumptions behind the associated relationships between concepts are clarified as examining meta-consistency involves an analysis of the meta-properties of concepts. As a result, incorrect relationships defined between concepts may be found. The measure examines the proportion of the concepts in the ontology that have meta-consistency with the total number of concepts in the ontology.

$$\text{ratio of meta-consistent concepts}(\mathcal{O}_c) = \frac{|\text{meta-consistent}(\mathcal{O}_c)|}{|\mathcal{O}_c|} \quad (2.13)$$

where  $\text{meta-consistent}(\mathcal{O}_c)$  is the set of concepts determined as meta-consistent.

**Tangledness** Gangemi et al. [2005] propose tangledness, which considers the proportion of multiple parents for each concept to the set of concepts in the ontology. This measure assumes multiple inheritance in an ontology.

$$t_{\text{Gangemi}}(\mathcal{O}_c) = \frac{|\mathcal{O}_c|}{|\{c \in \mathcal{O}_c : p(c) > 1\}|} \quad (2.14)$$

where  $p(c)$  is the number of parents for concept  $c \in \mathcal{O}_c$ , and  $\{c \in \mathcal{O}_c : p(c) > 1\}$  is the set of concepts that have more than one parent in an ontology.

We find this definition of tangledness  $t_{\text{Gangemi}}$  to be counter-intuitive. The measure of the tangledness of an ontology with the above definition ranges from 1, which denotes that each concept has multiple parents, to infinity to denote no tangledness. A more intuitive definition is simply the inverse of what is defined such that:

$$t(\mathcal{O}_c) = \frac{|\{c \in \mathcal{O}_c : p(c) > 1\}|}{|\mathcal{O}_c|} \quad (2.15)$$

where  $p(c)$  is the number of parents for concept  $c \in \mathcal{O}_c$ , and  $\{c \in \mathcal{O}_c : p(c) > 1\}$  is the set of concepts that have more than one parent in an ontology.

The revised tangledness measure  $t$ , ranges from 0 to 1, where 0 denotes no tangledness and 1 that each concept in the ontology has multiple parents.

**Consistency measures** Below, are the measures by Gómez-Pérez [2001] for the consistency criterion. They include measures of circularity errors and inconsistent classes. *Circularity error* is a measure of the consistency ontology evaluation criteria, that is, whether cycles occur in an ontology. This assumes a taxonomy where cycles are not allowed, which implies a tree. Gómez-Pérez [2001] propose three kinds of circularity errors and we present them below. See Figure 2.12 for examples of circularity errors.

$$\text{Circularity errors at distance of } 0 = \text{cycles}(\mathcal{O}, 0) \quad (2.16)$$

where  $\text{cycles}(\mathcal{O}, 0)$  is the number of cycles between a concept in an ontology with itself.

$$\text{Circularity errors at distance of } 1 = \text{cycles}(\mathcal{O}, 1) \quad (2.17)$$

where  $\text{cycles}(\mathcal{O}, 1)$  is the number of cycles between a concept and an adjacent concept.

$$\text{Circularity errors at distance of } n = \text{cycles}(\mathcal{O}, n) \quad (2.18)$$

where  $\text{cycles}(\mathcal{O}, n)$  is the number of cycles between a concept and another at  $n$  concepts away.

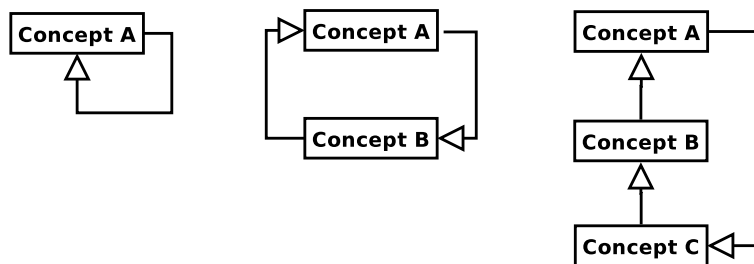


Figure 2.12: Examples of circularity error at a distance of 0, 1, and  $n = 3$

There is also a set of measures for *inconsistent classes* that Gómez-Pérez [2001] proposes, although sometimes, these inconsistencies are required. They are *subclasses with common classes*, and *classes with common instances*, which we present equations for below:

$$\begin{aligned} \text{subclasses with common classes}(\mathcal{O}) = |\{(c, d, e) : c \in \mathcal{O}_c, d \in \mathcal{O}_c, e \in \mathcal{O}_c \\ \wedge (d \neq e) \wedge (isa(c, d) \wedge isa(c, e))\}| \quad (2.19) \end{aligned}$$

where  $isa(c, d)$  is the direct subclass-of relation between concepts  $c$  and  $d$  in an ontology  $\mathcal{O}$ .

The measure of *subclasses with common classes* appears to be similar to the *tangledness* measure. Both measures involve the same set of classes, however, what is being counted is slightly different. Tangledness is the proportion of the number of classes in the ontology that have multiple parents compared to the total number of classes in the ontology. The *subclasses with common classes* measure is a count of the number of subclasses that have common classes.

$$\begin{aligned} \text{classes with common instances}(\mathcal{O}) = |\{(i, c, d) : i \in \mathcal{O}_i, c \in \mathcal{O}_c, d \in \mathcal{O}_c \\ \wedge (c \neq d) \wedge (instance\_of(i, c) \wedge instance\_of(i, d))\}| \quad (2.20) \end{aligned}$$

where  $instance\_of(i, c)$  is the instance-of relation between an instance  $i$  and a concept  $c$  in an ontology  $\mathcal{O}$ .

**Conciseness measures** Gómez-Pérez [2001] proposes some conciseness measures. They are grouped by measures for identical definitions and grammatical redundancy errors found in an ontology. The *measures for identical definitions* are:

$$\begin{aligned} \text{semantically identical concepts}(\mathcal{O}_c) = |\{(c, d) : c \in \mathcal{O}_c, d \in \mathcal{O}_c, c \neq d \\ \wedge \text{semantically\_identical}(c, d)\}| \quad (2.21) \end{aligned}$$

$$\begin{aligned} \text{semantically identical instances}(\mathcal{O}_i) = |\{(i, j) : i \in \mathcal{O}_i, j \in \mathcal{O}_i, i \neq j \\ \wedge \text{semantically\_identical}(i, j)\}| \quad (2.22) \end{aligned}$$

where *semantically\_identical*( $x, y$ ) is a function which detects whether two entities have the same formal definition but with different names. The only difference between two concepts or instances is the name used for each concept or instance, respectively.

The above measures may have a negative connotation and suggest that synonyms are not desirable in some ontologies. However, this is not always the case, there are ontologies which have a need for synonyms, such as medical ontologies like SNOMED. These measures are not appropriate for such ontologies.

A particular issue with the above measures is that semantically identical concepts and instances are not always so simple to determine. When dealing with ontologies developed using open world semantics such as the W3C OWL encoded ontologies, semantically identical concepts and instances may not be known or computable. Additionally, there may be rival definitions of particular concepts which also complicates the issue of determining semantically identical concepts and instances.

The following *measures for grammatical redundancy errors* relate to detecting multiple definitions of subclass-of and instance-of relations. They are:

$$\begin{aligned} & \textit{redundant\_relationships}(\mathcal{O}_c, \textit{isa}) \\ &= |\{(c, d) : c \in \mathcal{O}_c, d \in \mathcal{O}_c \wedge \textit{repeated\_relationships}(c, d, \textit{isa})\}| \quad (2.23) \end{aligned}$$

$$\begin{aligned} & \textit{redundant\_relationships}(\mathcal{O}, \textit{instance\_of}) \\ &= |\{(i, c) : \langle \mathcal{O}_c, \mathcal{O}_r, \mathcal{O}_i \rangle \in \mathcal{O}, i \in \mathcal{O}_i, c \in \mathcal{O}_c \\ & \quad \wedge \textit{repeated\_relationships}(i, c, \textit{instance\_of})\}| \quad (2.24) \end{aligned}$$

where *repeated\_relationships*( $c, d, \textit{isa}$ ) is function to detect subclass-of relationships that have been defined more than once between concepts  $c$  and  $d$ , and *repeated\_relationships*( $x, y, \textit{instance\_of}$ ) is function to detect the instance-of relations that have been defined more than once between instance  $i$  and concept  $c$  in  $\mathcal{O}$ .

**Completeness measures** Gómez-Pérez [2001] proposes completeness measures. These measures aim to determine the incompleteness of individual definitions in an attempt to subsequently determine whether the ontology as a whole is incomplete. The measures proposed by Gómez-Pérez [2001] are *Incomplete concept classification*, *Subclass partition omission* and *Exhaustive subclass partition omission* and we present them below.

$$\begin{aligned} & \textit{Incomplete concept classification}(\mathcal{O}_c, \mathcal{F}^*) \\ &= |\{c : \forall \langle \mathcal{F}_c, \mathcal{F}_i, \mathcal{F}_r \rangle \in \mathcal{F}^*, \exists (c \in \mathcal{F}_c) \wedge (c \notin \mathcal{O}_c)\}| \quad (2.25) \end{aligned}$$

that is, the number of subclasses for concept  $c$  that are not explicitly expressed as being disjoint.

This measure may also be applied for an ontology and compared to a point of view, that is, a frame of reference as shown in the equation below:

$$\textit{Incomplete concept classification}(\mathcal{O}_c, \mathcal{F}_c) = |\{c : (c \in \mathcal{F}_c) \wedge (c \notin \mathcal{O}_c)\}| \quad (2.26)$$

Incompleteness may also be deduced by examining the subclass partitions and the measures proposed for this are presented below.

$$\textit{Subclass partition omission}(\mathcal{O}_c) = |\{\forall c, d : \{subtree_c \cap subtree_d \neq 0\}\}| \quad (2.27)$$

that is, the number of subclasses for concepts  $c$  and  $d$  that are not explicitly expressed as being disjoint, where  $subtree_c$  and  $subtree_d$  are the set of all indirect and direct subclasses subclasses under concepts  $c$  and  $d$  respectively.

$$\begin{aligned} & \textit{Exhaustive subclass partition omission}(\mathcal{O}_c, \mathcal{F}^*) \\ &= |\{\forall \mathcal{F}_r : (\mathcal{F}_{cr} \subseteq \mathcal{F}_r \Rightarrow \exists \mathcal{O}_{cr} \subseteq \mathcal{O}_r) : (\mathcal{F}_{cr} - \mathcal{O}_{cr})\}| \quad (2.28) \end{aligned}$$

that is, cardinality of the set of subclass relationships omitted, where  $isa(\mathcal{F}_{cr})$  is the function that returns the set of all subclass relationships, where  $\langle \mathcal{F}_c, \mathcal{F}_i, \mathcal{F}_r \rangle \in \mathcal{F}^*$ .

In addition to the above measures, recall may also be used to measure completeness, by using it over all relevant frames of reference to obtain a measure of *average recall*. The range

of values for average recall is from 0, which denotes none of entities in any frame of reference is in the ontology, to 1 denoting that all entities in all relevant frames of reference is modelled in the ontology. We propose the measure of average recall for each of the components of an ontology below:

$$\text{average-recall}(\mathcal{O}_c, \mathcal{F}^*) = \frac{\sum_{\langle \mathcal{F}_c, \mathcal{F}_i, \mathcal{F}_r \rangle \in \mathcal{F}^*} \text{recall}(\mathcal{O}_c, \mathcal{F}_c)}{|\mathcal{F}^*|} \quad (2.29)$$

$$\text{average-recall}(\mathcal{O}_c, \mathcal{F}^*) = \frac{\sum_{\langle \mathcal{F}_c, \mathcal{F}_i, \mathcal{F}_r \rangle \in \mathcal{F}^*} \text{recall}(\mathcal{O}_c, \mathcal{F}_c)}{|\mathcal{F}^*|} \quad (2.30)$$

$$\text{average-recall}(\mathcal{O}_i, \mathcal{F}^*) = \frac{\sum_{\langle \mathcal{F}_c, \mathcal{F}_i, \mathcal{F}_r \rangle \in \mathcal{F}^*} \text{recall}(\mathcal{O}_i, \mathcal{F}_i)}{|\mathcal{F}^*|} \quad (2.31)$$

$$\text{average-recall}(\mathcal{O}_r, \mathcal{F}^*) = \frac{\sum_{\langle \mathcal{F}_c, \mathcal{F}_i, \mathcal{F}_r \rangle \in \mathcal{F}^*} \text{recall}(\mathcal{O}_r, \mathcal{F}_r)}{|\mathcal{F}^*|} \quad (2.32)$$

$$\text{average-recall}(\mathcal{O}_{cr}, \mathcal{F}^*) = \frac{\sum_{\langle \mathcal{F}_c, \mathcal{F}_i, \mathcal{F}_r \rangle \in \mathcal{F}^*, \mathcal{F}_{cr} \in \mathcal{F}_r} \text{recall}(\mathcal{O}_{cr}, \mathcal{F}_{cr})}{|\mathcal{F}^*|} \quad (2.33)$$

$$\text{average-recall}(\mathcal{O}_{ir}, \mathcal{F}^*) = \frac{\sum_{\langle \mathcal{F}_c, \mathcal{F}_i, \mathcal{F}_r \rangle \in \mathcal{F}^*, \mathcal{F}_{ir} \in \mathcal{F}_r} \text{recall}(\mathcal{O}_{ir}, \mathcal{F}_{ir})}{|\mathcal{F}^*|} \quad (2.34)$$

## Discussion

Despite having numerous definitions of measures available in literature, a remaining problem is that not all aspects of an ontology may be measurable. As with evaluation criteria, there may be parts of an ontology that are not measurable, for example, we cannot absolutely prove whether an ontology is complete. There may also be other aspects that are difficult to measure in an ontology, for example, determining adequate expandability.

Measures that are feasible but done in isolation are not as meaningful compared with measures put into the context of indicators and benchmarks from application requirements or needs. Pressman [2005] defines an *indicator* as:



“a metric or combination of metrics that provides insight into the software process, a software project or the product itself. An indicator provides insight that enables the project manager or software engineers to adjust the process, the project, or the product to make things better”.

Being able to measure something in an ontology does not necessarily mean it is of significance in the context of its usage. Some possible ontology evaluation measures in isolation are mere numbers, however, put into context of indicators and benchmarks, these measures can be meaningful, for example, comparing various ontologies for adequate coverage in a domain or performance measures in an application deployment. The coverage or performance measures taken in the context of the application give meaning to the measures taken.

Ontology evaluation measures that use a frame of reference can be problematic. If the frame of reference is fully specified, then there is no need to specify the ontology, thus nullifying the need for an evaluation measure. This can also be the case for recall, for if we know the answer, then there is no need for a query (from an Information Retrieval perspective). Thus, evaluation against a frame of reference does not seem operationalizable. However, in the real world, there remain cases that require the evaluation of newly created or existing ontologies. For example, how do we know whether the set of relationships in an ontology has been adequately defined? Thus, the need for measures which use some kind of comparison remains.

A possible solution to this problem is to approximate a frame of reference using various techniques such as natural language processing, and statistical text analysis. An example of a text analysis approach is proposed by Brewster et al. [2004] as discussed earlier in this chapter. Statistical text analysis is well researched in the information retrieval literature. Utilising some of these techniques, it may be possible to determine whether a concept in the ontology is equivalent to a concept in the frame of reference.

Another problem is that though there are aspects of existing research into ontology evaluation that relate to each other, often methods for ontology evaluation are discussed and implemented independently of others. For example, Tartir et al. [2005] propose measures for ontology evaluation, of which some measure identical aspects as measures found in Gangemi

et al. [2005]. Thus, there is overlapping work in this area. The rows in Table 2.9 represent measures proposed by different authors but are equivalent. Furthermore, with current ontology evaluation work, few examples are found outlining detailed examples of evaluation for real world data.

<b>Tartir et al. [2005]</b>	<b>Gangemi et al. [2005]</b>	<b>Gómez-Pérez [2001]</b>
Inheritance richness	Fanout	
Cohesion	Modularity	
	Logical adequacy	Consistency measures
	Meta-logical adequacy	Semantic inconsistency

Table 2.9: Equivalent measures proposed in different work found in literature (rows indicate the equivalent measures).

### 2.4.3 Validating measures

The field of software measurement matured as solutions to determining suitable measures, their validation, and usage in measurement processes became increasingly important in software engineering. It was in this environment, foundational ideas were proposed for the practice of software measurement. Fenton [1991] proposed a wide range of ideas including the use of empirical analysis. A criticism of Baker et al. [1990] at the time was that software metrics lacked “theoretical foundations”, which motivated their work in proposing a philosophy for software measurement. Also, Kitchenham et al. [1995] proposed a framework for software measurement validation.

Piattini et al. [2001] propose three general steps for metric definition and validation in the context of validating measures for data models used in Software Engineering. They are:

- 1. Metric definition:** Metrics are defined with clear goals.
- 2. Theoretical validation:** The proposed metric is analysed to confirm whether the “attribute being measured is reflected in the measurement”.
- 3. Empirical validation:** The method of validation here is by using the proposed metric

in a set of controlled experiments involving real data.

Figure 2.13 gives a diagram showing the steps described above.

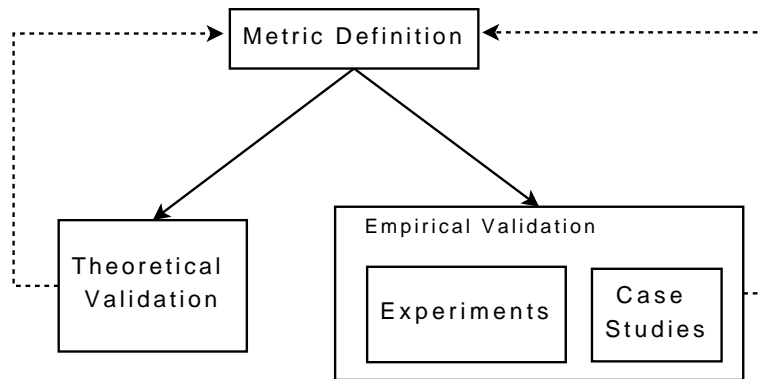


Figure 2.13: Steps for metric definition adapted from Piattini et al. [2001]

Kaner and Bond [2004] assert that software engineering presents an approach to measurement that “under-emphasizes measurement validity”, and that many metrics “do not capture the essence of whatever it is they are supposed to measure”. At the heart of the problem, software metrics regiments “don’t even recognize what attributes are supposedly being measured”. A framework is then proposed for a detailed analysis of software attributes to obtain a set of metrics where more meaningful data are produced.

Currently, ontology evaluation is experiencing a similar situation. There are currently many measures available, although with few accompanying validation analyses. Only recently has the practice of rigorous definition and validation of ontology measures been adopted [Yao et al., 2005; Orme et al., 2006; 2007]. Vrandecic and Sure [2007] also recognise the fact that measurement is important to ontology quality assurance. They go on to further state that metrics often consider structure rather than semantics. In that paper, ontology normalisation was introduced to align structural measures with semantic measures, in a bid to improve the accuracy of measurement.

## 2.5 Discussion

We discussed various approaches, methodologies, criteria and numerous measures that have been proposed in literature. Yet, there is no definitive methodology for evaluating ontologies. In particular, the OntoClean methodology presented in Section 2.3.1, does not focus on evaluating the ontology in with regard to requirements of an application. OntoMetric, which was introduced in Section 2.3.2, is a methodology for ontology selection but does not help with evaluating ontologies in the case of building new ontologies.

In considering existing research in ontology evaluation, it appears that some aspects are related, although this has not been explicitly noted in literature. From literature, there are two general approaches to ontology evaluation. One approach is to evaluate using some pre-determined requirements or competency questions, such as On-To-Knowledge, Methontology, and the formal method for ontology engineering proposed by Grüninger and Fox [1995]; Uschold and Grüninger [1996]. Another approach is to simply outline criteria and measures that are considered useful for evaluation, such as those proposed by Gruber [1995]; Gómez-Pérez [1996]; Hovy [2002]; Gómez-Pérez [2001]; Brewster et al. [2004]; Guarino [2004]; Gangemi et al. [2005]; Tartir et al. [2005].

Both approaches are valid and useful, however, by themselves, each of the above approaches have drawbacks. A requirements-based or competency-based evaluation without the use of specific measures, can be subjective, inconsistent and sometimes ambiguous as often, human judgements are relied upon for validation. On the other hand, without the analysis of ontology requirements of a suitable ontology for a given application, a criteria- and measures-based approach alone may lack significance and relevance to the ontology-driven application. What one application may consider an error or inconsistency, such as the presence of cycles, may not necessarily apply to another application.

Existing ontology engineering methodologies, such as the On-To-Knowledge, Methontology and On-To-Knowledge, both recognise that for ontologies that are used in applications, appropriate evaluation methods should arise from application requirements. Each application can have immensely varied needs and requirements. It follows then that measures deemed important for one application may not necessarily be important for others. Fur-

thermore, appropriate evaluation measures also vary with the given application. Therefore, requirements-based ontology evaluation incorporating a relevant set of measures is critical for determining the appropriate ontology evaluation activities that are concrete, relevant and meaningful. Unfortunately, both Methontology and On-To-Knowledge do not explore this aspect of evaluation. However, the GQM methodology that we presented in 2.3.3 may be useful in helping applications explore appropriate ontology evaluation methods based on their requirements. Thus, in the next chapter, we present a requirements-oriented methodology for ontology evaluation, which aims to address gaps in ontology evaluation with an adapted version of the GQM methodology.

It has been recognised in ontology evaluation that proposed measures need to be validated and thoroughly tested. Consequently, some work has begun recently which draws on the software measurement literature. Yao et al. [2005]; Orme et al. [2006; 2007] adopt the framework for metric validation of Kitchenham et al. [1995] in proposing and validating coupling measures for ontology evaluation. In Section 2.4.3, the importance of empirical validation was highlighted for software engineering and we contend that this is also needed for ontology evaluation. Thus we seek to explore the use of empirical means for validation in Chapter 6 of this thesis.

## 2.6 Summary

In this chapter, we presented a detailed description of ontologies, how they are specified, their role in applications and the possible levels of ontology granularity. We outlined several existing ontology engineering methodologies and examined their methods of evaluation, but found that the evaluation phases proposed in each had limitations. We also examined current evaluation methodologies for ontologies, but we concluded that they were also inadequate. Hence, there is currently no definitive methodology for evaluating ontologies. As such, there is a need for ontology evaluation that is based on the requirements of applications for identifying relevant measures to address each requirement.

Having recognised similarities between ontologies and software, this lead us to survey the literature on software measurement. We found that aspects of the GQM methodology

proposed in software engineering may be adapted to develop a new ontology evaluation methodology, and this is presented in the next chapter.

## Chapter 3

# The ROMEIO methodology

*O Romeo, Romeo! Wherefore art thou Romeo?*

— William Shakespeare

Ontology evaluation assesses ontologies with regard to a set of desirable qualities or ontology requirements, whether it is the process of ontology engineering or the task of ontology selection. It depends on the definition of measures for examining ontology qualities, as well as the correct use of these measures to address ontology requirements. These ontology requirements may be drawn from an analysis of desired competencies or qualities of an ontology. However, there is currently no suitable method to obtain appropriate mappings of these ontology requirements with related measures. A part of this problem is that a set of requirements for a given ontology may differ from one application to another. For example, a method to evaluate an ontology for one application based on a criterion, such as *minimal ontological commitment*, may not apply to another application, such as one requiring broad *coverage* of a particular domain. Thus, a methodology to determine an appropriate ontology evaluation method is needed for mapping a set of ontology requirements to suitable measures.

It is with this motivation that we propose a *Requirements Oriented Methodology for Evaluating Ontologies* (ROMEIO). The ROMEIO methodology maps ontology requirements of an application to appropriate measures using questions. *Requirements* describe specific competencies or qualities of a suitable ontology for a given application. *Questions* are then used to

determine if a given requirement has been met. Subsequently, for each question, one or more ontology evaluation *measures* are used to determine how the question has been answered, and hence if the requirement has been met. The product of the ROMEO methodology is a method of evaluation with mappings of requirements to questions and questions to measures.

In Section 3.1, we give an overview of the ROMEO methodology and elaborate on its components in the subsequent sections. We specifically consider *requirements* in Section 3.2, *questions* in Section 3.3 and *measures* in Section 3.4. We also discuss this methodology in Section 3.5 and compare it with existing methodologies.

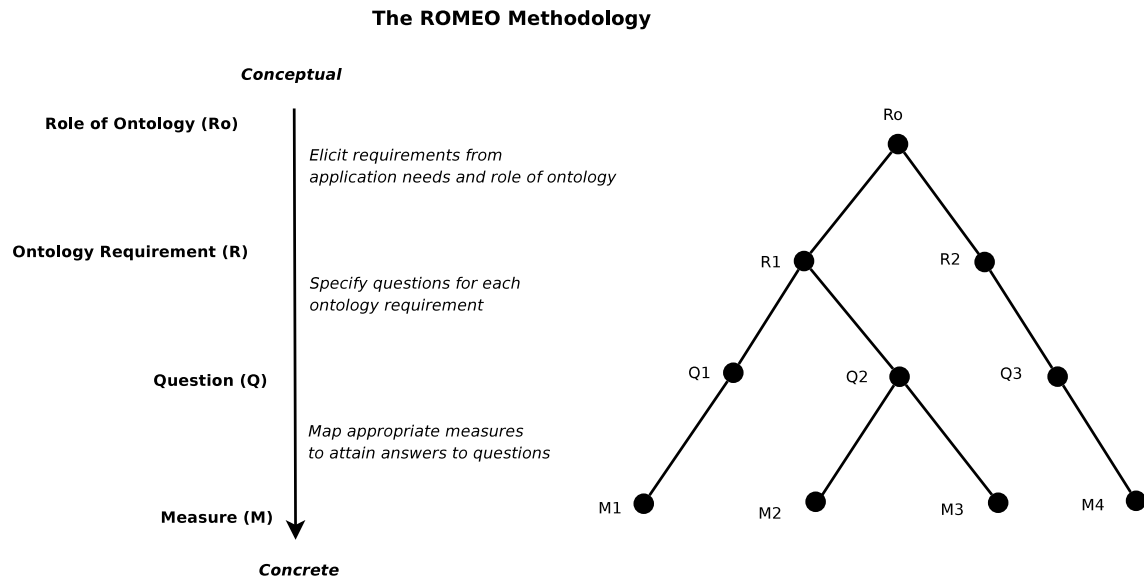
### 3.1 The ROMEO methodology

The ROMEO methodology establishes the relationship between a set of ontology requirements and a set of related measures used in ontology evaluation. ROMEO links these requirements to known ontology evaluation measures through a set of questions. Figure 3.1 shows the components involved in ROMEO, which begins from the intended set of *roles* of the ontology and links to the corresponding set of ontology *requirements* to its respective set of *questions* and *measures*. The *role* of the ontology describes the ways in which the ontology is used in an application. *Ontology requirements* specify the competencies, capabilities, functionalities and qualities that are needed from a suitable ontology for the given application. *Questions* relate requirements to measures, and may be based on aspects of an ontology evaluation criterion. *Measures* are quantifiable but not all measures are applicable to all questions, thus, appropriate measures are selected for each question.

### 3.2 Ontology requirements

An *ontology requirement* reflects a specific competency or quality that a given ontology must possess in the context of its role in the application. The process of defining a set of ontology requirements involves establishing the roles of the ontology, to establish how it is used in an application and this is discussed in Section 3.2.1. The roles of the ontology also help to distinguish applicable ontology requirements from other application requirements.





Ontology requirements may be drawn from relevant application requirements of ontology-driven applications, as some application requirements may be relevant in describing specific ontology requirements of the application. We present some guidelines for obtaining ontology requirements in Section 3.2.2.

To aid the specification of *ontology requirements*, a customised template is provided based on the *GQM goal description template*. We refer to it as the *requirements description template* and present it in Table 3.1. The use of the requirements description template helps to constrain the definition of ontology requirements into a standard specification. The template also allows the aspects of the requirement and its related information, such as, the context, users, and the motivation for the requirement to be elaborated upon. A description with the motivation for the ontology requirement should be included to specify its relevance to the evaluation of candidate ontologies for the given application.

### 3.2.1 Establishing the roles of the ontology

Defining the roles of an ontology helps to give an understanding of how the ontology is used in the context of an application. This step requires a brief description of the application

Table 3.1: Requirements description template

<b>Requirement:</b>	<i>Description of the requirement</i>
<b>Analyze:</b>	<i>Candidate ontologies being evaluated</i>
<b>For the purpose of:</b>	<i>Ontology evaluation, selection, or refinement</i>
<b>With respect to:</b>	<i>Quality or aspect being considered</i>
<b>From the viewpoint of:</b>	<i>Stakeholders and those affected parties from the adoption of the ontology</i>
<b>In the context of:</b>	<i>A specific ontology role</i>
<b>Motivation:</b>	<i>Description of the motivation for the requirements. Include reasons for the importance of the requirement and aspects being considered to the evaluation of candidate ontologies</i>

and the qualities of a suitable ontology, and a discussion of each role. Table 3.2 presents a recommended ROMEO template to use in explicitly stating the application, ontology and its roles.

Table 3.2: Ontology role template

<b>Application:</b>	<i>Brief description of the application</i>
<b>Ontology:</b>	<i>Ontology specificity, type of ontology of suitable ontologies</i>
<b>Role(s):</b>	<i>Functions that the ontology provides or enables in the application and its relevance</i>

By eliciting the roles of an ontology in the ontology-driven application, we can use them to determine an appropriate set of ontology requirements. The roles also help to decide whether certain requirements apply as there may be application requirements which are relevant but are not necessarily ontology requirements.

### Running example

In this chapter, we use a running example of an ontology-driven application in describing the ROMEO methodology – a search engine that utilises an ontology to enhance the performance of query expansion.

**Application:** For the search engine application, there are different techniques that may optimise search performance such as query expansion, indexing schemes, ranking schemes, and document processing techniques. An ontology may help with improving query expansion<sup>1</sup>. Andreou [2005] presents a specific implementation of Lucene for using ontologies to enhance query expansion based on relationships between concepts found in the Wordnet ontology.

**Role of the ontology:** In this search engine application, ontologies are used to help select candidate terms for query expansion. Concepts found in a given ontology that have a direct relationship to the matching terms in the initial query are incorporated in the final query. The search engine takes these sets of terms – both initial query terms and the final query terms, ranks documents in the collection in order of highest similarity, and presents them to the user. Figure 3.2 illustrates the search engine with the ontology driven query expansion module.

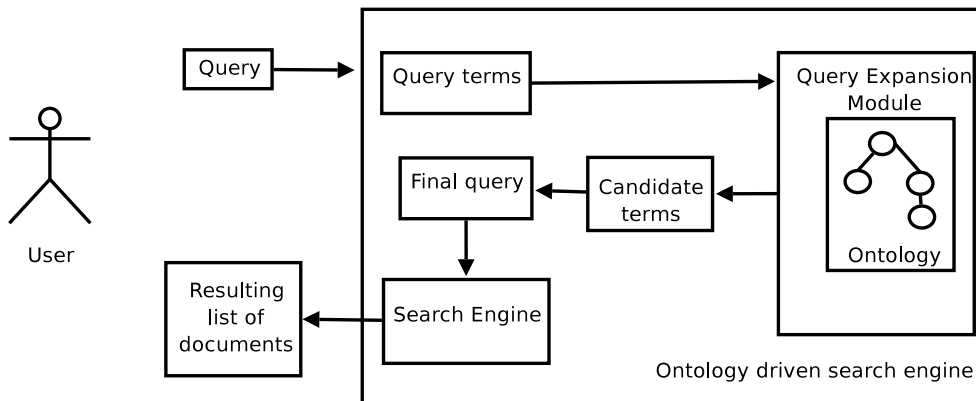


Figure 3.2: Ontology driven search engine

**Ontology:** Suitable ontologies for this search engine application are general ontologies, as they cover a broad range of subject areas in multiple domains and possess large vocabularies.

<sup>1</sup>Query expansion is when a given query to a search engine is taken as the *initial query* and relevant terms from a ranked list of candidate terms are added to be used as the *final query*.

Using the recommended ROMEO template, the role of the ontology for the running example is specified with respect to the application of a search engine which utilises ontologies for enhancing query expansion and is presented in Table 3.3.

*Table 3.3: Example application context and role of ontology*

<p><b>Application:</b> Search engine which relies on ontologies for query expansion</p> <p><b>Ontology:</b> General knowledge ontology which contains a large vocabulary and richly defined relationships.</p> <p><b>Role:</b> Boost effectiveness of query expansion module in selecting candidate terms for query expansion.</p>
--

### 3.2.2 Obtaining a set of ontology requirements.

A variety of sources may be used in obtaining the set of ontology requirements. We describe the following cases that may occur below:

**Case 1: Existing ontology requirements.** In the event that a set of *ontology requirements* exists, we adopt it as our ontology requirements.

**Case 2: Existing application requirements.** A set of ontology requirements may be derived from existing application requirements. There are, however, differences between ontology requirements and application requirements; application requirements are specific to a given application, whereas ontology requirements are specific for ontologies in the context of one or more ontology-driven applications. The set of application requirements may be examined to determine which requirement is affected by the content and the quality of the adopted ontology. There may be aspects of a given application requirement that are relevant with respect to how ontologies are used in the application. These application requirements may be drawn from interviews with application developers or from a review of the relevant documentation.

**Case 3: No application requirements or documentation exist.** For the case where application requirements have not been specified and documented, an analysis of the

application and its requirements may be necessary for the sole purpose of determining a set of ontology requirements.

Although, it is essential to the effectiveness of ROMEO, it is outside the scope of the ROMEO methodology to perform a complete requirements engineering analysis for the application. ROMEO assumes that ontology requirements have been elicited and validated beforehand and this methodology proceeds to determine relevant measures based on these requirements. However, there are established methods for eliciting and validating requirements [Nuseibeh and Easterbrook, 2000].

The guiding principle in determining what is included in the set of ontology requirements, is that requirements should outline specific *competencies* and *qualities* of a suitable ontology in the context of its role in a given application. The set of ontology requirements may vary greatly depending on the role of the ontology for a given application. Without a clear idea on the role of the ontology, it may be difficult to make decisions about which requirements apply.

### Running example

In the example of the search engine application, we will present existing application requirements as:

- ‘to efficiently index documents’,
- ‘to present relevant documents in effectively answering a given query’.

In this case, there may be relevant ontology requirements from the above application requirements. We use the role of the ontology to help determine whether a given application requirement is relevant for the use of ontologies. In this case, the role of the ontology is *for boosting effectiveness of query expansion in search engines*.

The first requirement of the *efficient indexing of documents*, does not affect the usage of the ontology, as ontologies are not used here to build an index; this requirement is thus excluded.

The second requirement of presenting relevant documents for a given query, is affected by the ontology used in this application to determine relevant documents. More specifically, the semantics defined in an ontology, help with the task of query expansion and consequently impact how relevant documents are determined. This requirement is thus applicable and is considered. However, as this is an application requirement, further refinement is needed for it to become an ontology requirement by outlining some competencies or ontology qualities.

For this example, the specific competency of the ontology is *having a rich definition of concepts and correct relationships defined*. The assumption here is that search engine tasks are general and may not be domain specific. However, in order to boost effectiveness of the query expansion component in the search engine, a suitable ontology needs to have an adequate range of concepts defined. In addition, a suitable ontology would also have definitions of relationships between concepts that are correct with respect to the world being modelled in order to correctly infer related concepts for query expansion. The set of ontology requirements for this application are *having a rich definition of concepts* and *having correct relationships between concepts*. The set of ontology requirements from our running example is given below in Table 3.4. and 3.5. The purpose of carrying out the ontology evaluation is clarified in each of those tables. For this example, the purpose is to select an appropriate ontology. Using the template, we also elaborate on the components of the application affected and the context, that is for search tasks carried out on a search engine. Lastly, we elaborate on the motivation for this ontology requirement.

### 3.3 Questions

After defining a set of ontology requirements to use for ontology evaluation, one or more questions are specified for each ontology requirement identified. Questions help to explore the various aspects of a given requirement and in turn provide a deeper understanding of the requirement.

A question is specified such that its associated requirement is considered to be satisfied when the question is answered; each question should consider aspects of a specific ontology quality, criterion or characteristic. The aim is to interpret a given requirement, formalise it

Table 3.4: Example: Ontology requirement 1

<b>Requirement OR1:</b> <i>Rich definition of concepts</i>	
<b>Analyze:</b>	Ontology
<b>For the purpose of:</b>	Selecting an ontology
<b>With respect to:</b>	Having rich definition of concepts
<b>From the viewpoint of:</b>	Query expansion component
<b>In the context of:</b>	Search Task
<b>Motivation:</b>	Given the search engine application requires the ontology to boost effectiveness of query expansion module, the most suitable ontology for use in query expansion in search engines would need to be rich in its definition of concepts (as the domain for search engines is general, however, in some cases may require specialised information).

Table 3.5: Example: Ontology requirement 2

<b>Requirement OR2:</b> <i>Correctness of relationships between concepts</i>	
<b>Analyze:</b>	Ontology
<b>For the purpose of:</b>	Selecting an ontology
<b>With respect to:</b>	Having correct relationships between concepts
<b>From the viewpoint of:</b>	Query expansion component
<b>In the context of:</b>	Search Task
<b>Motivation:</b>	A suitable ontology ought to have its definition of relationships between concepts correctly modelled with respect to the world being modelled to be able to correctly infer related concepts for query expansion.

into a set of questions and provide a basis for the inclusion of relevant measures in answering each question. With regard to the content of the question, we may base it on an aspect of an ontology evaluation criterion. Questions allow an ontology engineer to map specific aspects of an ontology evaluation criterion that relate directly to a given requirement, rather than trying to map a whole ontology evaluation criterion.

We provide the ROMEO methodology template for questions in Table 3.6. The template collects the set of questions for a given requirement and also prompts a discussion to justify

the included set of questions.

Table 3.6: Question template

<b>Questions for requirement:</b>	<i>Requirement description</i>
<b>Q1:</b>	<i>Question addressing the given requirement</i>
<b>Discussion:</b>	
	A brief discussion giving reasons why the questions are relevant to the requirement.

### 3.3.1 Criteria-questions

To aid the mapping of requirements to questions, we have listed some template questions for each criterion below in Tables 3.7, 3.8, 3.9 and 3.10, which we refer to as *criteria-questions*. These criteria-questions serve as a starting point for specifying questions. In specifying a set of questions for a particular ROMEO analysis, these criteria-questions may be adapted to suit a given ontology requirement.

In specifying questions, a new ontology evaluation criterion and associated questions may be encountered. As this list is not exhaustive, there may be situations where additional criteria-questions may be defined based on an aspect of an existing ontology evaluation criterion. Questions for a particular ROMEO analysis may also not necessarily be from this list of criteria-questions, and may not fit with any existing ontology evaluation criterion. Consequently, new ontology evaluation criteria and questions may be encountered.

#### Running example

The specific qualities of the ontology requirement OR1 and OR2 for the search engine application, are of a rich definition of concepts and correctness in the definition of relationships between concepts. We have specified questions for each ontology requirement.

The first question Q1 elaborates on the ontology requirement OR1 of adequate coverage of concepts. The search engine application may encounter diverse queries, thus requiring a



Table 3.7: Criteria-questions: clarity, consistency, and conciseness.

Criteria-question
<i>Clarity</i>
Does the ontology have concepts objectively defined? Does the ontology have instances objectively defined? Does the ontology have relationships between concepts objectively defined? Does the ontology have relationships between instances objectively defined?
<i>Consistency</i>
Does the ontology include two or more concepts that shares the same set of children? Does the ontology include two or more concepts that shares the same set of instances? How many cycles are found in the ontology? How many circularity errors are found in the ontology?
<i>Conciseness</i>
How many concepts are modelled with the same name? How many identical concepts are modelled using different names? How many instances are modelled with the same name? How many identical instances are modelled using different names? How many redundant subclass-of relationships are found in the ontology? How many redundant instance-of relationships are found in the ontology?

broad coverage of general knowledge. Q1 modifies the coverage criteria-question of “*Do the concepts in the ontology adequately cover concepts in the domain?*”, so that it accommodates a more general sense of coverage, rather than for a specific domain. Q1 is then adapted to become “*Do the concepts in the ontology adequately cover the set of concepts required for general knowledge?*”

The second question Q2, considers the other ontology requirement OR2 – the quality of its relationships. To be able to boost effectiveness of the query expansion component, the integrity of the existing definitions of relationships between concepts is important, that is, its correctness with respect to the world being modelled. Q2 uses the correctness criteria-questions of “*Does the ontology capture relationships between concepts of the domain correctly?*” Table 3.11 and 3.12 give the questions specified for our example application with a discussion using the ROMEO methodology template for questions.

Table 3.8: Mappings between criteria-question to measures: expandability and correctness

Criteria-question
<i>Expandability</i>
Do the set of concepts defined allow for future definitions of subclasses?
<i>Correctness</i>
Does the ontology capture concepts of the domain correctly?
Does the ontology capture instances of the domain correctly?
Does the ontology capture relationships between concepts of the domain correctly?
Does the ontology capture instance-of relationships of the domain correctly?
Does the ontology capture relationships between instances of the domain correctly?

Table 3.9: Criteria-questions: completeness and coverage

Criteria-question
<i>Completeness</i>
Does the ontology have concepts missing with regards to frame of reference?
Does the ontology have instances missing with regards to frame of reference?
Does the ontology have relationships between concepts missing with regards to frame of reference?
Does the ontology have relationships between instances missing with regards to frame of reference?
<i>Coverage</i>
Do the concepts in the ontology adequately cover the concepts in the domain?
Do the instances in the ontology adequately cover the instances in the domain?
Do the relationships between concepts in the ontology adequately cover the relationships between concepts in the domain?
Do the relationships between instances in the ontology adequately cover the relationships between instances in the domain?

Table 3.10: Criteria-questions: minimal ontological commitment

Criteria-question
<i>Minimal Ontological Commitment</i>
Does the ontology define any concepts that are overstated for the domain?
Does the ontology define any instances that are overstated for the domain?
Does the ontology define any relationships between concepts that are overstated for the domain?
Does the ontology define any relationships between instances that are overstated for the domain?

Table 3.11: Questions for OR1 in the example application

<b>Question for OR1: <i>Rich definition of concepts</i></b>	
<b>Q1:</b>	Do the concepts in the ontology adequately cover the set of concepts required for general knowledge?
<b>Discussion:</b>	
Queries may be diverse and cover a variety of topics; we specify Q1 to examine the <i>coverage</i> of concepts in a given ontology to its respective domain.	

Table 3.12: Questions for OR2 in the example application

<b>Question for OR2: <i>Correctness of relationships between concepts</i></b>	
<b>Q2:</b>	Does the ontology capture relationships between concepts of the domain correctly?
<b>Discussion:</b>	
For the query expansion technique to work effectively, the relationships between the concepts defined in the ontology need to be defined with assumptions that are as close as possible to real world assumptions; we use a question based on <i>correctness</i> for Q2.	

### 3.4 Measures

Measures seek to quantify and provide answers to the question with specific measurements. Each question is typically associated with one measure which is able to answer the question sufficiently. We provide the ROMEO methodology template for measures in Table 3.13. The template should be used to include details of the *possible range of values*, and an *optimal range of values* for the measure. In selecting appropriate measures, a *discussion* should be included to outline reasons for each measure used in the template. This discussion may also include a description of what the measure does and how it answers the associated question as well as include an accompanying example.

Table 3.13: Measures template

<b>Measures for question:</b> <i>Question</i>	
<b>M1:</b>	<i>Name of the measure</i>
	<b>Optimal value(s):</b> <i>Value indicating that the given question has been answered</i>
	<b>Range:</b> <i>Possible range of measurements</i>
<b>Discussion:</b>	
<i>A brief discussion on the reasons the set of measures listed apply to the given question.</i>	
<i>Also, a discussion on the optimal value and range expected with the set of measures listed.</i>	

### 3.4.1 Suggested mappings between criteria-questions and existing measures

To aid the ROMEO analysis for appropriate measures, we have provided a table of mappings between template questions and to a set of measures (where applicable). This provides a starting point for users of ROMEO in determining appropriate mappings to measures.

Table 3.14, presents mappings of criteria-questions to existing measures for the clarity, consistency and conciseness criteria. Clarity, introduced by Gruber [1995], currently has no measures for it in literature. It is up to the ontology engineer to determine whether this criterion has been met. The mappings of consistency and conciseness criteria-questions to the respective measures are proposed by Gómez-Pérez [2001]. For the conciseness criteria-questions, we have included the measure of precision proposed by Guarino [2004]. The reason for this is that precision measures how much of the concepts, instances and relationships in frame of reference is present in the ontology. Thus a concise ontology, which does not contain concepts, instances and relationships that are irrelevant, maps to an ontology that has high precision with respect to the frame of reference, that is, it does not model concepts, instances and relationships outside of a frame of reference.

Table 3.15, presents mappings of criteria-questions to relevant existing measures for the expandability and conciseness criteria. The expandability criteria-question relates to whether an ontology can be extended further to describe more fine-grain concepts and relationships

Table 3.14: Mappings between criteria-question to measures: clarity, consistency, and conciseness.

Criteria-question	Measure(s)
<i>Clarity</i>	
Does the ontology have concepts objectively defined?	–
Does the ontology have instances objectively defined?	–
Does the ontology have relationships between concepts objectively defined?	–
Does the ontology have relationships between instances objectively defined?	–
<i>Consistency</i>	
Does the ontology include two or more concepts that share the same set of children?	<i>subclasses with common classes</i> ( $\mathcal{O}$ )
Does the ontology include two or more concepts that share the same set of instances?	<i>classes with common instances</i> ( $\mathcal{O}$ )
How many cycles are found in the ontology?	<i>circularity errors at distance 0, 1, n</i>
How many circularity errors are found in the ontology?	<i>circularity errors at distance 0, 1, n</i>
<i>Conciseness</i>	
How many identical concepts are modelled using different names?	<i>semantically identical concepts</i> ( $\mathcal{O}_c$ )
How many identical instances are modelled using different names?	<i>semantically identical instances</i> ( $\mathcal{O}_i$ )
How many redundant subclass-of relationships are found in the ontology?	<i>redundant relationships</i> ( $\mathcal{O}_c, isa$ )
How many redundant instance-of relationships are found in the ontology?	<i>redundant relationships</i> ( $\mathcal{O}, instance\_of$ )
Does the ontology model concepts outside of the frame of reference?	<i>precision</i> ( $\mathcal{O}_c, \mathcal{F}_c$ )
Does the ontology model relationships outside of the frame of reference?	<i>precision</i> ( $\mathcal{O}_r, \mathcal{F}_r$ )
Does the ontology model instances outside of the frame of reference?	<i>precision</i> ( $\mathcal{O}_i, \mathcal{F}_i$ )

while maintaining the current definitions within the ontology. However, this question has no existing measures proposed for it and is left up to the experience and knowledge of an ontology engineer to determine. Correctness is about whether the right concepts, relationships and instances have been modelled according to the frame of reference or universe of discourse. We propose mappings of existing measures of the *ratio of metaconsistent concepts* and *precision* for the set of correctness criteria-questions. The measure of *ratio of metaconsistent concepts* applies to whether concepts are captured correctly for a given frame of reference, such as a domain. The measure of *precision* is also mapped as it is the measure of whether the concepts in an ontology has been captured according to the frame of reference. Thus *precision* is able to be applied to both conciseness and correctness.

Table 3.15: Mappings between criteria-question to measures: expandability and correctness

Criteria-question	Measure(s)
<i>Expandability</i>	
Do the set of concepts defined allow for future definitions of subclasses?	–
<i>Correctness</i>	
Does the ontology capture concepts of the domain correctly?	<i>ratio of meta-consistent concepts</i> ( $\mathcal{O}_c$ )
Does the ontology capture instances of the domain correctly?	<i>precision</i> ( $\mathcal{O}_c, \mathcal{F}_c$ ) <i>precision</i> ( $\mathcal{O}_i, \mathcal{F}_i$ )
Does the ontology capture relationships between concepts of the domain correctly?	<i>precision</i> ( $\mathcal{O}_{cr}, \mathcal{F}_{cr}$ )
Does the ontology capture relationships between instances of the domain correctly?	<i>precision</i> ( $\mathcal{O}_{ir}, \mathcal{F}_{ir}$ )

Table 3.16 present the mappings of criteria-questions of completeness, coverage and minimal ontological commitment. Gómez-Pérez [2001] proposes some mappings for completeness which we include in this table. However, we also include recall as a measure for the respective completeness criteria-questions. For the completeness criterion, relevant measures must account for the relevant frames of reference of the world being modelled to help determine whether an ontology is incomplete. For coverage, the same set of measures are used to ad-

dress the set of coverage criteria-questions. The exception is that it is only compared with a given frame of reference, which may be a domain. There are no existing measures applicable for minimal ontological commitment.

### Running example

In our running example, Q1 examines whether the coverage of concepts is adequately defined in the ontology compared to the general knowledge that is to be modelled. Coverage is not about completeness, but the adequacy with regard to the extent that an ontology models the domain and in this case, the extent of the set of concepts modelled in a general knowledge domain.

The specific measure we map to this question is the recall of the set of concepts in the ontology and a frame of reference and we use the recall measure defined in Equation 2.8. A frame of reference in this example refers to the general knowledge that needs to be modelled for query expansion. This measure is appropriate for coverage as it does not favour helps to determine whether an ontology includes a set of concepts which are relevant in the domain. The range of values is from 0 to 1. The optimal value of 1 indicates that the ontology adequately covers the frame of reference completely with relevant concepts. The value of 0 indicates that the ontology does not model any relevant concepts for the frame of reference. Using the ROMEO methodology template for measures, we give express this example in Table 3.17.

Q2 considers whether the relationships between concepts in the ontology have been correctly defined. The motivation for Q2 is for ontologies used in query expansion, in that the relationships between concepts must be correctly defined in order for the inferences which use these relationships to be effective. The measure of the *precision of the set of relationships between concepts in the ontology with the domain*, defined in Equation 2.3, is appropriate as it examines whether the relationships in the ontology have been defined correctly as compared to a representation of the relationships defined in the domain that the ontology models. The range of values is between 0 and 1. The optimal value is 1, indicating that the set of relationships defined in the ontology matches precisely with the domain being modelled. This

Table 3.16: Mappings between criteria-question to measures: completeness, coverage and minimal ontological commitment

Criteria-question	Measure(s)
<i>Completeness</i>	
Does the ontology have concepts missing with regard to the relevant frames of reference?	<i>Incomplete concept classification</i> ( $\mathcal{O}, \mathcal{F}^*$ ) <i>average-recall</i> ( $\mathcal{O}_c, \mathcal{F}^*$ )
Does the ontology have subclasses concepts missing from a given parent concept with regard to the relevant frames of reference?	<i>Ex. subclass partition omission</i> ( $\mathcal{O}_c, \mathcal{F}^*$ )
Does the ontology have instances missing with regard to the relevant frames of reference?	<i>average-recall</i> ( $\mathcal{O}_i, \mathcal{F}^*$ )
Does the ontology have relationships between concepts missing with regard to the relevant frames of reference?	<i>average-recall</i> ( $\mathcal{O}_{cr}, \mathcal{F}^*$ )
Does the ontology have relationships between instances missing with regard to the relevant frames of reference?	<i>average-recall</i> ( $\mathcal{O}_{ir}, \mathcal{F}^*$ )
<i>Coverage</i>	
Do the concepts in the ontology adequately cover the concepts in the domain?	<i>Incomplete concept classification</i> ( $\mathcal{O}, \mathcal{F}$ ) <i>recall</i> ( $\mathcal{O}_c, \mathcal{F}_c$ )
Do the instances in the ontology adequately cover the instances in the domain?	<i>recall</i> ( $\mathcal{O}_i, \mathcal{F}_i$ )
Do the relationships between concepts in the ontology adequately cover the relationships between concepts in the domain?	<i>recall</i> ( $\mathcal{O}_{cr}, \mathcal{F}_{cr}$ )
Do the relationships between instances in the ontology adequately cover the relationships between instances in the domain?	<i>recall</i> ( $\mathcal{O}_{ir}, \mathcal{F}_{ir}$ )
<i>Minimal Ontological Commitment</i>	
Does the ontology define any concepts that are overstated for the domain?	–
Does the ontology define any instances that are overstated for the domain?	–
Does the ontology define any relationships between concepts that are overstated for the domain?	–
Does the ontology define any relationships between instances that are overstated for the domain?	–



Table 3.17: Measures of Q1

<b>Measures for Q1:</b> Do the concepts in the ontology adequately cover the set of concepts required for general knowledge?	
<b>M1:</b>	<i>recall</i> ( $\mathcal{O}_c, \mathcal{F}_c$ )
	Optimal value(s): 1
	Range: 0–1
<b>Discussion:</b>	
We use a specific measure of coverage – the recall of concepts in the ontology $\mathcal{O}_c$ with concepts in the domain being considered $\mathcal{F}_c$ . This measure is important to examine whether the ontology is sufficiently detailed with regard to the domain. Values for the recall range from 0 and 1. The optimal value is 1 which indicates that it adequately covers the frame of reference completely with relevant concepts.	

measure helps to determine if there are any relationships between concepts in the ontology that should not be modelled with respect to the domain being modelled. Using the ROMEO methodology template for measures, we give express this example in Table 3.18.

### 3.5 Discussion

As we have seen in Sections 3.2, 3.3 and 3.4, ontology evaluation using ROMEO is driven by ontology requirements for an application. ROMEO seeks to associate them with relevant measures through a set of questions. The resulting product of ROMEO is a set of mappings between requirements to questions, and questions to measures. The set of ontology evaluation measures can be used as a basis for determining the suitability of an ontology for a given application. ROMEO is also a tool for ontology refinement as it is able to identify a set of applicable measures from ontology requirements and in turn, these measures can be used to measure aspects of the ontology that are deficient.

ROMEO is flexible and is able to accommodate the inclusion of additional measures should other measures be proposed in the future. As application requirements may change

Table 3.18: Measures of Q2

<b>Measures for Q2:</b> Does the ontology capture relationships between concepts of the domain correctly?	
<b>M2:</b>	<i>precision</i> ( $\mathcal{O}_{cr}, \mathcal{F}_{cr}$ ) Optimal value(s): 1 Range: 0–1
<b>Discussion:</b>	
<p>We use a specific measure of correctness – precision of the set of relationships between concepts <math>\mathcal{O}_{cr}</math> with respect to the frame of reference or domain being modelled <math>\mathcal{F}_{cr}</math>. Measurements of ‘precision of concepts’ measure gives the proportion of concepts defined in the ontology that overlaps with the concepts in a given domain. Precision values range from 0 (which indicates no concepts overlap) and 1 (every concept in the domain matches). This measure is important to ensure that the relationships defined in the ontology are correctly modelled, for query expansion to be applied appropriately.</p>	

over time, ROMEO allows for a systematic approach for reassessing which measures apply, thus, allowing for room to adapt or update the appropriate set of measures as more applicable measures are proposed.

ROMEO allows for the reuse of mappings in a ROMEO analysis. In ROMEO, the individual mappings that are determined between requirements to questions (which may be based on existing ontology evaluation criteria), and questions to ontology evaluation measures may be reused from one application to another. If there are requirements that are common between different applications, the resulting question ought to be similar, if not the same. However, mappings of questions to measures are more reusable as they are decoupled from a given application. In Chapter 6, we consider empirical validation experiments of the mappings between questions and measures in addition to the ROMEO methodology.

ROMEO is adapted from the GQM methodology. Although, the use of GQM as the basis of determining appropriate measures is similar to the application of GQM to knowledge bases proposed by Lethbridge [1998], ROMEO is used for the content-based evaluation of ontologies

of an application. Lethbridge [1998] uses GQM to derive a set of measures for knowledge bases according to general tasks in knowledge engineering. ROMEO differs in that it adapts GQM for ontology engineers to associate requirements with already existing measures using questions for determining an appropriate ontology evaluation method for the ontology-driven application.

### Comparison between GQM, OntoMetric and ROMEO

GQM, OntoMetric and ROMEO are all methodologies that rely on the analysis performed by a person. This gives rise to human error or bias and may result in inappropriate evaluation. In the case of ROMEO, there may be cases where an incorrect interpretation of application requirements has occurred, or where the ontology engineer has an incorrect understanding of the effect of the measures to apply. However, ROMEO and GQM provide a framework for the users to justify each action taken as analysis occurs at each stage and thus invoking sensible reasoning. In the case of ROMEO, the ontology engineer can review and correct the decisions made at each step.

The main difference between GQM and ROMEO is that, while GQM is a general methodology for process improvement, ROMEO is a methodology for ontology evaluation. The GQM methodology is driven by a prior analysis of *goals*, while the ROMEO methodology is driven by a prior analysis of requirements. *Goals* in GQM and *requirements* in ROMEO are used in the respective methodologies to determine which evaluations to carry out. *Questions* in ROMEO are also used in a similar way as it is in GQM. A question in GQM defines a specific aspect that is relevant for measurement; a question in ROMEO reflects a specific ontology quality or aspect of an ontology criterion that is relevant for ontology evaluation. Questions may be derived from the suggested list of criteria-questions. Measures in ROMEO and metrics in GQM are equivalent, although in ROMEO, there is a list of existing ontology evaluation measures to choose from. Figure 3.3 illustrates how ROMEO is similar to GQM.

Both OntoMetric and ROMEO adopt a top-down approach for determining appropriate ontology evaluation measures. Referred to as *dimensions* [Lozano-Tello and Gómez-Pérez, 2004], the OntoMetric methodology considers a wider range of issues for ontology selection,

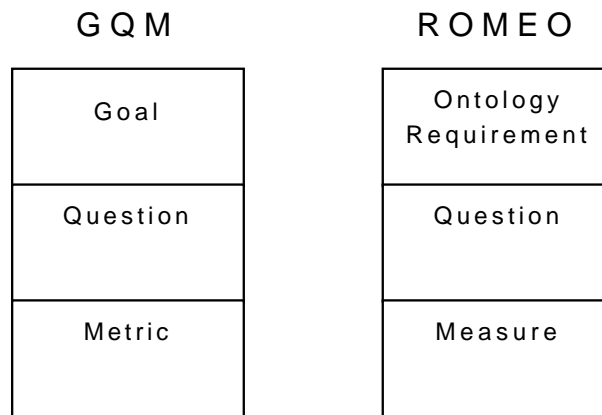


Figure 3.3: GQM and ROMEO

such as the dimension of language used to build the ontology, and the dimension of the cost of utilising a given ontology. However, a similarity that both the ROMEO methodology and OntoMetric methodology share is that they both consider the evaluation of the content of a given ontology.

In OntoMetric, ontology evaluation is carried out by having an *objective* formulated, various *criteria* are associated with each objective and through a series of steps, a set of *measures* of the relevant *characteristics* is determined. Using the analysis of each objective, we begin with the multilevel tree of characteristics (MTC), and from it a decision tree is formed by pruning and customising it. The MTC has a complete set of dimensions, criteria, and characteristics. Each candidate ontology is then compared using this decision tree. The top-down approach of breaking down the more conceptual *objectives* into specific *characteristics* and *measures* is similar to how ROMEO considers requirements and its relevant measures.

The difference between OntoMetric and ROMEO, however, is in the way ontology criteria, characteristics and qualities are associated with specific measures. In ROMEO, this is achieved through the specification of *questions*. Whereas in OntoMetric, an entire *criterion* is associated with an *objective*, and also requires a complete understanding of the criterion to apply it effectively. We have found that the use of questions offers a more flexible way to determine relevant ontology qualities or characteristics. For example, a *criteria-question* may be used initially and then customised for use in that specific requirement. Questions are

also not limited by the set of criteria-questions available. Users of ROMEO may specify a question outside of the existing set of criteria-questions which suits a particular requirement.

OntoMetric considers many issues with regard to the adoption of an ontology, such as cost, and specification language and tools used. ROMEO, however, is a methodology for evaluating the *content* of an ontology. We argue that content will remain an important part of evaluation whereas other aspects such as ontology languages used may not necessarily be as important as some of them may become widely used, such as OWL as a common ontology specification language.

Regarding the use of measures, OntoMetric adopts a *linguistic scale* for each measure considered. In comparison, the ROMEO methodology associates relevant measures to each question identified. The measures are used to carry out quantitative measurements and collect data for answering a question. Thus they help to determine whether a given requirement is met. Linguistic scales in OntoMetric are assigned by the user to designate values of an ontology characteristic for a candidate ontology (refer to Section 2.3.2). Therefore, in this context, the quantitative measurements used in a ROMEO analysis are better than the use of linguistic scales proposed by OntoMetric as the quantitative measurements are less subjective than linguistic scales.

Table 3.19 summarises our discussion on the comparisons between GQM, OntoMetric and ROMEO.

### 3.6 Summary

In this chapter, we have proposed an ontology evaluation methodology – ROMEO – which uses high level requirements from an application to determine the appropriate evaluation method to apply. ROMEO is a way of addressing requirements (which may be difficult to measure) with a set of specific and focused evaluation measures (which are quantitative and measurable). These requirements are used to determine a set of questions which may be selected from a list of ontology evaluation criteria-based questions. In turn, a focused set of measures is selected to answer these questions. We have also compared the ROMEO methodology with other methodologies such as GQM and OntoMetric and distinguished

<b>Comparison</b>	<b>GQM</b>	<b>OntoMetric</b>	<b>ROMEO</b>
Analysis relies on human judgements	•	•	•
Determines evaluation using goals/objectives	•	•	•
Customises appropriate evaluation method for an application	•	•	•
Incorporates existing ontology evaluation criteria and measures		•	•
Use in ontology selection		•	•
Ontology evaluation based on content		•	•
Uses quantitative measures	•		•
Use in ontology refining			•
Use in building ontologies			•

*Table 3.19: Comparison between evaluation methodologies*

ROMEO as a content-based ontology evaluation methodology. In the following chapters, we present case studies, which we apply the ROMEO methodology on. We will consider Lonely Planet and Wikipedia as applications which utilise ontologies and present ontology evaluation using the ROMEO methodology based on their ontology requirements.

## Chapter 4

# Lonely Planet

*It is not down in any map; true places never are.*

— Herman Melville

Lonely Planet<sup>1</sup> (LP) is an organisation that publishes travel media. It began with the publication of a single travel guidebook, *Across Asia on the Cheap*, and has expanded into what is today a large travel publication and media company. Regarding their guidebooks, around 650 books have been published concerning travel in various geographic locations around the world. However, as the number of publications increased, issues of content management arose, for example, the issue of consistency in vocabulary between content in books. Additionally, the issues arising from the process and the scale of publication of travel content brought about the need for ontologies. Furthermore, problems were encountered regarding building appropriate ontologies and the reuse of ontologies.

In this chapter, we consider Lonely Planet as a case study and apply the ROMEO methodology for determining an appropriate method for the evaluation of suitable ontologies for their application. We introduce the application in Section 4.1, examine issues that arise from it in more detail, and seek to understand the role of ontologies for this application. We then apply ROMEO to obtain an appropriate method of evaluating ontologies for this application. In Section 4.2, we elicit a set of ontology requirements from the discussion of the specific

---

<sup>1</sup><http://www.lonelyplanet.com>

role of ontologies in the application. In Section 4.3, we obtain ROMEO questions for each ontology requirement. A set of appropriate measures for each of these questions is presented in Section 4.4. The outcome of our analysis using ROMEO for the Lonely Planet case study is presented in Appendix B.<sup>2</sup>

#### 4.1 Content management for Lonely Planet

Lonely Planet produces a number of travel publications, media shows and digital content. Firstly, their publications include books that cover a range of popular travel destinations around the world. Secondly, Lonely Planet also produces travel programmes made for TV and DVD. Thirdly, Lonely Planet has an online website with limited travel content available publicly, discussion forums for users to post questions and an online bookstore where both print and digital content may be purchased.

With such a large operation, issues regarding the content management of Lonely Planet's travel information emerged, which motivated the need for ontologies. For this case study, we conducted a number of interviews with the Lonely Planet staff who were involved in the development of ontologies for Lonely Planet. The purpose of the interviews was to gain an understanding of the relevant issues for this application. These issues include achieving consistency between books, modelling travel information adequately and resolving issues between printed content for digital media.

This section describes the application of content management for Lonely Planet publications and the role of ontologies. Sections 4.1.1 and 4.1.2 describe travel guidebooks and digital publications, and we discuss some issues that arise in relation to the authoring and management of travel content. Also, in a preliminary investigation of off-the-shelf ontologies, further issues were raised regarding their suitability, which we present these in Section 4.1.3. We consider the roles of ontologies for this application in Section 4.1.4.

---

<sup>2</sup>Note: this chapter presents work that has been adapted from the original analysis from Yu et al. [2005b] in light of the ROMEO methodology.



#### 4.1.1 Travel guidebooks and their issues

Of the three publications described earlier, Lonely Planet's travel books are the most popular, in particular, the guidebooks. There are different types of guidebooks for travellers seeking information about continents, countries, or cities. For continents, there are four *Shoestring guides* that cater for large trips around a given continent, for example, the continent of Asia. Table 4.1 gives the number of publications for available for each continent. As can be seen in Table 4.1, for each continent, guidebooks vary in their scope of coverage and degree of detail ranging from the country level to the city level. There are also guidebooks that combine nearby countries, which is called a *Multi-country guide*, for example, the *Western Europe travel guide*. Regarding guidebooks about cities, there are about 40 *City guides*, 21 *Encounter* books and over 20 *Best-of city guides*. *City guides* give comprehensive details about a given city, whereas the latter two kinds of books give a brief overview of a city and cater for very short trips. In addition to the guidebooks described above, there are also over 130 book titles for general travel books which cover various topical travel information as well as activity guides, maps and pictorials. These books are not necessarily structured according to a country or city.

Table 4.1: Lonely Planet guidebook publications

Continent	Country	Multi-Country	City	Encounter	Best-of
Europe	42	21	18	10	15
Americas	26	18	14	5	1
Asia	17	4	7	6	3
Africa and Middle East	23	5	3	4	1
India and Central Asia	6	7	-	-	-
Australia and Pacific	10	8	4	1	1

Source: <http://shop.lonelyplanet.com> (Accessed January 2008)

With the broad and increasing range of travel guidebooks produced by Lonely Planet, many issues arise with regard to content management, such as consistency between books, managing updates, and the process of authoring. We highlight these issues in the application of content management of Lonely Planet guidebooks below.

### **Achieving consistent vocabulary**

The current process of book authoring and editing is such, that for a given publication, there may be many people involved. A particular problem is that people may be inconsistent with their use of terms and names. An example of this is the use of “*Things to do*” and “*Out and about*”, found in multiple Lonely Planet books to refer to the same activity. An example of the inconsistent use of names of places include, Mumbai and Bombay (although in general Mumbai is preferred over Bombay). However, the choice of terms used may be important, especially when locating places. Thus, the issue here is in achieving a consistent use of terms across publications, that is, *a shared and consistent shared vocabulary*.

### **Achieving consistent book structure**

Different authors also structure books differently and independently of each other. They may also place different emphasis on a given set of topics over others. Reasons for a variation in structure may arise from the nature of the places themselves. Some places are associated with specific travel activities that travellers tend to be more interested in, such as shopping in Singapore. Other places have general travel activities associated with it, such as travelling in France. Consequently, this is reflected in what is emphasised in each individual guidebook. What is needed is a *consistent structure for the guidebooks*, such that readers and viewers of Lonely Planet media do not have their experience impeded going from one book to another.

### **Achieving consistent content across guidebooks**

Content in a guidebook may depend on the intended audience of a given guidebook. Some guidebooks, such as the Encounter and Best-of guidebooks, are written for travellers who have shorter trips. Other guidebooks, such as country and city guidebooks, are written for travellers who spend more time in that particular country or city.

As more books are published, content management across the guidebooks becomes more complex. Descriptions of a particular geographic location may be found in many different books, for example, descriptions about the city of Paris can be found in at least five different series of guidebooks:

- *France Travel Guide* (January 2003)
- *Western Europe Travel Guide* (January 2003)
- *Paris City Guide* (October 2002)
- *Europe “On a shoestring” Guide* (February 2002)
- *Paris Condensed* (February 2002)

Furthermore, the descriptions of a location in a particular book may overlap with descriptions in other guidebooks. For example, partial descriptions about Paris and some of its restaurants are found in each of the books listed above.

Because of the overlapping nature of the guidebooks, the need here is in being able to reuse descriptions found in a given book and adapt it for a different book with related content. An example of this the reuse of a set of museum descriptions in the Paris City guidebook for inclusion in another book concerning that city, such as the France Country guidebook. Thus, the issue here is of *consistency of travel content across guidebooks*.

An aspect of this issue is that there may be conflicts encountered between updates. That is, as each book is updated, other books may be affected and require to be updated in subsequent revisions to maintain consistency across the collection. Generally, a given travel guidebook and its revised editions tend to have an asynchronous update schedule, that is, each book may be researched, written (or updated) and published on different dates. Some guidebooks are revised with shorter update cycles due to their popularity, such as the Paris City guide (which is revised and updated every two years).

#### 4.1.2 Digital content and its issues

Digital media, such as online publications and download-able travel information, may leverage content from the printed guidebooks. An example of this is the Lonely Planet online website that provides brief information about travel destinations, which has content included from the printed guidebooks. Other examples are guides for digital devices, such as selected book chapters in PDF format, PlayStation Portable device (PSP) edition of travel guides and digital maps. However, these digital publications are not as extensive as the published

media. Table 4.2 lists these digital travel media.

Table 4.2: *Lonely Planet digital travel content*

<b>Buy per chapter</b>		<b>Digital guides</b>	
USA	22	Audio (language learning)	3
South America	17	Guides for PSP	6
Mexico	6	City breaks (urban guide)	8
Central America	8	Maps	8
Caribbean	3		
Canada	4		

Source: <http://shop.lonelyplanet.com> (Accessed January 2008)

Digital services, such as products for mobile devices (such as PSP and Palm devices) and the Lonely Planet website, impose a constraint for up-to-date information. An example of such expectations is regarding the delivery of addresses and opening hours of accommodations, restaurants and entertainment venues to a traveller's mobile device. Thus, the issue with digital services here is of *ensuring information is up-to-date*. The problem is, that information used in digital services is leveraged from guidebooks, which may include details that are out of date. That is, there may be an 18-month gap between the gathering of updated information to the publication dates of subsequent updated editions of a particular guidebook, for example, the listings of restaurants, which may be due to the rate of change and development of a given place, especially countries in Asia. With print media, expectations for guidebooks are more lenient, as readers may be aware of the above and may be willing to accommodate for outdated information. Therefore, some information from printed guidebooks may not be entirely suitable to drive some digital services.

### 4.1.3 Previous experience of Lonely Planet in reusing ontologies

Preliminary investigations of candidate ontologies in the geographic domain had specific issues. Developers at Lonely Planet sought ontologies to aid the application of content management of travel information with some of the issues described above. Although, many

geographical ontologies are available, the problem that the developers at Lonely Planet encountered was finding a suitable ontology for the application. Specifically, the problems regarding the use of existing ontologies were the lack of appropriate representation of geographic places and an inadequate level of granularity. We will discuss these below.

### **Appropriate representation of geographic places**

As discussed above, there is the need for an ontology to have appropriate representation of geographic places. This need springs from having to accommodate imprecise relations present when modelling geography, specifically, the relationships between places. This especially includes relationships that are vague, where it is difficult to define a boundary. Thus, this requires the flexible classification of geographic items.

Existing taxonomies that Lonely Planet considered for reuse in the travel domain were found to be inadequate with regard to the geographical relationships that were modelled. In particular, they strictly defined places in a logical and rigid manner. However, there are cases where it can be difficult to define items due to the imprecise nature of the definitions themselves. A typical way these taxonomies classify places is that they split up a geographic location to distinguish one region from another, for example, separating the city of Paris in France from the surrounding regions according to the boundaries defined in a map. However, some places may possess relationships that are difficult to specify. We will use two examples to illustrate this below.

**Example: Paris** In the city of *Paris* in France, there are strict boundaries drawn in maps that define the tourism region (see Figure 4.1 for the map of Paris). However, often people loosely associate places in the surrounding regions with Paris, for example, La Défense is often associated with Paris although it is not part of the city of Paris; it is in a nearby region. Travellers may be interested in La Défense during their visits to Paris, although this does not correspond with the classification in some existing taxonomies and models of the city. As such, there is a requirement in this domain for flexibility in classifying places and for accommodating these *fuzzy boundaries*.

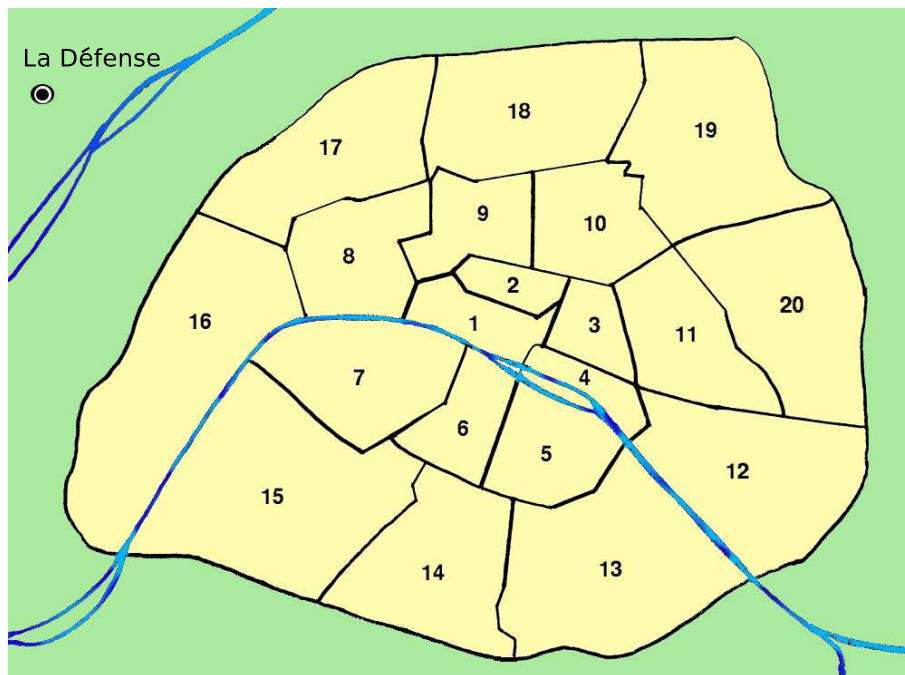


Figure 4.1: Map of the districts in Paris and La Défense

**Example: Lake Tahoe** *Lake Tahoe* is located on the border between the states of *California* and *Nevada* in the United States. Figure 4.2 shows a map of *Lake Tahoe* and where the state borders are located.

In modelling *Lake Tahoe*, some geographical thesauri, taxonomies and gazetteers, do not capture the right set of geographical relationships that are involved. *Lake Tahoe* crosses the state border lines of *California* and *Nevada*. However, in an example found in the Alexandria Digital Library Gazetteer [Alexandria Digital Library Gazetteer], *Lake Tahoe* is modelled under the state of *California* in the *Placer County*. It does not model *Lake Tahoe* as being part of *Nevada* other than modelling the city of *Stateline* in *Nevada* as part of the *South Lake Tahoe California* topographic map. Consequently, in producing travel information, locations around *Lake Tahoe*, may be difficult to classify.

The problem is that some places have imprecise relationships and we may want to be able to accommodate. In the specific case of *Lake Tahoe*, it may be important to do so as both

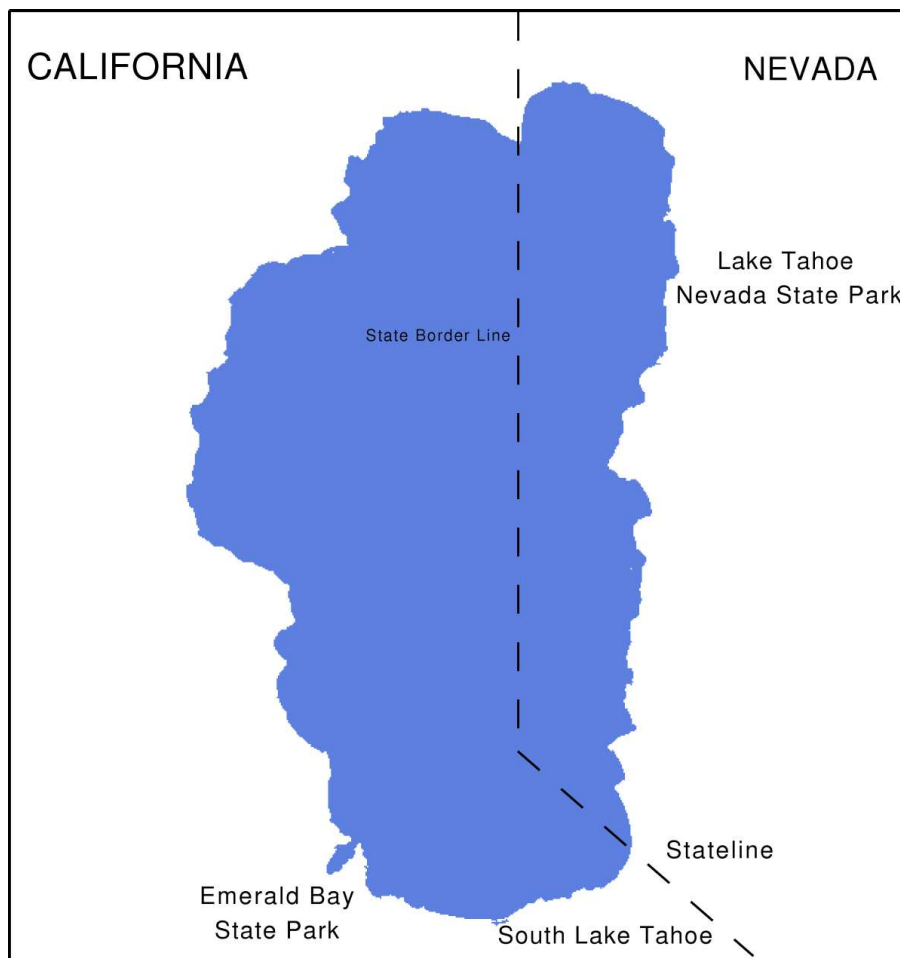


Figure 4.2: Map of Lake Tahoe in the United States

sides of the lake may have relevant places of interest. For example, the *Lake Tahoe Nevada State Park*, which is located in *Nevada*. However, another place of interest around Lake Tahoe is the *Emerald Bay State Park*, which is located in *California*. For a book about *California*, if we were to strictly classify places according to the book content, it may be difficult to include *Lake Tahoe Nevada State Park*, although it is just across the lake. Conversely, for a book about Nevada, it may be difficult to include *Emerald Bay State Park*.<sup>3</sup>

<sup>3</sup>As a side note, the Lonely Planet guide on Reno in Nevada includes Emerald Bay State Park in California as a related attraction.

### Right level of content granularity

We now consider the issue of content granularity. Geographic areas are able to be modelled with great level of detail with geographic tools such as Global Positioning Systems (GPS). However, such a level of detail may not be required in the travel domain. Users of the Lonely Planet travel content are predominantly travellers or tourists and may only be interested in information relevant to travel, such as descriptions of airports, reputable accommodation, places of interest, shopping areas and places to eat.

Developers at Lonely Planet found that off-the-shelf ontologies either had too much detail or had not modelled enough concepts. For example, the *Getty Thesaurus of Geographic Names* (TGN) [TGN], lacked detail with regard to the modelling of geographic relationships such as geographic relationships regarding *Lake Tahoe*. TGN is a structured vocabulary of geographic names for indexing art and architecture, and in particular, classifies Lake Tahoe under the level of its associated country, the United States. TGN neither classifies *Lake Tahoe* to be in *California* or in *Nevada*. However, TGN models *Stateline* as a place under the county of *Douglas* in the state of *Nevada* in the country of United States, and it has included *Lake Tahoe* as an alternate name for *Stateline*. Thus, the developers at Lonely Planet had to either remove irrelevant concepts and relationships or model additional concepts and relationships from such ontologies for it to be suitable.

The problem described above is determining the right level of granularity required. If the information is too coarse-grained, relevant information may be omitted. Conversely, if it is too fine-grained, some information may end up unused and effort would have been wasted in modelling that information and removing it. Thus, there is a need to capture the *appropriate* level of granularity in the travel domain.

#### 4.1.4 Roles of the suitable ontology

From the issues presented in Sections 4.1.1, 4.1.2 and 4.1.3, we observe various issues arising from the application of content management in Lonely Planet. There are roles that ontologies are able to play to address the issues highlighted above. However, among those issues lie application needs that may not apply to the use of ontologies and we distinguish them below.



In Section 4.1.1, an issue was raised with regard to the authoring and editing of guidebooks, that is, the issue of agreement of the vocabulary used across travel publications. An ontology may help with the process of creating and revising travel content to provide a consistent vocabulary which can be shared among book authors and editors. Thus, a role of a suitable ontology is as a shared controlled vocabulary to maintain a consistent use of terms.

The issue of the consistency of the structure of guidebooks was raised in Section 4.1.1. This issue has aspects that may be beyond the scope of what ontologies are able to address. In particular, ontologies alone may not be able to address the issue of consistency of the structure of guidebooks as it may require the scrutiny of the editors at Lonely Planet to decide on how the books are structured. As such, we do not consider this as a role.

In Section 4.1.1, the issue of consistency of content across guidebooks. This involved maintaining consistency when descriptions in a guidebook is reused in other books and upon updates of those descriptions in subsequent revisions of the guidebook. This aspect of this issue is related to the issue of ensuring information is up-to-date raised in Section 4.1.2. However, both issues may be beyond the scope of what ontologies are able to address. The issue of reusing guidebook content is dependant on decisions of the editors and does not relate to ontologies. The issue of ensuring information is up-to-date is an issue of tracking updated details and may be best handled using databases and versioning systems.

In Section 4.1.3, we described Lonely Planet's experience with various off-the-shelf ontologies and various issues were raised with regard to the adequate modelling of concepts and relationships of travel information in existing ontologies. As shown earlier in the example of Lake Tahoe in Section 4.1.3, a need of the application is to be able to model geographic places in a flexible manner. As this may be addressed with ontologies, we adopt this as an additional role of the ontology.

In summary, there are two roles that ontologies play in addressing issues in this application. They are having a shared controlled vocabulary of terms, and modelling geography and travel content. The first role of having a shared controlled vocabulary, is defined by the application and associated tasks. Consequently, the resulting content of the ontologies with regard to this role is determined by Lonely Planet. Table 4.3 summarises the Lonely Planet

application, the purpose of the ontology and the roles we identified.

<p><b>Application:</b> Lonely Planet Content Management</p> <p><b>Ontology:</b> Ontologies in the travel domain</p> <p><b>Roles of the ontology:</b></p> <ol style="list-style-type: none"> <li>1. Shared controlled vocabulary</li> <li>2. Adequately represent geographic places and associated content</li> </ol>
--

*Table 4.3: Lonely Planet application context and role of an ontology*

We use the roles identified in this section to derive a set of ontology requirements from this case study in applying the ROMEIO methodology for ontology evaluation which we present next.

## 4.2 ROMEIO ontology requirements for Lonely Planet

In this section, we present ontology requirements as part the ROMEIO methodology. First we identify which ontology requirements are needed based on each of the roles we identified in the previous section. For each of the ontology requirements in this section, we present them using the ROMEIO template for requirements and discuss them further.

### 4.2.1 Identifying ontology requirements

We use the roles of the ontology we discussed in the previous section, to elicit a set of applicable ontology requirements and present them below:

**Role 1: Shared controlled vocabulary** The ontology’s role here is in providing a reference point in maintaining a shared controlled vocabulary used by authors and editor in presenting travel content. The relevant detail here is maintaining a controlled vocabulary for the use of preferred terms and conventions used of names, places and activities (across the guidebooks). Thus, the specific ontology requirement here is the *controlled vocabulary of names, places and terms*.

**Ontology requirement:** Controlled vocabulary of names, places and terms

**Role 2: Adequately represent geographic places and associated content** With regard to the role of *appropriate representation of geographic places and associated content*, the ontology must be able to represent geographic relationships adequately. In particular, the ontology is to accommodate the imprecise nature of places, that is, flexibly model these geographic relationships between places and the ontology referred to here applies to the travel content across the collection of guidebooks. This directly relates to the content of the ontology. Thus, we take this to be the ontology requirement, that is, *having a set of relationships defined to flexibly classify geographic places*.

**Ontology requirement:** Flexibly classify geographic places

Another aspect of this role relates to the level of specificity of an ontology for the collection of travel content as a whole and an ontology for each guidebook. Two guidebooks may have differing levels of specificity, for example, a guidebook on Paris may have a higher level of specificity with regards to the geographic places in the city compared to a guidebook on France. Thus the role of the set of ontologies is to capture the appropriate level of specificity, or in other words, capturing *appropriate level of content granularity*, for each guidebook. This includes capturing the right level of content granularity so that the ontology is usable and without modelling too much detail so that there is not too much unrelated information in the ontology. Thus, the ontology requirement here is of an ontology with *appropriate granularity* for each guidebook.

**Ontology requirement:** Appropriate granularity

Thus, the appropriate set of ontology requirements from a discussion of the role of the ontology is:

- Controlled vocabulary of names, places and terms
- Flexibility in classifying geographic items
- Appropriate granularity

Table 4.4: LP ontology requirement: Controlled vocabulary of names, places and terms

<b>Requirement OR1:</b> Controlled vocabulary of names, places and terms	
<b>Analyze:</b>	Lonely Planet Ontology
<b>For the purpose of:</b>	The controlled usage of preferred names and terms
<b>With respect to:</b>	Lonely Planet content management of travel information
<b>From the viewpoint of:</b>	Guidebook authors and editors of travel information
<b>In the context of:</b>	A controlled vocabulary for travel content
<b>Motivation:</b>	
Motivating this requirement is maintaining a controlled vocabulary for authors and editors in providing a controlled usage of preferred names and terms. A specific scenario is avoiding the definition of redundant concepts that refer to an existing concept or instance, for example, the name of a place. However, for each concept in the vocabulary, there may be references to alternate names for other acceptable terms, for example, English terms for transliterated place names in Europe, which may have different permutations of terms referring to the same place.	

Having identified appropriate ontology requirements, we will elaborate on them further to obtain ontology requirements in carrying out the ROME0 analysis for this case study next.

#### 4.2.2 Ontology requirement 1: Controlled vocabulary of names, places and terms

The adoption of particular terms in Lonely Planet travel content differs due to multiple people authoring content. The ontology aims to help the application achieve consistency in how terms are used across travel content in guidebooks. This requires an ontology that defines a set of concepts and instances as a controlled vocabulary in a way that does not include redundant concepts and instances defined, that is, no two concepts or instances in the ontology are equivalent. However, where there may be alternate names and terms used for a given concept or instance, the ontology is required to specify a preferred one to adopt. Table 4.4 presents this requirement using the ROME0 requirement template.

Table 4.5: LP ontology requirement: Flexible classification of geographic items

<b>Requirement OR2:</b> Flexible classification of geographic items	
<b>Analyze:</b>	Lonely Planet Ontology
<b>For the purpose of:</b>	Flexible classification of geographic places
<b>With respect to:</b>	Lonely Planet Content Management of travel information
<b>From the viewpoint of:</b>	Guidebook authors and editors of travel information
<b>In the context of:</b>	Appropriate representation of geographic places
<b>Motivation:</b>	The relationships defined in the ontology are important in addressing the need to accommodate imprecise relationships in the geographic domain, specifically, the relationships between places that may be vague, for example, relationships of <i>regions surrounding Paris</i> . A suitable ontology would accommodate these kinds of relationships from the geographic domain.

### 4.2.3 Ontology requirement 2: Flexible classification of geographic items

Motivating this requirement is how concepts relate to one another in an ontology. In this case, the way relations are defined is important, specifically, accommodating for imprecise relations. An example of such an ontology is found in Jones et al. [2003]; Fu et al. [2005], where both precise and imprecise geographic relations, such as relationships of places *beside* London and places *near* London, feature in the SPIRIT geographic ontology for the application of spatial information retrieval.

For this requirement, the candidate ontology would need to have the appropriate geographic relationships (including the imprecise ones) defined in the ontology. Thus, coverage of the relationships required in the domain is important. Table 4.5 presents this requirement using the ROMEIO requirement template.

### 4.2.4 Ontology requirement 3: Appropriate granularity

The requirement here is about having an appropriate level of detail in the ontology for a given guidebook. In this domain, we may not require fine-grained geographic information in the ontology to the level of GPS co-ordinates. Conversely, having too little detail may

Table 4.6: *LP ontology requirement: Appropriate granularity*

<b>Requirement OR3:</b> Appropriate granularity	
<b>Analyze:</b>	Candidate ontologies
<b>For the purpose of:</b>	Evaluating suitable ontologies
<b>With respect to:</b>	Lonely Planet Content Management of travel information
<b>From the viewpoint of:</b>	Guidebook authors and editors of travel information
<b>In the context of:</b>	Modelling appropriate level of content granularity
<b>Motivation:</b>	
A suitable ontology will have enough detail modelled for it to be usable in the context of aiding the management of content for tasks in Lonely Planet. This means the right concepts and instances must be in the ontology. The problem to avoid is in either having too much detail or not enough detail in the ontology. Too much detail results in an ontology where the concepts and instances included are unused and perhaps redundant. Too little detail in the ontology results in an ineffective ontology for the application.	

result in an inadequate ontology. Table 4.6 presents this requirement using the ROMEIO requirement template.

### 4.3 ROMEIO questions for Lonely Planet

In considering the ontology requirements outlined in the previous section, we present questions to address those requirements below. Each set of questions is presented using the ROMEIO template for questions for a given requirement. Questions may be specified based on a related criterion that is related and thus an appropriate criteria-question may be used for that aspect of the requirement.

#### 4.3.1 Questions for ‘Controlled vocabulary of names, places and terms’

For this requirement, the candidate ontology would need to ensure that *semantic redundancy* is avoided. An applicable criteria-question can be found in the set of questions for the criterion of *conciseness*. We give them below using the ROMEIO template for questions.

Table 4.7: Questions for OR1

<b>Questions for OR1:</b> Controlled vocabulary of names, places and terms	
<b>Q1:</b>	How many identical concepts are modelled using different names? (conciseness)
<b>Q2:</b>	How many identical instances are modelled using different names? (conciseness)
<b>Discussion:</b>	
Q1 and Q2 are criteria-questions of the <i>conciseness</i> criterion as the problem that we seek to avoid is <i>semantic redundancy</i> . These questions consider concepts and instances that may have been defined as different concepts and instances, but are equivalent. The questions are applicable because they examine whether there is redundancy in the ontology with regard to the concepts and instances defined in the ontology.	

#### 4.3.2 Questions for ‘Flexible classification of geographic items’

The requirement of *flexible classification of geographic items* is about determining the required relationship types for the set of concepts in the given ontology. The ability of the ontology to capture relations used in the travel domain also plays a part. For example, accommodating imprecise relationships that capture a fuzzy aspect found in geography. Thus, coverage of the relationships modelled in the ontology is important in having concepts, such as ‘near-to’ or ‘beside’ relationships of places, being expressed in the ontology.

The criteria-questions we apply for this requirement are of the *coverage* criterion and we present it in Q3 found in Table 4.8. Specifically, this question considers the coverage of the relationships between concepts in the ontology compared to the relationships between concepts in the travel and geographic domain.

Table 4.8: Questions for OR2

<b>Questions for OR2:</b> Flexible classification of geographic items	
<b>Q3:</b>	Do the relationships between concepts in the ontology adequately cover the relationships between concepts in the domain? (coverage)
<b>Discussion:</b>	
	Q3 considers the relationships which exist between concepts – that is, the coverage of relationship types defined in the ontology. Here, we are concerned with having the various types of relationships in the travel and geographic domain which ought to be defined in the ontology, especially, relationships that may be imprecise.

### 4.3.3 Questions for ‘Appropriate granularity’

This requirement is about obtaining an ontology that has an appropriate the level of detail or granularity for the Lonely Planet application. None of the suggested criteria-questions from Chapter 3 adequately map to this requirement, although, the criteria of coverage and conciseness partially apply.

Appropriate granularity, in part, relates to coverage. A suitably granular ontology includes concepts and instances that have sufficient coverage of the domain being modelled. This requirement also relates to modelling the content of the ontology to the right level of granularity but not beyond that. That is, the ontology has conciseness such that it excludes redundant or irrelevant concepts and instances beyond a level of granularity. Taking the two criteria into account, we specify Q4 and Q5 to address this requirement of appropriate granularity in Table 4.9 for concepts and instances in an ontology, respective.

## 4.4 ROMEO measures for Lonely Planet

In this section, we associate measures with questions. Each measure seeks to answer the question as quantitatively as possible and is presented using the ROMEO template for measures. Accompanying each description of the measure is a discussion summarising its intent and expected measurements relevant in seeking to answer the question. We will also give the



Table 4.9: Questions for OR3

<b>Questions for OR3:</b> Appropriate granularity	
<b>Q4:</b>	Does the ontology have an appropriate level of granularity with regard to its <i>concepts</i> compared with the domain being modelled?
<b>Q5:</b>	Does the ontology have an appropriate level of granularity with regard to its <i>instances</i> compared with the domain being modelled?
<b>Discussion:</b>	
<p>The right level of detail in the ontology is important in this requirement, specifically, the set of concepts and instances that are included. A set of concepts is a prerequisite for being able to define instances, thus it is important to examine its granularity. We specify question Q4 for this reason. Q5 examines whether the granularity of an ontology is appropriate by examining the included set of instances in the ontology with regard to the set of instances in the travel domain.</p>	

range of possible values, the optimal values, and additional definitions of the measure used where possible.

#### 4.4.1 Measures for ‘How many identical concepts are modelled using different names?’

For this question, we are concerned with measuring the number of concepts that are equivalent with another for the set of concepts defined in the ontology. The appropriate measure for this question is a count of *the number of identical formal definitions of classes*, that is the measure of *semantically identical concepts* defined in Equation 2.21. Using the ROMEO table for measures, we discuss the measure in Table 4.10.

Table 4.10: Measures for Q1

<b>Measures for Q1:</b> How many identical concepts are modelled using different names?
<p><b>M1:</b> <i>semantically identical concepts</i> (<math>\mathcal{O}_c</math>)  Range of values: <math>0 -  \mathcal{O}_c </math>  Optimal value(s): 0</p> <p><b>Discussion:</b></p> <p>M1 considers any semantic equivalence between concepts in the ontology <math>\mathcal{O}_c</math>. Ideally, there should not be any concepts that are equivalent. In the event that there are a set of concepts that are equivalent, they require a solution where a preferred concept is used in the ontology to satisfy its requirement, which may require removing one or more concepts from the ontology.</p>

#### 4.4.2 Measures for ‘How many identical instances are modelled using different names?’

The appropriate measure for this question is the number of *semantically identical instances* presented in Equation 2.22. We discuss this measure in Table 4.10 using the ROMEIO template.

#### 4.4.3 Measures for ‘Do the relationships between concepts in the ontology adequately cover the relationships between concepts in the domain?’

For this question, we are concerned with capturing the relationship types for the given domain. The appropriate measure for this question is recall of relationships between concepts of the domain as the aim is to obtain a good coverage of relationship types in this domain. We use the Equation 2.11 for recall of the relationships between concepts in an ontology with a frame of reference.

For the recall measure, the range of values is between 0 and 1. A low recall measurement means that we have not captured many of the concepts in the domain. A high recall value

Table 4.11: Measures for Q2

<b>Measures for Q2:</b> How many identical instances are modelled using different names?
<p><b>M2:</b> <i>semantically identical instances</i> (<math>\mathcal{O}_i</math>)</p> <p>Range of values: <math>0 -  \mathcal{O}_i </math></p> <p>Optimal value(s): 0</p> <p><b>Discussion:</b></p> <p>M2 considers any semantic equivalence between instances in the ontology. Again, ideally, there should not be any instances that are equivalent. In the event that there are a set of instances that are equivalent, they require a solution where a preferred instance is used in the ontology to satisfy its requirement, which may require removing one or more instances from the ontology.</p>

means that we have captured many of the concept in the domain. Thus, an optimal value for recall is 1. Table 4.12 gives the description for the measure in the ROMEO template.

Table 4.12: Measures for Q3

<b>Measures for Q3:</b> Do the relationships between concepts in the ontology cover the relationships between concepts in the domain?
<p><b>M3:</b> <i>recall</i> (<math>\mathcal{O}_{cr}, \mathcal{F}_{cr}</math>)</p> <p>Range of values: 0–1</p> <p>Optimal value(s): 1</p> <p><b>Discussion:</b></p> <p>The aim is to measure the coverage of the required relationships in the ontology <math>\mathcal{O}_{cr}</math> with the frame of reference <math>\mathcal{F}_{cr}</math>, which we take to be the domain in this case. For example, <i>city-of</i>, <i>near-by</i>, <i>capital city-of</i> relationships. We do this using the recall measure. The desired value of this measure is 1, which indicates total coverage over the domain.</p>

#### 4.4.4 Measures for ‘Does the ontology have an appropriate level of granularity with regard to its concepts compared with the domain being modelled?’

An appropriate measure is required here for the purpose of measuring determining *appropriate granularity*. Two existing measures that are candidates for this are precision and recall, as they map to criteria-questions for conciseness and coverage respectively. In particular, *precision* measures how much of the domain is in the ontology, while *recall* determines how much of the ontology is in the domain. However, neither precision or recall help with measuring granularity. An ontology that is precise may define concepts in a given domain accurately, but not have adequate coverage of a given domain. An ontology that has recall over a given domain may have all concepts and instances in a given domain but may model numerous irrelevant concepts and instances, that is, it may be imprecise.

The measure we propose to answer this question is the *F-measure*, which is used in information retrieval [Baeza-Yates and Ribeiro-Neto, 1999]. The F-measure is the harmonic mean of precision and recall. The F-measure gives some sense of adequacy and balance between modelling enough of the domain without modelling too much, thus giving an indication of appropriate granularity. We have adapted the F-measure in information retrieval for ontology evaluation in the context of indicating appropriate granularity of concepts and give the specific definition for it along with precision and recall below in Equation 4.2. Hence the F-measure extends the work of Guarino [2004].

$$F\text{-measure}(\mathcal{O}_c, \mathcal{F}_c) = \frac{2}{\frac{1}{\text{recall}(\mathcal{O}_c, \mathcal{F}_c)} + \frac{1}{\text{precision}(\mathcal{O}_c, \mathcal{F}_c)}} \quad (4.1)$$

where  $\mathcal{O}_c$  denotes the set of concepts defined in the ontology and  $\mathcal{F}_c$  denotes the set of concepts of the frame of reference, which we take as the domain.

Table 4.13 gives the description of the measure in the appropriate ROMEO template for measures. The range of values for the F-measure is between 0 and 1, with 0 indicating inadequate granularity and 1, an appropriate level of granularity without the inclusion of

other concepts outside of the domain being modelled in the ontology. Thus, the optimal value for this measure is 1.

Table 4.13: Measures for Q4

<p><b>Measures for Q4:</b> Does the ontology have an appropriate level of granularity with regard to its concepts compared with the domain being modelled?</p>
<p><b>M4:</b> <math>F\text{-measure}(\mathcal{O}_c, \mathcal{F}_c)</math>  Range of values: 0–1  Optimal value(s): 1</p>
<p><b>Discussion:</b></p> <p>The F-measure may help to measure whether a given ontology has an appropriate level of granularity with regard to the concepts modelled in an ontology <math>\mathcal{O}_c</math> compared with a frame of reference <math>\mathcal{F}_c</math>, in this case the domain – 0 indicates an inadequate granularity and 1, an appropriate level of granularity without the inclusion of other concepts outside of the domain being modelled in the ontology. The optimal value is 1.</p>

#### 4.4.5 Measures for ‘Does the ontology have an appropriate level of granularity with regard to its instances compared with the domain being modelled?’

Similar to the previous mapping to Q4, the F-measure also applies here for this question. The difference for this question is the focus on the adequate level of granularity of the instances in the ontology for the domain being modelled. In Equation 4.2, we present the definitions relevant for the F-measure for mapping to this question.

$$F\text{-measure}(\mathcal{O}_i, \mathcal{F}_i) = \frac{2}{\frac{1}{\text{recall}(\mathcal{O}_i, \mathcal{F}_i)} + \frac{1}{\text{precision}(\mathcal{O}_i, \mathcal{F}_i)}} \quad (4.2)$$

where  $\mathcal{O}_i$  denotes the set of instances defined in the ontology and  $\mathcal{F}_i$  denotes the set of instances for the frame of reference, in this case the domain.

Table 4.14 gives the description of the measure in the appropriate ROMEO template for measures. The range of values for the F-measure is between 0 and 1, with 0 indicating inadequate granularity and 1, an appropriate level of granularity without the inclusion of other instances outside of the domain being modelled in the ontology. Thus, the optimal value for this measure is 1.

Table 4.14: Measures for Q5

<p><b>Measures for Q5:</b> Does the ontology have an appropriate level of granularity with regard to its instances compared with the domain being modelled?</p>
<p><b>M5:</b> <math>F\text{-measure}(\mathcal{O}_i, \mathcal{F}_i)</math>  Range of values: 0–1  Optimal value(s): 1</p> <p><b>Discussion:</b></p> <p>The aim here is to measure the granularity of instances included in an ontology <math>\mathcal{O}_i</math> compared with instances in the frame of reference <math>\mathcal{F}_i</math>, that is the domain being modelled. The measure used here is the F-measure – 0 indicates an inadequate granularity and 1, an appropriate level of granularity without the inclusion of other concepts outside of the domain being modelled in the ontology. The appropriate ontology will include instances that has an appropriate level of granularity for the domain being modelled. Ideally, the instances in the ontology should have the right level of granularity with a F-measure value of 1, compared with instances in the domain being modelled.</p>

In Chapter 6, we validate this measure with its associated question.

## 4.5 Summary

In this chapter, an application with an emerging need for an ontology – Lonely Planet, was considered. The role of ontologies for this application was examined from a case study of the management of Lonely Planet travel content. From an understanding of the role of the

ontology in this application, we applied the ROMEIO methodology for ontology evaluation and presented a set of ontology requirements, a set of questions for each ontology requirement and a set of appropriate measures for each ROMEIO question. Three requirements were identified and we summarise them below.

A controlled vocabulary was found to be a requirement and thus associated questions and measures were mapped to it. A question used was from the set of criteria-questions for conciseness, and specifically considered equivalent concepts and instances, which may have differing names or terms. These aimed to reduce the occurrence of semantic redundancy. Appropriate measures were mapped to those questions to examine those details.

The requirement of a flexible classification of geographic places was identified. Specifically, the application required an ontology that was able to model places appropriately and in a flexible way. Thus, questions used considered the *coverage of relationships between concepts* to examine whether the ontology included the set of geographic relationships required for the application at hand. The measure of *recall* of relationships of concepts in the ontology compared with a representation of the domain was mapped.

Lastly, the ontology was required to be able to provide an appropriate level of content granularity. Thus, through a newly defined set of questions, we considered the *appropriate level of granularity with regard to the concepts and instances in an ontology compared with the domain being modelled*. We proposed a new measure for ontology evaluation – the F-measure, which was adapted from information retrieval. The F-measure was used to address the set of questions for this requirement to indicate whether an ontology has an appropriate level of granularity with regard to its set of concepts and instances for the domain being modelled. Additionally, we present the validation of this mapping in Chapter 6. Table 4.15 summarises the analysis carried out on Lonely Planet using ROMEIO.

Table 4.15: Summary of the ROMEO analysis of ontology requirements for Lonely Planet

<b>Requirement:</b>	Controlled vocabulary of names, places and terms
<b>Question:</b>	How many identical concepts are modelled using different names?
<b>Measure:</b>	<i>semantically identical concepts</i> ( $\mathcal{O}_c$ )
<b>Question:</b>	How many identical instances are modelled using different names?
<b>Measure:</b>	<i>semantically identical instances</i> ( $\mathcal{O}_i$ )
<b>Requirement:</b>	Flexible classification of geographic items
<b>Question:</b>	Do the relationships between concepts in the ontology adequately cover the relationships between concepts in the domain?
<b>Measure:</b>	<i>recall</i> ( $\mathcal{O}_{cr}, \mathcal{F}_{cr}$ )
<b>Requirement:</b>	Appropriate granularity
<b>Question:</b>	Does the ontology have an appropriate level of granularity with regard to its concepts compared with the domain being modelled?
<b>Measure:</b>	<i>F-measure</i> ( $\mathcal{O}_c, \mathcal{F}_c$ )
<b>Question:</b>	Does the ontology have an appropriate level of granularity with regard to its instances compared with the domain being modelled?
<b>Measure:</b>	<i>F-measure</i> ( $\mathcal{O}_i, \mathcal{F}_i$ )



## Chapter 5

# Wikipedia

*All knowledge has its origins in our perceptions.*

— Leonardo da Vinci

Wikipedia is a multi-lingual online encyclopedia comprising articles from volunteer contributions on a variety of topics. These articles are continuously being discussed, updated, revised by the community and the editors. In addition to the article text itself, each article has a variety of metadata attached, such as links to other related articles, external web pages. Furthermore, articles may belong to one or more categories and these categories are organised to form a *category structure*. The category structure is primarily used to find information in related articles through browsing it.

In this chapter, we consider the application of browsing with the Wikipedia category structure as a case study for applying the ROMEIO methodology. In Section 5.1, Wikipedia and its category structure is described, and we examine the set of guidelines for the Wikipedia category structure in order to determine the roles of a suitable ontology in this application. With those roles in mind, we use the ROMEIO methodology for evaluating ontologies for Wikipedia. In particular, we identify a set of ontology requirements in Section 5.2. In Section 5.3, we obtain ROMEIO questions for each ontology requirement. Lastly, in Section 5.4, we associate appropriate measures with each question. Refer to Appendix C for the outcome of our analysis using ROMEIO for Wikipedia and its category structure.<sup>1</sup>

---

<sup>1</sup>Note: this chapter presents work that has been adapted from the original analysis from Yu et al. [2007]

## 5.1 Wikipedia and its categories

Wikipedia is a free, online encyclopedia which was introduced in 2001. Wikipedia has grown to a large collection of articles on various topics and it now has instantiations in 253 different languages. These articles are created, verified, updated and maintained by editors and volunteers worldwide.

Wikipedia deviates in many ways from a conventional encyclopedia. Wikipedia relies on a *wiki* for the authoring content, which enables documents to be authored in collaborative fashion using simple formatting markup. Wikipedia allows for a variety of access methods to its collection of articles, for example, search, portals, and its category structure. Regarding Wikipedia's category structure, it also deviates from conventional category structures in that it is defined in a bottom up manner by allowing users to attach a set of category associations to articles.

In this section, we describe the application of browsing Wikipedia articles using categories, the role of ontologies for this application, which will be used to determine an appropriate ontology evaluation method for this application using the ROMEO methodology from Section 5.2 onwards. We limit our discussion to the English instantiation of Wikipedia. Section 5.1.1 gives a brief background to Wikipedia, the article collection for the English instantiation of Wikipedia, and various methods for navigating the article collection. We also discuss the methods of exploring the article collection, and highlight a method of browsing articles, that is, using the Wikipedia category structure. Section 5.1.3 describes the current category structure for Wikipedia as a way of browsing the article collection, and we examine the specific role of the ontology, that is, the category structure, for this application.

### 5.1.1 Wikipedia content, policies and guidelines

Since its introduction, Wikipedia has grown into a large collection of articles on topics ranging from science and technology to art and popular culture. The English version of Wikipedia<sup>2</sup> in light of the ROMEO methodology.

---

<sup>2</sup><http://en.wikipedia.org>

alone has over 2.3 million articles.<sup>3</sup>

The content in Wikipedia is open in nature, which allows the authoring of articles to be contributed by volunteers worldwide. Access to the Wikipedia articles is freely available and its content is able to be discussed and modified by anyone around the world. While anyone may be able to make contributions to Wikipedia, the main process in creating article content for Wikipedia is through a process of consensus building and discussion before being officially accepted. Negotiation between the author of the new content and the community takes place on designated areas to develop consensus on each article page. Thus, content may change frequently as the discussion of article content progresses.

### **Wikipedia policies and guidelines**

While, the open nature of Wikipedia means that anyone can contribute to the body of knowledge in the encyclopedia, its content is subject to guidelines and policies set by editors. These official Wikipedia policies and guidelines are defined to support the goal of creating a free encyclopedia. They help to address specific issues such as violations of use and govern what is acceptable content and quality. Violations of use include *vandalism* and *sock puppeting*. Examples of vandalism are *blanking* (removing significant parts of an article), *profanity* and *graffiti*. A *sock puppet* is an alternate Wikipedia account used for the purpose of avoiding scrutiny and including misleading information. Violations are recognised by Wikipedia and actively dealt with according to the processes and guidelines put in place by Wikipedia's editors.

There are many guidelines for the different parts of Wikipedia. However, these guidelines can be summarised as the *five pillars of Wikipedia* and are presented below:

1. **Wikipedia is an encyclopedia:** Its content should be referenced by external sources and ought to be reliable, accurate and factual. Original research and personal opinions are not appropriate.
2. **Wikipedia has a neutral point of view:** Aims to have articles that are verifiable, authoritative but not biased in any way.

---

<sup>3</sup><http://en.wikipedia.org/wiki/Wikipedia> as of January 2008.

3. **Wikipedia is free content:** Article text is available under GNU Free Documentation License. This implies that article content is subject to further editing and available for redistribution to the community.
4. **Wikipedia has a code of conduct:** Contributors to Wikipedia are expected to act in good faith and treat others with respect.
5. **Wikipedia does not have firm rules:** Rules and guidelines are provided, however, may be broken if it is sensible and appropriate and upholds the above pillars.

Additionally, there are three core policies that are overriding with regard to the content of Wikipedia and govern what is acceptable. The core policies especially apply with regard to authoring the content of articles and are presented below.

1. **Verifiability:** Wikipedia content ought to be verified from a published third party source that is reliable. This means that readers and editors should be able to defend article content with reliable sources. Examples of reliable sources include the peer reviewed publications of the *American Journal of Public Health*, and the *IEEE Intelligent Systems Journal*.
2. **No original content:** Another policy of Wikipedia is that it does not allow original research to be published. This policy requires citations from reliable sources that are directly related to the article content. This means that content should be factual, rather than opinions, speculations, or unpublished ideas.
3. **Neutral Point of View (NPOV):** NPOV requires the fair and unbiased representation of content in Wikipedia. Articles ought to have a balanced view that does not condemn or condone, endorse or discourage a given point of view. Where multiple or conflicting perspectives exist, these are to be presented fairly, irrespective of the popularity of a given view.

Both the guidelines and policies shape Wikipedia as a free encyclopedia and help to address issues that may arise.

### 5.1.2 Navigating and exploring articles

Another key aspect of Wikipedia, is providing appropriate navigation methods for the article collection. There are numerous ways of exploring the Wikipedia article collection. A common method of accessing article content is by using the search tool in Wikipedia to match on article title and content. Alternatively, users may issue a query on Wikipedia content using an online search engine.

Users of Wikipedia can also navigate Wikipedia using other features provided by the Wikipedia web interface. The Wikipedia web interface features functionality that allows users to view articles, the *recent changes link* to view articles that were recently modified and the *random articles* function to navigate to a randomly selected Wikipedia article.

Another navigation tool in Wikipedia is a *reference list*, which groups related content. Specific reference lists include:

- **Overview lists:** A list giving an overview of Wikipedia's coverage topics and article content
- **Featured content:** A list of articles, portals, images, and topics determined to be some of the best by Wikipedia's editors
- **Good articles:** A list of articles which are considered to have good quality but are not featured article quality.
- **Portals:** Includes a useful list of entry-points to specific topics or areas in Wikipedia

Users may also navigate using *an index*, which is designed to help the reader find information using terms and their associated content. The two main indices in Wikipedia are the *Alphabetical index of articles by title* and the *Wikipedia category structure*. The *Alphabetical index of articles* is an index of articles sorted alphabetically. The *category structure* is an index of major categories, arranged by subject. The Wikipedia category structure will be the main topic of discussion for the remainder of this chapter.

### 5.1.3 Wikipedia categories

Categories are used to organise articles in Wikipedia topically. However, the category structure was not part of the original design of Wikipedia. There were inadequacies with regard to the reliance on the links in an article, the alphabetical indexing of articles, the use of the search tool and reference lists to help users to find articles [Original Wikipedia Guidelines for Category Proposals and Implementations]. Thus a category structure was proposed in late 2003 as an additional method for finding articles. It was subsequently implemented in early 2004.

Categories are created by annotating an article with the category title, as opposed to specifying a category structure independently from articles. Subcategories may be created by annotating a given category page with the parent category. Each category has one or more associated articles and can have multiple parent and children categories. The intention of this was for users to associate related categories and articles with each other. Table 5.1 gives a sense of its size with the number of categories and articles of the entire category structure.<sup>4</sup>

Measure	Value
Number of articles	1,079,246
Number of categories	111,287
Average articles per category	25.7

*Table 5.1: Size of the Wikipedia category structure*

#### Design of category structure

The Wikipedia category structure allows for the navigation of information that would otherwise be difficult to locate by submitting a set of query terms to a search engine, or if the user's information need of a user is vaguely defined. Brent Gulanowski, a contributor who initiated the need for a category structure, is quoted below from an archived discussion page

<sup>4</sup>Source: Category statistics from <http://labs.systemone.at/wikipedia3> and Article statistics from <http://download.wikimedia.org/enwiki> accessed August 2007

in Wikipedia regarding the reason for proposing a category structure for Wikipedia.<sup>5</sup>

“The primary reason for proposing this system is to ensure that a user of the Wikipedia can see, at a glance, all pages which relate to a particular page that they are reading or a topic that they are interested in. Users should be spared the necessity of following links or performing fruitless searches in the cases where their unfamiliarity with a subject means they do not even know the terms for which they are searching.”

In addition to helping with navigation, the Wikipedia category structure may help authors and editors to group articles and maintain a certain consistency in style and granularity of content, for example, the headings and content for articles on film actors, and marking articles for review by editors.

### **Wikipedia category structure as an ontology**

Overall, the Wikipedia category structure does not conform to a tree structure or strict hierarchy as articles may belong to more than one category, and categories may also have multiple parents (although parts of the category structure may be tree-like). The category structure may intersect to allow multiple schemes to co-exist. Parts of the category structure may implement multiple classification systems such that they overlap. These overlapping trees or classifications are allowed in Wikipedia to provide different but valid views on a given set of categories.

We may consider the Wikipedia category structure as a *simple ontology* using the definition we presented in Chapter 2, as it includes concepts and relationships between concepts, that is, a set of categories and subcategory relationships. Also, relationships exist between a given article and a set of categories. There are however, no complex relationships and logical constraints in the Wikipedia category structure. It is a general ontology, in the sense that it has coverage over multiple domains. At the highest level, the root is a category called “Fundamental” and has five subcategories. The subcategories are: Information, Nature, Society,

---

<sup>5</sup>Source: [http://en.wikipedia.org/wiki/User:Brent\\_Gulanowski/Categorization](http://en.wikipedia.org/wiki/User:Brent_Gulanowski/Categorization) Accessed March 2008

Structure, and Thought. These are abstract and may be taken as upper level concepts, but they encompass general topic areas that a given Wikipedia article may fall under. Thus, the Wikipedia category structure is a large, general, simple ontology, which is utilised by users and editors, and is constantly evolving.

As the Wikipedia category structure is used for browsing, the role of the category structure therefore, is to help users navigate articles through an exploratory mode of browsing between categories, despite the fact that only some users navigate in this manner. We also include the role of the administration and editing of articles. We outline the purpose of the ontology, that is the Wikipedia category structure, and the application at hand using the appropriate ROMEO template in Table 5.2.

<p><b>Application:</b> Browsing Wikipedia.</p> <p><b>Ontology:</b> Wikipedia Category Structure.</p> <p><b>Role of the ontology:</b></p> <ul style="list-style-type: none"> <li>• Aid user navigation of the collection of articles through browsing</li> <li>• Aid administration and editing of the collection of articles</li> </ul>
---

*Table 5.2: Application context and purpose of ontology for Wikipedia*

## 5.2 ROMEO ontology requirements for Wikipedia

In this section, we outline the ontology requirements using the ROMEO template introduced in Chapter 3. The ontology requirements gathered in the previous section are highly focused around the activity of browsing for relevant articles. We present each of the requirements below using the template and give its context and the motivation for each requirement, specifically in Sections 5.2.2, 5.2.3, 5.2.4, 5.2.5, and 5.2.6.



### 5.2.1 Identifying ontology requirements

For this case study, ontology requirements are elicited directly from the online documentation of the guidelines for the Wikipedia category structure. We specifically use a set of online documentation that describe Wikipedia and its category structure. These online documents have been established through discussion and scrutiny of the Wikipedia editorial process, which involves Wikipedia authors, users and editors. As a result of this process, the requirements and purpose of the Wikipedia category structure was documented and guidelines established creating, organising and implementing categories and can be found online.<sup>6</sup>

We will present a set of relevant guidelines for Wikipedia category structure below, however, for more detail, excerpts of the sections from the relevant Wikipedia guidelines about the category structure is found in Appendix D. For this case study, the scope of discussion for a set of ontology requirements includes guidelines relating to the desired quality of content and structure for the Wikipedia category structure. The set of ontology requirements should support the specific role of aiding users to navigate article content and uphold the five Wikipedia pillars, which were presented in Section 5.1.1.

In eliciting a set of ontology requirements, we exclude the editorial role of the ontology in the ROMEO analysis. We focus on the evaluation of the content of the category structure that is used for browsing articles, rather than categories that are used for administrative and editorial purposes.

The overall requirement with regard to the content and structure of the ontology is to *“make decisions about the structure of categories and subcategories that make it easy for users to browse through similar articles”* [Wikipedia Online Editing Guidelines for Categories]. From this requirement, there are several guidelines that editors outline in the set of online documentation regarding category structure. We summarise specific guidelines regarding the content of the Wikipedia category structure below and determine whether they are applicable as ontology requirements.

---

<sup>6</sup><http://en.wikipedia.org/wiki/Wikipedia: Categorization>

**Categories do not form a tree.** Editors of Wikipedia do not impose a strict hierarchy on the Wikipedia category structure so that any interesting and potentially useful category associations may be incorporated [Wikipedia Online Editing Guidelines for Categories]. The category structure is intended to be more like a directed acyclic graph that includes the intersection of various categories, although in practice, cycles may exist. This allows for the ability to accommodate multiple views which co-exist. However, the usefulness of such an intersection of categories is dependent on having an adequate level of intersection without including too much intersection that it impedes user navigation. Thus, we consider this to be an ontology requirement, specifically, of having an adequate level of category intersection.

**Ontology requirement:** Adequate level of category intersection.

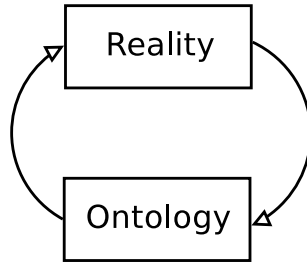
**Categories should be appropriately grouped.** This guideline of appropriate grouping of categories is about ensuring that the category structure is factual, rich, interesting, and useful for browsing articles. Standard classifications, such as those used in science, may be incorporated and adapted to accommodate alternate views, as long as they achieve the main purpose of aiding users browse. Another suggested method for grouping categories is a functional one where related categories are grouped on a function or theme, for example, the category of ‘*World War II*’ has parent categories of ‘*Contemporary French history*’ and ‘*Nuclear warfare*’ – both are different classifications but are combined to give an interesting categorisation using a common intersecting category.

Another aspect of this guideline is the consistency over the set of category groupings. For example, if the category of ‘*World War II*’ has parent categories of ‘*Contemporary French history*’ and ‘*Nuclear warfare*’, it would be also relevant for ‘*Contemporary German history*’ to be a parent category of ‘*World War II*’ as well.

As the above guideline relates to the structure and content of the category structure, we consider as an ontology requirement.

**Ontology requirement:** Categories should be appropriately grouped.

**Cycles should usually be avoided.** This guideline relates to the relationships between a set of categories and its subcategories. Cycles may occur for various reasons. The simplest way for a cycle to occur is when a given category is made a subcategory of its child. In Figure 5.1, an example is given of the *Reality* category defining a subcategory relationship to the *Ontology* category. The *Ontology* category also defines a subcategory to the *Reality* category and thus a cycle has occurred.



*Figure 5.1: Example of a cycle between categories of Reality and Ontology*

However, other more complex cycles involving many categories in the chain may occur and may be difficult to detect. The Wikipedia guidelines documents an instance of resolving a 22 category long cycle.

Although there may be cases where cycles are useful, they generally impede the usability of the Wikipedia category structure, for example, with users getting lost in sequential browsing of a particular cycle of categories. It also impedes certain automated processing of the subcategory using computer programs [Wikipedia Online Editing Guidelines for Categories]. Thus, the need here is to detect and avoid cycles that are not useful. As this relates to the structure of the category structure, we consider this to be an ontology requirement.

**Ontology requirement:** Cycles should be avoided.

**Appropriate categorisation of articles.** The requirement here is to associate a given article to the right set of categories. The guidelines recommend restraint on the number of categories attributed to an article [Wikipedia Online Editing Guidelines for Categories]. An article may have too many categories associated with it due to the following reasons:

1. Possible lack of an appropriate set of categories for the given article
2. Incorrect association of categories to the article
3. NPOV, that is, association of articles to categories based on opinion or bias
4. Spam, that is, deliberate vandalism of the article

Item 1 relates to a possible lack of content in the Wikipedia category structure. An inadequate set of categories may lead to articles being placed in too many categories. As such, the specification of new categories to may be required. This item forms an ontology requirement of ensuring the set of categories available is complete and correct.

**Ontology requirement:** Ensure the set of categories is complete.

Item 2 relates to the incorrect association of a set of categories to a given article. Items 3 and 4 violate Wikipedia's policies of ensuring NPOV and verifiability respectively. These items relate to *correctness* of the defined relationships between article and category within the category structure. Thus, this is an ontology requirement, specifically, of ensuring the set of categories associated with an article is correct. An indirect effect of correctly associating articles to categories is that it may help to identify erroneous categories (assuming that most poorly formulated categories can be identified as categories with little or no articles associated). These erroneous categories may subsequently be removed, hence improving the quality of the category structure.

**Ontology requirement:** Ensure categories associated in articles are correct.

From the set of guidelines and policies for the Wikipedia category structure, we have distinguished a set of ontology requirements. However, we do acknowledge that the above analysis is a simplified view of the overall set of issues faced in organising the Wikipedia category structure. The issues faced by the Wikipedia editors are complex, involving multiple interpretations and views in each domain being captured into a set of categories. There may also be other issues that the guideline and policy documents may not fully capture. Thus, there is scope for obtaining other possible ontology requirements from further analysis. For the purposes of considering the Wikipedia category structure as a case study in applying the ROMEO methodology, we limit our discussion to the identified ontology requirements above.

### 5.2.2 Ontology requirement 1: Adequate level of category intersection

The original guideline from the previous section is *categories do not form a tree*. The guideline was to have a category structure that does not strictly adhere to a tree. The structure was intended to give an intersection of domains to make it interesting, sensible and useful for browsing. We have refined this requirement to read as ‘*adequate level of category intersection*’. Table 5.3 presents the ROMEO analysis for this requirement and further elaborates on the requirement.

Table 5.3: *Ontology requirement 1*

<b>Requirement OR1:</b> Adequate level of category intersection.	
<b>Analyze:</b>	Wikipedia Category Structure
<b>For the purpose of:</b>	Evaluating the Category Structure
<b>With respect to:</b>	Adequate level of category intersection
<b>From the viewpoint of:</b>	Wikipedia Users, Editors and Authors
<b>In the context of:</b>	Browsing Wikipedia Categories
<b>Motivation:</b>	
Having an intersecting category structure facilitates user navigation of articles based on the given information need through the ability to browse alternate but somewhat related domains. In doing so, users may browse useful articles that they did not expect to encounter. For example, if we consider the <i>Food and Drink</i> category, it may intersect with categories such as <i>Culture</i> , <i>Health</i> , and <i>Digestive system</i> . All are related and may be in mind of users as they browse.	

### 5.2.3 Ontology requirement 2: Categories should be appropriately grouped

This requirement is about the quality of the categorisation and how categories should be grouped, that is, the organisation of parent and child categories for each category. Table 5.4 presents the ROMEO analysis for this requirement.

Table 5.4: *Ontology requirement 2*

<b>Requirement OR2:</b> Categories should be appropriately grouped	
<b>Analyze:</b>	Wikipedia Category Structure
<b>For the purpose of:</b>	Evaluating the Category Structure
<b>With respect to:</b>	Categories should be appropriately grouped
<b>From the viewpoint of:</b>	Wikipedia Users, Editors and Authors
<b>In the context of:</b>	Browsing Wikipedia Categories
<b>Motivation:</b>	The issue here is regarding the quality of subcategories and parent categories associated with each category. A given category structure should have an appropriate set of subcategories and parent categories specified such that it facilitates the browsing of categories and articles according to the information need.

#### 5.2.4 Ontology requirement 3: Avoiding cycles in the category structure

This requirement is about avoiding the occurrences of cycles. Applying the ROME template in Table 5.5, we describe the motivation for this requirement as presented in gathering requirements from the guidelines in the previous section.

Table 5.5: *Ontology requirement 3*

<b>Requirement OR3:</b> Avoiding cycles in the category structure	
<b>Analyze:</b>	Wikipedia Category Structure
<b>For the purpose of:</b>	Evaluating the Category Structure
<b>With respect to:</b>	Avoiding cycles in the category structure
<b>From the viewpoint of:</b>	Wikipedia Users, Editors and Authors
<b>In the context of:</b>	Browsing Wikipedia Categories
<b>Motivation:</b>	Wikipedia does not prevent cycles in its category structure but the Wikipedia editors strongly discourage them. In general, they are not helpful for users as it can be confusing. In addition, cycles may impede some automated processes in their use of the category structure.

### 5.2.5 Ontology requirement 4: Ensure the set of categories is complete

This requirement is about ensuring that the ontology has a complete set of categories defined. Applying the ROMEIO template in Table 5.6, we describe the motivation for this requirement based on the discussion of its respective guideline in the previous section.

*Table 5.6: Ontology requirement 4*

<b>Requirement OR4:</b> Ensure the set of categories is complete	
<b>Analyze:</b>	Wikipedia Category Structure
<b>For the purpose of:</b>	Evaluating the Category Structure
<b>With respect to:</b>	Ensuring the set of categories is complete
<b>From the viewpoint of:</b>	Wikipedia Users, Editors and Authors
<b>In the context of:</b>	Browsing Wikipedia Categories
<b>Motivation:</b>	
A motivation for this requirement is that the association of a large number of categories to a given article may indicate a poorly defined set of categories. The issue here is resolving a lack of completeness in the category structure. Thus, a complete category structure ensures that the set of categories defined allows an article to associate an appropriate set of categories to it.	

### 5.2.6 Ontology requirement 5: Ensure categories associated in articles are correct

Applying the ROMEIO template in Table 5.7, we describe the motivation for this requirement based on the discussion of its respective guideline in the previous section.

## 5.3 Questions

In considering the ontology requirements outlined in the previous section, we present questions that help to address those requirements below. Accompanying each question is a discussion of how we arrived at those questions and some qualities to consider for a given requirement.

Table 5.7: *Ontology requirement 5*

<b>Requirement OR5:</b> Ensure categories associated in articles are correct	
<b>Analyze:</b>	Wikipedia Category Structure
<b>For the purpose of:</b>	Evaluating the Category Structure
<b>With respect to:</b>	Ensure categories associated in a given article are correct
<b>From the viewpoint of:</b>	Wikipedia Users, Editors and Authors
<b>In the context of:</b>	Browsing Wikipedia Categories
<b>Motivation:</b>	A motivation for this requirement is that the association of a large number of categories to a given article may indicate associating a set of categories incorrectly, for example, due to spam, bias or human error. The issue here is resolving the correctness of categories associated for each article, especially those with too many categories associated with it.

### 5.3.1 Questions for ‘Adequate level of category intersection’

This requirement is not just about how intersected an ontology is. A highly intersected ontology would reduce the quality of the ontology and would not be useful for users browsing it. Conversely, an ontology with very little category intersection for Wikipedia may not be as rich and interesting for users browsing it. The key for this requirement in determining whether there is adequate intersectedness in the category structure.

Thus, the question presented for this requirement in Table 5.8 emphasises having adequate intersection. *Intersectedness* does not appear in the list of existing ontology evaluation criteria and there is no defined criteria-question for this quality. It was discovered in applying the ROMEO methodology for this case study.

### 5.3.2 Questions for ‘Categories should be appropriately grouped’

This requirement is about correctly modelling the category structure, specifically, the relationships between categories and its subcategories as well as the relationships between each category and its parent categories. Regarding this requirement, an issue to avoid is given in the relevant set of Wikipedia guidelines, specifically, *incorrect categorisation* in [Wikipedia Online Editing Guidelines for Categories].



Table 5.8: Questions for OR1

<p><b>Questions for OR1:</b> Adequate intersection of categories.</p> <p><b>Q1:</b> Does the category structure have an adequate intersection of categories?</p> <p><b>Discussion:</b></p> <p>An adequate degree of intersectedness is sought in Q1. Q1 considers the degree of intersectedness in the category structure. Having not enough category intersection may result in an increased amount of browsing, which impedes the task of finding information. Conversely, if a given category is completely connected to all other concepts in the category structure, then the resulting category structure is as good as an index of all available categories, which is not useful.</p>
--

*Incorrect categorisation* refers to categories which have been defined in a non-factual manner and do not match with the view of the domain. This may occur due to erroneous modelled relationships between parent and children categories, or where a category has been incorrectly associated with one or more articles under it. This issue is about the correctness and therefore, we adopt the appropriate correctness criteria-question for this requirement and present this in Table 5.9.

### 5.3.3 Questions for ‘Avoiding cycles in the category structure’

For ontologies, avoiding cycles refers to resolving a consistency issue. Cycles occur because of inconsistent association of concepts in defining relationships. The appropriate criteria-question for cycles is presented in Table 5.10. Q3 is about detecting the cycle and once detected, the ontology engineer can then go about resolving those cycles.

### 5.3.4 Questions for ‘Ensure a complete set of categories’

For the Wikipedia category structure, ensuring a complete set of categories directly relates to the ontology evaluation criterion of completeness. In the context of this ontology requirement, a reason that a large number of categories are associated with a given article may be indicate

Table 5.9: Questions for OR2

<b>Questions for OR2:</b> Categories should be appropriately grouped
<p><b>Q2:</b> Does the ontology capture concepts of the domain correctly? (correctness)</p> <p><b>Discussion:</b></p> <p>In the Wikipedia documentation, a specific issue was highlighted with regard to the categories: <i>incorrect categorization</i>. Incorrect categorization occurs when there is a discrepancy between a category and its associated categories (such as subcategories of a category or parent categories of a given category) with respect to the world being modelled in. We have used a correctness criteria-question for Q2, which examines whether the relationships between categories have been correctly captured.</p>

Table 5.10: Questions for OR3

<b>Questions for OR3:</b> Avoiding cycles in the category structure
<p><b>Q3:</b> How many cycles are found in the ontology? (consistency)</p> <p><b>Discussion:</b></p> <p>This question examines whether the ontology has cycles. Cycles impede Wikipedia users in browsing and is undesirable in this application, as noted in the online Wikipedia guidelines.</p>

the poor definition of the set of categories for the category structure. The right category simply may not exist, thus, users associate multiple categories to the article being considered. To resolve this, we map the appropriate completeness criteria-question in Table 5.11.

*Table 5.11: Questions for OR4*

<b>Questions for OR4:</b> Ensure a complete set of categories
<b>Q4:</b> Does the ontology have concepts missing with regard to the relevant frames of reference? (completeness)
<b>Discussion:</b>
Q4 examines incompleteness of the ontology, that is, concepts that are missing with regard to the relevant frames of reference using a completeness criteria-question. These frames of reference may be a collection of a set of all possible concepts that are defined within a given domain.

### 5.3.5 Questions for ‘Ensure categories associated in articles are correct’

This requirement directly maps to the criterion of correctness, specifically of the relationships between a given article and an associated category. Thus, we specify a customised correctness criteria-question for this requirement in Table 5.12.

*Table 5.12: Questions for OR5*

<b>Questions for OR5:</b> Ensure categories associated in articles are correct
<b>Q5:</b> Is the set of categories correctly associated with a given article? (correctness)
<b>Discussion:</b>
Q5 examines the correctness of the set of categories associated for a given article based on a correctness criteria-question. The issue being investigated is ensuring that each article is associated with appropriate categories. This helps users view relevant articles as they are browsing each category.

## 5.4 Measures

In this section, we associate a set of measures with questions that were specified in the previous section. Each measure seeks to answer the question quantitatively. Accompanying each description of the measure is a discussion summarising its intent, an optimal value or range of values and a range of possible measurements for the measures proposed.

### 5.4.1 Measures for ‘Does the category structure have an adequate intersection of categories?’

For this question, the appropriate measure is *tangledness* and we use the formula for tangledness defined in Chapter 2 in Equation 2.15. Tangledness gives the measure of multiple parents to the number of categories in the structure.

The optimal values for tangledness of a category structure for Wikipedia may vary and is dependent on a given domain. Table 5.13 gives specific measurements taken relating to tangledness for the Wikipedia category structure.

Measure	Value
Number of categories	111,287
Number of parent categories	23,978
Categories with multiple parents	7,788
Tangledness	0.69

*Table 5.13: Some measures for the Wikipedia category structure*

Overall, the Wikipedia category structure has a tangledness value of 0.69. However, this tangledness value differs between the different subtrees. From an initial investigation of a set of subtrees, the tangledness value may vary between 0.2 and 0.8. Thus, what may be an optimal value in one domain may not apply in another domain. Because of this, we propose an optimal range rather than a single value to denote a desirable tangledness value of 0.2 to 0.8 . We present this measure and its discussion in Table 5.14.

Table 5.14: Measures for Q1

<b>Measures for Q1:</b> Does the category structure have an adequate intersection of categories?	
<b>M1:</b>	Tangledness: $t(\mathcal{O}_c)$ Optimal value(s): Between 0.2 and 0.8 Range: 0–1
<b>Discussion:</b>	
<p>Tangledness measures the ratio of multiple parent categories. This measure may help us understand how intersected the category structure is. Measurements for this measure are values between 0 and 1, indicating the level of category intersection (0 indicating no intersection and 1 indicating a completely intersected ontology). Using this measure, we will evaluate what the threshold for an appropriate value for indicating the adequate intersection of categories. Optimal value for this measure is yet to be determined, but having considered some categories and its tangledness value, between 0.2 and 0.8 is the proposed threshold.</p>	

### 5.4.2 Measures for ‘Does the ontology capture concepts of the domain correctly?’

For this question, we propose to use the meta-consistency ratio measure by Gangemi et al. [2005]. We use the definition presented in Equation 2.13. This measure gives the proportion of consistent categories to the number of categories in the structure. Thus, the range of values is between 0 and 1. In applying this measure for this application, the optimal value corresponds to 1 as this indicates that the entire subtree is meta-logically consistent. This can be used to evaluate the correctness of the relationships between concepts in the Wikipedia domain. Table 5.15 outlines this measure in the appropriate ROMEO template.

Table 5.15: Measures for Q2

<b>Measures for Q2:</b> Does the ontology capture the relationships between concepts of the domain correctly?	
<b>M2:</b>	<i>ratio of meta-consistent concepts</i> ( $\mathcal{O}_c$ )
	Optimal value(s): 1
	Range: 0–1
<b>Discussion:</b>	
This measure gives the ratio of concepts that are <i>correct</i> to the total number of concepts in the ontology $\mathcal{O}_c$ and may involve an OntoClean analysis of each relationship. In particular, the OntoClean analysis may help examine the meta-consistency ratio using meta-properties such as essence and rigidity to indicate incorrectly modelled categories and their subcategories.	

For this measure, the key component of the measure is *the number of meta-consistent concepts in the ontology*. This may be achieved by carrying out an OntoClean analysis on each concept in the ontology and determining the number of concepts that are found to be correct.

### 5.4.3 Measures for ‘How many cycles are found in the ontology?’

The appropriate measure for this question is the number of circularity errors. Gómez-Pérez [2001] gives three kinds of circularity errors that may be used to detect cycles, that were presented in Section 2.4.2. They are:

1. *Circularity error at distance of 0* (that is, with a given node)
2. *Circularity error at distance of 1* (that is, a cycle detected with a given node)
3. *Circularity error at distance of  $n$*  (that is, a cycle detected at  $n$  nodes away from a given node)

Equations 2.16, 2.17 and 2.18 give definitions for each of the respective measures above. The range of values of measures M3–5 can be large, thus the expected values are  $0-\infty$ . However, the optimal value for these measures is 0 as we require no cycles to be in the ontology. We incorporate it in Table 5.16 in Measures M3, M4, and M5.

Gómez-Pérez [2001] does not give any specific ways of detecting circularity errors at distance of 0, 1 and  $N$ . We propose the use of the Floyd [1967] cycle-finding algorithm, which lists all simple cycles in a directed graph to detect circularity error at any distance. We present the algorithm below in Algorithm 1.

### 5.4.4 Measures for ‘Does the ontology have concepts missing with regard to the relevant frames of reference?’

From the suggested table of mappings in Table 3.16, the measure of *incomplete concept classification* and *recall* map to this question. The measure of incomplete concept classification proposed by Gómez-Pérez [2001], is presented in Equation 2.25. However, it is up to the ontology engineer to determine this by manual inspection.

The recall measure considers the level of completeness with regard to the concepts defined in the ontology to a frame of reference, that is a reference point. The frame of reference is chosen based on the specific set of concepts for the domain being examined. We use the recall definition presented in Equation 2.29. The optimal value for this measure is 1. This indicates an ontology with a complete set of concepts defined with regard to a frame of reference.

Table 5.16: Measures for Question 3

<b>Measures for Q3:</b> Are there any cycles currently?	
<b>M3:</b>	Circularity error at distance 0 Optimal value(s): 0 Range: 0– $\infty$
<b>M4:</b>	Circularity error at distance 1 Optimal value(s): 0 Range: 0– $\infty$
<b>M5:</b>	Circularity error at distance N Optimal value(s): 0 Range: 0– $\infty$
<b>Discussion:</b>	
<p>Simply detecting a cycle is sufficient to answer the question. Upon detection, we examine and resolve those cycles. M3, M4 and M5 are proposed as measures for errors at various distances from a given node. Range of values for M3, M4, and M5 are non-negative integers. We can employ the use of cycle detection algorithms to assess circularity errors such as those proposed by Floyd [1967].</p>	



---

**Algorithm 1** The Floyd cycle-finding algorithm

---

Let  $initial := choice(n)$ Let  $old := initial$ **repeat**

Write old

 $new := choice(initial)$     **if**  $used[new] || step[old, new]$  **then**

return FAILURE

**else**        **if**  $new == initial$  **then**

return SUCCESS

**end if**    **end if**     $used[new] = true$      $old := new$ **until** no more nodes to process

---

Table 5.17: Measures for Question 4

<b>Measures for Q4:</b> Does the ontology have concepts missing with regard to the relevant frames of reference?	
<b>M6:</b>	<i>Incomplete concept classification</i> ( $\mathcal{O}, \mathcal{F}^*$ )
<b>M7:</b>	<i>average-recall</i> ( $\mathcal{O}_c, \mathcal{F}^*$ )
	Optimal value(s): 1
	Range: 0–1
<b>Discussion:</b>	
<p>There are two measures that map to this question. The measure of <i>Incomplete concept classification</i> (M6) involves determining whether there are concepts left out of the ontology from the frames of references, although this requires manually inspection between <math>\mathcal{O}</math> and <math>\mathcal{F}^*</math>. The recall measure (M7) can be used to answer the question on completeness by determining whether the set of concepts in the ontology <math>\mathcal{O}</math> are complete with respect to the relevant frames of reference <math>\mathcal{F}^*</math>. These frames of reference may be gathered from analysis of a domain using representative text corpora. The measurements of recall may range from 0 to 1. A recall value of 1 indicates a complete ontology and a value of 0 indicates that the ontology has no matching categories with the frames of reference.</p>	

### 5.4.5 Measures for ‘Is the set of categories correctly associated with a given article?’

For this question, two specific measures are mapped. The first measure considers the number of non-related associations between a given category and article. However, because a related association between a given category and article does not exclude bias and a non-NPOV, we map the second measure to determine the number of NPOV violations with regard to each association between article and category. Both measures have an optimal value of 0 as there should not be any violations and incorrect associations between articles and categories.

*Table 5.18: Measures for Question 5*

<b>Measures for Q5:</b> Is the set of categories correctly associated with a given article?	
<b>M8:</b>	Number of non-related associations between a given category and article Optimal value(s): 0 Range: 0–∞
<b>M9:</b>	Number of NPOV violations of a relationship between given category and article Optimal value(s): 0 Range: 0–∞
<b>Discussion:</b>	
<p>The two measures mapped to this question are very specific to the application requirements of Wikipedia for its category structure. M8 examines non-related relationships between a given category and article. Articles belonging to a category ought to have sufficient references in it to show it belongs to the category it is associated with. Thus, M8 may require verifying the relationship between category and article using reliable external sources. M9 examines the number of NPOV policy violations with regard to the relationship between a given article and category from the set of categories associated with an article. These violations include non-neutral point-of-views of a relationship between an article and a category which may display bias, or a personal opinion that has no factual basis.</p>	

Although we have proposed measures M8 and M9 for Q5, they require manual inspection

to distinguish non-related associations between a given category and article, and NPOV violations respectively. Given the size of the category structure for Wikipedia, it may not be feasible to measure this. Rather, a more realistic approach may be to allow users and editors to raise categories that require improvements with respect to the categorisation of articles and NPOV violations.

## 5.5 Summary

In this chapter, we considered Wikipedia as an application that uses an ontology for browsing articles – the Wikipedia category structure. We explored the requirements of this category structure and from its ontology requirements, we applied a ROMEIO analysis. We elicited requirements using existing documentation from Wikipedia itself as they were available and well documented. Also, since the category structure has been in use for more than three years, guidelines for it are very specific, thus requiring little further analysis.

Using the ROMEIO methodology, we mapped a set of questions to each requirement identified in the earlier step. This case study used a mix of criteria-questions and questions that were not criteria-questions. The questions that were specified apart from the set of criteria-questions, related to requirements of *adequate level of category intersection*, and *ensure categories associated in articles are correct*. The reason for this was because they were more specific to the requirements of case study themselves and may not necessarily apply for other ontologies. The ROMEIO analysis for Wikipedia is summarised in Table 5.19. Additionally, we present the validation of the mapping between the questions of ‘Does the category structure have an adequate intersection of categories?’ and the tangledness measure in Chapter 6.

In the next chapter, we will present an empirical method for validating selected mappings of questions to measures.

Table 5.19: Summary of the ROMEO analysis of ontology requirements for Wikipedia

<b>Requirement:</b>	Adequate level of category intersection
<b>Question:</b>	Does the category structure have an adequate intersection of categories?
<b>Measure:</b>	Tangledness: $t(\mathcal{O}_c)$
<b>Requirement:</b>	Categories should be appropriately grouped
<b>Question:</b>	Does the ontology capture the relationships between concepts of the domain correctly?
<b>Measure:</b>	<i>ratio of meta-consistent concepts</i> ( $\mathcal{O}_c$ )
<b>Requirement:</b>	Avoiding cycles in the category structure
<b>Question:</b>	How many cycles are found in the ontology?
<b>Measures:</b>	Circularity Error at distance 0, 1 and N
<b>Requirement:</b>	Ensure a complete set of categories
<b>Question:</b>	Does the ontology have concepts missing with regard to the relevant frames of reference?
<b>Measures:</b>	<i>Incomplete concept classification</i> ( $\mathcal{O}, \mathcal{F}^*$ ) <i>recall</i> ( $\mathcal{O}_i, \mathcal{F}^*$ )
<b>Requirement:</b>	Ensure categories associated in articles are correct
<b>Question:</b>	Is the set of categories correctly associated with a given article?
<b>Measures:</b>	Number of non-related associations between a given category and article Number of NPOV violations of a relationship between given category and article

## Chapter 6

# Empirical validation

*The proof of the pudding is in the eating.*

— Miguel de Cervantes

*It does not happen all at once. There is no instant pudding.*

— W. Edwards Deming

In Chapters 4 and 5, we used the ROMEO methodology to establish a set of mappings from requirements to questions, and from questions to measures. In this way, an appropriate ontology evaluation method is obtained for each application. However, this analysis is theoretical and based on some logical reasoning. Thus, these mappings require validation and we propose the use of empirical means to verify that a given mapping corresponds to the expected behaviour. We especially focus on the two new mappings discovered using ROMEO for the two case studies – a mapping from the Lonely Planet case study, which included a new measure we propose, and another mapping of the tangledness measure to a question of intersectedness from the Wikipedia case study.

In this chapter, we outline a process for the empirical validation of ROMEO mappings, and carry out this process on the two new mappings discovered from the case studies. This process examines the performance of tasks that are carried out using a set of experiments. Experiments provide a platform for comparing ontologies, which vary according to a specific ontology characteristic. In particular, we focus on mappings between questions and measures

in this chapter as they are not application specific and may be reusable in future applications of the ROMEO methodology. In Section 6.1, we outline a process for the empirical validation of ROMEO mappings. Drawing from the case studies presented in the Chapters 4 and 5, we carry out two validation experiments on mappings between questions and measures from each case study and present these in Sections 6.2 and 6.3, respectively.

## 6.1 The validation process

The validation process is used to verify a mapping of a requirement to a question or a question to a measure, that is established using the ROMEO methodology. The process validates a mapping using a set of tasks carried out in a controlled experimental setup to compare the performance of a given task on a set of ontologies. The validation environment may use an ontology-driven application to benchmark the performance of each task or direct measures to compare each ontology for each task. This step of validation is separate from the ROMEO methodology. However, it complements the ROMEO methodology in that it helps to validate the mappings used in a ROMEO analysis. Carrying out these validation experiments allows the ontology engineer to observe the actual working of the measures, that is, whether the right measures are used to measure the ontology characteristics from the ROMEO question and measure mapping.

### 6.1.1 The validation environment

The validation of a given mapping may be carried out using one of two types of validation experimental setup. The first type of experimental setup examines aspects of the ontology directly like a white-box test, for example, matching the concepts of an ontology with concepts in a given domain. Figure 6.1 illustrates the overview of this type of validation, where a set of tasks is performed in a validation environment on a set of ontologies that is varied according to a specific ontology characteristic. We then analyse measurements from the validation experiment. We use the first type of experimental setup in a case study presented later in Section 6.2.

Another type of experimental setup compares the performance of a set of tasks carried

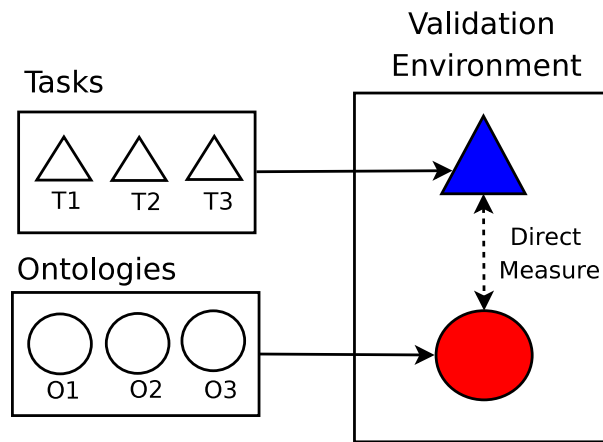


Figure 6.1: White-box validation process

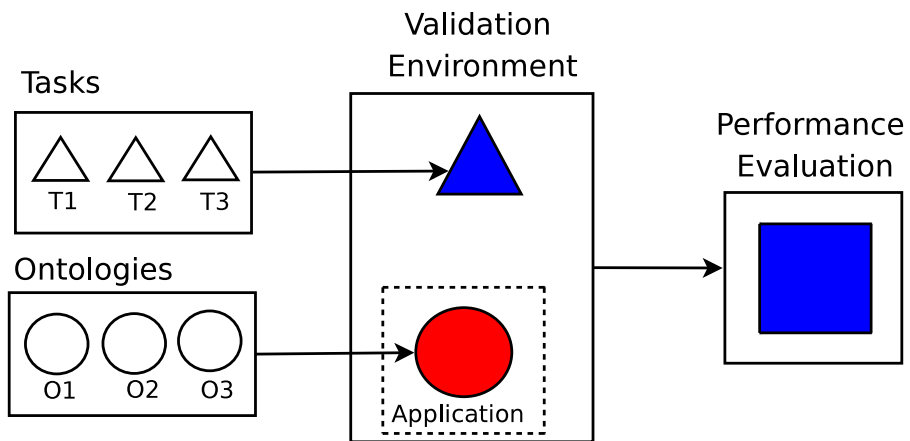


Figure 6.2: Black-box validation process



out on an ontology-driven application like a black-box test. Figure 6.2 illustrates this type of validation process, where a set of tasks is performed in a validation environment, which includes the use of a set of ontologies that is varied according to a specific ontology characteristic. The performance of each task is then benchmarked against each other. We then compare the effect that each ontology had on the tasks performed. We specifically examine whether an ontology that varies on a specific characteristic helps with the tasks. An experiment conducted in this way performs an indirect measurement of the ontology characteristic by varying the base ontology. This seeks to observe the performance of the base ontology and its variants in a given application environment. We use the black-box validation process to validate a mapping in Section 6.3.

### 6.1.2 Obtaining comparable ontologies

The set of ontologies used in empirical validation experiments ought to be comparable, for example, if we were validating the measure of an ontology's coverage, comparable ontologies should describe the same domain, rather than a variety of domains. They should vary according to the characteristic being considered in a way that is independent of any other characteristics.

The problem with obtaining comparable ontologies from a collection of existing ontologies is that they are not always available. Furthermore, concepts and relations in the ontology may be vastly different, as they may include aspects of other domains or have different levels of detail. Hence, we may not be able to make a fair comparison between these existing ontologies.

A solution to the above problem of obtaining comparable ontologies, is to vary an ontology to derive a set of ontologies. This solution takes a candidate ontology and systematically alters it to be varied according to a given characteristic or measure, for example, a given ontology may be altered to have an increased size. The drawback is that this approach may not be possible in every case. Not all ontology characteristics can be varied systematically. For example, ontology breadth is difficult to alter unless more children are introduced for a set of concepts or there is an increase in the set of concepts to allow for more breadth.

Nevertheless, for characteristics that are able to be varied systematically, the set of ontologies produced in this way can be used to compare the performance of ontologies in a set of empirical validation experiments.

### 6.1.3 Select appropriate tasks and benchmarking standards

Tasks are carried out in an experiment and each task involves the use of an ontology. Tasks should be selected that reflect the roles of the ontology in the given application, for example, for an application that uses ontologies for navigation, choose tasks which examine the effectiveness and efficiency of navigation using the ontology. Defining more than one task to examine the effect of each ontology may be necessary in order to explore the mapping in depth.

For performance comparisons, there must be a means of judging good and bad performance. In certain cases, initiatives in relevant fields of study may provide a medium to benchmark systems. In such cases, experiments may be conducted using the tasks, datasets and relevance judgements provided by these initiatives. For all other cases, the definition of a clear specification of success and failure on a given attribute is required for performance comparisons.

## 6.2 Validating granularity mapping

In this section, we validate a mapping of a question to a measure derived from the ROMEO analysis performed for the Lonely Planet application in Chapter 4. There we identified measures for the appropriate level of granularity of concepts and instances in an ontology. We propose a series of experiments to validate a mapping between the question of appropriate granularity for instances and the F-measure of the instances in the ontology with the domain, which is shown in Table 6.1.

This question examines whether an ontology has an appropriate level of granularity regarding its instances for the domain being modelled, that is, having enough detail regarding the defined set of instances, yet not too much for the domain being considered. The specific

<b>Question:</b>	(Q5) Does the ontology have an appropriate level of granularity with regard to its instances compared with the domain being modelled?
<b>Measure:</b>	(M5) $F\text{-measure}(\mathcal{O}_i, \mathcal{F}_i)$

Table 6.1: Mapping of Q5 to M5 from the Lonely Planet case study in Chapter 4

measure that maps to this question is the F-measure, that is the harmonic mean of precision and recall (as detailed in Equation 4.2).

In this section, we outline the setup for the validation experiment, the set of ontologies used and how they were obtained, and the set of text corpora used to define the tasks.. We present results from this experiment and discuss them in the context of validation.

### 6.2.1 Experimental setup

For this validation experiment, we utilise an experimental setup which examines the characteristics of the ontology directly. The experiment involves the comparison between instances defined in the ontology and instances found in a given domain. The ontologies used are derived from Lonely Planet’s in-house ontology of geographic places, specifically, this ontology describes the country of France and the city of Paris in France. The set of tasks is defined by processing text corpora from the full text of a set of Lonely Planet guidebooks in order to obtain a different set of terms for each task in the travel domain. We consider the terms in each text corpus to include the set of instances in this domain. If this mapping between question and measure is valid, we expect that ontologies with a finer level of granularity ought to perform tasks that possess an increased amount of detail or granularity better than with other tasks. Figure 6.3 illustrates the validation process for measuring an appropriate level of granularity using ontologies describing Countries and Cities, and in particular, France and Paris. We describe the ontologies and tasks defined in more detail below.

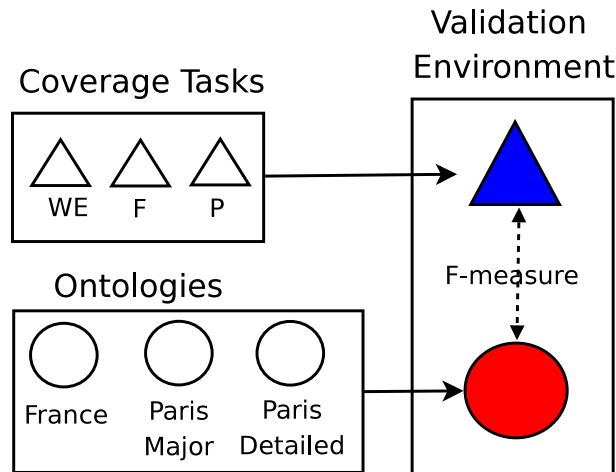


Figure 6.3: White-box validation process for the F-measure using Paris and France ontologies

### Ontologies used

Lonely Planet utilises an ontology for maintaining a consistent vocabulary within the organisation. The content of the ontology includes the names of places and activities used in their guidebooks. In particular, the ontology contains the names of geographic places around the world such as countries, provinces and cities. The geographic part of the ontology models every continent and country in the world. However, it does not model fine levels of detail for many of the countries. For example, most of the provinces of France are modelled in the ontology, but it does not model city- or town-level details. Thus, the modelling for this geographic part of the ontology is *course-grained*.

For this validation experiment, we adopt the geographic sections of the ontology as the base ontology that we vary to obtain a set of varied ontologies. This set of ontologies is obtained by adding increasing levels of detail to the base ontology, specifically, with regard to the geographic area of France and Paris. Thus, we are able to achieve different levels of granularity. The first ontology models instances of the regions in France (such as Brittany and Corsica), the major cities and towns (such as Paris, Nantes and Marseille), which is referred to as  $\mathcal{O}_{France}$ . We increase the granularity of  $\mathcal{O}_{France}$  by adding instances of geographic locations in Paris, in particular, locations in Paris that have major metro and train stations

(such as Bastille and Place d'Italie), which is referred to as  $\mathcal{O}_{ParisMajor}$ . The last ontology in the set of varied ontologies, is obtained by increasing the granularity to include all geographic locations in Paris taken from the Paris metro and train stations (such as Rue des Boulets and Porte de Versailles), as more detailed instances of places of Paris. This ontology is referred to as  $\mathcal{O}_{ParisDetailed}$ . Thus, the set of ontologies is such that  $\mathcal{O}_{France} \subset \mathcal{O}_{ParisMajor} \subset \mathcal{O}_{ParisDetailed}$ .

## Tasks

In validating the ROME mapping, we use a set of text corpora from the Lonely Planet travel guidebooks. A given text corpus used should have varying levels of granularity regarding travel in order to assess whether the F-measure does help to determine the appropriate granularity for this domain. As the text corpora we use vary with regard to their granularity, we use each text corpus and compare it against the set of varied ontologies.

This experiment takes travel text of Paris from three books with differing levels of detail to define the set of tasks performed. Task 1 uses the text corpus of the Paris sections in *Western Europe Travel Guide* (WE), which has the least geographic detail. Task 2 uses the text corpus of the Paris sections in the *France Travel Guide* (F), which has more detail than the text corpus used in Task 1. Task 3 uses the text corpus from *Paris City Guide* (P), which has most detail. For each task, the set of instances in an ontology is compared with terms extracted from the corresponding text corpus.

From this set of text corpora, we are able to extract terms, which we take to include the instances for this domain. We specifically applied stopping and then stemming (using the Porter [1980] stemmer) to the text corpus used for each task to extract the set of terms from a text corpus. Stopping and stemming were applied on the text labels from the instances of each ontology to extract the set of terms from an ontology.

### 6.2.2 Results

Below, we present results from the set of experiments described earlier, in which we use the F-measure to compare the granularity between instances in the ontology and instances from

the domain.

Table 6.2: Results for ontology and corpus size, number of overlapping terms, precision, recall and the F-measure

Corpus	$ F $	Ontology	$ O $	$ O \cap F $	Precision	Recall	F-measure
<i>WE-Paris</i>	2792	France	739	265	0.3586	0.0949	0.1501
		Paris Major	852	360	<b>0.4225</b>	0.1289	0.1976
		Paris Detailed	1268	<b>533</b>	0.4204	<b>0.1909</b>	<b>0.2626</b>
<i>France-Paris</i>	7142	France	739	448	0.6062	0.0627	0.1137
		Paris Major	852	559	<b>0.6561</b>	0.0783	0.1399
		Paris Detailed	1268	<b>831</b>	0.6554	<b>0.1164</b>	<b>0.1976</b>
<i>Paris City</i>	11953	France	739	535	0.7240	0.0448	0.0843
		Paris Major	852	648	<b>0.7606</b>	0.0542	0.1012
		Paris Detailed	1268	<b>957</b>	0.7547	<b>0.0801</b>	<b>0.1448</b>

Table 6.2 shows a count of the set of terms extracted from each ontology and text corpus, the set of overlapping terms and the associated measures. From this, we observe that the sizes of the three text corpora differ greatly – the Paris chapter in the *Western Europe Travel Guide* being the smallest and the *Paris City Guide* having the most terms. When we add the names of the major metro and train stations to the base ontology ( $\mathcal{O}_{France}$ ) to form  $\mathcal{O}_{ParisMajor}$ , there is a 15% increase in the number of terms (113 terms) compared with  $\mathcal{O}_{France}$ . When we include the names of all of the metro and train stations in the ontology in  $\mathcal{O}_{ParisDetailed}$ , there is an increase of nearly 50% (416 terms) in the number of terms compared with  $\mathcal{O}_{ParisMajor}$ . However, we observe that this does not necessarily produce an equivalent increase in the number of overlapping terms extracted from the ontology and the text corpus. Also, in Table 6.2, the results highlighted in bold show which is the most suitable ontology for a given corpus using each measure.

The graph in Figure 6.4 shows results for granularity using the F-measure. Graphs (b) and (c) in Figure 6.5 show results for recall and precision respectively. Precision considers how much of ontology overlaps with the domain, while Recall measures how much of the domain overlaps with the ontology. We observe that recall improves when more detail is added to the ontology, and that precision improves when adding the names of the major

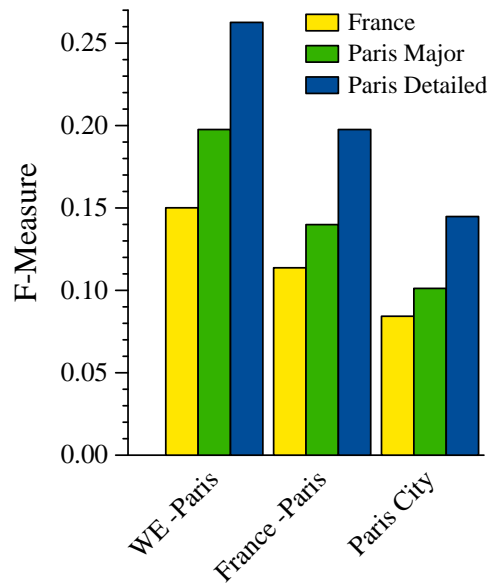


Figure 6.4: (a) Results showing granularity using the F-measure

Paris metro and train stations. However, when all of the names of the stations were added, that is in  $\mathcal{O}_{ParisDetailed}$ , precision starts to decrease slightly. This may indicate that the level of detail was too much for all three text corpora. Although, given the sizes of the ontologies and the difference in precision, it may be a significant difference. A key difference between  $\mathcal{O}_{ParisMajor}$  and  $\mathcal{O}_{ParisDetailed}$  is the number of overlapping concepts between  $\mathcal{O}_{ParisMajor}$  and the respective text corpora. The F-measure combines aspects of both recall and precision. We observed that the F-measure score increases as the granularity of the ontology increases, and seems to be dominated by the recall component shown in Graph (b) in Figure 6.5.

### Outcomes of validation experiment

An appropriate ontology should contain sufficient level of detail to model all instances down to the appropriate level of granularity for the application, while excluding unnecessary detail. We expected to observe two behaviours using the F-measure to determine adequate granularity. First, we expected tasks that used a text corpus that was coarse-grained, for

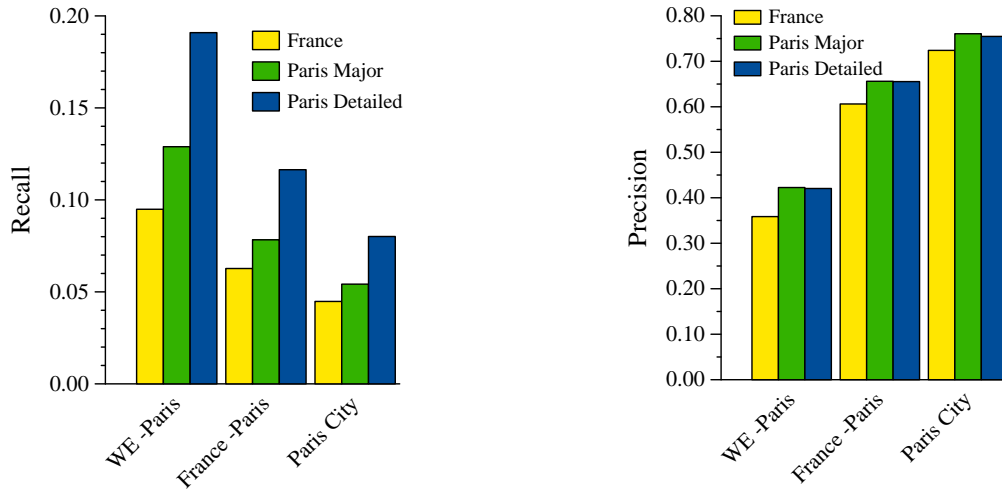


Figure 6.5: (b) Recall and (c) Precision

example, Task 1 (WE), to perform better against an ontology that is also coarse-grained, for example,  $\mathcal{O}_{France}$ , compared with an ontology that is fine-grained, for example,  $\mathcal{O}_{ParisDetailed}$ . Secondly, we expected that tasks which used a fine-grained text corpus, for example, Task 3 (Paris City), performed better against a given ontology that is also fine-grained, for example,  $\mathcal{O}_{ParisDetailed}$ , compared with an ontology that is coarse-grained, for example,  $\mathcal{O}_{France}$ .

Another observation is that the three text corpora used in the set of experiments are not the most varied in terms of geographic granularity. In other words, the set of text corpora are about the same level of specificity. From Table 6.2, we observe that the number of overlapping terms between the respective ontology and text corpora are relatively stable. For example, the finest grained ontology  $\mathcal{O}_{ParisDetailed}$  varies between 533 to 957 terms across the set of text corpora, which is not a big variation given the size of the text corpora. It is worth noting at this point that the text corpora is not restricted to geographic place detail, but may include terms related to activities at particular geographic locations.

What the results show then, is that  $\mathcal{O}_{ParisDetailed}$  is the most suitable ontology for this level of specificity, while the other ontologies are not, as determined by the measures of Number of Overlapping Terms, Recall, and the F-measure. In terms of the validation aims,



the F-measure is validated and identified as an appropriate measure as determined by the respective ROMEO analysis.

From the results of this experiment, the other two measures – Number of Overlapping Terms and Recall, are equally valid candidate measures for this case study. However, given a set of text corpora with more varied levels of geographic specificity, we would expect that F-measure would be the better measure to use as it combines both precision and recall. This may be an avenue of future work.

### 6.3 Validating intersectedness mapping

In this section, we validate a mapping of a question regarding *adequate category intersection* to the measure of *tangledness*. This mapping is taken from the ROMEO analysis performed on the Wikipedia application in the Chapter 5 and is shown below in Table 6.3.

<b>Question:</b>	(Q1) Does the category structure have an adequate intersection of categories?
<b>Measure:</b>	(M1) Tangledness: $t(\mathcal{O}_c)$

Table 6.3: Mapping of Q1 to M1 from the Wikipedia case study in Chapter 5

This question examines the adequate intersection of categories in the ontology, or in this case, the category structure. The appropriate measure for this question is tangledness, as determined in the ROMEO analysis considered in Chapter 5 with Wikipedia. The definition of the *tangledness* measure is the proportion of nodes in the graph that have more than one parent (as defined in Equation 2.15).

This validation experiment examines the mapping between the question of adequate category intersection with the tangledness measure. This experiment involves a group of users browsing an information space using a given category structure – much in the same way users would do when browsing categories from Wikipedia. The set of tasks used includes different browsing task types. As we are examining tangledness, the set of ontologies we use in this experiment varies in tangledness. If this mapping is valid, we will observe a corresponding variation of the performance on the tasks carried out.

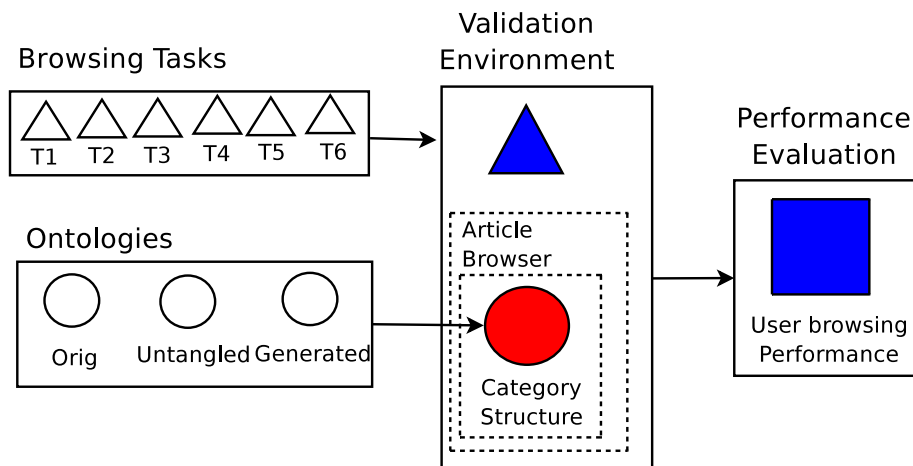


Figure 6.6: Black-box validation process for the tangledness measure on Wikipedia category structure

In this section, we present the experimental design for the evaluations that are undertaken, specifically, we describe the dataset used and the ontologies that were taken from Wikipedia’s category structure. Lastly, we present outcomes from a user study we undertook for validating our mapping of the adequate category intersection question to the tangledness measure.

### 6.3.1 Experimental setup

The experimental setup used here compares the performance of a set of ontologies that vary on tangledness. This experiment involves a user study that observes the performance of users browsing Wikipedia articles using the set of varied ontologies. The goal was to examine the *browsability* of an ontology in completing a range of tasks to find information in the set of Wikipedia articles, that is, the ability of users to locate information using the category structure for articles for a given information need by browsing. The browsing environment allows users to be able to explore the a given category structure. The set of Wikipedia articles is taken directly from the Wikipedia database, which is available online<sup>1</sup>. Figure 6.6 summarises our task based evaluation for validating this ROMEO mapping.

<sup>1</sup><http://download.wikimedia.org/enwiki>

## Ontologies used

For this user study, we used the category structure and the associated set of articles from the English language version of Wikipedia. The Wikipedia *category structure* is obtained from System One's RDF representation of it, also available online<sup>2</sup>. This was constructed from the Wikipedia database dated March 2006, in which each article and category is represented as an RDF triple with category and inter-article relations. The relations represented in the Wikipedia categories are *category-subcategory*, *category-article* and *article-article relations*. For a given category, no restrictions are put on the number of parent and subcategories. There may be multiple parent and child categories. Also, there are no restrictions on the number of categories to associate an article with (as long as they are related). However, there are some limitations with regard to the Wikipedia categories. Some categories are administrative in nature, for example, '*Sporting stubs*'. An article in such a category has no information written for it but has been linked from another article previously. Also, a given article may not have any categories associated with it. This means that some articles are not viewable from navigating the category structure. Despite this, the Wikipedia category structure is a rich organisation, and is used here as the basis for the validation experiment.

In processing the categories, we traversed the subtree in breadth-first search fashion starting from the category '*Categories*', which we take to be the root of the content section, to obtain a set of measures of the Wikipedia category structure. We present these measures in Table 6.4.

From Table 6.4, we observe that the Wikipedia categories have a ratio of about 1:10 with regard to the number of categories and their associated articles. Also, we find that the category structure is not deep considering the number of articles and categories, with the number of levels as 14. Instead, we find it to be quite broad with an average breadth of 8559.5 in a given level. The overall Wikipedia category structure is also quite tangled with 69% of all Wikipedia categories having multiple parents.

For this user study, we needed to obtain a set of ontologies which vary on tangledness. Additionally, these ontologies had to be based on the original subtree, semantically reason-

---

<sup>2</sup><http://labs.systemone.at/wikipedia3>

Measure	Value
Number of categories	111287
Number of articles	1079246
Average articles per category	25.7
Levels	14
Categories with multiple parents	7788
Number of parents	23978
Average no. parents	2.0
Maximum parents for any given child	56
Number of leaf categories	87309
Average no. children	4.64
Maximum children	1760
Avg breadth	8559.5
Maximum breadth	33331
Avg depth	5.8
Maximum depth	13
Fanout factor	0.78
Tangledness	0.69

Table 6.4: Measures for the Wikipedia category structure

able, utilised all the categories in the subtree and was comparable to the original subtree. We were faced with two options – either vary the original Wikipedia subtree or generate a subtree category structure according to an automated technique – which was a variation on a document clustering technique. We carried out both methods for producing untangled ontologies in this experiment. We present the two methods below.

**Method for removing tangledness** Removing tangledness meant removing occurrences of multiple parents in a given category. The specific algorithm we use is *Dijkstra’s algorithm* for finding a single-source shortest path tree. This is the most appropriate shortest path algorithm available as we know the root of the subtree. Where there were more than one parent candidate category, we chose the category that was most similar to the category being considered. For the similarity measure here, we used the *cosine similarity* from *TF-*

*IDF* measures of the article titles within the categories considered. We found this kept the subtree semantically equivalent. Using this method, we obtain a varied subtree for each domain.

**Method for generating subtrees** For a given subtree of the Wikipedia category hierarchy, we removed all category relations from it and applied a document clustering technique over the categories contained in the base subtree. We used *partitional-based criterion-driven document clustering* based on features gathered from a combination of the category title and associated article information [Zhao and Karypis, 2005] provided in the Cluto clustering toolkit<sup>3</sup>. Algorithm 2 describes the pseudocode for varying the category subtree.

---

**Algorithm 2** Varying a subtree

---

```

Let  $N :=$  maximum number of elements in cluster
Add root of subtree to queue  $q$ 
repeat
  Let  $cl :=$  next item in  $q$ 
  Obtain clusters  $I$  for  $cl$  from elements in its cluster
  for all  $i$  in  $I$  do
    Nominate element in  $i$  as representative category  $r$  for  $i$ 
    Add  $r$  as subcategory of  $cl$ 
    Let  $clustersize :=$  number of elements in cluster  $cl - 1$ 
    if  $clustersize \geq N$  then
      Add  $i$  to queue
    end if
  end for
until queue has no more clusters to process

```

---

We retain the category title as the concept name and performed clustering using the following as input parameters: category title, category title and the associated article titles, and category title and the associated article text. We also varied the clustering technique

<sup>3</sup><http://glaros.dtc.umn.edu/gkhome/views/cluto>

based on the number of features considered and also the resulting number of clusters on each clustering event. We used the *cosine similarity function* for this.

Using the two methods discussed above, we obtained two varied subtrees for a given domain. Table 6.5 presents the original and varied subtrees we obtained. We use Subtrees *a*, *b*, and *c* to look at the effects on tangledness.

Label	Subtree
<i>a</i>	Wikipedia original
<i>b</i>	Wikipedia original (remove tangledness)
<i>c</i>	Generated (untangled)

Table 6.5: Subtrees used in validation experiments

## Tasks and domains

We outline the tasks and domains we use in our user studies below. First, we discuss some background to a browsing activity from literature, and determine what types of tasks are appropriate to perform in the validation experiment.

**Browsing** A browsing activity is different from a *search* activity. Both have goals in mind, however, Baeza-Yates and Ribeiro-Neto [1999] differentiate *search* from *browse* by the clarity of user goals. In *search*, users enter into a system keywords that are related to their information need. They are then presented with a list of results the system returns as relevant and users can decide to select one of the results or refine their search query. In comparison to this, *browsing* is a different type of behaviour. There may not be a specific query as such associated. However, answers to user goals and information needs can be readily recognised in a browsing activity. Thus, it is the clarity and mode of accessing this information that differs in *browsing*.

Marchionini [1995] outlines the range of browsing types from a *directed* browse to an *undirected* browse. A directed or intentional browsing behaviour is usually associated with tasks that are *closed or specific*. These refer to a task where there are usually not more

than a few answers to the information need. On the other hand, an undirected browse is exploratory in nature and this browsing behaviour is associated with tasks that are more *open or broad*. These refer to a task where there may be many answers to the information need.

The efficiency of browsing is affected by the user’s knowledge of the domain and the specificity of the browse task. It is characterised by movement. Thompson and Croft [1989] describe browsing as an “informal or heuristic search through a well connected collection of records in order to find information relevant to one’s need”. In a browsing activity, users evaluate the information that is currently displayed, its value to their information need, and what further action to take.

The type of browsing that users would perform on the Wikipedia category structure is “Structure Guided Browsing” [Baeza-Yates and Ribeiro-Neto, 1999] and can encompass broad and undirected browsing to directed and specific browsing. Thus, the tasks that were used incorporated the different kinds of browsing behaviour which was discussed above.

**Browsing Tasks** Each participant was given a set of tasks to complete within a 10 minute duration (inclusive of pre- and post-task questions)<sup>4</sup>. The given tasks were domain specific, and hence would not be comparable in other domains. We chose to use domains that were as separate from each other as possible so as to reduce the learning effect from completing tasks on a given domain. Also, we chose three levels of specificity regarding the nature of the tasks (see Table 6.6). We proposed Tasks 1 to 3 and Tasks 4 to 6 to have increasing levels of specificity, from broad to specific, in their respective domains X and Y. For example, *International racing competitions* (Task 1) covered a broad range of possible answers within the Racing Sport domain (X). Whereas *Makers of F1 racing cars* (Task 3) was a very specific task type in the same domain.

Table 6.7 below outlines the task sequence for each user for the experiment we used to compare various aspects of the generated subtrees. In Table 6.7, the original Wikipedia subtree *a* is compared with the same subtree altered to remove multiple parents *b*, hence

---

<sup>4</sup>The handout given to users is included in Appendix E

Domain	Task description
Racing Sport (X)	T1: International racing competitions
	T2: Racing sports without wheeled vehicles
	T3: Makers of F1 racing cars
Foods (Y)	T4: Non-alcoholic beverages
	T5: Different cuisines of the world
	T6: Wine regions in Australia

Table 6.6: Tasks used for comparing ontologies for browsing articles

being *untangled*. We considered an additional subtree for this experiment, which appears as Subtree *c*. This subtree was generated using a document clustering technique.

For this experiment, we used the *Latin squares* method of determining in what order the participants use the subtrees to be compared. We did this to remove the learning factor of users progressing from one subtree to another in a given domain. Using this configuration each user has a unique task sequence. We also applied blocking on the domain. Lastly, we rotated the domain after nine users.

### 6.3.2 Analysis of varied ontologies

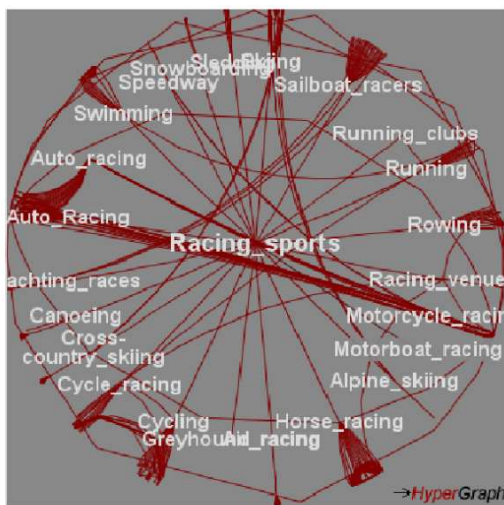
After varying the subtree for each of the two domains, we took measurements on these to analyse the changes and present them in Table 6.8. The Racing sports domain (X) has 1185 categories. The Foods domain (Y) has 652 categories. These were ideal sizes for the time given to each user to browse through in that they were sufficiently large such that users would probably not look at all categories. The average number of articles per category is 15 and 20.

We observe that for each domain, Subtree *b* does not have any multiple parents. Having an untangled subtree reduces the total number of parents compared with the Wikipedia original subtree (Subtree *a*). The generated subtree (*c*) had fewer levels as they were generally broader than the others. The effect of this is presenting the user with about twice as many narrower category links compared with the other subtrees. Figures 6.7 and 6.8 presents a visualisation of the original and untangled subtrees used in each domain.



Participant	X			Y		
	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>
user 1	t1	t2	t3	t4	t5	t6
user 2	t2	t3	t1	t5	t6	t4
user 3	t3	t1	t2	t6	t4	t5
	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>
user 4	t1	t2	t3	t4	t5	t6
user 5	t2	t3	t1	t5	t6	t4
user 6	t3	t1	t2	t6	t4	t5
	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>
user 7	t1	t2	t3	t4	t5	t6
user 8	t2	t3	t1	t5	t6	t4
user 9	t3	t1	t2	t6	t4	t5
	Y			X		
	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>
user 10	t4	t5	t6	t1	t2	t3
user 11	t5	t6	t4	t2	t3	t1
user 12	t6	t4	t5	t3	t1	t2
	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>
user 13	t4	t5	t6	t1	t2	t3
user 14	t5	t6	t4	t2	t3	t1
user 15	t6	t4	t5	t3	t1	t2
	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>
user 16	t4	t5	t6	t1	t2	t3
user 17	t5	t6	t4	t2	t3	t1
user 18	t6	t4	t5	t3	t1	t2

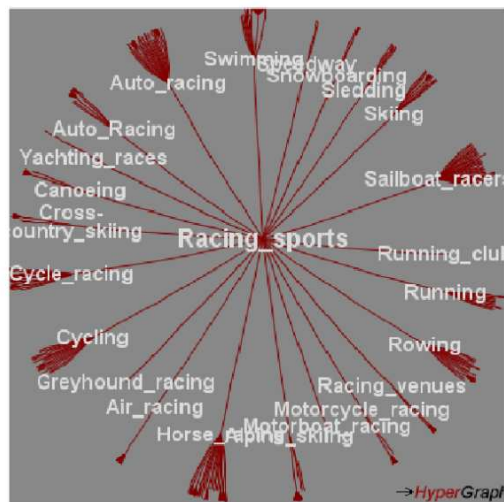
Table 6.7: Task sequence of Tasks 1 to 6 (t1-t6) for each user in validation user experiment comparing Subtree *a* (base), *b* (untangled) and *c* (generated)



**Base**



**Base**



**Base (Untangled)**



**Base (Untangled)**



**Autogen**



**Autogen**

Figure 6.7: Racing Sports subtrees

Figure 6.8: Foods subtrees

Measure	Racing Sports (X)			Foods (Y)		
Number of categories	1185			642		
Number of articles	18178			12630		
Avg articles per category	15.3			19.7		
	Subtree			Subtree		
	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>
Levels	7	7	4	9	9	3
Number of parents	305	213	292	187	135	51
Categories with multiple parents	293	0	0	132	0	0
Average no. parents	1.3	0.9	0.9	1.2	0.9	0.9
Maximum no. parents for a given child	5	1	1	4	1	1
Leaf nodes	880	972	893	455	507	580
Average children	4.9	5.6	4.1	4.2	4.8	12.4
Maximum children	54	53	20	48	48	51
Tangledness	0.25	0.00	0.00	0.21	0.00	0.00

Table 6.8: Characteristics of the set of subtrees used

### 6.3.3 Benchmarking

To benchmark the performance of users with regard to browsing and marking relevant articles for a given task, we observed the browsing efficiency and effectiveness of users. For efficiency, we looked at the number of backtracking actions a user does. Included are the number of clicks a user made to:

1. go back to the **previous** category
2. go back to the **top** category
3. click on a **past category or article** from history links

For effectiveness, we considered the number of relevant articles users marked for each task. For each article marked, we evaluated the marked article as:

**Not relevant:** does not relate to the task

**Somewhat-relevant:** has bits of relevant information

**Mostly-relevant:** most of article is relevant

**Definitely-relevant:** all of article is relevant

### Significance testing

For the significance testing, we used a two-tailed unpaired unequal variance t-test. The p-value indicates the probability that the values for the users' performance for the specific comparison are from the same distribution. We may consider the performance of a given subtree to be different from another with statistical significance if the p-value is lower than 0.05. That is, there is less than 5% chance that the two distributions are from the same population.

#### 6.3.4 Results

The users participating in this study ranged from 20 to 35 years of age, and included both females and males. From the pre-study questionnaire, the sample of users had an equal mix of English as a the first language and English as a second language, and were moderately proficient with using the web to find information and on average performed this activity many times a day with about 5 to 15 minutes spent per session. From the pre-study questionnaire, it was noted that on average the sample of users indicated that they had a moderate but not expert knowledge of domain. Users found the interface fairly easy to use however, on average, they indicated that the duration allocated for the tasks overall were not sufficient.

We present the results of the user study experiment below. Specifically, we present the major performance comparisons for tasks that users completed using the set of ontologies. The results for the user study experiments carried out are summarised in the graphs in Figures 6.9, 6.10, and 6.11. Figure 6.9 presents an overview of the main differences between the results of the tasks undertaken by users. The graph shows the results for relevant articles found and the amount of backtracking clicks performed by users in completing the set of tasks. Figure 6.10 outlines detailed results showing a comparison of the average clicks of users for a given task on a given system and the breakdown of those clicks into average backtracks made, average number of category clicks, average number of article clicks and other clicks. "Other clicks" refer to clicks users made to mark relevant articles and view marked articles. Figure 6.11 shows a comparison between systems on a given task for relevant articles retrieved by a user. The breakdown on each bar includes articles ranging from definitely relevant to

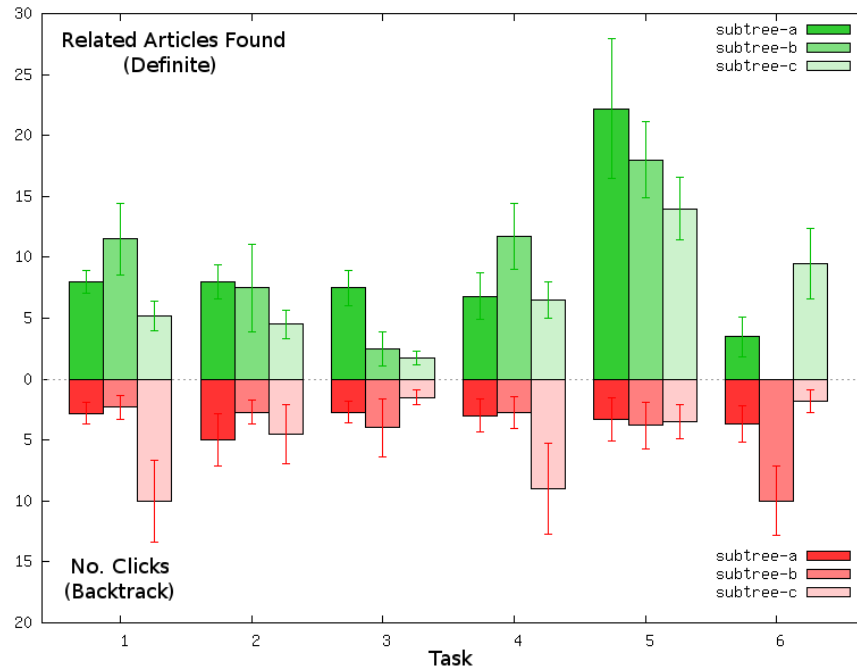


Figure 6.9: Relevant articles found (definite) vs. Backtracking clicks for Subtrees *a* (base), *b* (untangled), and *c* (generated)

not relevant at all.

### Best method for obtaining untangled ontology

Both Subtree *b* and *c* are ontologies that are not tangled, that is, do not have any multiple parents. For Subtree *b*, the resulting category organisation for each domain is based on the original Wikipedia category structure. Subtree *c* is a generated category organisation based on a clustering technique. The resulting category organisation for Subtree *c* was selected from several versions that is varied on different parameters and the best subtree was chosen on the basis of the number of *category-subcategory* relations in the generated subtree (*c*) that appeared in the original Wikipedia subtree (*a*). This tended to yield broad subtrees using our method for generating subtrees. We also observed that broader subtrees had an effect in reducing subtree depth. The results from Figures 6.10, 6.11 show that users generally performed the set of tasks better using Subtree *b* than with Subtree *c*. The exception

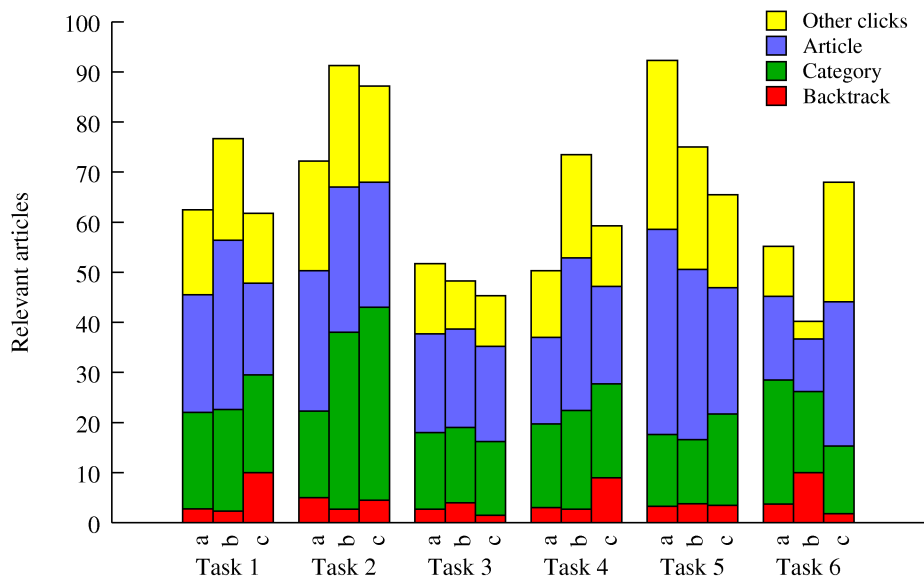


Figure 6.10: Average number of clicks by users for Subtrees a (base), b (untangled), and c (generated)

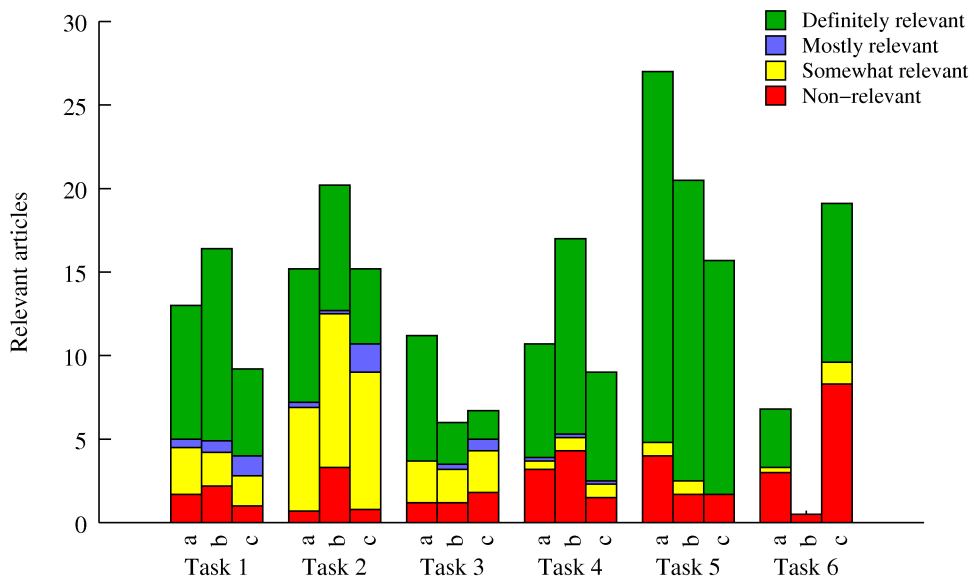


Figure 6.11: Number of relevant articles retrieved for Subtrees a (base), b (untangled), and c (generated)

is the results for users performing Task 6 and we account for this in further discussion of results later in this section. Because of the difference in performance, we limit our analysis to comparing tangledness using the better ontology (Subtree *b*) with the original ontology (Subtree *a*).

### Comparing Subtree *a* (base) and Subtree *b* (untangled)

**Results for each task** We summarise our findings in Tables 6.9 and 6.10, and give further details on the main measures for each subtree and include significance tests. The major rows in Table 6.9 are grouped by individual tasks followed by task types in Table 6.10.

**Results for each task type** The minor rows are grouped into backtracking clicks and measures of relevant articles found. We sum up measures of definitely-, mostly- and somewhat-relevant articles found and present this in the table as relevant articles found. The first column shows the task comparisons. We list the corresponding averages for each measure on each subtree beside it. The last column in each of Tables 6.9 and 6.10, present p-values for the significance tests carried out on each measure.

**Specific observations** A table of results with more specific observations from the experiment of significant differences and is presented in Table 6.11 below. For each observation that is included in Table 6.11, results for both Subtree *a* and *b* are presented as well as the p-value from the t-test conducted. The last column in this table shows the p-values from the significance test using the two-tailed unpaired unequal variance t-test.

Overall, we found that in comparison to Subtree *b* (untangled), Subtree *a* (original) enabled users to be more efficient in finding relevant articles. Of the set of articles viewed, the number of definitely-relevant answers found in Subtree *a* was 6% better than Subtree *b*. This was statistically significant.

Looking at the individual tasks in Table 6.9, the main differences found between Subtree *a* and Subtree *b* were highlighted in Task 3 and Task 6. Of the set of tasks, Task 3 and 6 were the most specific.

Table 6.9: Results and significance tests for Subtrees a (base) and b (untangled)

Task	Measure	Subtree		t-test
		a	b	a-b
1	% backtrack	4.48%	3.00%	0.4641
	Definitely	8	11.5	0.2782
	Mostly	0.5	0.7	0.6867
	Somewhat	2.8	2	0.3813
	Relevant	11.3	14.2	0.4237
	Non-relevant	1.7	2.2	0.6890
2	% backtrack	6.93%	2.96%	0.3243
	Definitely	8	7.5	0.8996
	Mostly	0.3	0.2	0.6643
	Somewhat	6.2	9.2	0.4524
	Relevant	14.5	16.8	0.5638
	Non-relevant	0.7	3.3	0.1350
3	% backtrack	5.22%	8.28%	0.6076
	Definitely	7.5	2.5	0.0309*
	Mostly	0	0.3	0.3409
	Somewhat	2.5	2	0.5490
	Relevant	10	4.8	0.0339*
	Non-relevant	1.2	1.2	1.0000
4	% backtrack	5.96%	3.67%	0.3928
	Definitely	6.8	11.7	0.1758
	Mostly	0.2	0.2	1.0000
	Somewhat	0.5	0.8	0.5413
	Relevant	7.5	12.7	0.1909
	Non-relevant	3.2	4.3	0.5849
5	% backtrack	3.58%	5.07%	0.6880
	Definitely	22.2	18	0.5374
	Mostly	0	0	-
	Somewhat	0.8	0.8	1.0000
	Relevant	23	18.8	0.5361
	Non-relevant	4	1.7	0.2198
6	% backtrack	6.70%	24.88%	0.0065*
	Definitely	3.5	0	0.0571
	Mostly	0	0	-
	Somewhat	0.3	0	0.1449
	Relevant	3.8	0	0.0437*
	Non-relevant	3	0.5	0.0646

\* denotes  $p < 0.05$



Table 6.10: Results and significance tests combining tasks for Subtrees a (base) and b (untangled)

Task	Measure	Subtree		t-test
		a	b	a-b
1 & 4	% backtrack	5.17%	3.33%	0.2462
	Definitely	7.4	11.6	0.0663
	Mostly	0.3	0.4	0.7314
	Somewhat	1.7	1.4	0.6985
	Relevant	9.4	13.4	0.1193
	Non-relevant	2.4	3.3	0.4981
2 & 5	% backtrack	5.07%	3.91%	0.6713
	Definitely	15.1	12.8	0.6082
	Mostly	0.2	0.1	0.6591
	Somewhat	3.5	5.0	0.5348
	Relevant	18.8	17.8	0.8140
	Non-relevant	2.3	2.5	0.8979
3 & 6	% backtrack	5.93%	15.82%	0.0223*
	Definitely	5.5	1.3	0.0066*
	Mostly	0.0	0.2	0.3282
	Somewhat	1.4	1.0	0.4919
	Relevant	6.9	2.4	0.0169*
	Non-relevant	2.1	0.8	0.0831

\* denotes  $p < 0.05$

Measure	Wikipedia		t-test
	Orig (a)	Untangled (b)	
<b>Overall</b>			
% Definitely-relevant articles / Total no. articles	40%	34%	0.0375*
<b>Domain X</b>			
% Categories with marked articles / Number of Visited categories	55.4%	41.7%	0.0143*
<b>Task 3</b>			
Categories with marked articles	3.5	1.7	0.0199*
Definitely-relevant articles	7.5	2.5	0.1133
<b>Task 6</b>			
% backtracking	6.7%	24.9%	0.0041*
Path length	16.2	7.5	0.0010*
Unique categories visited	24.8	16.2	0.0101*
Marked articles	6.8	0.5	0.0041*
Relevant articles	10.8	0	0.0692

\* denotes  $p < 0.05$

Table 6.11: Results for Subtree a vs. b

In Task 3, users were asked to find articles about ‘*Formula one car makers*’. Here we found that users marked articles from a more diverse set of categories using Subtree *a* than with Subtree *b*. On average, users marked articles in 3.5 subcategories when browsing with Subtree *a* compared to users marking articles in 1.7 subcategories when browsing with Subtree *b*. This was statistically significant according to the t-test.

On average, users browsing with Subtree *a* had 7.5 definitely relevant articles compared to users browsing with Subtree *b*, which had 2.5. However, this was not statistically significant. Upon closer inspection of the category structure, the Formula One section of Subtree *a* had many categories with multiple parents that were related which explains how users were able to browse more effectively. We also found that users performed three times better using Subtree *a* in finding more relevant articles. Figures 6.12 and 6.13 shows the contrast between the two subtrees for the relevant section of the ontology. The lines in the respective figures denote a subcategory-parent category relationship between two categories. In Figure 6.12, we observe a more tangled structure around the ‘Auto-racing’ category and the key category of ‘Formula One Constructors’, which indicates multiple entry points to those categories. In Figure 6.13, we observe less tangledness and a more linear sequence in navigating to the key category of ‘Formula One Constructors’.

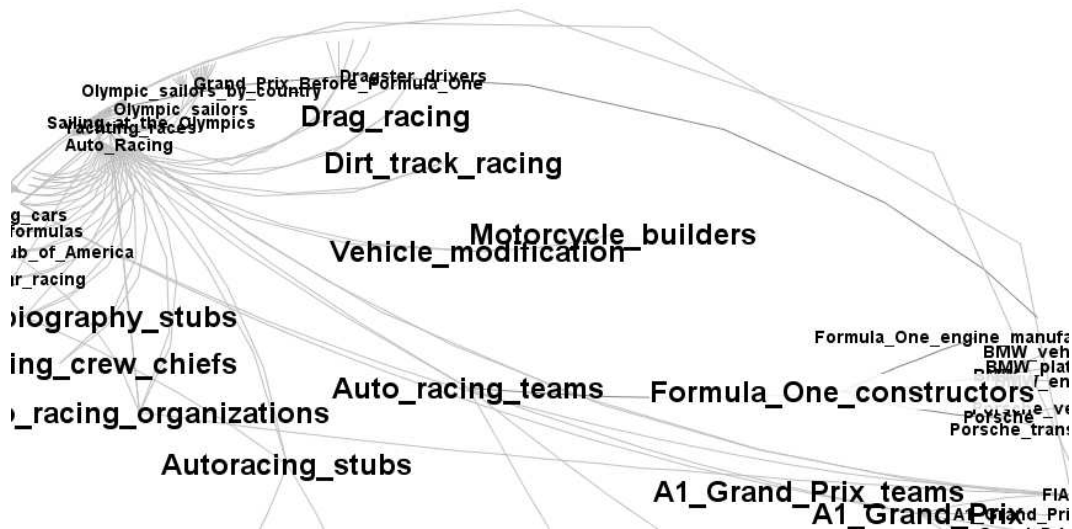


Figure 6.12: Visualisation of an excerpt from Subtree *a* for Racing Sports

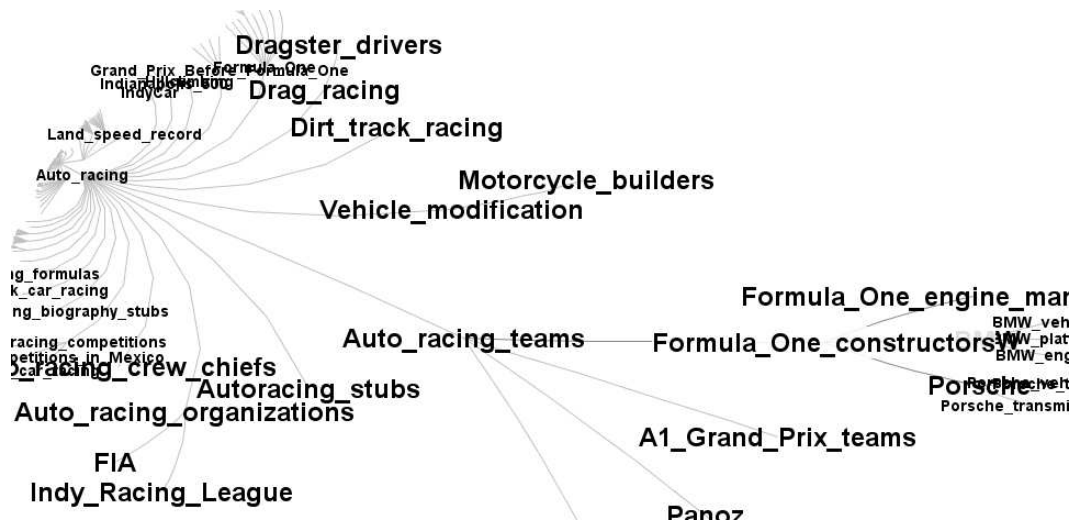


Figure 6.13: Visualisation of an excerpt from Subtree *b* for Racing Sports

In Task 6, users were asked to find wine regions in Australia. Subtree *a* did significantly better than Subtree *b*. Using Subtree *a*, 4 out of the 6 users found related articles for this task – of which 3 users found definitely relevant articles, while all users using Subtree *b* failed to find any related articles.

In observing users perform Tasks 3 and 6, the key was in finding the specific **gateway** category. This gateway category opened up relevant categories which were often clustered together around the gateway category. For Task 6, this gateway category was more difficult to find in Subtree *b*. This was because there were relations missing from the categories where users were looking. The key category for Task 6 in Subtree *b* was located in a related but obscure category called ‘*Herbs and Medicinal herbs*’. In contrast, users performing the task on Subtree *a* tended to find the key category *Wine* as a multiple parent of ‘*Grape varieties*’ which helped them perform this task well.

### 6.3.5 Outcome of validation experiment

In this section, we carried out user studies on the browsing of Wikipedia articles using a set of ontologies which varied on tangledness. This was to validate the performance of ontologies based on the ROMEO mapping identified in Chapter 5. The results from the validation

experiment showed that tangledness impacts on a user's performance in browsing articles using a given category structure. Overall, Subtree *a*, which had a degree of tangledness, enabled users to perform tasks more efficiently than Subtree *b*. For the specific task types, specifically Task 3 and 6, users performed with greater effectiveness and efficiency using Subtree *a* to browse the categories to find relevant articles than they did using Subtree *b*. Thus, in carrying out this empirical validation experiment, the ROMEO mapping between the question of "Does the category structure have an adequate intersection of categories?" and the measure of tangledness is found to be valid.

#### 6.4 Summary

In this chapter, we proposed a process for the empirical validation of mappings which may be produced by the application of the ROMEO methodology. This process involves validation experiments using task-based performance comparisons on ontologies that vary on key characteristics that are essential for that mapping. We presented validation experiments for a selected set of mappings specifically between questions and measures, as they are able to be reused in future applications of the ROMEO methodology.

In Section 6.2, experiments for the validation of a ROMEO question to its measure regarding having an appropriate level of granularity was presented. Specifically, we considered the mapping of a granularity question to the F-measure. We used text corpora as representations of the domain and compared it to the ontologies that were derived from the geographic sections of the ontology from Lonely Planet. We varied the Lonely Planet ontology of places in France to obtain more detailed ontologies and compared them with the set of Lonely Planet text corpora for the same domain but having different levels of detail. For the case of validating the granularity question to its measures, we found that the results were inconclusive as to whether the measures of Precision, Recall or F-measure adequately answer the question of granularity. It was suggested that the text corpora used may have been too course-grained. For future work, corpora of the same domain may be used in additional validation experiments of this mapping.

In Section 6.3, we validated the mapping of the question of adequate category intersection

to its measure of tangledness. The aim of the user study conducted for validation was to find relevant articles in Wikipedia using two ontologies for browsing – the original tangled ontology and an untangled version. We found that using the Dijkstra algorithm for shortest path produced an appropriate method to untangle a given ontology. This method was better than obtaining an ontology using a clustering algorithm on the set of categories. From the validation experiment, we found that for some tasks, tangledness did make a difference in finding related articles. This was especially the case for tasks that required articles with content that is more specific in nature. Thus, the mapping between the tangledness measure and the adequate category intersection question was validated.

## Chapter 7

# Conclusions and future work

*Everything that can be counted does not necessarily count;  
everything that counts cannot necessarily be counted.*

— Albert Einstein

In this thesis, we addressed two fundamental questions regarding ontology evaluation. We focused on defining an appropriate methodology for evaluating an ontology according to a set of requirements from an application, and how aspects of an ontology evaluation method may be validated. We proposed the ROMEIO methodology, which is capable of determining an appropriate evaluation method for a given set of ontology requirements. A process to validate aspects of the evaluation method was also presented, and we carried out validation experiments on new mappings that we discovered by applying ROMEIO to the two case studies conducted in this thesis.

This chapter summarises the research contributions of this thesis. Section 7.1 summarises the research contributions of ROMEIO and its application of ontology evaluation for the case studies of Lonely Planet and Wikipedia. Section 7.2 summarises the research contribution of the proposed validation process and the mappings that were validated using this process. In the respective sections, we also discuss limitations and suggest possible future research.

## 7.1 The ROMEO methodology for ontology evaluation

Determining a suitable method that incorporates a relevant set of measures for concrete, relevant and meaningful evaluation activities is critical for determining the appropriate ontology for ontology-driven applications. In examining existing methods in Chapter 2, we did not find an suitable ontology evaluation methodology that associated relevant measures based on the set of ontology requirements.

We examined two existing ontology evaluation methodologies: OntoClean and OntoMetric. However, we found that they had limitations with regards to the evaluation of ontology content. OntoClean evaluates only one part of an ontology, that is, the correctness of subsumption relationships between concepts but does not address other aspects of the ontology. OntoMetric helps with ontology selection but not with building new ontologies, as evaluating ontology content is not the sole focus. Additionally, OntoMetric adopts a linguistic scale for each measure, which can be subjective and does not allow for comparisons using actual quantitative measurements. Furthermore, OntoMetric does not help guide ontology engineers in determining which measures are appropriate for an ontology requirement. This analysis is left up to the ontology engineer to decide.

Aside from existing evaluation methodologies, many methods used to evaluate ontologies in ontology engineering methodologies are also inadequate. Approaches using pre-determined requirements or competency questions, such as On-To-Knowledge and Methontology, do not use specific measures; rather human judgments are relied upon for validation. Furthermore, approaches using criteria or measures outlined in current methodologies may not relate directly to the requirements of an ontology-driven application.

In this thesis, we introduced a new methodology for ontology evaluation called ROMEO to help ontology engineers evaluate ontologies according to a set of requirements from an application. This was presented in Chapter 3. ROMEO provides the ontology engineer with tools to determine an appropriate method for ontology evaluation using a systematic series of steps. Ontology requirements are identified from the application at hand. Each requirement is then mapped to a set of questions, which helps to determine whether the requirement has been met. In turn, each question is then mapped to an appropriate set of measures.



The research contribution of ROMEIO is that, in contrast to existing methods of ontology evaluation, ROMEIO directs the ontology engineer to obtain a more relevant method of ontology evaluation that is concrete and meaningful. The ontology evaluation method established using ROMEIO is specific to the ontology requirements of a given application, which relates these requirements to a focused set of applicable ontology evaluation measures. The mappings between requirements and questions, and questions to measures in ROMEIO allow ontology evaluation measures specific to the ontology requirements of the application to be obtained. Other benefits that come from using ROMEIO are that it allows for existing measures to be mapped to ontology requirements using questions and allows for the inclusion of new measures as they are proposed. ROMEIO is applicable for evaluating ontologies for both ontology engineering and ontology selection, whereas existing evaluation methodologies do not help with both.

A key component of ROMEIO is the specification of questions. ROMEIO allows the ontology engineer to map questions to address relevant aspects of an ontology evaluation criterion. Each question is written to examine one aspect of an ontology evaluation criterion. We found questions help to associate relevant aspects of a criterion to an ontology requirement, as there may be aspects which do not apply. This, in turn, helps the ontology engineer to map specific measures to the questions identified, thus obtaining a mapping between requirements to measures.

There remains the issue of knowing how to respond to the outputs from the identified measures, that is, the measurements taken. In some instances, measurements taken may highlight areas of refinement where an ontology does not answer some questions arising from the ROMEIO methodology. Although the ROMEIO methodology proposed in this thesis does not provide recipes for ontology engineers to improve their ontologies, a benefit of establishing measures from applying the ROMEIO methodology is that there is the ability to execute regression testing and benchmarking, despite any changes made. Thus, it does provide a useful framework for iterative refinement of ontologies. For example, tangledness is high, the ontology engineer can experiment with removal of nodes, and run regression tests.

In this thesis, we used the ROMEIO methodology to obtain appropriate ontology evalu-

ation methods for two case studies of applications that use ontologies. The case studies we considered were Lonely Planet and Wikipedia. In considering Lonely Planet as a case study in Chapter 4, we examined the ontology requirements related to the application of content management of travel information. In considering Wikipedia in Chapter 5, we considered issues regarding the browsing of articles using the category structure. We discussed how we gathered a set of ontology requirements for the category structure, which we took to be the ontology in this application.

From the two case studies, we demonstrated that ROMEO helps with the task of determining an appropriate method of ontology evaluation for ontology driven-applications. In the two case studies, we detailed the process in obtaining the relevant mappings for each case study and showed how applicable measures are identified using ROMEO. The measures selected are not arbitrarily chosen or based on general evaluation principles, but relate directly to the set of ontology requirements of a given application. Furthermore, these mappings are justified with reasons supporting the choices made using the ROMEO methodology.

In each case study, it was shown that *existing ontology evaluation measures* could be mapped to ontology requirements using questions using the ROMEO methodology. In the Lonely Planet case study, we mapped the ‘number of identical concepts with different names measure’ to the requirement of a ‘controlled vocabulary of names, places and terms’ using the question of ‘How many identical concepts are modelled using different names?’. In the Wikipedia case study, we mapped the number of circularity errors measure to the requirement of ‘avoiding cycles in the category structure’ using the question ‘How many circularity errors are found in the ontology?’

In the Lonely Planet case study, we showed that *new ontology evaluation measures* can be identified by performing an analysis using ROMEO. Using ROMEO, we mapped two questions to the ontology requirement of having ‘appropriate granularity’. They were: ‘Does the ontology have an appropriate level of granularity with regards to its concepts compared with the domain being modelled?’ and ‘Does the ontology have an appropriate level of granularity with regards to its instances compared with the domain being modelled?’ We found that none of the existing ontology evaluation measures were able to address these

two questions. As a consequence, we proposed a new ontology evaluation measure – the F-measure, which extends the work of Guarino [2004] who proposed precision and recall for ontology evaluation. The F-measure we introduce is adapted from information retrieval and we argued that it is suitable for determining the appropriate granularity of an ontology.

The ontologies we considered in the case studies demonstrate that ontologies have a key role to play in information driven applications. Because of the nature of the case studies considered, the measures and approaches taken were data-driven and included techniques from the Information Retrieval and Language Processing domains. A key contribution in this thesis was the presentation of new measures inspired from those domains – Precision, Recall and F-measure. Thus, the case studies and the approaches we took to obtain measures for ontology evaluation may serve as an important example and the measures may be reused in other information driven applications.

Although we primarily considered case studies of application ontologies in this thesis, ROMEO may be also used in the evaluation of all kinds of ontologies, such as upper, middle, and domain ontologies. Using ontology requirements from applications that use those types of ontologies, the ROMEO methodology may be applied to determine appropriate questions and measures for those requirements.

As more ontologies are being developed and utilised, ontology evaluation is of utmost importance for applications that require the use of ontologies. Without the right ontology evaluation methods, it is difficult to determine whether the right ontology is being used and ontology engineers may need to rely on instinct and experience to evaluate ontologies that are being developed. As such, ROMEO is beneficial for ontology engineers seeking to obtain an appropriate method for evaluating ontologies to meet the requirements of ontology-driven applications. ROMEO is also useful to ontology engineers seeking to reassess existing evaluation regimes in place.

### **Future work on ROMEO mappings and ontology evaluation measures**

As ROMEO is a new methodology, a limitation is that the set of known mappings of questions to measures is incomplete, as mappings of evaluation criteria and their associated measures

have not been completely explored. Therefore further work should explore the mappings between questions and their associated measures, as they are not dependent on an application and thus are more reusable than the mappings of requirements to questions. Furthermore, collating these mappings into a table of known mappings will aid future ontology evaluation analyses using ROMEO.

A specific difficulty is that some measures may not have any associated ontology evaluation criteria. There are also criteria that are difficult to quantify. Nevertheless, we propose that as any future analysis of ontology evaluation is carried out using ROMEO, the aforementioned table of known mappings be populated with any new mappings found. With this further work, a table of these mappings may be completed to aid with ontology evaluation using ROMEO.

More broadly, we propose future work on establishing measures for existing ontology evaluation criteria that are difficult to quantify or do not have any measures associated. For example, the criterion of minimal ontological commitment does not currently have any measures associated with it, thus in establishing a complete knowledge of measures that map to a given ontology evaluation criterion, future ontology evaluation activities may be more effective.

## 7.2 Empirical validation of ontology evaluation methods

Scientific disciplines and software engineering have used empirical analysis for the validation of measures to ensure that what is being measured corresponds to the attributes being examined. Whether using ROMEO or other methodologies, ontology evaluation methods obtained by the application of human judgments and analysis may be subjective and require validation. In this thesis, we outlined a process using empirical analysis for the validation of aspects of an ontology evaluation method in Chapter 6. Using this validation process, mappings between ontology requirements to ROMEO questions, and ROMEO questions to measures may be validated.

We used this validation process in task-based evaluations to examine the set of mappings of a question and its associated measures in task-based evaluations and compared the per-

formance of tasks which utilise different ontologies. We compared ontologies with each other for the purpose of determining whether measures correspond to its associated question. We specifically conducted two validation experiments to examine a subset of these mappings drawn from the case studies in Chapters 4 and 5.

In the first experiment, we examined the mapping between the question of ‘Does the ontology have an appropriate level of granularity with regards to its instances compared with the domain being modelled?’ with the ‘F-measure of the instances in the ontology with the domain’. We obtained a set of varied ontologies by extending the base ontology of places in France, with instances of places found in Paris, thus increasing the degree of granularity of each ontology. This validation process was used to examine whether the granularity of instances in each ontology was appropriate for each task in the travel domain describing the city of Paris in France. We used various text corpora from books about travel in France and Paris as representations of the France geographical domain. However, the results of that experiment were inconclusive with regards to the F-measure, which may be due to the fact that the text corpora used were too coarse-grained.

In the second experiment, we sought to validate the mapping of ‘Does the category structure have an adequate intersection of categories?’ with the tangledness measure. The set of tasks used in this validation experiment required users to browse different ontologies to find relevant articles. We compared two ontologies: a base ontology that had tangledness, and an untangled ontology, which uses a version of the base ontology with tangledness removed. The results from this validation experiment showed that the performance of user browsing differed depending on the tangledness in a given ontology. A set of tasks performed using the base ontology with tangledness, helped users find more related articles than users who browsed using an untangled ontology. This was especially the case for tasks that required users to find articles to satisfy a narrower information need. We validated the mapping between the tangledness measure and the question of ‘Does the category structure have an adequate intersection of categories?’

In both experiments, we experienced difficulties in obtaining a set of comparable ontologies. Existing ontologies available from publicly available sources may differ from each other

with regards to the level of specification of their structure and content. To overcome this, we obtained comparable ontologies for validation experiments by varying a base ontology systematically over a given measure and we demonstrated this for the two experiments that were carried out. In the first experiment, we used the geographic sections of the ontology obtained from Lonely Planet as the base ontology. We derived a set of varied ontologies from that base ontology by adding increasing levels of detail using more detailed geographic places in France and Paris and we were thus able to compare ontologies according to their levels of granularity. In the second experiment, we processed the base ontology using Dijkstra's algorithm for shortest path to obtain the untangled ontology, which ensured that each concept in the ontology had, at most, one parent (except for the root concept of the ontology). We were thus able to compare ontologies on their tangledness.

The validation of ontology evaluation methods using the process we presented in this thesis confirms that what is being measured corresponds to the attributes being examined. Thus, once validated, the aspect of the ontology evaluation method can be used in subsequent ontology evaluation methods. This process helps ontology engineers to have confidence that the right measures have been mapped in a given ontology evaluation method. This is especially important in ROMEQ as the mappings obtained are based on the analysis of an ontology engineer.

### **Future work on validating mappings**

We propose future work to address the empirical validation of the question of granularity to the F-measure. As discussed, the set of text corpora used in the experiment lacked enough variance in the level of geographic specificity, which meant that three measures – Number of Overlapping terms, Recall, and the F-measure, were appropriate measures. Future work includes determining the best measure for granularity given a text corpora with greater levels of specificity. In particular, the Wikipedia categories and articles about Paris may be used as text corpora to further explore granularity measures in the Paris domain.

Furthermore, further work may be conducted to confirm the validated mapping in Chapter 5, that is, the mapping of the tangledness measure to the question of 'Does the category

structure have an adequate intersection of categories?’ A limitation of our validation was its scope; we only considered tasks and domains relating to *Racing Sports* and *Food*. Further work may explore other domains in confirming the validation of this mapping.

Future work may also be carried out to validate the remaining mappings that were identified from the case studies, as they were not addressed in this thesis. This provides an additional set of validated mappings between questions and their associated measures, which may be useful in future ontology evaluation activities using ROMEEO.

More broadly, the validation process introduced in this thesis may be used to validate mappings in the table of known mappings proposed in Section 7.1. We propose that mappings between questions and measures should be validated using the validation process. With this further work, a validated table of these mappings may be completed and aid the evaluation of ontologies using ROMEEO. The establishment of a validated set of known mappings assures the ontology engineer of the validity of each mapping and thus instills confidence in using these mappings in mapping suitable evaluation methods in the future.

### 7.3 Summary

This thesis presented the ROMEEO methodology for aiding the process of content-based ontology evaluation using requirements from ontology-driven applications. ROMEEO uses the ontology requirements of an application as the basis for determining an appropriate method for evaluation. This is achieved by mapping applicable questions to the requirements and those questions to their measures. We examined the ontology requirements from two case studies – *Lonely Planet* and *Wikipedia*, and used ROMEEO for determining appropriate ontology evaluation methods for those applications. We found that a different set of evaluation measures was mapped to each application according to their requirements using ROMEEO.

A validation process was also presented. Requirements are dependent on the application and thus would differ from one application to another. We proposed the validation of mappings between questions and their measures, as they may be reusable in subsequent ROMEEO analyses. Experiments were carried out to validate two of such mappings, although one of the mappings we considered required additional experiments with other datasets and we propose

this as future work. We also proposed additional experiments to be carried out on the set of mappings that are yet to be validated. Populating a table of mappings with known mappings is also proposed and may be of benefit to future ontology evaluation activities using the ROMEIO methodology.

With the growing use of ontologies in applications in other domains, such as the life sciences, helping to determine the suitability of an ontology for the requirements of an application or domain is important. The ROMEIO methodology provides a means for obtaining a method of ontology evaluation for ontologies used in knowledge-intensive applications. Validating a mapping in ROMEIO using the validation process outlined in this thesis gives empirical evidence to ascertain whether the mapping corresponds to the expected behaviour. The ROMEIO methodology helps ontology engineers find a suitable ontology evaluation method given ontology requirements of an application, and the process of validation proposed in this thesis helps to provide validation mappings used to determine an appropriate ontology evaluation method.



# Glossary

<b>competency questions</b>	outline a list of questions a particular ontology should be able to answer, 27
<b>criteria-question</b>	A question prescribed in the ROMEIO methodology that is based on existing ontology evaluation criteria, 76
<b>criterion</b>	a quality used for making judgement or comparison, 5
<b>domain</b>	refers to a domain of discourse, which encompasses knowledge and rules in a given discipline, a field of study or subject area, 11
<b>frame of reference</b>	a point of reference used in the evaluation of ontologies, such as through the use of a set of competency questions or domain analysis, 27
<b>GQM</b>	the Goal-Question-Metric methodology proposed by Basili and Weiss [1984], 42

<b>indicator</b>	“a metric or combination of metrics that provides insight into the software process, a software project or the product itself. An indicator provides insight that enables the project manager or software engineers to adjust the process, the project, or the product to make things better” [Pressman, 2005], 61
<b>measure</b>	“a quantitative indication of extent, amount, dimensions, capacity, size of some attribute of a product or process” [Pressman, 2005], 5
<b>method</b>	“a formal procedure for producing some result” [Pfleeger and Atlee, 2005], 4
<b>methodology</b>	a collection of procedures, phases or processes, metrics, and measures with an underlying rationale or philosophy associated with it [Avison and Fitzgerald, 2002], 4
<b>Ontology</b>	the study of existence and being in Philosophy, 11
<b>ontology</b>	“an explicit specification of a conceptualisation” [Gruber, 1995], 11
<b>ontology engineering</b>	the design, building, testing and maintenance of ontologies, 3
<b>ontology evaluation</b>	examines ontologies using methodologies, tools, methods, and measures to examine a given set of ontology criteria, 4
<b>ontology evaluation criteria</b>	general qualities an ontology ought to possess, 5
<b>ontology evaluation measure</b>	quantifies some aspect of an ontology, 5

<b>ontology requirement</b>	the expression of desired quality or competency of an ontology, 4
<b>ontology selection</b>	the selection of an appropriate ontology from a pool of ontologies, 3
<b>requirement</b>	an expression of desired behaviour, 4
<b>ROMEIO</b>	Requirements Oriented Methodology for Evaluating Ontologies, 67
<b>validation</b>	“the process of ensuring that the measure is a proper numerical characterization of the claimed attribute” [Fenton, 1991], 7

## Appendix A

# ROMEIO templates and suggested mappings

### A.1 ROMEIO template

*Table A.1: Ontology role template*

**Application:** *Brief description of the application*

**Ontology:** *Qualities of suitable ontology - specificity, and desirables*

**Role:** *function the ontology provides or enables, and its relevance to the application*

Table A.2: Requirements template

<b>Requirement:</b>	
<b>Analyze:</b>	<i>Candidate ontologies being evaluated</i>
<b>For the purpose of:</b>	<i>Ontology evaluation, selection, or refinement</i>
<b>With respect to:</b>	<i>Quality or aspect being considered</i>
<b>From the viewpoint of:</b>	<i>Stakeholders and those affected parties from the adoption of the ontology</i>
<b>In the context of:</b>	<i>A specific ontology role</i>
<b>Motivation:</b>	<i>Description of the motivation for the requirements. Include reasons for the importance of the requirement and aspects being considered to the evaluation of candidate ontologies</i>

Table A.3: Question template

<b>Questions for requirement: <i>Requirement description</i></b>	
<b>Q1:</b>	<i>Question 1 that addresses the given requirement</i>
<b>Q2:</b>	<i>Question 2 that addresses the given requirement</i>
<b>Discussion:</b>	
A brief discussion on why the questions are relevant to the requirement.	

Table A.4: Measures template

<b>Measures for question: <i>Question</i></b>	
<b>M1:</b>	<i>Name of the measure</i>
	<b>Optimal value(s):</b> <i>Value indicating that the given question has been answered</i>
	<b>Range:</b> <i>Possible range of measurements</i>
<b>Discussion:</b>	
<i>A brief discussion on the reasons the set of measures listed apply to the given question. Also, a discussion on the optimal value and range expected with the set of measures listed.</i>	

## A.2 Suggested mappings between questions and measures

Table A.5: Clarity, consistency, and conciseness mappings

Criteria-question	Measure(s)
<i>Clarity</i>	
Does the ontology have concepts objectively defined?	–
Does the ontology have instances objectively defined?	–
Does the ontology have relationships between concepts objectively defined?	–
Does the ontology have relationships between instances objectively defined?	–
<i>Consistency</i>	
Does the ontology include two or more concepts that shares the same set of children?	<i>subclasses with common classes</i> ( $\mathcal{O}$ )
Does the ontology include two or more concepts that shares the same set of instances?	<i>classes with common instances</i> ( $\mathcal{O}$ )
How many circularity errors are found in the ontology?	<i>circularity errors at distance 0, 1, n</i>
<i>Conciseness</i>	
How many identical concepts are modelled using different names?	<i>semantically identical concepts</i> ( $\mathcal{O}_c$ )
How many identical instances are modelled using different names?	<i>semantically identical instances</i> ( $\mathcal{O}_i$ )
How many redundant subclass-of relationships are found in the ontology?	<i>redundant relationships</i> ( $\mathcal{O}_c, isa$ )
How many redundant instance-of relationships are found in the ontology?	<i>redundant relationships</i> ( $\mathcal{O}, instance\_of$ )
Does the ontology model concepts outside of the frame of reference?	<i>precision</i> ( $\mathcal{O}_c, \mathcal{F}_c$ )
Does the ontology model relationships outside of the frame of reference?	<i>precision</i> ( $\mathcal{O}_r, \mathcal{F}_r$ )
Does the ontology model instances outside of the frame of reference?	<i>precision</i> ( $\mathcal{O}_i, \mathcal{F}_i$ )

Table A.6: Expandability and correctness mappings

Criteria-question	Measure(s)
<i>Expandability</i>	
Do the set of concepts defined allow for future definitions of subclasses?	–
<i>Correctness</i>	
Does the ontology capture concepts of the domain correctly?	<i>ratio of meta-consistent concepts</i> ( $\mathcal{O}_c$ )
Does the ontology capture instances of the domain correctly?	<i>precision</i> ( $\mathcal{O}_c, \mathcal{F}_c$ ) <i>precision</i> ( $\mathcal{O}_i, \mathcal{F}_i$ )
Does the ontology capture relationships between concepts of the domain correctly?	<i>precision</i> ( $\mathcal{O}_{cr}, \mathcal{F}_{cr}$ )
Does the ontology capture relationships between instances of the domain correctly?	<i>precision</i> ( $\mathcal{O}_{ir}, \mathcal{F}_{ir}$ )

Table A.7: Completeness, coverage and minimal ontological commitment mappings

Criteria-question	Measure(s)
<i>Completeness</i>	
Does the ontology have concepts missing with regard to the relevant frames of reference?	<i>Incomplete concept classification</i> ( $\mathcal{O}, \mathcal{F}^*$ ) <i>average-recall</i> ( $\mathcal{O}_c, \mathcal{F}^*$ )
Does the ontology have subclasses concepts missing from a given parent concept with regard to the relevant frames of reference?	<i>Ex. subclass partition omission</i> ( $\mathcal{O}_c, \mathcal{F}^*$ )
Does the ontology have instances missing with regard to the relevant frames of reference?	<i>average-recall</i> ( $\mathcal{O}_i, \mathcal{F}^*$ )
Does the ontology have relationships between concepts missing with regard to the relevant frames of reference?	<i>average-recall</i> ( $\mathcal{O}_{cr}, \mathcal{F}^*$ )
Does the ontology have relationships between instances missing with regard to the relevant frames of reference?	<i>average-recall</i> ( $\mathcal{O}_{ir}, \mathcal{F}^*$ )
<i>Coverage</i>	
Do the concepts in the ontology adequately cover the concepts in the domain?	<i>Incomplete concept classification</i> ( $\mathcal{O}, \mathcal{F}$ ) <i>recall</i> ( $\mathcal{O}_c, \mathcal{F}_c$ )
Do the instances in the ontology adequately cover the instances in the domain?	<i>recall</i> ( $\mathcal{O}_i, \mathcal{F}_i$ )
Do the relationships between concepts in the ontology adequately cover the relationships between concepts in the domain?	<i>recall</i> ( $\mathcal{O}_{cr}, \mathcal{F}_{cr}$ )
Do the relationships between instances in the ontology adequately cover the relationships between instances in the domain?	<i>recall</i> ( $\mathcal{O}_{ir}, \mathcal{F}_{ir}$ )
<i>Minimal Ontological Commitment</i>	
Does the ontology define any concepts that are overstated for the domain?	–
Does the ontology define any instances that are overstated for the domain?	–
Does the ontology define any relationships between concepts that are overstated for the domain?	–
Does the ontology define any relationships between instances that are overstated for the domain?	–



Table A.8: *Intersectedness and appropriate granularity mappings*

Criteria-question	Measure(s)
<i>Intersectedness</i>	
Does the category structure have an adequate intersection of categories?	Tangledness: $t(\mathcal{O}_c)$
<i>Appropriate granularity</i>	
Does the ontology have an appropriate level of granularity with regard to its concepts compared with the domain being modelled?	$F\text{-measure}(\mathcal{O}_c, \mathcal{F}_c)$
Does the ontology have an appropriate level of granularity with regard to its instances compared with the domain being modelled?	$F\text{-measure}(\mathcal{O}_i, \mathcal{F}_i)$

## Appendix B

# ROMEIO analysis: Lonely Planet

### B.1 Role of the ontology

**Application:** Lonely Planet Content Management

**Ontology:** Ontologies in the travel domain

**Roles of the ontology:**

1. Shared controlled vocabulary
2. Adequately represent geographic places and associated content

*Table B.1: Lonely Planet application context and role of an ontology*

## B.2 Requirements

Table B.2: LP Ontology Requirement: Controlled vocabulary of names, places and terms

<b>Requirement OR1:</b> <i>Controlled vocabulary of names, places and terms</i>	
<b>Analyze:</b>	Lonely Planet Ontology
<b>For the purpose of:</b>	The controlled usage of preferred names and terms
<b>With respect to:</b>	Lonely Planet content management of travel information
<b>From the viewpoint of:</b>	Guidebook authors and editors of travel information
<b>In the context of:</b>	A controlled vocabulary for travel content
<b>Motivation:</b>	Motivating this requirement is maintaining a controlled vocabulary for authors and editors in providing a controlled usage of preferred names and terms. A specific scenario to avoiding the definition of redundant concepts which refer to an existing concept or instance, for example, the name of a place. However, for each concept in the vocabulary, there may be references to alternate names for other acceptable terms, for example, English terms for transliterated place names in Europe, which may have different permutations of terms referring to the same place.

Table B.3: LP Ontology Requirement: Flexible classification of geographic items

<b>Requirement OR2:</b> <i>Flexible classification of geographic items</i>	
<b>Analyze:</b>	Lonely Planet Ontology
<b>For the purpose of:</b>	Flexible classification of geographic places
<b>With respect to:</b>	Lonely Planet Content Management of travel information
<b>From the viewpoint of:</b>	Guidebook authors and editors of travel information
<b>In the context of:</b>	Appropriate representation of geographic places
<b>Motivation:</b>	The relationships defined in the ontology are important in addressing the need to accommodate imprecise relationships in the geographic domain, specifically, the relationships between places that may be vague, for example, relationships of <i>regions surrounding Paris</i> . A suitable ontology would accommodate these kinds of relationships from the geographic domain.

Table B.4: LP Ontology Requirement: Appropriate granularity

<b>Requirement OR3: Appropriate granularity</b>	
<b>Analyze:</b>	Candidate ontologies
<b>For the purpose of:</b>	Evaluating suitable ontologies
<b>With respect to:</b>	Lonely Planet Content Management of travel information
<b>From the viewpoint of:</b>	Guidebook authors and editors of travel information
<b>In the context of:</b>	Modelling appropriate level of content granularity
<b>Motivation:</b>	<p>A suitable ontology will have enough detail modelled for it to be usable in the context of aiding the management of content for tasks in Lonely Planet. This means the right concepts and instances must be in the ontology. The problem to avoid is in either having too much detail or not enough detail in the ontology. Too much detail results in an ontology where the concepts and instances included are unused and perhaps redundant. Too little detail in the ontology results in an ineffective ontology for the application.</p>

## B.3 Questions

Table B.5: Questions for OR1

<b>Questions for OR1:</b> <i>Controlled vocabulary of names, places and terms</i>	
<b>Q1:</b>	How many identical concepts are modelled using different names? (conciseness)
<b>Q2:</b>	How many identical instances are modelled using different names? (conciseness)
<b>Discussion:</b>	
<p>Q1 and Q2 are criteria-questions of the <i>conciseness</i> criterion as the problem that we seek to avoid is <i>semantic redundancy</i>. These questions consider concepts and instances that may have been defined as different concepts and instances, but are equivalent. The questions are applicable because they examine whether there is redundancy in the ontology with regard to the concepts and instances defined in the ontology.</p>	

Table B.6: Questions for OR2

<b>Questions for OR2:</b> <i>Flexible classification of geographic items</i>	
<b>Q3:</b>	Do the relationships between concepts in the ontology adequately cover the relationships between concepts in the domain? (coverage)
<b>Discussion:</b>	
<p>Q3 considers the relationships which exist between concepts – that is, the coverage of relationship types defined in the ontology. Here, we are concerned with having the various types of relationships in the travel and geographic domain which ought to be defined in the ontology, especially, relationships that may be imprecise.</p>	

Table B.7: Questions for OR3

<b>Questions for OR3:</b> Appropriate granularity	
<b>Q4:</b>	Does the ontology have an appropriate level of granularity with regard to its <i>concepts</i> compared with the domain being modelled?
<b>Q5:</b>	Does the ontology have an appropriate level of granularity with regard to its <i>instances</i> compared with the domain being modelled?
<b>Discussion:</b>	
<p>The right level of detail in the ontology is important in this requirement, specifically, the set of concepts and instances that are included. Concepts are a prerequisite for being able to define instances, thus its granularity is important to examine and we specify question Q4 for this reason. Q5 examines whether the granularity of an ontology is appropriate by examining the included set of instances in the ontology with regard to the set of instances in the travel domain.</p>	

## B.4 Measures

Table B.8: Measures for Q1

<b>Measures for Q1:</b> How many identical concepts are modelled using different names?	
<b>M1:</b>	<i>semantically identical concepts</i> ( $\mathcal{O}_c$ )
	Range of values: $0 -  \mathcal{O}_c $
	Optimal value(s): 0
<b>Discussion:</b>	
M1 considers any semantic equivalence between concepts in the ontology $\mathcal{O}_c$ . Ideally, there should not be any concepts that are equivalent. In the event that there are a set of concepts that are equivalent, they require a solution where a preferred concept is used in the ontology to satisfy its requirement, which may require removing one or more concepts from the ontology.	

Table B.9: Measures for Q2

<b>Measures for Q2:</b> How many identical instances are modelled using different names?	
<b>M2:</b>	<i>semantically identical instances</i> ( $\mathcal{O}_i$ ) Range of values: $0 -  \mathcal{O}_i $ Optimal value(s): 0
<b>Discussion:</b>	
M2 considers any semantic equivalence between instances in the ontology. Again, ideally, there should not be any instances that are equivalent. In the event that there are a set of instances that are equivalent, they require a solution where a preferred instance is used in the ontology to satisfy its requirement, which may require removing one or more instances from the ontology.	

Table B.10: Measures for Q3

<b>Measures for Q3:</b> Do the relationships between concepts in the ontology cover the relationships between concepts in the domain?	
<b>M3:</b>	<i>recall</i> ( $\mathcal{O}_{cr}, \mathcal{F}_{cr}$ ) Range of values: 0–1 Optimal value(s): 1
<b>Discussion:</b>	
The aim is to measure the coverage of the required relationships in the ontology $\mathcal{O}_{cr}$ with the frame of reference $\mathcal{F}_{cr}$ , which we take to be the domain in this case. For example, <i>city-of</i> , <i>near-by</i> , <i>capital city-of</i> relationships. We do this using the recall measure. The desired value of this measure is 1, which indicates total coverage over the domain.	



Table B.11: Measures for Q4

<b>Measures for Q5:</b> Does the ontology have an appropriate level of granularity with regard to its instances compared with the domain being modelled?	
<b>M5:</b>	$F\text{-measure}(\mathcal{O}_i, \mathcal{F}_i)$ Range of values: 0–1 Optimal value(s): 1
<b>Discussion:</b>	
<p>The aim here is to measure the granularity of instances included in an ontology <math>\mathcal{O}_i</math> compared with instances in the frame of reference <math>\mathcal{F}_i</math>, that is the domain being modelled. The measure used here is the F-measure – 0 indicates an inadequate granularity and 1, an appropriate level of granularity without the inclusion of other concepts outside of the domain being modelled in the ontology. The appropriate ontology will include instances that has an appropriate level of granularity for the domain being modelled. Ideally, the instances in the ontology should have the right level of granularity with a F-measure value of 1, compared with instances in the domain being modelled.</p>	

## Appendix C

# ROMEIO analysis: Wikipedia

### C.1 Role of the ontology

**Application:** Browsing Wikipedia.

**Ontology:** Wikipedia Category Structure.

**Role of the ontology:** Aid user navigation of the collection of articles through browsing

*Table C.1: Application context and purpose of ontology for Wikipedia*

## C.2 Requirements

Table C.2: *Ontology Requirement 1*

<b>Requirement OR1:</b> <i>Adequate level of category intersection.</i>	
<b>Analyze:</b>	Wikipedia Category Structure
<b>For the purpose of:</b>	Evaluating the Category Structure
<b>With respect to:</b>	Adequate level of category intersection
<b>From the viewpoint of:</b>	Wikipedia Users, Editors and Authors
<b>In the context of:</b>	Browsing Wikipedia Categories
<b>Motivation:</b>	
<p>Having an intersecting category structure facilitates user navigation of articles based on the given information need through the ability to browse alternate but somewhat related domains. In doing so, users may browse useful articles that they did not expect to encounter. For example, if we consider the <i>Food and Drink</i> category, it may intersect with categories such as <i>Culture</i>, <i>Health</i>, and <i>Digestive system</i>. All are related and may be in mind of users as they browse.</p>	

Table C.3: *Ontology Requirement 2*

<b>Requirement OR2:</b> <i>Categories should be appropriately grouped</i>	
<b>Analyze:</b>	Wikipedia Category Structure
<b>For the purpose of:</b>	Evaluating the Category Structure
<b>With respect to:</b>	Categories should be appropriately grouped
<b>From the viewpoint of:</b>	Wikipedia Users, Editors and Authors
<b>In the context of:</b>	Browsing Wikipedia Categories
<b>Motivation:</b>	
<p>The issue here is regarding the quality of subcategories. A given category structure should have an appropriate set of subcategories specified such that it facilitates the browsing of categories and articles according to the information need.</p>	

Table C.4: *Ontology Requirement 3*

<b>Requirement OR3:</b> <i>Avoiding cycles in the category structure</i>	
<b>Analyze:</b>	Wikipedia Category Structure
<b>For the purpose of:</b>	Evaluating the Category Structure
<b>With respect to:</b>	Avoiding cycles in the category structure
<b>From the viewpoint of:</b>	Wikipedia Users, Editors and Authors
<b>In the context of:</b>	Browsing Wikipedia Categories
<b>Motivation:</b>	<p>Wikipedia does not prevent cycles in its category structure but the Wikipedia editors strongly discourage them. In general, they are not helpful for users as it can be confusing. In addition, cycles may impede some automated processes in their use of the category structure.</p>

Table C.5: *Ontology Requirement 4*

<b>Requirement OR4:</b> <i>Ensure the set of categories is complete</i>	
<b>Analyze:</b>	Wikipedia Category Structure
<b>For the purpose of:</b>	Evaluating the Category Structure
<b>With respect to:</b>	Ensuring the set of categories is complete
<b>From the viewpoint of:</b>	Wikipedia Users, Editors and Authors
<b>In the context of:</b>	Browsing Wikipedia Categories
<b>Motivation:</b>	<p>A motivation for this requirement is that the association of a large number of categories to a given article may indicate a poorly defined set of categories. The issue here is resolving a lack of completeness in the category structure. Thus, a complete category structure ensures that the set of categories defined allows an article to associate an appropriate set of categories to it.</p>

Table C.6: *Ontology Requirement 5*

<b>Requirement OR5:</b> <i>Ensure categories associated in articles are correct</i>	
<b>Analyze:</b>	Wikipedia Category Structure
<b>For the purpose of:</b>	Evaluating the Category Structure
<b>With respect to:</b>	Ensure categories associated in a given article are correct
<b>From the viewpoint of:</b>	Wikipedia Users, Editors and Authors
<b>In the context of:</b>	Browsing Wikipedia Categories
<b>Motivation:</b>	A motivation for this requirement is that the association of a large number of categories to a given article may indicate associating a set of categories incorrectly, for example, due to spam, bias or human error. The issue here is resolving the correctness of categories associated for each article, especially those with too many categories associated with it.

### C.3 Questions

Table C.7: *Questions for Ontology Requirement 1*

<b>Questions for OR1:</b> <i>Adequate intersection of categories.</i>	
<b>Q1:</b>	Does the category structure have an adequate intersection of categories?
<b>Discussion:</b>	An adequate degree of intersectedness is sought in Q1. Q1 considers the degree of intersectedness in the category structure. Having not enough category intersection may result in an increased amount of browsing, which impedes the task of finding information. Conversely, if a given category is completely connected to all other concepts in the category structure, then the resulting category structure is as good as an index of all available categories, which is not useful.

Table C.8: Questions for Ontology Requirement 2

<b>Questions for OR2:</b> <i>Categories should be appropriately grouped</i>
<b>Q2:</b> Does the ontology capture concepts of the domain correctly? (correctness) <b>Discussion:</b>  In the Wikipedia documentation, a specific issue was highlighted with regard to the categories: <i>incorrect categorization</i> . Incorrect categorization occurs when there is a discrepancy between a category and its associated categories (such as subcategories of a category or parent categories of a given category) with respect to the world being modelled in. We have used a correctness criteria-question for Q2, which examines whether the relationships between categories have been correctly captured.

Table C.9: Questions for Ontology Requirement 3

<b>Questions for OR3:</b> <i>Avoiding cycles in the category structure</i>
<b>Q3:</b> How many circularity errors are found in the ontology? (consistency) <b>Discussion:</b>  This question examines whether the ontology has cycles. Cycles impede Wikipedia users in browsing and is undesirable in this application, as noted in the online Wikipedia guidelines.

Table C.10: Questions for Ontology Requirement 4

<b>Questions for OR4:</b> <i>Ensure a complete set of categories</i>
<b>Q4:</b> Does the ontology have concepts missing with regard to the relevant frames of reference? (completeness) <b>Discussion:</b>  Q4 examines incompleteness of the ontology, that is, concepts that are missing with regard to the relevant frames of reference using a completeness criteria-question. These frames of reference may be a collection of a set of all possible concepts that are defined within a given domain.

*Table C.11: Questions for Ontology Requirement 5*

<b>Questions for OR5:</b> <i>Ensure categories associated in articles are correct</i>
<b>Q5:</b> Is the set of categories correctly associated with a given article? (correctness)
<b>Discussion:</b>
<p>Q5 examines the correctness of the set of categories associated for a given article based on a correctness criteria-question. The issue being investigated is ensuring that each article is associated with appropriate categories. This helps users view relevant articles as they are browsing each category.</p>

## C.4 Measures

Table C.12: Measures for Question 1

<b>Measures for Q1:</b> Does the category structure have an adequate intersection of categories?	
<b>M1:</b>	Tangledness: $t(\mathcal{O}_c)$ Optimal value(s): Between 0.2 and 0.8 Range: 0–1
<b>Discussion:</b>	
<p>Tangledness measures the ratio of multiple parent categories. This measure may help us understand how intersected the category structure is. Measurements for this measure are values between 0 and 1, indicating the level of category intersection (0 indicating no intersection and 1 indicating a completely intersected ontology). Using this measure, we will evaluate what the threshold for an appropriate value for indicating the adequate intersection of categories. Optimal value for this measure is yet to be determined, but having considered some categories and its tangledness value, between 0.2 and 0.7 is the proposed threshold.</p>	



Table C.13: Measures for Question 2

<b>Measures for Q2:</b> Does the ontology capture the relationships between concepts of the domain correctly?	
<b>M2:</b>	<i>ratio of metaconsistent concepts</i> ( $\mathcal{O}_c$ ) Optimal value(s): 1 Range: 0–1
<b>Discussion:</b>	
This measure gives the ratio of concepts that are <i>correct</i> to the total number of concepts in the ontology $\mathcal{O}_c$ and may involve an OntoClean analysis of each relationship. In particular, the OntoClean analysis may help examine the meta-consistency ratio using meta-properties such as essence and rigidity to indicate incorrectly modelled categories and their subcategories.	

Table C.14: Measures for Question 3

<b>Measures for Q3:</b> Are there any cycles currently?	
<b>M3:</b>	Circularity error at distance 0 Optimal value(s): 0 Range: 0– $\infty$
<b>M4:</b>	Circularity error at distance 1 Optimal value(s): 0 Range: 0– $\infty$
<b>M5:</b>	Circularity error at distance N Optimal value(s): 0 Range: 0– $\infty$
<b>Discussion:</b>	
Simply detecting a cycle is sufficient to answer the question. Upon detection, we examine and resolve those cycles. M3, M4 and M5 are proposed as measures for errors at various distances from a given node. Range of values for M3, M4, and M5 are non-negative integers. We can employ the use of cycle detection algorithms to assess circularity errors such as those proposed by Floyd [1967].	

Table C.15: Measures for Question 4

<b>Measures for Q4:</b> Does the ontology have concepts missing with regard to the relevant frames of reference?	
<b>M6:</b>	<i>Incomplete concept classification</i> ( $\mathcal{O}, \mathcal{F}^*$ )
<b>M7:</b>	<i>recall</i> ( $\mathcal{O}_i, \mathcal{F}^*$ )
	Optimal value(s): 1
	Range: 0–1
<b>Discussion:</b>	
<p>There are 2 measures that map to this question. The measure of <i>Incomplete concept classification</i> (M6) involves determining whether there are concepts left out of the ontology from the frames of references, although this requires manually inspection between <math>\mathcal{O}</math> and <math>\mathcal{F}^*</math>. The recall measure (M7) can be used to answer the question on completeness by determining whether the set of concepts in the ontology <math>\mathcal{O}</math> are complete with respect to the relevant frames of reference <math>\mathcal{F}^*</math>. These frames of reference may be gathered from analysis of a domain using representative text corpora. The measurements of recall may range from 0 to 1. A recall value of 1 indicates a complete ontology and a value of 0 indicates that the ontology has no matching categories with the frames of reference.</p>	

Table C.16: Measures for Question 5

<b>Measures for Q5:</b> Is the set of categories correctly associated with a given article?	
<b>M8:</b>	Number of non-related associations between a given category and article Optimal value(s): 0 Range: 0– $\infty$
<b>M9:</b>	Number of NPOV violations of a relationship between given category and article Optimal value(s): 0 Range: 0– $\infty$
<b>Discussion:</b>	
<p>The two measures mapped to this question is very specific to the application requirements of Wikipedia for its category structure. M8 examines non-related relationships between a given category and article. Articles belonging to a category ought to have sufficient references in it to show it belongs to the category it is associated with. Thus, M8 may require verifying the relationship between category and article using reliable external sources. M9 examines the number of NPOV policy violations with regard to the relationship between a given article and category from the set of categories associated with an article. These violations include non-neutral point-of-views of a relationship between an article and a category which may display bias, or a personal opinion that has no factual basis.</p>	

## Appendix D

# Requirements gathering: Wikipedia

We present the following excerpts of Wikipedia guidelines, policies and requirements that we used for applying ROMEO in Chapter 5. They were accessed using the following links:

- [http://meta.wikimedia.org/wiki/Categorization\\_requirements](http://meta.wikimedia.org/wiki/Categorization_requirements) (Accessed May 2008)
- <http://en.wikipedia.org/wiki/Wikipedia:Categorization> (Accessed May 2008)

### D.1 Excerpts from Meta:Categorization requirements

#### D.1.1 Goals (or, “Why implement categories?”)

Currently the wiki allows readers to find information in a couple of ways (more?):

- follow links in the articles, starting at the main page
- search for a term, resulting in either an exact match, or a list of partial matches

The wiki has its own search utility, but web users can also search on Google using the “site:wikipedia.org” search criterion.

These methods of finding information will fail in the following cases:

- searching with a term or terms not found in the document
- searching with too general a term or terms, resulting in more hits than can be checked in a reasonable time

- user does not know the information exists

The last case is strange, but is valid if you consider that many users might be interested in browsing for knowledge related to some subject with which they are familiar, but which no link-following strategy has allowed them to discover. In other words, the chain of links from the article currently being viewed to the potentially useful article is too long, too convoluted, or simply lost in the complexity of the network.

Other means than categories, for providing short-cuts to articles, already exist. One way is contributor-created lists. These have to be kept up to date by manual editing. This is usually quite reliable, but there is a time delay involved.

A possible solution to the problem of keeping useful lists of articles up to date is to try to find a way to partly automate the process. Some automatic processes already exist. Overly simple list generation methods tend to be comprehensive, but suffer the same problem as lists of search results. Moreover, on wikipedia, the alphabetical listing seems to have been turned off. The page loads, but this message occurs at the top: “Sorry! This feature has been temporarily disabled because it slows the database down to the point that no one can use the wiki.” This demonstrates that automatic organization methods are not only hard to design in a useful way, but also in an efficient way.

Another interesting way to see a comprehensive list of articles connected with a given article is to generate localized site maps, or link graphs. These grow exponentially, however, so you can only get to two or maybe three links before they become unreadable. An article itself is already a one-degree of separation link graph. As mentioned, there is no way to be certain that all articles which a user would expect to be related would show up within three links. Network graphs are also a terribly complicated thing to diagram. They are effectively impossible on a public access server.

An answer to the weaknesses of inter-article links is to define some kind of logical entity which is not an article, but a hub of connection for articles: a category. Categories are a means of partitioning the articles into various sets based on some kind of relationship, usually taken to be subject matter. Defining subjects, and hence categories, is a challenge. The task can be made both easier and more difficult by putting various types of constraints on the

nature of and means of defining categories.

Since many categorization schemes already exist in library and information science, it is common to suggest use of one of these. They have two weaknesses:

1. They are designed for books stored in stacks in a library, not for articles
2. They change all the time

Also, the size of most such category schemes is prohibitive, they would require some level of expertise both to implement and to use, and many categories in such schemes would be empty of articles in the wiki. Finally, adding an article to a predetermined set of categories requires the tedious task of searching for the correct category that seems to fit.

The alternative is to allow wiki contributors to define their own categories. This is fairly simple; they just need a means to add a name or other identifier to an article that represents the category (or categories) in which the article should be placed. Specifying parent–child category relationships is more complicated, but could possibly be done using a special interface. Also, there is a question of what happens if two categories have the same unqualified name (where the qualified name includes the list of the descendents starting at some root category).

If a category scheme is implemented, the software would have to be extended to provide facilities to take advantage of them: to allow users to look at lists of articles which fit in categories, and lists of sub–categories of categories. If these lists are too long or too complicated to read, or if their generation places too great a strain on resources, they will not be usable.

### **D.1.2 Theoretical basis**

It would be desirable, if not absolutely necessary, to consider some kind of scientific or mathematical model for a categorization scheme. As categories of articles correspond closely to sets of elements, it seems natural to consider and adhere to aspects of set theory when developing a scheme. Because relations between sets can be described using graphs, where vertices are the sets and directed edges are the subset relationship, graph theory should also be useful.

In addition, it is also desirable that any categorization scheme includes an algorithm for converting a group of uncategorized articles (which could be called a set of uncategorized articles) into a categorized group of articles (articles which belong to the categorized set, which is to say that they are members of sets which are in a set called “category”). Such an algorithm can thus be judged on its suitability in light of various qualities: programmability (how well, and to what extent, it can be automated), synchronization, parallelism, recursiveness, and reliability. It is also important whether the algorithm can be performed reliably and naturally by human beings, especially non-specialists, and, moreover, in numbers but working mostly independently.

Algorithms also have the benefit that they can define, implicitly or explicitly, constraints on the way categories are created, how they are named, and the like.

## D.2 Excerpts from *Wikipedia: Categorization*

This page is considered an editing guideline on Wikipedia. It is a generally accepted standard that all editors should follow. However, it is not set in stone and should be treated with common sense and the occasional exception. When editing this page, ensure that your revision reflects consensus. When in doubt, discuss first on the talk page.

This page in a nutshell:

- Categories help users navigate through Wikipedia via multiple taxonomies
- Categories are for defining characteristics, and should be specific, neutral, inclusive and follow certain conventions.

For detailed technical information on how to use categories, see *Help:Category*. For everything you ever wanted to know about categories, see *Wikipedia: Categorization FAQ*.

### D.2.1 When to use categories

Categories (along with other features like cross-references, lists, and infoboxes) help users find information, even if they don't know that it exists or what it's called.

Every page in the article namespace should belong to at least one category. The categories to be included, which serve as classifications, should be the significant (useful) topics to which the subject of the article most closely belongs to as a member, and where users are most likely to look if they can't remember the name of the thing they are trying to look up. For example:

Article: *Michael Jackson*

Useful category: *Category:Pop singers*

Not as useful: *Category:Musicians whose first name starts with M*

Questions to ask to determine whether it is appropriate to add an article to a category:

- If the category does not already exist, is it possible to write a few paragraphs or more on the subject of the category, explaining it?
- If you go to the article from the category, will it be obvious why the article was put in the category? Is the category subject prominently discussed in the article?



- Does the category fit into the overall category system? Categories that don't fit are often deleted. To familiarize yourself with the types of categories that routinely get deleted read *Wikipedia:Overcategorization*.

If the answer to any of these questions is no, then the category is probably inappropriate. Note that it is always appropriate to add articles to categories that fit into well established taxonomies. For example, every article about a musical album is categorized in some *[[Category:Artistname albums]]* category, which is in turn categorized in *Category:Albums by artist*.

## D.2.2 Guidelines

### Some general guidelines

1. Categories are mainly used to browse through similar articles. Make decisions about the structure of categories and subcategories that make it easy for users to browse through similar articles.
2. An article will often be in several categories. Restraint should be used as categories become less effective the more there are on any given article. Use the *Too many categories* tag if you feel the article is in too many categories.
3. Usually, articles should not be in both a category and its subcategory. For example, *Golden Gate Bridge* is in *Category:Suspension bridges*, so it should not also be in *Category:Bridges*. However, there are occasions when this guideline can and should be ignored. For example, *Robert Duvall* is in *Category:Film actors* as well as its subcategory *Category:Best Actor Academy Award winners*. See #5 for another exception. For more about this see *Wikipedia:Categorization and subcategories*.
4. Check to see where siblings of the article reside. If there are  $0 < x < 1$  few if any articles in a category, the article probably belongs in one of the subcategories.
5. Articles should be placed in categories with the same name. However, the article and the category do not have to be categorized the same way. The article can also be placed in categories populated with similar articles. The category can be put into categories

populated with similar subcategories. For an example of this see *George W. Bush* and *Category:George W. Bush*.

6. There are often occasions when articles might ideally be moved from a category to two or more of its subcategories, but not all of the subcategories exist. In such cases consider creating the additional subcategories, but if you decide not to do so, leave the articles in the parent category for the time being.
7. Categories appear without annotations, so be careful of NPOV when creating or filling categories. Categories that are not self-evident, or are shown through reliable sources to be controversial, should not be included on the article; a list might be a better option.
8. An article should normally possess all the referenced information necessary to demonstrate that it belongs in each of its categories. Avoid including categories in an article if the article itself doesn't adequately show it belongs there. For example, avoid placing a category for a profession or award unless the article provides some verification that the placement is accurate. Use the `{{Category unsourced}}` tag if the article is in a category but no sources demonstrate the category is appropriate.
9. If you don't know where to put an article, don't worry about it. Instead of adding a category, add the `{{uncategorized}}` template to bring attention to the article. Editors who love to categorize articles will find a good home for your article.
10. Bend the rules above when it makes sense, especially when it is the best solution that can be found under the circumstances.

### **Categories vs. lists vs. info boxes**

Categories are not the only way to organize articles. For alternative methods of grouping articles, and the circumstances in which they should be used, see *Wikipedia:Categories, lists, and series boxes*.

### Categories applied to articles on people

A separate wikipedia page *Wikipedia:Categorization of people* was created to help you in designing, applying and checking categories that are used for articles on people.

### Categories do not form a tree

Each Wikipedia article can appear in more than one category, and each category can appear in more than one parent category. Multiple categorization schemes co-exist simultaneously. In other words, categories do not form a strict hierarchy or tree structure, but a more general directed acyclic graph (or close to it; see below).

Nevertheless, parts of the category graph will be tree-like, and it may be convenient to think of parts of the category graph as being like multiple overlapping trees. When applying the guidelines above, consider each tree to be independent of the overlapping trees. A person browsing through a hierarchy should find every article that belongs in that hierarchy. This can lead to a good deal of debate as to what the hierarchies actually are. To clarify the structure of the hierarchy and help people browse through it, you can add a classification to each category. For more about this, see *Wikipedia:Classification*.

### Cycles should usually be avoided

Although the MediaWiki software does not prevent cycles (loops), these usually should be avoided. Cycles can be confusing to some readers, they can challenge some automated searching processes, and they can grow quite large. For example, in January 2006 a 22-member category cycle was discovered and eliminated.

However, acceptable loops also exist. Self-referencing systems such as the meta-fields naturally create cycles that provide many examples. This type of cycle involves making a category one of its own subcategories. A real-world example of a self-referencing system is ‘education about education’, such as:

*Classification: Education: Social sciences: Academic disciplines: Academia: Education: ...*

Another type of cycle involves making two categories subcategories of each other. Loops such as these can be avoided by linking the categories manually to each other by adding “See

also:Category:Foo” to each category page. For an example of this see *Category:World Trade Center* and *Category:September 11, 2001 attacks*.

### D.2.3 Grouping categories

A set of related categories often forms a hierarchy or a nexus. This can take several different forms, all of which are welcome and encouraged:

- A taxonomic grouping. For example, *Category:South Asian countries* is part of a geographical hierarchy. *Category:Academic disciplines* catalogs divisions between fields of study.
- A functional grouping. Examples: *Category:Ancient Rome*, *Category:World War II*, and *Category:Commercial item transport and distribution*. These bring together articles and subcategories from different fields or taxonomies (history, war, culture, people, companies, industries, technologies) that have an interesting common thread.
- Hybrid forms. For example, *Category:Art* is both part of the taxonomy of *Category:Academia* and a cross-reference point for lots of things that have little in common except that they have something to do with art.
- Offshoot forms. For example, *Category:Film stubs* contains subcategories of *Category:Comedy film stubs* for comedy films, and *Category:Drama film stubs* for drama films.

## Appendix E

# Wikipedia Browsing Experiment

### E.1 User handouts

**Section I. Pre-study questionnaire**

Please circle the options that best answers the following questions

1. Is English your first language?

Yes                      No

2. a) In an average week, how many separate occasions do you spend finding information on the web?

1-5 times a week              Once a day              Many times a day              Every hour

b) What is the average duration per session?

~1-5 minutes              ~5-15 minutes              ~30 minutes              ~1 hour              > 1 hour

*Figure E.1: Pre-study questionnaire*

**Section II. Interface description**

The screenshot shows the Wikipedia interface for the 'Travel' category. At the top, the current category is 'Travel'. Below it, there are links for broader categories (Tourism, Transportation) and narrower categories (Airlines, Airline tickets, Hitchhiking, Hospitality service, Open Travel Alliance). A list of articles in the category is shown, including 'A Cook's Tour', 'A Journey Beyond the Three Seas', 'American Tourister', 'Backpacking (Canada)', 'Backpacking (Christian)', 'Backpacking (travel)', 'Backpacking (United States)', 'Babur', 'Bareboat Charter', 'Basilio Vitorino', 'Bicycle touring', 'Bookstore Tourism', 'Business class', 'CARNet', 'Carte de Passage', 'Code sharing', 'Computer reservations system', 'Continental pass', and 'Doppeldecker'. The selected article, 'Travel agency', is displayed in a large text area. The history trail at the bottom shows 'History: Tourism >>> Travel agency'.

A: Current category you have navigated to

B: Actions you can make  
*'Home'* – takes you back to the starting category (i.e. the entry point of your navigation)  
*'Mark relevant'* – allows you to mark the current viewed article as relevant to the task at hand  
*'View marked'* – lists the articles you have marked

C & D: Links to broader and narrower categories to the current category

E: **List of articles** linked to the current category

F: The text of the selected article is displayed here when clicked in E

G: Links to previously clicked on categories/articles in your session

Figure E.2: Interface description

**Training Task description:**

*Mark relevant articles of tourist attractions.*

PRE-task questions:

1. What is your level of knowledge with this task?

No prior knowledge      1            2            3            4            Expert

2. From your own knowledge, under which subject areas would you expect to find relevant articles in?

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Start browsing**

*Figure E.3: Training task*

POST-task questions:

1. In using this category organisation to complete the task, I felt...

Uncertain/Confused	1	2	3	4	Confident/Clear
--------------------	---	---	---	---	-----------------

2. Overall, did the system help in getting comprehensive answers?

Found nothing	1	2	3	4	Comprehensive
---------------	---	---	---	---	---------------

3. How satisfied were you in getting answers to the task using the category organisation?

Frustrated	1	2	3	4	Very satisfied
------------	---	---	---	---	----------------

4. How would you rate the quality of the category organization?

No organisation	1	2	3	4	Well organized
-----------------	---	---	---	---	----------------

**Comments:**

---

---

---

Figure E.4: Post-task questionnaire



**Task description:**

*T1: Mark relevant articles of any international racing competitions.*

*PRE-task questions:*

1. What is your level of knowledge with this task?

No prior knowledge      1            2            3            4            Expert

2. From your own knowledge, under which subject areas would you expect to find relevant articles in?

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Start browsing**

*Figure E.5: Task 1 description*

**Task description:**

*T2: Mark articles of racing sports that do not involve vehicles with wheels.*

*PRE-task questions:*

1. What is your level of knowledge with this task?

No prior knowledge      1            2            3            4            Expert

2. From your own knowledge, under which subject areas would you expect to find relevant articles in?

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Start browsing**

Figure E.6: Task 2 description

**Task description:**

*T3: Mark relevant articles relevant articles describing makers of F1 racing cars.*

*PRE-task questions:*

1. What is your level of knowledge with this task?

No prior knowledge      1            2            3            4            Expert

2. From your own knowledge, under which subject areas would you expect to find relevant articles in?

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Start browsing**

*Figure E.7: Task 3 description*

**Task description:**

*T4: Mark relevant articles describing non-alcoholic beverages.*

PRE-task questions:

1. What is your level of knowledge with this task?

No prior knowledge	1	2	3	4	Expert
--------------------	---	---	---	---	--------

2. From your own knowledge, under which subject areas would you expect to find relevant articles in?

---

---

---

**Start browsing**

Figure E.8: Task 4 description

**Task description:**

*T5: Mark articles that describe different cuisines of the world.*

*PRE-task questions:*

1. What is your level of knowledge with this task?

No prior knowledge      1            2            3            4            Expert

2. From your own knowledge, under which subject areas would you expect to find relevant articles in?

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Start browsing**

*Figure E.9: Task 5 description*

**Task description:**

*T6: Mark articles describing wine regions found in Australia.*

*PRE-task questions:*

1. What is your level of knowledge with this task?

No prior knowledge      1            2            3            4            Expert

2. From your own knowledge, under which subject areas would you expect to find relevant articles in?

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Start browsing**

*Figure E.10: Task 6 description*

POST-study questionnaire:

1. Regarding the length of each task, it was...

Too Short/Long	1	2	3	4	Sufficient Time
----------------	---	---	---	---	-----------------

2. Regarding the interface for browsing for task information, it was...

Difficult to use	1	2	3	4	Easy to use
------------------	---	---	---	---	-------------

**Study Completed**

Thanks for your participation!

Hope you found it enjoyable – please accept my sincere appreciation for taking out time to help out.

Figure E.11: Post-study questionnaire

# Bibliography

- C. Alexakos, B. Vassiliadis, K. Votis, and S. Likothanassis. A multilayer ontology scheme for integrated searching in distributed hypermedia. In S. Sirmakessis, editor, *Adaptive and Personalized Semantic Web*, volume 14 of *Studies in Computational Intelligence*, pages 75–83. Springer, 2006.
- Alexandria Digital Library Gazeteer. URL <http://middleware.alexandria.ucsb.edu/client/gaz/adl>. Accessed March 2008.
- A. Andreou. Ontologies and query expansion. Master’s thesis, University of Edinburgh, 2005.
- D. Avison and G. Fitzgerald. *Information Systems Development: 3rd Ed.* McGraw Hill Higher Education, 2002.
- R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press / Addison-Wesley, 1999.
- A. L. Baker, J. M. Bieman, N. Fenton, D. A. Gustafson, A. Melton, and R. Whitty. A philosophy for software measurement. *Journal of Systems and Software*, 12(3):277–281, 1990.
- V. Basili, G. Caldiera, and H. Rombach. Goal question metric approach. In *Encyclopedia of Software Engineering*, pages 528–532. John Wiley & Sons Inc., 1994.
- V. R. Basili and D. M. Weiss. A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering*, 10(6):728–738, November 1984.



- R. Batres, M. West, D. Leal, D. Price, K. Masaki, Y. Shimada, T. Fuchino, and Y. Naka. An upper ontology based on ISO 15926. *Computers & Chemical Engineering*, 31(5–6): 519–534, May 2007.
- S. Bechhofer, R. Stevens, and L. P.W. Ontology driven dynamic linking of biology resources. In *Pacific Symposium on Biocomputing*, pages 79–90, 2005.
- T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5): 35–43, May 2001.
- T. Bittner and B. Smith. Granular partitions and vagueness. In *Proceedings of the international conference on Formal Ontology in Information Systems (FOIS)*, pages 309–320, NY, USA, 2001. ACM Press.
- T. Bittner and B. Smith. *A Theory of Granular Partitions*, chapter 7, pages 117–151. Foundations of Geographic Information Science. Taylor and Francis, London, UK, 2003.
- J. Brank, M. Grobelnik, and D. Mladenic. A survey of ontology evaluation techniques. In *Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005)*, 2005.
- T. D. Breaux and J. W. Reed. Using ontology in hierarchical information clustering. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, pages 111–112, Washington, DC, USA, January 2005. IEEE Computer Society.
- K. Breitman, M. A. Casanova, and W. Truszkowski. *Semantic Web: Concepts, Technologies and Applications*, chapter 8: Methods for Ontology Development, pages 155–173. Springer-Verlag, London, 2007.
- C. Brewster, H. Alani, S. Dasmahapatra, and Y. Wilks. Data driven ontology evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal, 2004. European Language Resources Association.
- A. Burrow. Negotiating access within wiki: a system to construct and maintain a taxonomy

- of access rules. In *Proceedings of the 15th ACM conference on Hypertext and Hypermedia*, number 15, pages 77–86, New York, USA, 2004. ACM Press.
- J. Cavano and J. McCall. A framework for the measurement of software quality. In *Proceedings of the software quality assurance workshop on Functional and performance issues*, pages 133–139, 1978.
- O. Corcho, M. Fernández-López, and A. Gómez-Pérez. Evaluation experiment for the editor of the WebODE ontology workbench. In *Proceedings of the OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002*, pages 129–134, 2002.
- O. Corcho, M. Fernández-López, A. Gómez-Pérez, and A. López-Cima. Building legal ontologies with METHONTOLOGY and WebODE. In *Law and the Semantic Web*, pages 142–157, 2003.
- E. J. Davidson. *Evaluation Methodology Basics*. Sage Publications, London, UK, 2005.
- J. Davies, D. Fensel, and F. van Harmelen, editors. *Towards the Semantic Web: Ontology-Driven Knowledge Management*. Wiley, 2003.
- S. Domingo and P. Eklund. Evaluation of concept lattices in a web-based mail browser. In *13th International Conference on Conceptual Structures (ICCS)*, volume 3596 of *Lecture Notes in Computer Science*, pages 281–294, Kassel, Germany, July 2005. Springer-Verlag.
- J. S. Dong, C. H. Lee, H. B. Lee, Y. F. Li, and H. Wang. A combined approach to checking web ontologies. In *Proceedings of International Conference on World Wide Web*, pages 714–722. ACM Press, 2004.
- N. Fenton. *Software Metrics: A Rigorous Approach*. Chapman and Hall, 1991.
- R. Floyd. Non-deterministic algorithms. *Journal of the ACM*, 14(4):636–644, 1967.
- F. Fonseca, M. Egenhofer, P. Agouris, and G. Camara. Using ontologies for integrated geographic information systems. *Transactions in Geographic Information Systems*, 6(3): 231–257, 2002a.

- F. Fonseca, M. Egenhofer, C. Davis, and G. Câmara. Semantic granularity in ontology-driven geographic information systems. *Annals of Mathematics and Artificial Intelligence*, 36(1-2):121–151, 2002b.
- G. Fu, C. Jones, , and A. Abdelmoty. Ontology-based spatial query expansion in information retrieval. In *Proceedings of 4th International Conference on Ontologies, Databases and Applications of Semantics (ODBASE)*, Lecture Notes in Computer Science, pages 1466–1482, Ayia Napa, Cyprus, 2005. Springer-Verlag.
- F. Gandon. Engineering an ontology for a multi-agents corporate memory system. In *Proceedings of the International Symposium on the Management of Industrial and Corporate Knowledge*, pages 209–228, 2001.
- A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider. Sweetening ontologies with DOLCE. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, volume 2473 of *Lecture Notes in Computer Science*, pages 166–181, Sigüenza, Spain, October 2002. Springer-Verlag.
- A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann. Ontology evaluation and validation. Technical report, Laboratory for Applied Ontology, 2005. URL [http://www.loa-cnr.it/Files/OntoEval4OntoDev\\_Final.pdf](http://www.loa-cnr.it/Files/OntoEval4OntoDev_Final.pdf). Accessed January 2008.
- J. Gascuea, A. Fernandez-Caballero, and P. Gonzalez. Domain ontology for personalized e-learning in educational systems. In *Sixth International Conference on Advanced Learning Technologies*, pages 456–458, 2006.
- A. Gómez-Pérez. Evaluation of ontologies. *International Journal of Intelligent Systems*, 16:391–409, 2001.
- A. Gómez-Pérez. Evaluating ontology evaluation (in Why evaluate ontology technologies? Because they work!). *IEEE Intelligent Systems*, 19(4):74–81, 2004.

- A. Gómez-Pérez. Towards a framework to verify knowledge sharing technology. *Expert Systems With Applications*, 11(4):519–529, 1996.
- T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
- T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal Human-Computer Studies*, 43(5-6):907–928, 1995.
- M. Grüninger and M. Fox. Methodology for the design and evaluation of ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI'95*, 1995.
- N. Guarino. Towards a formal evaluation of ontology quality (in Why evaluate ontology technologies? Because they work!). *IEEE Intelligent Systems*, 19(4):74–81, 2004.
- N. Guarino. Understanding, building, and using ontologies a commentary to “Using explicit ontologies in KBS development”. *International Journal of Human-Computer Studies*, 46: 293–310, 1997.
- N. Guarino. Some ontological principles for designing upper level lexical resources. In *Proceedings of the 1st International Conference on Lexical Resources and Evaluation (LREC)*, May 1998.
- N. Guarino and C. Welty. Evaluating ontological decisions with ontoclean. *Communications of ACM*, 45(2):61–65, 2002.
- P. Haase, A. Hotho, L. Schmidt-Thieme, and Y. Sure. Collaborative and usage-driven evolution of personal ontologies. In *Proceedings of the 2nd European Semantic Web Conference*, volume 3532 of *Lecture Notes in Computer Science*, pages 486–499, London, UK, 2005. Springer-Verlag.
- F. Hakimpour and A. Geppert. Global schema generation using formal ontologies. In *Proceedings of the 21st International Conference on Conceptual Modeling (ER)*, pages 307–321, 2002.

- I. Herman. *Web Ontology Language OWL*. World Wide Web Consortium (W3C), February 2004. URL <http://www.w3.org/2004/OWL>.
- H. Hotho, S. Staab, and G. Stumme. Ontologies improve text document clustering. In *Proceedings of 3rd IEEE International Conference on Data Mining (ICDM'03)*, pages 541–544, 2003.
- E. Hovy. Comparing sets of semantic relations in ontologies. In R. Green, C. A. Bean, and S. H. Myaeng, editors, *The Semantics of Relationships*, pages 91–110. Kluwer, 2002.
- E. Hovy. Using an ontology to simplify data access. *Communications of ACM*, 46(1):47–49, 2003.
- E. Hovy. Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC)*, 1998.
- E. Hyvönen, S. Saarela, and K. Viljanen. Ontogator: Combining view- and ontology-based search with semantic browsing. In *Proceedings of XML Finland*, October 2003.
- IEEE. Standard glossary of software engineering terminology, 1990. Std 610.121990.
- ISO9126. URL <http://www.iso.org>. Accessed February 2008.
- C. Jones, A. Abdelmoty, and G. Fu. Maintaining ontologies for geographical information retrieval on the web. In *On The Move to Meaningful Internet Systems: CoopIS, DOA, and ODBASE*, volume 2888 of *Lecture Notes in Computer Science*, pages 934–951. Springer-Verlag, 2003.
- C. Jones, A. Abdelmoty, D. Finch, G. Fu, and S. Vaid. The spirit spatial search engine: Architecture, ontologies and spatial indexing. In *Proceedings of the 3rd International Conference on Geographic Information Science*, volume 3234 of *Lecture Notes in Computer Science*, pages 125–139, Maryland, USA, 2004.

- C. Kaner and W. Bond. Software engineering metrics: What do they measure and how do we know? 10th International Software Metrics Symposium, September 2004. <http://www.kaner.com/pdfs/metrics2004.pdf>.
- C. Keet. *A Formal Theory of Granularity*. PhD thesis, Free University of Bozen-Bolzano, Italy, April 2008.
- M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of ACM*, 42:741–843, May 1995.
- B. Kitchenham, S. Pfleeger, and N. Fenton. Towards a framework for software measurement validation. *IEEE Transactions on Software Engineering*, 21(12):929–944, December 1995.
- H. Knublauch. Ontology-driven software development in the context of the semantic web: An example scenario with protege/owl. In *Proceeding of the International Workshop on the Model-Driven Semantic Web*, 2004.
- A. Kumar, B. Smith, and D. Novotny. Biomedical informatics and granularity: Conference papers. *Comparative and Functional Genomics*, 5(6-7):501–508, 2004.
- R. Lara, S. Han, H. Lausen, M. Stollberg, Y. Ding, and D. Fensel. An evaluation of semantic web portals. In *Proceedings of the Applied Computing International Conference*, 2004.
- D. Lenat. CYC: a large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.
- D. Lenat and R. Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.
- T. Lethbridge. Metrics for concept-oriented knowledge bases. *Software Engineering and Knowledge Engineering*, 8(2):161–188, June 1998.
- M. F. López, A. Gómez-Pérez, J. P. Sierra, and A. P. Sierra. Building a chemical ontology using Methontology and the Ontology Design Environment. *IEEE Intelligent Systems*, 14(5):37–45, January 1999.

- A. Lozano-Tello and A. Gómez-Pérez. OntoMetric: A method to choose the appropriate ontology. *Journal of Database Management*, 15(2):1–18, 2004.
- J. A. Macias and P. Castells. Dynamic web page authoring by example using ontology-based domain knowledge. In *Proceedings of the 8th International conference on Intelligent user interfaces*, pages 133–140, New York, USA, 2003. ACM Press.
- A. Maedche and S. Staab. Measuring similarity between ontologies. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management*, pages 251–263, London, UK, 2002. Springer-Verlag.
- A. Maedche, B. Motik, L. Stojanovic, R. Studer, and V. R. Ontologies for enterprise knowledge management. *IEEE Intelligent Systems*, 18(2):26–33, March/April 2003.
- G. Marchionini. *Information Seeking in Electronic Environments*. Cambridge University Press, 1995.
- D. L. McGuinness. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, chapter 6: Ontologies Come of Age, pages 171–195. MIT Press, 2002.
- D. L. McGuinness and F. van Harmelen. *OWL Web Ontology Language Overview*. World Wide Web Consortium (W3C), February 2004. URL <http://www.w3.org/TR/owl-features>.
- K. Moeller and D. Paulish. *Software Metrics: A Practitioner's Guide to Improved Product Development*. IEEE Press, 1993.
- I. Niles and A. Pease. Towards a standard upper ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, Ogunquit, Maine, October 2001.
- N. Noy, D. Rubin, and M. Musen. Making biomedical ontologies and ontology repositories work. *IEEE Intelligent Systems*, 19(6):78–81, November/December 2004.
- N. F. Noy and D. L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical report, Stanford Knowledge Systems Laboratory (KSL), March 2001. URL [http://protege.stanford.edu/publications/ontology\\_development/ontology101.pdf](http://protege.stanford.edu/publications/ontology_development/ontology101.pdf).

- B. Nuseibeh and S. Easterbrook. Requirements engineering: A roadmap. In *Proceedings of International Conference on Software Engineering (ICSE-2000) – Future of SE Track*, pages 35–46, Limerick, Ireland, June 2000. ACM Press.
- D. Oberle and P. Spyns. OntoWeb - A Semantic Web Community Portal. In *Fourth International Conference on Fourth International Conference on Practical Aspects of Knowledge Management (PACM)*, pages 189–200, 2002.
- OBI. Ontology for biomedical investigations. URL <http://obi.sourceforge.net>. Accessed February 2008.
- L. Obrst, H. Liu, and R. Wray. Ontologies for corporate web applications. *AI Magazine*, 24(3):49–62, 2003.
- Original Wikipedia Guidelines for Category Proposals and Implementations. URL [http://meta.wikimedia.org/wiki/Categorization\\_requirements](http://meta.wikimedia.org/wiki/Categorization_requirements). Accessed January 2008.
- A. Orme, H. Yao, and L. Etzkorn. Coupling metrics for ontology-based systems. *IEEE Software*, 23(2):102–108, 2006.
- A. Orme, H. Yao, and L. Etzkorn. Indicating ontology data quality, stability, and completeness throughout ontology evolution. *Software Maintenance and Evolution: Research and Practice*, 19(1):49–75, January 2007.
- S. Pfleeger and J. Atlee. *Software Engineering: Theory and Practice: 3rd Ed.* Prentice Hall, 2005.
- M. Piattini, M. Genero, and C. Calero. *Handbook of Software Engineering & Knowledge Engineering*, volume 2: Emerging Technologies, pages 325–342. World Scientific, 2001.
- D. M. Pisanelli, A. Gangemi, and G. Steve. Ontologies and information systems: the marriage of the century? In *Proceedings of Lyee Workshop*, 2002.
- R. Poli. Ontology and knowledge organization. In *Knowledge Organization and Change*, pages 313–319, Frankfurt, 1996. Indeks Verlag.



- M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
- R. S. Pressman. *Software engineering: a practitioner's approach*. McGraw-Hill, Inc., New York, USA, 6th edition, 2005.
- T. Saaty. How to make a decision: The analytic hierarchy process. *Interfaces*, 24(6):19–43, 1994.
- A. Schreiber, B. Dubbeldam, J. Wielemaker, and B. Wielinga. Ontology-based photo annotation. *IEEE Intelligent Systems*, 16(3):66–74, 2001.
- M. Scriven. *Evaluation thesaurus*. Sage Publications, Newbury Park, CA, 4th edition, 1991.
- J. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole, 2000.
- S. Staab, R. Studer, H. Schnurr, and Y. Sure. Knowledge processes and ontologies. *IEEE Intelligent Systems*, 16(1):26–34, 2001.
- L. Stojanovic, S. Staab, and R. Studer. elearning based on the semantic web. In *WebNet2001 - World Conference on the WWW and Internet*, pages 23–27, 2001.
- R. Studer, V. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *IEEE Trans. on Data and Knowledge Engineering*, 25(1–2):161–197, 1998.
- Y. Sure. Why evaluate ontology technologies? Because they work! *IEEE Intelligent Systems*, 19(4):74–81, 2004.
- S. Tartir, I. Arpinar, M. Moore, A. Sheth, and B. Aleman-Meza. OntoQA: Metric-based ontology quality analysis. In *Proceedings of the Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources at IEEE International Conference on Data Mining (ICDM)*, pages 45–53, Texas, USA, November 2005.
- TGN. Getty thesaurus of geographic names. URL <http://www.getty.edu/research/tools/vocabulary/tgn>. Accessed March 2008.

- Q. Tho, F. A. Hui, S.C., and T. Cao. Automatic fuzzy ontology generation for semantic web. *IEEE Transactions on Knowledge and Data Engineering*, 18(6):842–856, June 2006.
- R. Thompson and W. Croft. Support for browsing in an intelligent text retrieval system. *International Journal of Man-Machine Studies*, 30(6):639–668, 1989.
- M. Uschold and M. Grüninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.
- M. Uschold and M. King. Towards a methodology for building ontologies. In *IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.
- J. Vasconcelos, F. Gouveia, and C. Kimble. An organizational memory information system using ontologies. In *Proceedings of the 3rd Conference of Associacao Portuguesa de Sistemas de Informacao*, 2002.
- J. Völker, D. Vrandečić, and Y. Sure. Automatic Evaluation of Ontologies (AEON). In *Proceedings of the 4th International Semantic Web Conference (ISWC)*, volume 3729 of *Lecture Notes in Computer Science*, pages 716–731. Springer-Verlag, NOV 2005.
- D. Vrandečić and Y. Sure. How to design better ontology metrics. In *Proceedings of the 4th European Semantic Web Conference (ESWC)*, pages 311–325, 2007.
- C. Welty and N. Guarino. Supporting ontological analysis of taxonomic relationships. *Data and Knowledge Engineering*, 39(1):51–74, 2001.
- Wikipedia Online Editing Guidelines for Categories. URL <http://en.wikipedia.org/wiki/Wikipedia:Category>. Accessed January 2008.
- S. Yang, I. Chen, and N. Shao. Ontology enabled annotation and knowledge management for collaborative learning in virtual learning community. *Educational Technology & Society*, 7(4):70–81, 2004.
- H. Yao, A. Orme, and L. Eitzkorn. Cohesion metrics for ontology design and application. *Computer Science*, 1(1):107–113, 2005.

- J. Yu, J. Thom, and L. Alem. Group memory based on the task information. In *Recherche d'Information Assistée par Ordinateur (RIAO)*, pages 329–347, France, 2004.
- J. Yu, J. Thom, and A. Tam. Evaluating an ontology with OntoClean. In *Proceedings of 10th Australasian Document Computing Symposium (ADCS)*, page 84. School of Information Technologies, The University of Sydney, December 2005a.
- J. Yu, J. A. Thom, and A. Tam. Evaluating ontology criteria for requirements in a geographic travel domain. In *Proceedings of 4th International Conference on Ontologies, Databases and Applications of Semantics (ODBASE)*, volume 3761 of *Lecture Notes in Computer Science*, pages 1517–1534, Ayia Napa, Cyprus, 2005b. Springer-Verlag.
- J. Yu, J. Thom, and A. Tam. Ontology evaluation: Using Wikipedia categories for browsing. In *Proceedings of 16th Conference on Information and Knowledge Management (CIKM)*, pages 223–232. ACM Press, 2007.
- Y. Zhao and G. Karypis. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168, 2005.