

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT COMPUTERWETENSCHAPPEN
AFDELING INFORMATICA
Celestijnenlaan 200 A — B-3001 Leuven

Learnometrics: Metrics for Learning Objects

Promotor :
Prof. Dr. ir. E. DUVAL

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Xavier OCHOA

September 2008



KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT COMPUTERWETENSCHAPPEN
AFDELING INFORMATICA
Celestijnenlaan 200 A — B-3001 Leuven

Learnometrics: Metrics for Learning Objects

Jury :
Prof. Dr. ir. D. Vandermeulen, voorzitter
Prof. Dr. ir. E. Duval, promotor
Prof. Dr. ir. W. Van Petegem
Prof. Dr. B. Berendt
Prof. Dr. R. Rousseau
Prof. Dr. L. Carr
Prof. Dr. ir. E. Peláez

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Xavier OCHOA

U.D.C. 681.3∗H51

September 2008

©Katholieke Universiteit Leuven – Faculteit Ingenieurswetenschappen
Arenbergkasteel, B-3001 Heverlee (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

D/2008/7515/92
ISBN 978-90-5682-982-7

I believe the most striking quality of these regularities is that they are very sturdy. They seem to be highly resilient to ambiguity, and seem to be unique among simple functional forms in enjoying this quality. The implications of this are far reaching. It tells us that, if such a regularity adequately describes a phenomenon given proper conceptualization, then it would also tend to describe that phenomenon given erroneous or imperfect conceptualization. It is the sort of regularity one would expect to see emerge when one is investigating a phenomenon that is interesting, but regarding which we only *sort of* know what we are talking about. In other words, it is ideal for domains we all work in.

A. Bookstein, about Informetric Laws [Bookstein, 1997]

Preface

The concept of Learning Object has evolved from the need to reuse digital learning materials. Learning Object Technologies offer economic as well as pedagogical advantages. The learning materials are created just once, but used several times in different contexts, compensating the high cost of production. Also, high quality, thoughtfully designed, multimedia materials could be easily accessed by any instructor or learner. As favorable as they seem for learning, learning objects are not in mainstream use. While reusing existing multimedia resources and avoiding duplication clearly saves time and money, the effort required to identify, catalogue, store, search, retrieve, and finally to reuse a learning object is still significant [Downes, 2005] [Oliver, 2005] [Duval, 2004].

To build friendlier and smarter tools that would help instructors and learners to make easy and transparent use of learning objects, deep understanding on the different processes in the called Learning Object Economy [Campbell, 2003] is needed. This knowledge is not currently available due to a scarcity of empirical research on the workings of this economy.

This dissertation is divided in two parts. The first part will apply Informetrics studies to obtain insight into the main processes of the Learning Object Economy: supply and demand. These studies will be based on data collected from the use of real Learning Object systems.

Chapter 2 conducts the first detailed quantitative study of the process of publication of learning object. This process has been often discussed theoretically [Neven and Duval, 2002] [Sicilia et al., 2005], but never empirically evaluated. Several questions related to basic characteristics of the publication process are raised at the beginning of the chapter and answered through quantitative analysis. To consider a wide view of the publication process, this paper analyzes five types of repositories: Learning Object Repositories, Learning Object Referatories, Open Courseware Initiatives, Learning Management Systems and Institutional Repositories. Three repository characteristics are measured: size, growth and contributor base. The main findings are that the number of learning objects is distributed among repositories according to a power law, that the repositories mostly grow linearly in the number of objects and contributors and the number of

learning objects published by each contributor follows heavy-tailed distributions. This chapter also proposes and evaluates a simple model to explain the observed results. This model is based on three basic characteristics of the contributor base: publication rate, lifetime and growth. The evaluation of the model shows that it is a good approximation for the behavior of most common repositories. The chapter finally discusses the implications that these findings could have in the design and operation of Learning Object Repositories.

Chapter 3 presents the first quantitative analysis of the reuse of learning objects in real world settings. The data for this analysis were obtained from three sources: Connexions modules [Baraniuk, 2007], University courses and Presentation components [Verbert et al., 2006]. They represent the reuse of learning objects at different granularity levels. Data from other types of reusable components, such as software libraries, Wikipedia images and Web APIs, were used for comparison purposes. Three aspects of reuse are measured: 1) the percentage of objects reused from a collection, 2) the correlation between popularity and reuse and 3) the distribution of reuse among the objects. The result of this analysis is used to answer empirically several questions that until now have only been discussed theoretically [Littlejohn, 2003] [McNaught, 2003] [Collis and Strijker, 2004]. The chapter also presents a model of reuse that successfully explains the results of the analysis. Finally, the chapter discusses the implications that the findings of the quantitative analysis and the proposed model have in the field of Learning Object research.

The second part of the dissertation is focused on the proposal and evaluation of metrics extracted from descriptive, usage and contextual metadata. The main goal of these metrics is to improve the tools currently used to label and select learning objects.

Chapter 4 proposes metrics to estimate the quality of the metadata used to label the learning object before inserting it into a repository. Due to recent developments in automatic metadata generation [Meire et al., 2007] and interoperability between digital repositories [Simon et al., 2005], the production of metadata is now vastly surpassing manual quality control capabilities. Abandoning quality control altogether is problematic, because low quality metadata compromise the effectiveness of services that repositories provide to their users. To address this problem, this chapter, based on the Bruce & Hillman framework for metadata quality control [Bruce and Hillmann, 2004], proposes a set of quality metrics for metadata. Three experiments to evaluate the metrics are performed: 1) the degree of correlation between the metrics and manual quality reviews, 2) the discriminatory power between metadata sets and 3) the usefulness of the metrics as low quality filters. The implications of the results of the evaluation are discussed. Finally, the chapter proposes possible applications of the metrics to improve tools for the administration of learning object repositories.

Chapter 5 develops the concept of relevance in the context of learning object

search. It proposes a set of metrics to estimate the topical, personal and situational relevance dimensions. These metrics are derived mainly from usage and contextual information. An exploratory evaluation of the metrics is performed to measure their effectiveness. The metrics are combined by two methods: linear combination and RankNet [Richardson et al., 2006], to create a single ranking value. This chapter discusses the results of this evaluation, as well as its implications for learning object search engines.

Chapter 6 presents a Service Oriented Architecture on which a Metric Service could be provided to improve current Learning Object tools. It also describes how the metrics are calculated and discusses their scalability. Use cases of several projects that are currently using the Metrics Service are presented. This chapter closes with conclusions of the implementation experiences.

Finally, Chapter 7 concludes this dissertation with a summary of the main contributions of this research, a list of the of interesting open questions and final words about the significance of this research on the Learning Object Technology field.

Acknowledgement

The writing of this dissertation has been an exiting journey of learning and growth. I would like to thank all the people that have shared it with me in one way or another. In this short acknowledgement I wish to mention and offer my special thanks to the ones that have deeply contributed to the success of this work.

First, I want to thank my supervisor and friend Erik Duval. His guidance and wise advice in key moments of this path always leaded me to new findings. His knowledge, scientific intuition and curiosity have always inspired me. Working with him has been "serious fun".

I would also like to thank Enrique Pelez for his trust in me and the opportunity that he gave me to pursue my scientific interests. His work has not just created the environment where I was able to conduct my research, but also has inspired a whole generation of ecuadorian researchers.

My appreciation goes also to Wim van Petegem. Without his intervention and initial support this work would have not existed. His feedback during the whole process of this dissertation has been instrumental to keep this research "down to earth".

I want to express my sincere thanks to the members of my doctoral committee: Prof. Dr. Bettina Berendt, Prof. Dr. Ronald Rousseau and Prof. Dr. Les Carr. Their feedback and advice on the draft versions of this dissertation has greatly helped me to improve it. I have been honored of having them in my jury. My thanks also extend to Prof. Dirk Vandermeulen for chairing the committee.

I want to thank the current and former members of the HMDB group that have worked with me: Katrien Verbert, Riina Vourikari, Stefaan Ternier, Jehad Najjar, Michael Meire, Joris Klerkx, Bram Vandeputte, Martin Wolpers, Nik Corthaut, Sten Govaerts, Kris Cardinaels and Gonzalo Parra. They always welcomed me and made me part of the group. I have learnt a lot from each one of them.

I also want to thank my CTI colleagues, specially Katherine Chiluiza, Gonzalo Parra, Cristina Guerrero and Vicente Ordoez, for their help in this work.

I am grateful to Katrien Verbert and Jehad Najjar for providing me with very useful feedback in early versions of this dissertation.

I want to acknowledge the financial and administrative support provided by the

VLIR-ESPOL IUC program. Personal thanks to Annick Verheylezoon in Belgium and Catalina Vera in Ecuador for their time and efforts organizing the administrative details of my scholarship and trips.

Finally and more importantly, I want to deeply thank Maria Fernanda, my wife, and Eugenia, my daughter, for their love, patience and sacrifice during the latest four years. The only major downside of this journey was to be far from them. All this dissertation and the fruits that it could generate are dedicated to them.

Contents

Contents ix

List of Acronyms xiii

List of Figures xv

List of Tables xix

1 Introduction 1
- 1.1 Learnometrics . . . 1
- 1.2 Definitions . . . 4
  - 1.2.1 Learning Object . . . 4
  - 1.2.2 Learning Object Metadata . . . 5
  - 1.2.3 Learning Object Repository . . . 7
- 1.3 Learning Object Economy . . . 9
  - 1.3.1 Actors . . . 10
  - 1.3.2 Drivers, Enablers and Mediators . . . 11
- 1.4 Learning Object Life cycle . . . 13
- 1.5 Research Questions . . . 15
  - 1.5.1 Understanding the Publication of Learning Objects . . . 16
  - 1.5.2 Understanding the Reuse of Learning Objects . . . 16
  - 1.5.3 Quality Control for the Labelling Process . . . 17
  - 1.5.4 Relevance Ranking to Improve the Selection Process . . . 18
- 1.6 Outline . . . 19

2 Quantitative Analysis of the Publication of Learning Objects 21
- 2.1 Introduction . . . 21
- 2.2 Size Analysis . . . 24
  - 2.2.1 Number of Objects . . . 24
  - 2.2.2 Objects per Course . . . 30
- 2.3 Growth Analysis . . . 33

2.3.1 Content Growth . . . 33
2.3.2 Contributor Base Growth . . . 38
2.4 Contribution Analysis . . . 41
2.4.1 Contribution Distribution . . . 41
2.4.2 Lifetime and Publishing Rate . . . 43
2.5 Modeling Learning Object Publication . . . 49
2.5.1 Model Definition . . . 49
2.5.2 Model Validation . . . 51
2.5.3 Conclusions . . . 55
2.6 Implication of the Results . . . 56
2.7 Conclusions . . . 60

3 Quantitative Analysis of the Reuse of Learning Objects 63
3.1 Introduction . . . 63
3.2 Data Sources . . . 66
3.3 Quantitative Analysis . . . 67
3.3.1 Amount of Reuse . . . 68
3.3.2 Popularity vs. Reuse . . . 68
3.3.3 Distribution of the Reuse . . . 70
3.4 Interpretation of the Results . . . 72
3.4.1 Model . . . 72
3.4.2 Interpretation . . . 73
3.5 Implication of the Results . . . 75
3.6 Conclusion . . . 77

4 Metadata Quality Metrics for Learning Objects 79
4.1 Introduction . . . 79
4.2 Measuring Metadata Quality . . . 82
4.3 Quality Metrics for Metadata in Digital Repositories . . . 86
4.3.1 Completeness Metrics . . . 86
4.3.2 Accuracy Metrics . . . 88
4.3.3 Conformance to Expectation Metrics . . . 91
4.3.4 Consistency & Coherence Metrics . . . 94
4.3.5 Accessibility Metrics . . . 97
4.3.6 Timeliness Metrics . . . 100
4.3.7 Provenance Metrics . . . 102
4.4 Evaluation of the Quality Metrics . . . 103
4.4.1 Quality Metrics correlation with Human-made Quality Assesment . . . 103
4.4.2 Quality Metrics comparison between two metadata sets . . . 112
4.4.3 Quality Metrics as automatic low quality filter . . . 114
4.4.4 Studies Conclusions . . . 118
4.5 Implementation and Applications of Metadata Quality Metrics . . . 119

4.6 Related Work 122
4.7 Conclusions 123

5 Relevance Ranking Metrics for Learning Objects 125
5.1 Introduction 125
5.2 Current Status of Learning Object Ranking 127
5.2.1 Ranking based on Human Review 127
5.2.2 Ranking based on Text Similarity 129
5.2.3 Ranking based on User Profile 130
5.2.4 Current Approaches vs. Ideal Approach 130
5.3 Relevance Ranking of Learning Objects 131
5.4 Ranking Metrics for Learning Objects 133
5.4.1 Topical Relevance Ranking Metrics. 134
5.4.2 Personal Relevance Ranking Metrics 138
5.4.3 Situational Relevance Ranking Metrics 142
5.4.4 Ranking Metrics Comparison 144
5.5 Learning to (Learn)Rank 146
5.6 Validation Study 148
5.6.1 Study Setup 148
5.6.2 Results 150
5.6.3 Discussion of the Results 151
5.6.4 Study Limitations 154
5.7 Conclusion 155

6 Metrics Service Architecture and Use Cases 157
6.1 Introduction 157
6.2 Service Oriented Architecture 158
6.3 Implementation of the Metric Service 161
6.3.1 Metadata Quality Metrics 162
6.3.2 Ranking Metrics 165
6.3.3 Scalability 167
6.4 Uses Cases 169
6.4.1 OER Commons Metadata 169
6.4.2 MELT Project 171
6.4.3 MACE Project 171
6.4.4 Ariadne Finder 172
6.4.5 Early Feedback 174
6.5 Conclusions 175

7 Conclusions **177**
7.1 Main Contributions . . . 177
7.1.1 Publication of Learning Objects . . . 177
7.1.2 Reuse of Learning Objects . . . 179
7.1.3 Quality of Learning Object Metadata . . . 179
7.1.4 Relevance of Learning Objects . . . 180
7.2 Further Research . . . 181
7.2.1 Quantitative Studies . . . 181
7.2.2 Metrics for Learning Objects . . . 183
7.3 Final Words . . . 184

A Metadata Quality Metrics Interface **207**
A.1 Metadata Quality Metrics Description . . . 207
A.1.1 Repository Level . . . 208
A.1.2 Subset Level . . . 209
A.1.3 Instance Level . . . 209
A.2 Calculate All Metadata Quality Metrics . . . 210
A.2.1 Repository Level . . . 210
A.2.2 Subset Level . . . 211
A.2.3 Instance Level . . . 213
A.3 Calculate Selected Metadata Quality Metrics . . . 214
A.3.1 Repository Level . . . 214
A.3.2 Subset Level . . . 215
A.3.3 Instance Level . . . 215
A.4 Calculate a Metadata Quality Metric for Selected Fields . . . 216
A.4.1 Repository Level . . . 216
A.4.2 Subset Level . . . 218
A.4.3 Instance Level . . . 219

B Ranking Metrics Interface **223**
B.1 Ranking Metrics Description . . . 223
B.1.1 Global Level . . . 223
B.1.2 Result List Level . . . 224
B.2 Calculate Ranking Metrics . . . 225
B.2.1 Global Level . . . 225
B.2.2 Result List Level . . . 227

List of Acronyms

AIC	Akaike Information Criterion
ALOCOM	Abstract Learning Object Content Model
CAM	Contextualized Attention Metadata
DC	Dublin Core
DRM	Digital Rights Management
ESPOL	Escuela Superior Politecnica del Litoral
IEEE	Institute of Electrical and Electronics Engineers
IR	Institutional Repository
LMS	Learning Management System
LOM	Learning Object Metadata
LOR	Learning Object Repository
LORP	Learning Object Repository
LORF	Learning Object Referatory
LSA	Latent Semantic Analysis
MACE	Metadata for Architectural Contents in Europe
MELT	Metadata Ecology for Learning and Teaching
MIT	Massachusetts Institute of Technology
MLE	Maximum Likelihood Estimation
OAI-PMH	Open Access Initiative - Protocol Metadata Harvesting
OCW	Open CourseWare
OER	Open Educational Resources
ROAR	Registry of Open Access Repositories
SamgI	Simple Automated Generated Metadata Interface
SCORM	Sharable Content Object Reference Model
SPI	Simple Query Interface
SQI	Simple Query Interface
SVD	Simgle Value Decomposition
TFIDF	Term Frequency - Inverse Term Frequency
UGC	User Generated Content

List of Figures

1.1 Classification of Learnometrics 2
1.2 Informetric research pattern 3
1.3 Learning Object Economy. Taken from [Johnson, 2003] 10

2.1 Distribution and Leimkuhler curve of the Size of Learning Object Repositories 25
2.2 Distribution and Leimkuhler curve of the Size of Learning Object Referatories 26
2.3 Distribution and Leimkuhler curve of the Size of Open Courseware Initiatives 27
2.4 Distribution and Leimkuhler curve of the Size of Learning Management Systems 28
2.5 Distribution and Leimkuhler curve of the Size of Institutional Repositories 28
2.6 Empirical and Fitted Course Size Distributions 32
2.7 Empirical and Fitted Size Growth 36
2.8 Empirical and Fitted Contributor Base Growth 40
2.9 Empirical and Fitted Distribution of Number of Publications between Contributors 44
2.10 Comparison between Lifetime Distribution between Repository Types. 48
2.11 Empirical and Simulated Distributions of Publicantion and Growth Function 53

3.1 Scatter plots of the Reuse vs. Popularity in the Connexions and Freshmeat sets 69
3.2 Size-Frequency graphs of the data sets (points) and the best fitting Log-Normal distribution (line) 71

4.1 Mapping between the Bruce & Hillman and the Stvilia et al. frameworks. (Taken from [Shreeves et al., 2005]) 84

4.2 Procedure to establish the linking between instances, based on classifying concept . . . 98
4.3 Calculation of the Source Reputation and the Provenace of each instance (R represent the instances and S the sources) . . . 102
4.4 Screen were the reviewer is presented with the metadata of the object, the option to download and to rate its quality . . . 104
4.5 Comparison between the average quality score and the textual information content metric values) . . . 109
4.6 Range explanation. 4 ranges were selected from the quality metric value to indicate 4 groups (R1, R2, R3 and R4) of increasing metric value . . . 115
4.7 Distribution of human selection of lowest quality instances among Ranges of the Quality Metrics. R1 are the lowest metric values and R4 are the highest metric values. . . . 117
4.8 Visualization of the Textual Information Content of the ARIADNE Repository. Red (dark) boxes indicate authors that produce low quality descriptions. . . . 121
4.9 Visualization of the Completeness of the Manual Metadata set extracted from MIT OCW. Dark boxes represents instances that are incomplete. . . . 122

5.1 Calculation of the SimRank between courses for the Course-Similarity Topical Relevance Ranking (CST) . . . 136
5.2 Calculation of Internal Topical Relevance Ranking (IT) . . . 138
5.3 Results of the Kendall tau distance from the manual ranking of the individual metrics . . . 152
5.4 Results of the Kendall tau distance from the manual ranking of the combined metrics . . . 153

6.1 Architecture for Metrics Services . . . 160
6.2 Visualization of the Qtinfo metric of the OER Commons Harvested Metadata . . . 170
6.3 Architecture of the MELT Project including Metadata Quality Metrics Service . . . 172
6.4 Architecture of the MACE Project including Ranking Metrics Service 173
6.5 Finder and ARIADNE Next Architecture . . . 173
6.6 Ariadne Finder interface with sorted results . . . 174

A.1 Schema of the result of GetMQMetricsDescriptions . . . 208
A.2 Schema of the result of repositoryGetAllMQMetricsValues . . . 211
A.3 Schema of the result of SubsetGetAllMQMetricsValues . . . 212
A.4 Schema of the result of instanceGetAllMQMetricsValues . . . 213
A.5 Schema of the result of repositoryGetMQMetricValuePerField . . . 217

A.6 Schema of the result of subsetGetMQMetricValuePerField 218
A.7 Schema of the result of instanceGetMQMetricValuePerField . . . 220

B.1 Schema of the result of globalGetRankMetricsDescriptions 225
B.2 Schema of the result of globalGetRankingMetricValues 226

List of Tables

2.1 Summary of Number of Objects Analysis 30
2.2 Course Size Distribution 31
2.3 Results of the Growth Analysis of the Repositories 37
2.4 Result of the Analysis of the Contributor Growth in the Repositories 39
2.5 Analysis of Distribution of Contribution 45
2.6 Result of the Analysis of Publishing Rate 46
2.7 Result of the Analysis of Publishing Rate. ALT is measured in days 47
2.8 Results of the Simulation of the Distribution of Publications 54
2.9 Simulation of the Size of the Repositories 55

3.1 Percentage of reuse in the different data sets. 68
3.2 Log-Normal distribution fitted parameters for each data set and the Vuong test significace against competing distributions. 72

4.1 Review of different quality evaluation studies 82
4.2 Example of the calculation of Qwcomp for a 4-field metadata instances 88
4.3 Example of the Qaccu values for two metadata instances 90
4.4 Example of the calculation of Qcinfo for 2 metadata instances . . . 92
4.5 Example of the calculation of Qtinfo for text of different words and lenghts 94
4.6 Recommendations for values in the LOM Standard (v.1.0) 95
4.7 Example of Qcoh calculation for 2 metadata instances 97
4.8 Example calculation of the Flesch Index for different texts 99
4.9 Example calculation of Qtime 101
4.10 Inter Class Correlation values for the rates provided by the human reviewers. 0.7 is the critical point for ICC 106
4.11 Example of the average quality value assigned to 6 of the 20 sampled instances. The first 3 were obtained from manually generated metadata, the last 3 from automatic generated metadata 107

4.12 Example of the metric values assigned to 6 of the 20 sampled instances. The first 3 were obtained from manually generated metadata, the last 3 from automatic generated metadata 107
4.13 Correlation between the human quality evaluation and the quality metrics. Bold font represents that the correlation is significant at the 0.01 level (2-tailed). Italic font represents that the correlation is significant at the 0.05 level (2-tailed). 108
4.14 Multivariate regression analysis of the quality parameters in function of the quality metrics. The Explanatory metrics specify which metrics where selected in the model (Stepwise) and their explanation power. 110
4.15 Metric values for the Manual and Automatic metadata sets, the correlation between the values for a same instance and the result of the comparison of means using the Paired T-Test. In bold, the highest quality average for each metric. 113
4.16 Directives given to reviewers to select the lowest quality instance according to each metric 116
4.17 Consistency percentage for each Comparison Sets 118
4.18 Effectiveness percentage for each metric. This indicate the percentage of times that the metric agreed with the human most voted instance. It also presents the percentage disaggregated for the Manual and Automated Metadata sets. 119

5.1 Map of Duval's "quality in context" characteristics into Borlund's relevance dimensions. 132
5.2 Correspondence of the Ranking Metrics with the Quality Characteristics and Relevance Dimensions. S = Strong, M = Medium, W = Weak, A = After Adaptation 146
5.3 Source Data needed to calculate the Ranking Metrics. QT = Query Time, OL = Off-Line 147
5.4 Task performed during the study and their corresponding query phrase 150
5.5 Average distances between the manual ranking and the calculated metrics and the average improvement over the Base Rank 151

6.1 Scalability of the Metadata Quality Metrics 165
6.2 Scalability of the Relevance Ranking Metrics 168

A.1 Repository Level Metadata Quality Metrics Description 208
A.2 Subset Metadata Quality Metrics Description 210
A.3 Instance Metadata Qualiy Metrics Description 210
A.4 Repository Level Calculate All Metadata Quality Metrics 210
A.5 Subset Level Calculate All Metadata Quality Metrics 212

A.6 Instance Level Calculate All Metadata Quality Metrics 213
A.7 Repository Level Calculate Metadata Quality Metrics 215
A.8 Subset Level Calculate Metadata Quality Metrics 215
A.9 Instance Level Calculate Metadata Quality Metrics 216
A.10 Repository Level Calculate Metadata Quality Metric for Fields . . 217
A.11 Subset Level Calculate Metadata Quality Metric for Fields 218
A.12 Instance Level Calculate Metadata Quality Metric for Fields 220

B.1 Global Level Ranking Metrics Description 224
B.2 Result List Level Ranking Metrics Description 225
B.3 Global Level Get Ranking Metric Value 226
B.4 Codes for Time Period 226
B.5 Result List Level Get Ranking Metric Value 227
and

Chapter 1

Introduction

1.1 Learnometrics

During the lastest 15 years of research in the area, the foundations of Learning Objects Technology have been developed. There are standards that define the metadata that should describe a learning object [IEEE, 2002] and how to sequence them [Bohl et al., 2002]. Thanks to these standards, Learning Management Systems (LMS) are able to import and export learning objects of different granularity. There are several repositories worldwide where authors can publish the learning objects they create and search for learning objects published by peers [Neven and Duval, 2002]. Thanks also to standardization, these repositories can query each other and present the user with a considerable number of results [Simon et al., 2005] [Van de Sompel et al., 2004]. Several scientists have described a Learning Object Economy [Polsani, 2003] [Duncan, 2003] [Campbell, 2003] where learning material is shared, re-used and improved. The predicted result of this economy is wider, cheaper access to relevant and high-quality learning objects for teachers and learners.

Despite all the work on these foundations, a fully functional Learning Object Economy has not materialized yet. Learning Object Technologies are not in mainstream use among teachers and learners [Downes, 2005] [Oliver, 2005] [Duval, 2004]. Given the strong theoretical and technological infrastructure on which Learning Objects are based, one of the most common cited issues with Learning Object Technologies is the lack of maturity of the end-user tools [Dodani, 2002] [Duval and Hodgins, 2003] [Ochoa, 2005]. In order to improve the adoption of learning object technologies, smarter and friendlier end-user tools must be developed. These tools should capitalize on the vast amount of information that is present in the learning object metadata and other sources including context and usage. To be exploitable, that information should be automatically measured and

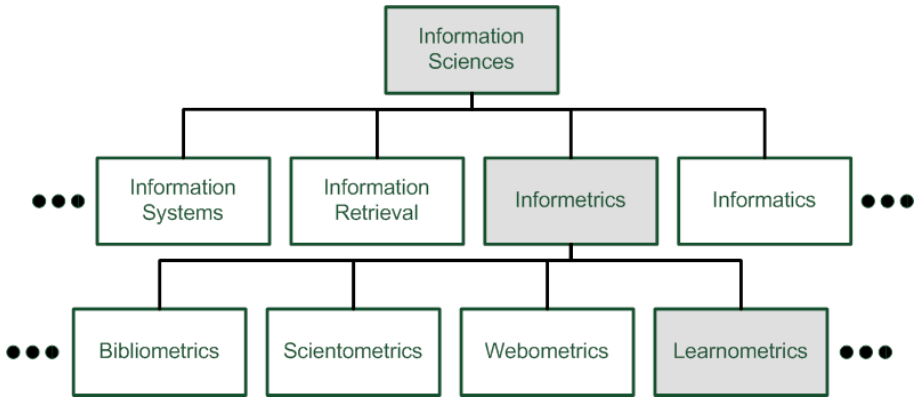


Figure 1.1: Classification of Learnometrics

processed to extract deep knowledge of the characteristics, relations, usefulness, behavior and recommended use of individual learning objects, as well as complete learning object repositories. Also, the measurement of the characteristics of different stages of the Learning Object life cycle [Collis and Strijker, 2004] would help us to understand how the Learning Object Economy really works.

This dissertation presents the measurement of several characteristics related to a learning object and the different processes that take place during its life cycle. We have called this initiative "Metrics for Learning Objects" or "Learnometrics" for short. This denomination was chosen to reflect the similarity of this study goal and methodology with the Informetric fields. For example, Bibliometrics, which is the scientific field focused on the measurement and analysis of texts and information [Broadus, 1987], Scientometrics, which measures the scientific process [Hood and Wilson, 2001] and Webometrics, that analyzes the behavior of the World Wide Web and the Internet [Almind and Ingwersen, 1997]. All these Informetric sciences fall under the umbrella of Information Sciences. A taxonomic classification of the proposed field of "Learnometrics" can be seen in Figure 1.1.

Informetrics is focused on measuring and understanding processes that create, publish, consume or adapt information. Moreover, it is common that after the process has been analyzed, useful metrics are developed to summarize characteristics of the process and then used to create tools that can have a practical application to improve the studied or a related process. Scientometrics, for example, has studied the scientific publication and citation processes. Extensive publication and citation data has been quantitatively analyzed. From these analyzes, it has been found that the number of publications per author and the number of citations per journal follows the Lotka law [Lotka, 1926]. Based on these findings, several scientists have suggested models that explain the publication and citation

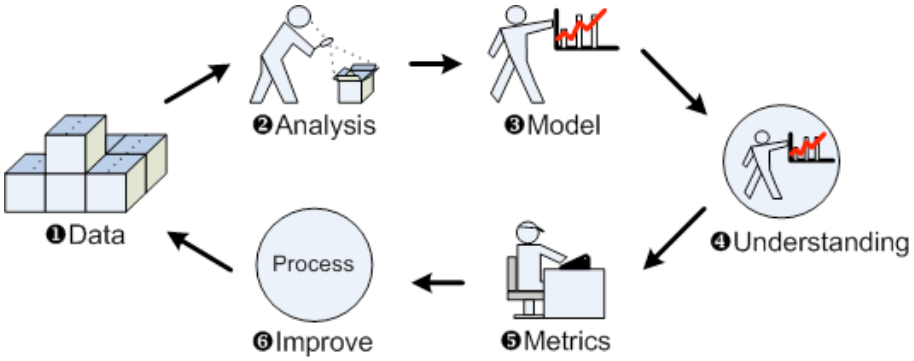


Figure 1.2: Informetric research pattern

process. "Success breeds success" [Egghe and Rousseau, 1995] and "Cumulative advantage" [Price, 1976] are two ways to express that the probability to publish a new scientific article or receive a new citation is proportional to how many articles the author has published before or how many citations the journal has already received. Practical metrics that have been extracted from these analyses are the Journal Impact Factor [Garfield, 1994] and the h-index [Hirsch, 2005]. Those metrics serve to summarize the scientific impact that a journal or a scientist has in a particular field. Moreover, these metrics, while not perfect [Jacsó, 2001], are often used as selection criteria in other scientific processes, such as when selecting a journal to publish research or selecting the most talented scientist to fulfill an academic position.

Another example of this pattern can be seen in Webometrics. The study of the links between Web pages leads to the discovery that the Web is a scale-free network, meaning that the number of links pointing to a web page is not Normally distributed, but also follows the Lotka law [Rousseau, 1997]. This discovery helped to create more accurate models for Web that differentiate from the original models based on random graphs. The idea that already heavily linked pages attract more links lead to the proposal of ranking metrics, such as HITS [Kleinberg, 1999] and PageRank [Page et al., 1998], to evaluate the relevance or importance of Web pages. These metrics are the core of current Web Search engines, that consistently present relevant results in the top places from a pool of more than 11.5 billion Web pages [Gulli and Signorini, 2005].

Figure 1.2 presents a visual representation of the Informetric research pattern. The following list presents the steps of this pattern:

1. Obtaining Data

2. Quantitatively analyze the Data

3. Create a Model of the Process that produced the Data
4. Use the Model to gain Understanding of the Process
5. Develop useful Metrics that summarize the Process characteristics
6. Use the Metric information to help/improve the same or a related Process

This Learnometric study follows a similar pattern. It quantitatively analyzes data from processes that take place during different points of the Learning Object life cycle. Based on that analysis, initial models are proposed to explain the observed results. The study also proposes small calculations (metrics) that can covert the data available about the learning objects into information that can be used to improve the effectiveness or usefulness of existing Learning Object end-user tools.

Before the specific goals and research questions of this dissertation are discussed, certain definitions needed to understand the Learning Object research field are presented. This chapter is organized as follows: Section 2 presents a series of basic definitions. Section 3 discusses the research visions of what the Learning Object Economy is and how it should behave. Section 4 presents the different processes involved in the life cycle of a Learning Object. Finally, Section 5 explains the main research issues and questions that this dissertation addresses and Section 6 presents how those issues and questions are covered in the different chapters.

1.2 Definitions

Learning Object Technologies have been suffering from their beginning from a strong disagreement on their basic definitions. This disagreement has caused critics [Friesen, 2004] and even unfair rejection [Amory, 2005] of Learning Object Technologies as a whole. Moreover, one of the main consequences of this lack of consensus in these definitions is the confusion between practitioners [Polsani, 2003]. The origin of this disagreement is the fuzzy nature of what can be called a Learning Object. This section presents the main competing definitions for Learning Object, Learning Object Metadata and Learning Object Repository, as well as the one selected as the working definitions that will be considered through the following chapters.

1.2.1 Learning Object

Defining what is a Learning Object is arguably the most discussed topic in Learning Object Technologies. While most researchers could easily differentiate what is and what is not a learning object, producing a consensus for its definition has been

very difficult. Perhaps the most broad, and at the same time more criticized, definition has been proposed by IEEE Learning Technologies Standard Committee (LTSC) [IEEE, 2002]. This definition states that a learning object is "any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning". While the purpose of this definition is to avoid excluding any existing learning object, it is also true that under that definition, anything can be a learning object [Liber, 2005]. A more concrete definition is proposed by Wiley [Wiley, 2002]. In this definition, a learning object is "any digital resource that can be reused to support learning".

Other definitions restrict what can be considered a learning object based on intrinsic characteristics of the resource. Rehak and Mason propose that a learning object should be reusable, accessible, interoperable and durable [Rehak and Mason, 2003]. Similarly, Downes considers that only resources that are shareable, digital, modular, interoperable and discoverable can be considered learning objects [Downes, 2004]. However, those definitions do not propose an objective way to measure those properties. For example, it is expected that the reusability of a learning object varies according to the context where it should be reused. A high quality learning object in Spanish would be very hard to reuse in an English speaking class. Also, some of those characteristics not only depend on the learning object. The discoverability of a learning object does not only depend on the object and its metadata, but on the features of the search engine. While the purpose of these definitions is to restrict the label of learning objects to certain type of learning material, the borders of the definitions are too fuzzy and do not provide any operational advantage over Wiley's definition.

Other definitions use less subjective characteristics to restrict even more what can be considered a learning object. L'Allier defines a learning object as a learning objective, a unit of instruction and a unit of assessment packed together [Polsani, 2003]. The Wisconsin Online Resource Center considers that only small units of content with duration between 2 and 15 minutes can be considered learning objects [Chitwood et al., 2000]. While these are more operational definitions, it is clear that much of the learning material that can be found online does not comply with these constrains.

The analyses performed during this dissertation need the most general and operational definition for learning objects. Therefore, we rely on Wiley's definition any time that the Learning Object concept is cited in the following chapters.

1.2.2 Learning Object Metadata

Any data that can be used to describe a learning object can be considered as learning object metadata. According to the IEEE Learning Technologies Standard Committee, the purpose of the metadata is to facilitate the "search, evaluation, acquisition, and use of learning objects". Therefore, a general definition of learning object metadata is any piece of information that can be used to search, evaluate,

acquire and use learning objects. For example, the title of a learning object would help to find a relevant learning object. A review created by a user would help to evaluate the relevance of the object for another user. The link pointing to actual resource, as well as the information about the copyrights of the object would help to acquire the object properly. Finally, the technical information about the object, such as file type or size, would help the user to select the right tools to use the object.

This definition is widely agreed upon within the community. However, there is a certain degree of confusion about how this definition relates to the learning object metadata standards. A Learning Object Metadata standard defines a set of data fields that describe a learning object. For example, a schema could include the title, the author, a classification and the publication date (similar to a library record). Another schema should define the title and duration and intended learner age as the most important metadata to store. The purpose of these schemas is to enable interoperability between different systems that contain learning object metadata. If both systems agree on a common metadata standard, then the exchange of information about learning objects is, at least in principle, possible.

The most commonly used metadata standard for learning objects is LOM (Learning Object Metadata) [IEEE, 2002]. This standard was sanctioned by the IEEE Lerning Technologies Standard Committee. LOM proposes around 50 different metadata fields grouped into nine categories:

- General: Description of general characteristics of the learning object (title, description, language, etc.)
- Life cycle: Entities that have created or altered the learning object and its current status (version, status, contributor, etc.)
- Meta-Metadata: Information about the metadata instance (language, contributor, lom version, etc.)
- Technical: Information about the technological characteristics and needs of the learning object (size, file type, format, requirements, etc.)
- Educational: Educational and Pedagogical characteristics and needs of the learning object (interactivity type, semantic density, difficulty, etc.)
- Rights: Copyright information and conditions of use (cost, restrictions, etc.)
- Relation: Relationship of the learning object with other learning objects (resource, kind, etc.)
- Annotation: Comments or reviews from users or systems (entity, description, etc.)

- Classification: Taxonomic path in a particular classification system (source, purpose, taxon path, etc.)

LOM, however, is not the only metadata standard used to describe learning objects. Standards used for other types of digital resources, such as Dublin Core (DC), are also used to describe learning objects. Dublin core is much more simple, but at the same time less expressive, than LOM. In its simple version, DC only considers 15 elements: title, creator, subject, description, publisher, contributor, date, type, format, identifier, source, language, relation, coverage and rights. These fields, being generic for any type of digital resource, do not capture any information about the potential educational use of the object. A working group [Weibel and Koch, 2000] is currently extending DC to include such educational elements in what is called the DC-Education proposal.

The analyses in this dissertation are standard agnostic. Due the lack of a better, commonly agreed term, "metadata instance" is used to describe the group of information that describes a given learning object. This metadata instance could conform or not to any of the metadata standards. This metadata instance could contain information not considered part of LOM or DC-Educational, such as usage, popularity and even metrics values.

1.2.3 Learning Object Repository

Learning Objects can be shared in several ways. They can be just published on the web, made available in online forums or even pass personally from user to user. This thesis however, concentrate in the most formal way of learning object sharing: Learning Object Repositories. To share an object in this way, the object is indexed in what is called a Learning Object Repository (LOR). In their most common form, LORs usually store the learning object itself and the metadata instances associated with it. These LORs provide some sort of indexation facility, where users can add new learning objects together with their metadata. Also, some sort of search or browsing facility is provided to provide access to the content of the repository.

An important sub-type of LORs are the Learning Object Referatories. These only store the metadata, while the object itself is stored elsewhere, usually on a server on the Web.

The most popular examples of Learning Object Repositories are:

- ARIADNE Knowledge Pool System[1] [Duval et al., 2001]: It originated from a European project to create a repository of learning materials in the region. It is based on a distributed architecture that enables each node to keep control of its own materials. With almost 12 years of existence, it can be considered one of the oldest still operative repositories.

[1] ARIADNE Foundation. http://www.ariadne-eu.org

- Connexions[2] [Baraniuk, 2007]: A repository born from the need to share materials for Digital Signal Processing that has expanded to other fields. It can be considered to be one of the newest and, currently, most successful repositories. It is based on a Creative Commons [CreativeCommons, 2003] license. This license enables the free sharing and adaptation of the material.
- Maricopa Learning Exchange [3]: A small size repository belonging to a small group of institutions. The focus of this repository is to provide packaged objects that can be easily reused in Learning Management Systems. The contributor base is restricted only to faculty of Maricopa Community Colleges.

The most popular examples of Learning Object Referatories are:

- MERLOT[4] [Malloy and Hanley, 2001]: A USA initiative to catalog learning material on the web. It is one of the oldest referatories. It is open for contribution, but it has a system where experts review the on-line material and provide extensive reviews and ratings. This model is unique among the community of LORs. This referatory currently is still growing strongly.
- INTUTE[5]: An initiative based on UK where a group of experts also catalog on-line materials. It is closed to external contribution and tries to keep an uniform level of quality. Due to its age and continuous funding, INTUTE is one of the biggest referatories.

These LORs describe themselves as Learning Object repositories or referatories. However, due to the fuzziness of the concept of Learning Object, what can be considered a Learning Object Repository is also not clear-cut. Current Open Courseware (OCW) initiatives provide digital material that can be reused in a learning setting. However OCW sites do not identify themselves as Learning Object Repositories, even if they comply with the definition given at the beginning of this subsection. Learning Management Systems also store a great amount of learning material that is shared in a small community of the teacher and the students of a course. Even if they are not open, these systems can also be considered as LORs. Finally any type of digital library, such as Institutional Repositories, where digital learning material can be stored, could also be considered under this definition. During this dissertation LORs, OCWs, LMSs and IRs are considered as Learning Object Repositories.

As conclusion, in the context of this dissertation, a LOR is consider in its widest definition. Any system that stores digital learning material and that provides some sort of indexing and searching or browsing interface for those materials.

[2]Connexions. http://www.cnx.org
[3]Maricopa Learning Exchange. http://www.mcli.dist.maricopa.edu/mlx
[4]MERLOT. http://www.merlot.org
[5]INTUTE. http://www.intute.ac.uk

1.3 Learning Object Economy

The main promise that fuels the research on Learning Object Technologies is the capacity to provide just the right content, to just the right learner, at just the right time, device and format [Hodgins, 2002]. However, this promise has not materialized yet. One pre-requisite to fulfil this promise is to have a healthy Learning Object Economy [Campbell, 2003].

In a Learning Object Economy, learning materials are produced by motivated individuals. These materials are made available to a global community of teachers and learners. The teachers or learners can find and access the most relevant materials for their particular context. If the materials can not be used "as is", the users can adapt them to their needs and publish the adapted version for others to use. The result of this iterative process is that learning materials of increasing quality are made available to a growing community of producers-consumers or prosumers [Toffler, 1981]. It is expected that this economy produces not only very popular and highly reused learning objects, but also a wide-range of materials tailored to several niches of individuals and contexts. In other words, it is expected that a long-tail market appears [Anderson, 2006]. This long-tail market will provide just the right material in just the right format.

The commodity being exchanged in this economy is the Learning Object. Producers, most commonly teachers or experts, create learning objects that can be shared with others. Consumers, most commonly other teachers or learners, reuse those materials in their own learning designs or experiences. It is important to note that in this market pure producers and consumers are rare. Usually, individuals play both roles. For example a teacher could produce the first lesson of a course because it was not available before, but can also fully reuse the second lesson from existing material and adapt existing content for the third lesson. Even students, through their assignments and evaluations, could be considered producers in this economy. For example, the class notes taken by one student can be reused by the students of the next year.

In September 2002, at San Francisco, CA, USA, a group of 15 experts in Learning Objects from diverse sectors of business, higher education, and government met to discuss the status and future of this economy [Johnson, 2003]. The main conclusion of this meeting was that the Learning Object Economy was, at that time, not pervasive. Much of the efforts in Learning Object Technologies were focused on reaching a "tipping point" or "critical mass" in terms of scale. While this conclusion seems to continue being true after six years, the most durable result of this meeting is the identification of actors, drivers, enablers and mediators involved in the Learning Object Economy. A summary of those conclusions [Johnson, 2003] is presented here.

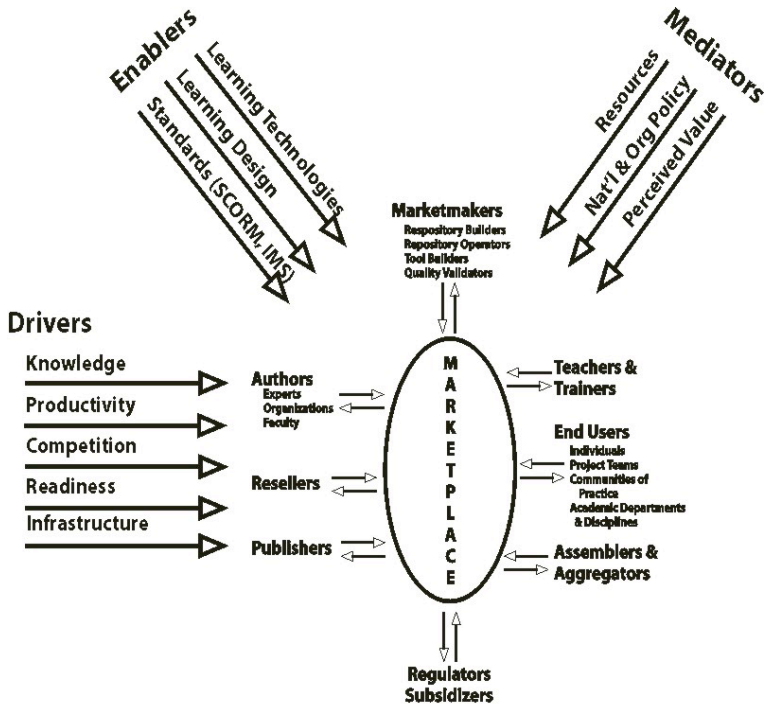


Figure 1.3: Learning Object Economy. Taken from [Johnson, 2003]

1.3.1 Actors

Eight groups of actors were identified in the Learning Object Economy (Figure 1.3): Market-Makers, Authors, Resellers, Publishers, Teachers, End Users, Assemblers and Regulators. These eight could be grouped into four bigger categories: Market-Makers, Contributors, Consumers and Policy-Makers. A brief description of each of these four categories follows:

- Market-Makers: These actors are in charge of providing the infrastructure where the Learning Object interchange could take place. Example actors in this group are the Organizations that provide Learning Object Repositories, Open Courseware sites, Learning Object Technologies researchers and trainers. While they decide on the technical and pedagogical aspects of the interchange, they do not control how the actual economy works.

- Contributors: This group is in charge of supplying the learning objects to the economy. This group encompasses the original author of the learning object, the publishers and resellers of learning material and assemblers of

existing materials and also regular users that bookmark relevant learning material on the web. The term Contributor is used instead of the most common term Producers to reflect that they not necessarily need ownership over the material in order to make it available to others.

- Consumers: This group involves the Teachers and End-Users (learners). They gain access to the learning material and reuse the material in their own learning designs or experiences. As mentioned before, these consumers could also have the role of Contributors if they adapt and share again the material.

- Policy-Makers: An often forgotten, but critical group of actors in the Learning Object Economy. These Policy-Makers set the rules by which the sharing takes place. A real example of this group are the authorities at MIT that decide that the learning objects of all their courses should be made available under a Creative Commons license. This policy caused an expansive movement of open access to learning materials [JOCW, 2006]. The rules set by these Policy-Makers could make or break the market.

One aspect that has always been criticized in the Learning Object Technology field is the separation between these four groups of actors [Rehak and Mason, 2003]. This dissertation tries to create information and metrics that could help Market-Makers to understand how Contributors and Consumers are interchanging objects in their repositories. These metrics could also be used by Policy-Makers to understand the effect of their policies in Market-Makers, Contributors, Consumers and the economy in general.

1.3.2 Drivers, Enablers and Mediators

Despite the unclear definitions, resistance to change and difficult to use end-user tools, researchers and practitioners are still trying to make the Learning Object Economy a reality. This effort is fueled by several external drivers that not only suggest but require the existence of a Learning Object Economy. The drivers identified in the San Francisco meeting are reproduced here from [Johnson, 2003]:

- Knowledge: The expanding body of knowledge is creating an immediate and recurring need for learning across organizations, sectors, and nations.

- Productivity: A demand for ever-increasing productivity requires people and organizations to work smarter.

- Competition: Intra-sector, national, and international competition for markets, for resources, or simply for “an edge” creates a rationale for rapid learning solutions.

- Readiness: A need to be prepared for unanticipated situations increases reliance on "just-in-time" learning.
- Infrastructure: A rapidly evolving information infrastructure provides a mechanism for quick access to a large amount of material.

The pushing effect of this drivers in the Learning Object Economy is clear. What is less clear, however, is to what extent they shape how the economy works. For example, to which point competition, instead of cooperation, has reduced the opportunities of free interchange of learning object among organizations. While understanding how this major forces affect the Learning Object Economy requires much diverse research than just quantitative analyses, this dissertation find interesting characteristics of the economy that can be the starting points for deeper and wider studies of the effect of the drivers.

Although the importance of the drivers, their pressure does not, by itself, brings this economy to reality. In order to develop a Learning Object Economy, several enablers should be in place. Again, the San Francisco meeting identified three of those enablers [Johnson, 2003]. They are summarized here:

- Learning Technologies: Learning objects need to be created, adapted, stored, found and inserted. Learning Technologies provide the tools and infrastructure for those process to happen. For example, Learning Management Systems seems to be one of the main enablers for the publication of learning objects.
- Learning Design: Learning Objects are just content. To have real learning experiences those contents need to be sequenced and administered in a pedagogically sound way. The adequate pedagogical theories and techniques need to be in place in order to assure that learning objects have real impact.
- Standards: If a global Learning Object Economy is the goal, there must be common-agreed standards that enable the sharing of learning objects between heterogeneous systems. IEEE LOM and SCORM [Bohl et al., 2002] are examples of these standards.

Beside the enablers, the success of a Learning Object Economy depends on characteristics of its internal markets. These characteristics, defined as Mediators, could facilitate or hinder the evolution of this Economy, depending on their abundance or scarcity. The identified Mediators at the San Francisco meeting are:

- Resources: The construction of the Learning Object economy depends on the amount of resources available. For example, if the motivated individuals do not have time to create or improve learning objects, the amount of learning objects produced will be small. If there are no funds available to create and operate Learning Object repositories, there will be no marketplace where the learning object exchange could take place.

- Policies: As mentioned above, the policies and regulations could make or break the market. Policies that encourage and compensate sharing of resources have an immediate positive effect in the fluidity of the Learning Object exchange [Baraniuk, 2007].
- Perceived Value: Maybe one of the most important mediators in the Learning Object Economy is the value proposition for the contributor. Money, recognition or career advancement are proposed forms of incentive [Campbell, 2003].

The analyses performed during this dissertation provide insight into how the presence or absence of the enablers and mediators alter the workings of the Learning Object Economy. For example, the results of the studies of chapter 1 suggest that the Perceived Value has a leading role in the productivity of the contributor base of a repository and chapter 2 offers an explanation to why the existence of the adequate Learning Technology tools improves the amount of reuse.

1.4 Learning Object Life cycle

The life cycle of learning objects has been analyzed by several researchers [Dalziel, 2002] [Collis and Strijker, 2004] [Cardinaels, 2007]. Perhaps the most cited and agreed upon life cycle definition is the one provided by Collins and Strijker in [Collis and Strijker, 2004] with six distinct stages. These stages are deeply interrelated with the circulation of the object in the Learning Object Economy. Given that the chapter of this dissertation will be focused in specific stages of this life cycle, these stages are briefly described here. They are also linked to the Learning Object Economy and to its different actors.

1. Obtaining:
   The first stage of the life cycle is obtaining or creating a learning object. This stage is usually executed by a Contributor. If the learning object is created, the contributor assembles a new learning resource from existing or, also, new digital materials. If obtained, the contributor discovers the learning object in online searches or analog sharing with partners. The usual reason behind this step is the personal or institutional need for the learning material. The most common example of this step is a teacher creating a slide presentation to use during class.

2. Labelling:
   In this stage, metadata is added to the learning object to describe it. While Collins and Strijker present it like a finite and separate stage, it can be consider as a pervasive process that is constantly adding information to the

description of the object each time that the object is used [Cardinaels, 2007]. In their traditional, static, interpretation, this stage is executed by the Contributor or by the Market-Makers. The Contributor can specify various metadata values, such as title, description, difficulty, etc. Market-Makers can also create or improve metadata for objects in their repositories. The reason behind this stage is not personal use as in the previous stage, but sharing the resource with a community. This is the reason why subjecting the contributor to a tedious form filling routine each time that she wants to share an object diminishes the appeal to contribute objects to the Learning Object Economy. Recently, however, automatic metadata generation for Learning Objects [Cardinaels et al., 2005], has helped Market-Makers to reduce or even eliminate the need to ask the contributor to submit metadata together with their objects. This automatic generation of metadata is also used to realize the more dynamic and pervasive nature of the labelling stage, where new information is added with each use of the object.

3. Offering:
   During this stage, the learning object is published so that it can be accessed by the community. In the case of LORs, the learning object, together with its metadata is inserted into the repository. The main actors involved in this stage are the Contributor and the Market-Maker. The contributor offers her material, the market-maker makes it available to the community. This offering not necessarily means that the object is made freely available to use inside the community, but only that its description (metadata) can be findable. If a learning object is never found or reused by another user, this is the last stage in its life cycle. Policy-Makers also play an important role in this stage. They define under which conditions the offering could take place.

4. Selecting:
   In this stage, the actor searches and selects the most relevant learning objects for her information needs. The actors directly involved in this stage are the Market-Makers and the Consumers. Market-Makers are in charge of providing some functionality to find learning objects. Consumers evaluate the offer provided by the Market-Makers and select what they consider are the most relevant. This stage is one of the most critical for the Learning Object Economy. Market-Makers act like brokers that connect the need of the Consumers with the offer of the Contributors. If consistently the repository is not able to present relevant objects, the Consumers withdraw from the market.

5. Using:
   This stage is the main goal of the Learning Object Economy. Here the Consumers use the selected objects inside their own learning design or experience. There are two ways in which the object could go through this stage:

"pure" reuse or repurposing. In pure reuse, the object is used "as is" by the Consumer. In the case of repurposing, the learning object is altered or decomposed by the Consumer to fit her specific needs. Other actors that influence the Using stage are the Contributor and the Policy-Makers. This influence is made by the licensing policy set on the object. Restrictive licensing would only allow the object to be reused and not repurposed. Other licences condition the reuse or repurposing of the learning object to certain communities or contexts (non-commercial use, for example).

6. Retaining:
   This is the final stage of the Learning Object life cycle. During its lifetime, a learning object could become outdated or no longer necessary. Also, new versions of the object could become available. The Consumer decides whether to maintain, discard or replace the learning object. The most common example of this phase is when a teacher is preparing a new semester or year of her course and evaluates the pertinence and validity of the objects that she used the previous year. When a new object is needed because the current one is outdated, a new Obtaining stage could start closing the life cycle.

The most important and visible processes involved in the learning Object life cycle, Labelling, Offering, Selecting and Using, will be studied in the following chapters. The measurements and metrics generated by the different analyses could be used to improve the tools used in those stages and to gain understanding on how those processes actually work.

1.5 Research Questions

Research in the area of Learning Object Technologies has produced the theoretical and technological infrastructure over which much of today's Learning Object Economy works. Fruits of this research are the Learning Object Metadata standard, Learning Object Repositories, interoperability solutions [Simon et al., 2005] [Van de Sompel et al., 2004], automatic metadata generation [Cardinaels, 2007] [Meire et al., 2007] and decomposition tools [Verbert et al., 2006]. However, little is actually known about how the processes involved in the Learning Object Economy are evolving and how that information could be made available to improve our current tools.

Given the amount of research needed to cover all processes involved in the Learning Object Economy, this dissertation will focus on the two main aspects of any market: supply and demand. In the first part of the dissertation, quantitative studies will be performed to the Offering and Using stages of the Learning Object life cycle. The second part of this dissertation, on the other hand, will analyze how to measure and improve the processes that preceded those studied in the first part:

Labelling and Selecting. The following subsections present the specific research questions that this dissertation tries to answer.

1.5.1 Understanding the Publication of Learning Objects

The first step to understand the Learning Object Economy is to measure and understand how learning objects are offered or published. A literature review on the topic is very discouraging. The only serious work that tries to quantitatively measure the publication of learning objects is [McGreal, 2007], which is, however, very superficial on its quantitative side and draws no conclusions from the results. This lack of research leads to an almost unexplored field with even the most basic questions unanswered. A list of the questions that this dissertation will address about the publication of Learning Objects are:

- What is the typical size of a repository? Is it related to its type (LORs, OCWs, LMSs or IRs)?
- How many learning objects are typically used in a course?
- How do repositories grow over time?
- What is the typical number of contributors for a repository? Is it related to its type?
- How does the number of contributors grow over time?
- How many learning objects does a contributor publish on average?

Answering these questions can help us to understand how many Learning Objects are published and where they are published. This information can guide the efforts made by Learning Object Technologies to provide the widest access to the largest collection of learning objects possible. The information gained about the growth of repositories can help in the capacity planning of existing and new repositories. Finally, knowing the characteristics of the contributors can help to design incentive programs that could solve the Value Perception of the publication of learning objects.

Chapter 2 presents a quantitative study over a wide variety of Learning Object Repositories designed to answer the abovementioned questions. Section 2.5 presents a model that can simulate the publication process of different types of repositories.

1.5.2 Understanding the Reuse of Learning Objects

Although reuse is the reason why much of Learning Object Technologies exist, little is quantitatively known about the Reuse process. Beside small scale experiment

in artificial settings [Schoner et al., 2005] [Elliott and Sweeney, 2008] [Verbert and Duval, 2007], there is practically no empirical data on how different factors affect the reusability of learning objects. Again, with an almost unexplored field, this dissertation proposes and aims to solve the following basic questions:

1. What percentage of learning objects is reused?
2. Is the amount of reuse in learning objects similar to other types of component reuse?
3. Does the granularity of a learning object affect its probability of reuse?
4. Is there a relation between the popularity of an object and its reuse?
5. What is the distribution of reuse among learning objects?
6. Is the distribution of reuse among learning objects similar to other types of component reuse?

The answers to these questions could help to test long held theoretical beliefs about the reusability of learning objects. The study of the popularity versus reusability could lead to the development of better measurements of the impact of Learning Object Repositories. Finally, gaining insight into the distribution of reuse among different objects can help theorists to study what differentiates a highly reused learning object from the others.

Chapter 3 collects available data about reuse from real world Learning Object Systems and other systems based on reusable components and performs a quantitative analysis to answer these questions. Section 3.4 presents a simple mathematical model that could explain the results.

1.5.3 Quality Control for the Labelling Process

The quality of metadata on learning objects stored in a LOR is an important issue for LOR operation [Barton et al., 2003] [Beall, 2005] and interoperability [Liu et al., 2001] [Stvilia et al., 2006]. Due to its importance, metadata quality assurance has always been an integral part of resource cataloging [Thomas, 1996]. Nonetheless, most LOR implementations have taken a relaxed approach to metadata quality assurance. As repositories grow and federate, quality issues become more apparent. The traditional solution for quality assurance, manually reviewing a statistically significant sample of metadata against a predefined set of quality parameters, similar to sampling techniques used for quality assurance of library cataloguing [Chapman and Massey, 2002], fails to scale to increasing amounts of learning objects being indexed manually or automatically. Some sort of automatical quality assurance mechanism should be created to cope with this problem.

The main research questions in the development of such automatic quality checker are:

1. How can the information contained in the metadata, the learning object itself and its context be transformed in quality metrics that can be processed by computers and understood by humans?
2. Does the metrics correlate with human evaluation?
3. Does the metrics discriminate between good and bad quality metadata?
4. Can the metrics be used to filter low quality records?

Chapter 4, based on current metadata quality frameworks, proposes and evaluates eleven metadata quality metrics. Section 4.5 proposes several applications of those metrics that can be considered first steps towards an automatic quality evaluation of learning object metadata. Section 6.3.1 is devoted to explain how those metrics could be implemented and deployed in real systems.

1.5.4 Relevance Ranking to Improve the Selection Process

In the early stages of the Learning Object Economy, LORs where isolated and only contained a small number of learning objects [Neven and Duval, 2002]. The search facility usually provided users with an electronic form where they could select the values for their desired learning object. The search engine then compared the values entered in the query with the values stored in the metadata of all objects and returned those which complied with those criteria. While initially this approach seems appropriate to find relevant learning objects, experience shows that it presents several problems, such as high cognitive load [Najjar et al., 2005], mismatch between indexers and searchers [Najjar et al., 2004], and low recall [Sokvitne, 2000]. Given these problems with the metadata based search, most repositories provided a "Simple Search" approach, based on the success of text based retrieval exemplified by Web Search engines [Chu and Rosenthal, 1996]. In this approach, users only need to express their information needs in the form of keywords or query terms. This approach seemed to solve the problems of metadata based search for small repositories. However, working with small, isolated repositories also meant that an important percentage of users did not find what they were looking for because no relevant object was present in the repository [Najjar et al., 2005]. If this technique is applied to large repositories, or to federated collections of repositories, the user is no longer able to review several pages of results in order to select the relevant objects. While doing a stricter filtering of results (increasing precision at expense of recall) could solve the oversupply problem, it could also lead again to the initial problem of scarcity. A proven solution for this problem is ranking or ordering the result list based on its relevance. In this way, it does not matter how long the list is, because the most relevant results will be at the top and the user can manually review them.

The main research questions to improve the selection process are:

1. What means relevance in the context of learning objects?
2. How existing ranking techniques can be used to produce metrics to rank learning objects?
3. How those metrics can be combined to produce a single ranking value?
4. Do the proposed metrics outperform simple text-based ranking?

Chapter 5 presents and evaluates the effectiveness of seven metrics based on the different dimensions of relevance. Chapter 6 also explains how these metrics can be implemented and integrated into current search engines and learning object tools.

1.6 Outline

This dissertation is divided in two parts. The first part deals with quantitative analysis of the publication (Chapter 2) and reuse (Chapter 3) of learning objects. These two chapters are observatory in nature, meaning that data is collected, analyzed and conclusions are obtained from the results of the analysis. These two chapters also present models to explain the process. The second part of the dissertation is experimental. Metadata Quality (Chapter 4) and Relevance Ranking (Chapter 5) metrics are defined, discussed and evaluated trough one or more experiments. Finally, Chapter 6 explains how those metrics can be implemented into a Service Oriented architecture and integrated into existing Learning Object systems. Four use cases are presented as an additional validation of the usefulness of the metrics. The dissertation concludes in Chapter 7 with a summary of contributions, discussion of the implication of the findings and many research questions that remain open for further research. The dissertation is accompanied by two Appendices describing the interfaces of the Metadata Quality and Relevance Ranking Metrics Services. Chapters 2, 3, 4 and 5 are based, in part or in whole, on material already published elsewhere. Among the most relevant papers on which these chapters are based are: [Ochoa and Duval, 2006c], [Ochoa and Duval, 2006b], [Ochoa and Duval, 2006a], [Ochoa and Duval, 2007b], [Ochoa and Duval, 2007a], [Ochoa and Duval, 2007a], [Ochoa and Duval, 2008a] and [Ochoa and Duval, 2008b].

Chapter 2

Quantitative Analysis of the Publication of Learning Objects

2.1 Introduction

Learning Object publication can be defined as the act of making a learning object available to a certain community. Strijker and Collis call this process "Offering" in their Learning Object Life cycle model [Collis and Strijker, 2004]. The publication process can take several forms. A professor can publish lectures notes for students in a Learning Management System (LMS). The same professor can decide to share objects with a broader community and publish them in a Learning Object Repository (LOR), such as ARIADNE [Duval et al., 2001] or Connexions [Baraniuk, 2007]. The University where this professor works can decide to start an Open Courseware (OCW) initiative [Malloy et al., 2002] and put the learning material of its courses freely available on the Web. Moreover, material already available online, can be discovered and re-published for other communities. For example, a student that found an interesting Web site to learn about basic Physics, could publish a link to that Web site on a Learning Object Referatory (LORF), such as Merlot [Malloy and Hanley, 2001] or SMETE [Agogino, 1999]. In all its different forms, Learning Object publication is the most important enabler of the Learning Object Economy [Campbell, 2003], because making the objects available is the first step to fuel the "share, reuse, improve and share again" philosophy behind this economy.

The publication of learning objects has been an important research issue since the definition of the field 15 years ago. These research efforts can be summarized

into three different research lines:

- Publishing Infrastructure: This line deals with the architecture of Learning Object repositories. These repositories are defined as computational systems containing some type of metadata database, a content store and a user interface for indexation and search [Richards et al., 2002]. Papers, such as [Duval et al., 2001] and [Richards et al., 2002], propose architectural decisions to store and make available learning objects in a scalable and distributed way. This line of research has produced a plethora of designs for learning object publication systems, ranging from centralized databases [Dong and Agogino, 2001] to peer to peer networks [Goth, 2005].

- Interoperability: The second line of research is focused on interoperability among heterogeneous repositories. The purpose of this research is to extend the availability of learning objects to larger communities than those to which it was originally published. The most important results of this line of research are the Learning Object Metadata (LOM)[1] standard [IEEE, 2002], a common metadata schema to describe learning objects, the Shareable Content Object Reference Model (SCORM)[2], a de-facto standard to create interoperable packages of learning objects, and the Simple Publishing Interface (SPI)[3] and Simple Query Interface (SQI)[4] [Ternier, 2008], standard protocols to insert and query learning objects from different tools into different repositories.

- Copyright and DRM: The third line of research is more oriented to legal aspects involved in the publication of learning materials such as copyright and digital right management (DRM). Papers, such as [Guth and Kppen, 2002], [Liu et al., 2003a] and [Liu et al., 2005], support the idea of including some type of copy protection system into learning objects at publishing time. More recent papers, such as [Downes, 2007], [Joyce, 2007] and [Duval et al., 2007], based on the success and popularity open access licenses like Creative Commons[5], push for open, but licensed, access to learning materials. Success stories, such as MIT OCW [Carson, 2004] and Connexions, and recent moves to Creative Commons licenses in ARIADNE and MERLOT seem to support the validity of open access.

One area of research that is practically unexplored is the study of the actual process and results of learning object publication. The research on technical and

[1]IEEE Learning Object Metadata Standard (LOM). http://ltsc.ieee.org/wg12/par1484-12-1.html

[2]Sharable Content Object Reference Model (SCORM). http://www.adlnet.org

[3]Simple Publishing Interface (SPI). http://ariadne.cs.kuleuven.be/lomi/index.php/SimplePublishingInterface

[4]Simple Query Interface (SQI). ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/WS-LT/CWA15454-00-2005-Nov.pdf

[5]Creative Commons Licenses. http://www.creativecommons.org

legal aspects lays the ground on which publication can take place. However, it does not provide any information about simple questions, such as how many learning objects are actually published, how they are distributed among different repositories or how repositories grow. Moreover, answers to these questions are not only relevant to measure the progress of the Learning Object Economy, but also to provide information on which decisions about architecture, interoperability strategies and planning for growth should be based. To our knowledge, the most prominent attempts to characterize learning object repositories and measure their characteristics are made by McGreal in [McGreal, 2007]. He provides a comprehensive survey of existing LORs and classifies them in various typologies. Unfortunately, his analysis is mostly qualitative and cannot be used to answer the questions mentioned above. Other relevant studies are [Neven and Duval, 2002] and [Sicilia et al., 2005], where different LORs are also qualitatively compared. In contrast with these earlier studies, this chapter will quantitatively analyze and compare different types of publication venues for learning objects . These types include Learning Object Repositories (LORP), Learning Object Referatories (LORF), Open Courseware Initiatives (OCW), Learning Management Systems (LMS). To provide some type of comparison and because their content can also be used for educational purposes, Institutional Repositories (IR) are also included in the studies. For simplicity, during this chapter, we will refer to all these systems as "repositories". The main goal of this chapter is to provide empirical answers to the following questions:

- What is the typical size of a repository? Is it related to its type?
- How many learning objects are typically used in a course?
- How do repositories grow over time?
- What is the typical number of contributors a repository has? Is it related to its type?
- How does the number of contributors grow over time?
- How many learning objects does a contributor publish on average?
- Is there a model that could explain the observed distributions?

To answer these questions data from different repositories is collected and analyzed. These answers will help us to gain insight into the actual process of learning object publication. Understanding how supply works in the Learning Object Economy will help the administrator of the repositories (Market-Makers) to design and plan the technological infrastructure needed to receive, store and share the published material. The methodologies used to find these answers can also help administrators to calculate the different characteristics of their particular repository and compare them against others. Policy-Makers can also use these answers

to evaluate which are the best approaches to encourage Contributors to publish their materials.

This chapter is structured as follows: Section 2 presents an analysis of the size distribution of different repositories. Section 3 analyzes the growth rate in objects, as well as contributors. Section 4 studies the distribution of contribution, publishing rate and engagement time. Section 5 presents a model to interpret the observed results. Section 6 answers the research questions and discusses their implications.

2.2 Size Analysis

In this section, we will analyze the size of different repositories. We define size as the number of objects present in the repository. In the first subsection, we compare the number of objects between repositories of the same type. In order to be able to compare the size between repository types, the second subsection decompose repositories that contains learning objects of large granularity (courses) into smaller objects.

2.2.1 Number of Objects

This analysis measures the size distribution of different types of repositories. We start with the study of 24 LORPs and 15 LORFs. These LORs were selected from the list compiled in [McGreal, 2007]. To avoid an unfair size comparison between repositories, only LORs that are not the result of the federation of other repositories, are publicly available and contain or link to learning objects of small and intermediate granularity (raw material or lessons) were analyzed. While McGreal already reported an estimate of the size, we measured each LOR through direct observation, on November 3rd - 4th 2007. Some inconsistencies, not due to natural growth, were found between the data reported by McGreal and the LORs size obtained from our observation. For example, Exploratorium Digital Library is reported in [McGreal, 2007] to have 100+ objects, while its web site presents 13,886 objects in the general search. These inconsistencies can be attributed to outdated information provided by the information section of the repositories.

The 24 LORPs have in total circa 100,000 learning objects, with an average size of circa 4,000 objects. A simple histogram of the data shows that the size distribution is not Normal, but highly skewed to the left. To analyze the distribution, we fit five known probabilistic distributions to the data: Lotka, Exponential, Log-Normal, Weibull and Yule. These distributions were selected because they have high skewness to the left and are commonly present in other Information Production Processes [Egghe and Rousseau, 2006]. The Maximum Likelihood Estimation (MLE) method [Goldstein et al., 2004] was used to obtain the distribution parameters. To find the best-fitting distribution, the Vuong test [Vuong, 1989] is

applied on the competing distributions. When the Vuong test is not statistically significant between two distributions, the distribution with less parameters is selected. This methodology is recommended by [Clauset et al., 2007] to select among heavy tail models instead of the more common Least-Squares Estimation and R^2 values used for Generalized Linear Models. In the specific case of LORP, the best fitting distribution is Exponential ($\lambda = 2.5\text{x}10^{-3}$). Figure 2.1 (left) presents the empirical (points) and fitted (line) Complementary Cumulative Distribution functions (CCDF) presented in logarithmic scales. In this graph, the X axis represent the number of objects present in the repository. The Y axis represent the inverse accumulated probability of the size ($P(X \geq x)$), that is, the probability that a repository has x or more objects. This skewed distribution of content size concentrates the majority of learning objects in few big repositories, while the rest of repositories contribute only a small percentage. Figure 2.1 (right) shows the Leimkuhler curve [Rousseau, 1988] [Burrell, 1992]. This curve is a representation of the concentration of object in the different repositories. The Y axis represent the cumulative proportion of objects published in the top x proportion of repositories. For example it is easy to see that the top 20% of the repositories (the biggest 5) contribute almost 70% of the total number of learning objects. The smaller 40% of the repositories combined contribute less than 3% of the objects.

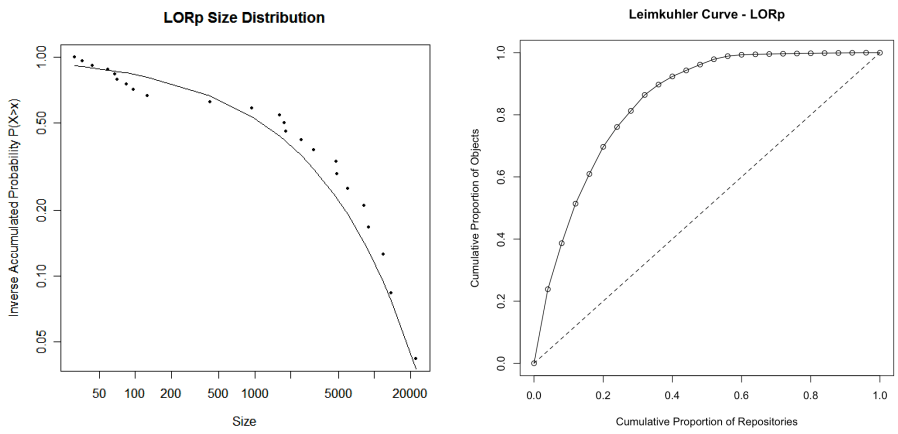


Figure 2.1: Distribution and Leimkuhler curve of the Size of Learning Object Repositories

The 15 studied LORFs offer in total circa 300,000 learning objects, with an average of circa 20,000 objects per referatory. The best fitting distribution is Exponential ($\lambda = 5.2\text{x}10^{-5}$). Figure 2.2 (left) presents the empirical (points) and fitted (line) CCDF. When the Leimkuhler curve is analyzed, the unequal

distribution is apparent. The biggest 20% (3 referatories) concentrate 66% of the 300,000 objects. The lower half contribute only 10% of the total (Figure 2.2 right).

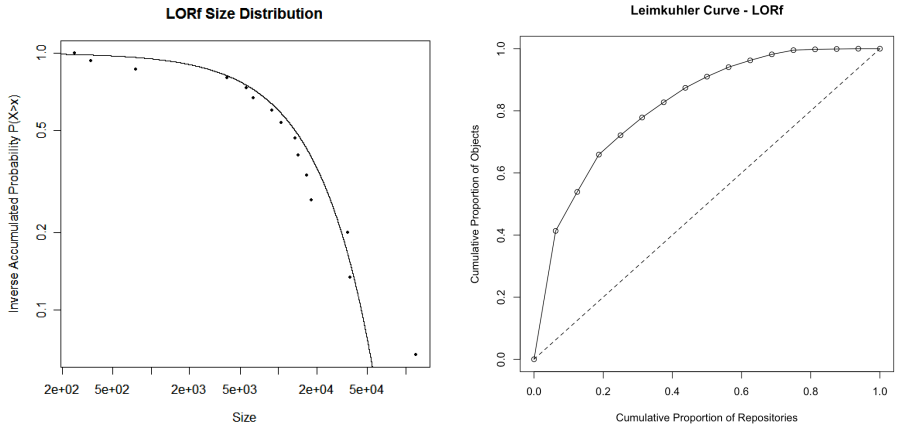


Figure 2.2: Distribution and Leimkuhler curve of the Size of Learning Object Referatories

To study the size of OCW initiatives, we collect a list of 34 Institutions providing their materials online from the Web site of the OCW Consortium[6]. The size was determined by the number of full courses offered online by each Institution. In total, 6,556 courses are available among all the studied sites, with an average of 193 courses per site. However, the average is a misleading value, because the size distribution is extremely skewed. The best-fitted distribution for OCW sites with more than 7 objects (tail) was Lotka with an $\alpha = 1.61$. The estimation of the start of the tail was performed minimizing the Kolmogorov-Smirnov statistic [Massey, 1951] for the Lotka law. The empirical and fitted CCDF can be seen in Figure 2.3 (left). This distribution leads to a very unequal concentration of courses. The Leimkuhler curve shows that the top 20% (7 sites) of the OCW sites offer almost the 90% of the courses, the remaining 27 sites just account for 10% (Figure 2.3 right). An analysis of the number of smaller granularity objects per OCW course is done in the following subsection.

Data about the size of LMSs is not normally available online. Most LMS implementations only allow registered users to have access to their contents. In order to obtain an estimation of the size of an LMS, we use some characteristics of Moodle [Cole and Foster, 2007], a popular Open Source LMS. Moodle allows guests to see the list of courses and only requires authentication to actually enter the course. Also, during deployment, there is the option to register the Moodle installation

[6]Open Courseware Consortium. http://www.ocwconsortium.org/

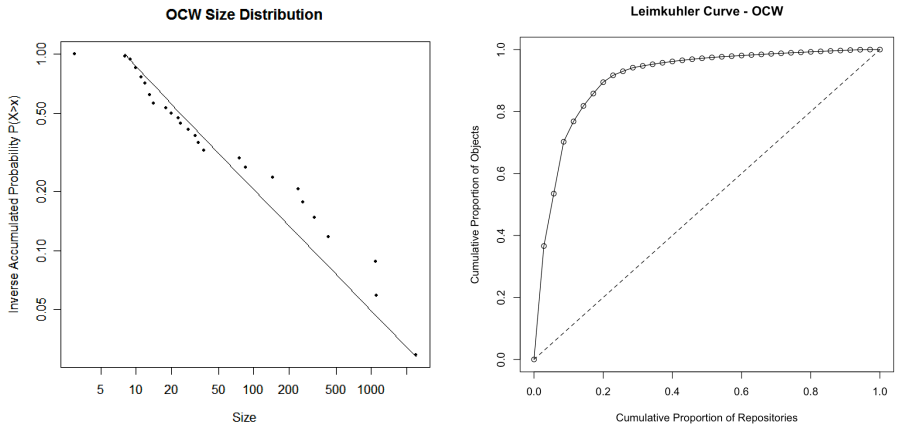


Figure 2.3: Distribution and Leimkuhler curve of the Size of Open Courseware Initiatives

with the Moodle.org site[7]. The link to registered installations is available on that site. We obtained a random sample of 2,500 from the circa 6,000 LMS sites listed on the Moodle site as installations in the United States. This country was selected because it had the largest number of installations. Through Web scraping, we downloaded the list of courses for each one of those installations. In those 2,500 Moodle sites, 167,555 courses are offered, with an average of 67 courses per site. The distribution that best fits the tail of the data (sites bigger than 70 courses) is Lotka with an estimated α of 1.95. The estimation of the start of the tail was determined through minimization of the Kolmogorov-Smirnoff. The empirical and fitted CCDF can be seen in Figure 2.4 (left). Again, this distribution concentrates most of the courses in just a few LMSs. The Leimkuhler curve shows, for example, that the top 20% LMSs (500 sites) offer more than 85% of the courses (Figure 2.4 right). An analysis of the number of smaller granularity objects per LMS course is done in the following subsection.

Finally, to establish the size distribution of IRs, we collect the list of repositories listed at the Registry of Open Access Repositories (ROAR)[8]. An automated service connected to this registry regularly harvests OAI-PMH enabled [Van de Sompel et al., 2004] IRs and provides information about their sizes. During data collection, 772 repositories with more than one object were measured. The total number of documents stored in those repositories was 7,581,175. There were, in average, 9,820 documents per repository. The tail (repositories with more than 3,304 documents)

[7]Moodle Web site. http://www.moodle.org
[8]Registry of Open Access Repositories. http://roar.eprints.org/

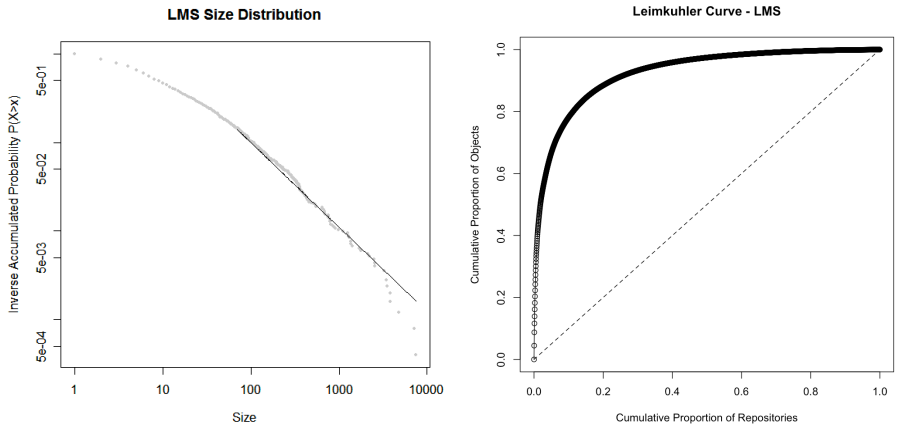


Figure 2.4: Distribution and Leimkuhler curve of the Size of Learning Management Systems

of the distribution was fitted by Lotka with an estimated α of 1.73. Figure 2.5 (left) presents the empirical and fitted CCDF. The highly skewed concentration of documents can be seen in the Leimkuhler curve in Figure 2.5 (right). For IRs, 20% (155) of the repositories concentrate circa 90% of the documents.

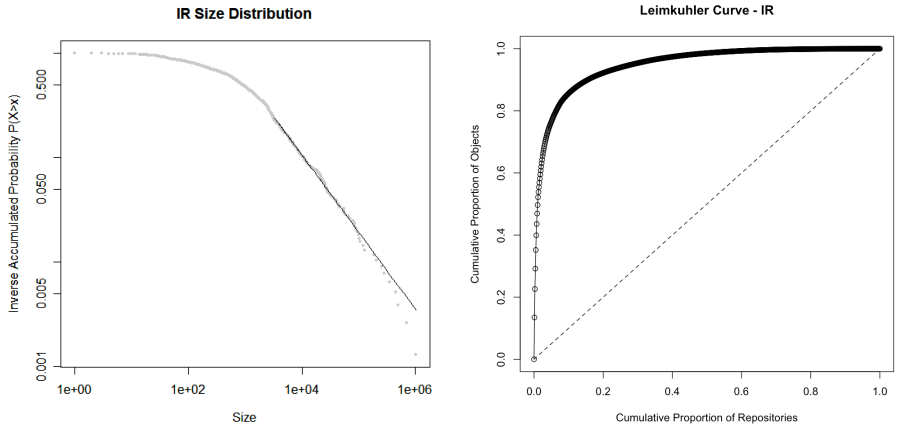


Figure 2.5: Distribution and Leimkuhler curve of the Size of Institutional Repositories

Table 2.1 presents the summary of the findings about the size of the different

types of repositories. From the Average Size column, the first conclusion that can be extracted is that the size of a repository is directly related to its type. The most interesting difference can be found between Learning Object Repositories and Referatories. LORFs are almost an order of magnitude bigger than LORPs. This difference can be explained by the level of ownership required to contribute to these repositories. To contribute to a Referatory, the user only needs to know the address of the learning resource on the web. Any user can publish any online learning object because its publication does not require permission from the owner of the material. On the other hand, publishing material in a Repository requires, if not being the author of the object, at least to have a copy of it. It can be considered non-ethical, or even illegal, to publish a copy of the object without having its ownership or at least the explicit consent of its owner. It will be safe to assume that, in the general case, the number of online learning objects that interest a user is larger than the amount of learning objects being authored by herself. Another type of repositories, comparable by its granularity to LORs, are the IRs. The average size, around 10,000, seems to be a mid-way between the LORF and the LORPs. However, because its power-law distribution, IRs could normally be at least 2 orders of magnitude bigger and 4 order of magnitude smaller than the average. In conclusion, IRs present a larger "range", with some IRs ten times bigger than the biggest LORF and others ten times smaller than the smallest LORP. In the large granularity group, OCWs and LMS, the difference is less significant. This similitude can be explained because OCWs are not more than the content of LMS published and made available. The largest and smallest OCWs and LMSs have also similar amount of courses.

If we consider the distribution of the sizes among different types of repositories (Table 2.1, forth and fifth columns) it is clear that the size of OCWs, LMSs and IRs is distributed according to a power law with an exponent between 1.5 and 2. This distribution, as mentioned above, produces a wide range of sizes. The variance of the Lotka distribution for those values of *alpha* is infinite, meaning that it is possible, at least theoretically, that extremely large or small repositories exists. Also, this distribution presents a heavy-tail with few big repositories and a lot of smaller ones. It is surprising that independently of the type of repository, the *alpha* parameters are similar. On the other hand, LORPs and LORFs present an exponential distribution. However, we argue that, in reality, they also follow a Lotka distribution, but the finding of the exponential is an artifact of the sampling method. In the case of OCWs, LMSs, and IRs, the considered repositories were sampled from lists that are not biased to consider only small or large repositories. Any repository, regardless of size, can publish itself in the sampled lists. In the case of LORPs and LORFs, there are no compilation lists available, and the considered repositories are only those that are known, biasing the sample against the expected large amount of small and relatively unknown repositories. An example of this sampling artifact can be seen in [Carr and Brody, 2007]. There, only the top IRs

from ROAR are considered and the size distribution that can be inferred from the presented graphs is distinctively exponential. If all the IRs of ROAR are considered, we found that the real distribution is actually a power law.

The final conclusion that can be extracted from the size analysis is that the distribution of learning objects is very unequal. Most of the resources, independently of the type of repository, are stored in just a few repositories. The concentration is a consequence of the power law distribution. The Pareto or 20/80 rule (also used to describe other heavy-tailed distributions, such as wealth [Wold and Whittle, 1957]) seems to be a good guide to look at this inequality. The lower value observed in the LORPs and LORFs can also be attributed to the bias toward bigger repositories in the sampling. A more detailed search for LORPs and LORFs will most certainly find small repositories. Adding these small repositories will increase the concentration at the top 20%. Despite this inequality distribution, however, no single repository of any type contains more than 40% of the available resources. The remaining long-tail [Anderson, 2006] with the 60% of resources is located in other repositories. This can be seen as a strong empirical corroboration of the need to interconnect repositories, either through query federation [Simon et al., 2005] or metadata harvesting [Van de Sompel et al., 2004].

Table 2.1: Summary of Number of Objects Analysis

Repository	Sampled	Average Size	Distribution	Parameters	Conc. (20%)
LORp	24	3,905	Exponential	l=2.5x10-3	70%
LORf	15	19,396	Exponential	l=5.2x10-5	66%
OCW	34	193	Lotka	a=1.61, xmin=8	90%
LMS	2,500	67	Lotka	a=1.95, xmin=70	85%
IR	775	9,820	Lotka	a=1.73, xmin=3304	90%

2.2.2 Objects per Course

To gain insight into the real size of course-based repositories, such as OCW and LMS, we sample four representative repositories that can be decomposed into their smaller components:

1. The first selected repository was the MIT OCW[9]. Together with their content, OCW MIT provides metadata about the whole course and its components. We harvested these metadata from their Web site. From OCW MIT, 1,796 courses were retrieved. When decomposed, MIT OCW courses generated a total of 42,527 learning objects.

[9]MIT OCW. http://ocw.mit.edu

Table 2.2: Course Size Distribution

Repository	Courses	Objects	Average	shape	scale
MIT OCW	1,796	42,527	24	1.03	26.5
OpenLearn OCW	405	10,644	27	0.71	18.1
Connexions LORP	268	5,242	20	0.78	15.4
SIDWeb LMS	1,445	23,370	16	0.65	9.46

2. The second repository was Open Learn[10] [McAndrew, 2006], an OCW intiatitive of the Open University in United Kingdom. OpenLearn provides an option to download an XML description of the course. We use those descriptions to extract the number of smaller learning objects included into the course. From OpenLearn, 405 courses were obtained, resulting in a total of 10,644 learning objects.

3. The third repository was Connexions[11]. This repository groups learning objects of intermediate granularity (modules) into higher granularity objects (collections). A collection could be considered as a course. From Connexions, we obtained, through Web scraping, 268 collections that include, in total, 5,242 modules.

4. The final selected repository was SIDWeb[12], a LMS of ESPOL, a polytechnic University at Ecuador. The author, being in charge of the development of this LMS, has complete access to its contents. The number of objects attached by ESPOL professors to each course was extracted from the SIDWeb database. In total, 1,445 courses were available. Those courses produced a total amount of 23,370 learning objects.

It is interesting to note that the histogram of the number of learning objects per course is not symmetrical, and therefore, not Normal. It presents a large skewness to the left. The distribution of learning objects per course for the four selected repositories was fitted with the same five distributions used in the previous analysis. Weibull distribution seems to be the best fit for all the data sets. Table 2.2 summarizes the fitted parameters as well as other characteristics of the distributions. As a visual aid, Figure 2.6 presents the empirical and fitted Probability Density function (PDF) of the course size distribution for the four repositories. The more intuitive PDF, similar in shape to an histogram, can be used for graphical representation thanks to the large amount of data.

From Table 2.2, it can be concluded that if the courses presented in an LMS or OCW are decomposed into smaller learning objects, they can be considered

[10]Open University Open Learn. http://openlearn.open.ac.uk
[11]Connexions. http://www.cnx.org
[12]SIDWeb LMS. http://www.sidweb.espol.edu.ec

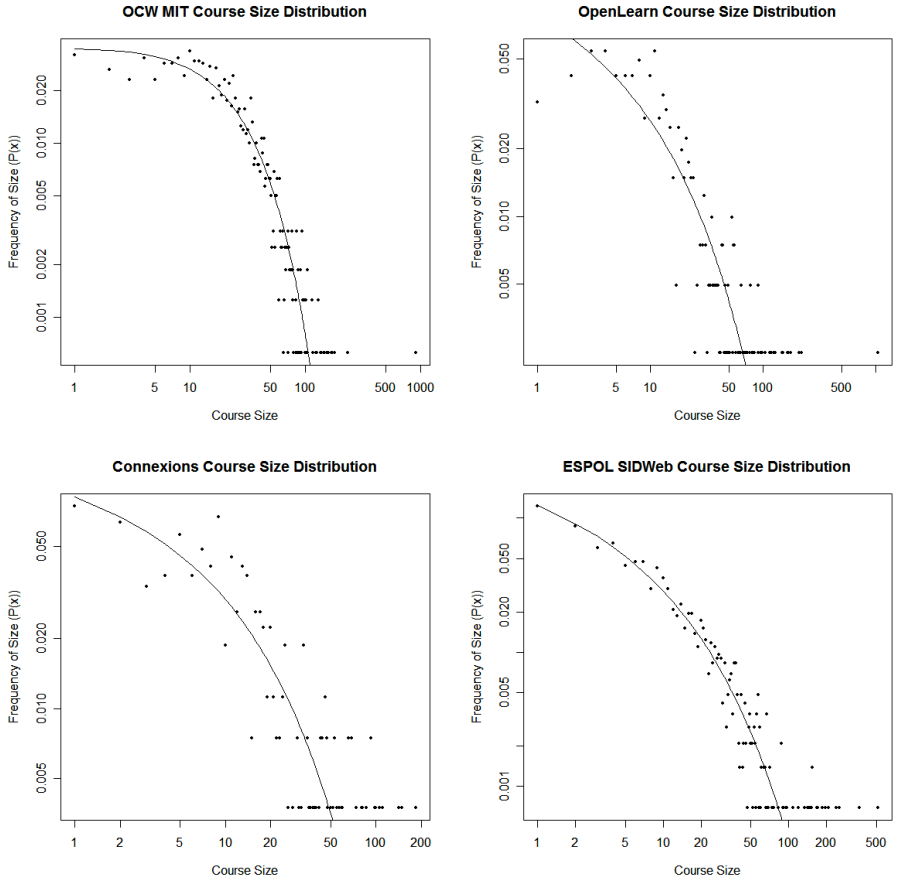


Figure 2.6: Empirical and Fitted Course Size Distributions

large LORPs. For example, MIT OCW contains two times more objects than the largest LORP studied in this chapter. Even not exceptionally big LMSs, such as SIDWeb, can be transformed in LORPs of considerable size.

Contrary to our initial belief, the distribution of the number of learning objects per course is not Normal. That means that a considerable amount of the courses are published with few learning objects. Courses with very few objects, in the case of SIDWeb and Connexions, can be attributable to abandoned or incomplete courses. However, on institutional sites, such as MIT OCW and OpenLearn, with a quality control before publication [JOCW, 2006], it is clear that the course was intentionally published with few objects. This can be an indication that, in a

considerable proportion of cases, the learning materials used in a course has not being digitalized, and therefore, they are not publishable on-line. However, the alternative explanation, that those courses actually use very few objects, could only be discarded with deeper research.

The Weibull distribution, nonetheless, is less unequal for small values than Lotka and Log-Normal. For example, the plot of the distribution in MIT OCW shows that the number of courses with 1 to 20 objects is almost the same. The same is true for OpenLearn. In the case of SIDWeb and Connexions, the higher the number of objects, the lower the probability to find such a course, but the decrease is much less steep than in Lotka distributions.

The Weibull parameters, specifically the shape, are very similar among sites, leading to similar distributions. An interesting consequence is that the average number of objects per course for all the sites is around 20. Also the extreme values fall into the same order. The biggest courses have between 200 and 900 objects. While we only analyze 4 sites, their diversity helps us to conclude that this value could be used as a rule-of-thumb to quantify courses in general.

The final conclusion of this analysis is that OCWs and LMSs (if openly accessible) can be considered sizeable sources of learning objects, comparable with big LORPs or average LORFs. Moreover, given its relative abundance in comparison with other types of repositories [Harrington et al., 2004], it can be easily concluded that, currently, OCWs and LMSs are the repositories where the majority of learning objects are stored. Tapping this vast resource could provide the needed boost to bootstrap the global Learning Object Economy.

2.3 Growth Analysis

In order to understand how repositories grow over time, this section analyzes several repositories of different types. The repository growth will be considered in two dimensions: growth in number of objects and growth in the number of contributors. The following subsections will present the analysis for each one of these dimensions.

2.3.1 Content Growth

To measure the growth in the number of objects, 15 repositories of different type were studied. They were selected based on how representative they are for their respective type in terms of size and period of existence. The availability of the object publication date was also a determinant factor. The selected repositories are:

- LORPs: ARIADNE[13], Maricopa Learning Exchange[14] and Connexions.
- LORFs: INTUTE[15], MERLOT[16] and FerlFirst[17].
- OCWs: MIT OCW and OpenLearn.
- LMSs: SIDWeb
- IRs-Large: PubMed[18], Research Papers in Economics[19] and National Institute of Informatics[20].
- IRs-University: Queensland[21], MIT[22] and Georgia Tech[23].

The collection of data for all the LORs, except INTUTE, consisted in obtaining the date of publication of all their objects. In the case of INTUTE, a sample with all the objects containing the word "Science" (approximately 10% of the repository) was obtained. The data for LORPs and LORFs were collected through Web scraping of the sites during the period between the 5th and the 8th of November 2007. In the case of OCWs and LMSs, the data of publication of all the courses were obtained through direct download. Finally, the selection criteria for the first three IRs (IRs-Large) was size, time of existence and current activity. These three factors were evaluated from the data provided by ROAR. The second three (IRs-University) were selected from University repositories of intermediate size with at least 3 years of existence. The monthly size of these repositories was obtained from data provided by ROAR.

The first variable analyzed was the average growth rate (AGR), measured in objects inserted per day. This value is obtained by dividing the number of objects in the repository by the time difference between the first and last publications. Results for this calculation can be seen in the fourth column (AGR) of Table 2.3. It is interesting to compare the AGR of different types of repositories. LORPs, for example, grow with a rate of 1 or 2 objects per day. OCWs and LMSs grow similarly with an unexpectedly high value of circa 1 course published per day. From the previous analysis on course size, that rate can be translated, on average, in 20 objects per day. In LORFs and IRs the variability is significantly higher. For example, big IRs grow more than 10 times faster than University IRs. This

[13]ARIADNE Foundation. http://www.ariadne-eu.org
[14]Maricopa Learning Exchange. http://www.mcli.dist.maricopa.edu/mlx
[15]INTUTE. http://www.intute.ac.uk
[16]MERLOT. http://www.merlot.org
[17]FerlFirst. http://ferl.becta.org.uk (decommissioned)
[18]PubMed repository. http://www.ncbi.nlm.nih.gov/pubmed/
[19]Research Papers in Economics repository. http://www.repec.org
[20]National Institute of Informatics repository. http://www.nii.ac.jp
[21]Repository of U. Queensland. http://espace.library.uq.edu.au/
[22]Repository of MIT. http://dspace.mit.edu/
[23]Repository of Georgia Tech. http://smartech.gatech.edu/dspace/

difference can be explained by the fact that big IRs are open to a wider base of contributors. On the other hand, the contributor base of University IRs is often restricted to researchers and students of that specific University. The difference between LORFs, however, could not be attributed to size of the contributor community, but to their dedication. INTUTE is a project that pays expert catalogers to find and index learning material on the web. Merlot and FerlFirst, however, rely on voluntary contributions from external users. A group of paid workers are expected to have a higher production rate than a group of volunteers of a similar size.

The AGR describes linear growth. To test the actual growth function six models were fitted against the data: linear ($at+b$), bi-phase linear with breakpoint (a_1t for $t < Breakpoint$ and $a_2t + b_2$ for $t \geq Breakpoint$), bi-phase linear with smooth transition ($ln(a*exp(bx)+c)$, exponential ($b*e^{at}$), logarithmic ($b*ln(at)$) and potential ($b*t^a$). These models were selected based on visual inspection of the size vs. time plot (Figure 2.7). We use Generalized Linear Model fitting with Least-Squares Estimation. The selection of the model was based on the Akaike information criterion (AIC) [Akaike, 1976], that not only takes into account the estimation power of the model, but also its simplicity (less estimated parameters). The result of the fitting indicates that most data sets were best explained by the linear bi-phase model (both the breakpoint and smooth versions). In PubMed, Connexions and OCW the growth is best explained by the potential function, but bi-phase linear is the second best. A visual inspection of the plots (Figure 2.7) shows indeed that, in most data sets, two regimes of linear growth can be easily identified, sometimes with a clear transition point (BP). This result suggests that growth is mainly linear, but the rate is not constant. Two different growth rates are identified in all the repositories. There is an initial growth rate (IGR) that is maintained until a "Breakpoint" (BP) is reached and then, a mature growth rate (MGR) starts. Table 2.3 reports the growth rates and breakpoint values for all the studied repositories.

In most cases, the change between IGR and MGR is positive, meaning that the rate increases with maturity. The most logical explanation is that at some point in time the repository reaches a critical mass of popularity and the contributor base starts to grow faster, and therefore, the total production rate increases. This hypothesis is tested in the following subsection when the contributor base growth is studied. However, in two LORs, the production rate decreases from IGR to MGR (Ariadne and FerlFirst). Having inside knowledge of Ariadne history, the inflection point represents the moment when the focus from the Ariadne community shifted from evangelization to attract new members towards interconnection with other repositories through the GLOBE consortium[24], decreasing the number of active submissions to the core repository. As such, Ariadne is moving from primarily being a repository to primarily being an integrator of repositories. For FerlFirst,

[24]Globe Consortium. http://www.globe-info.org

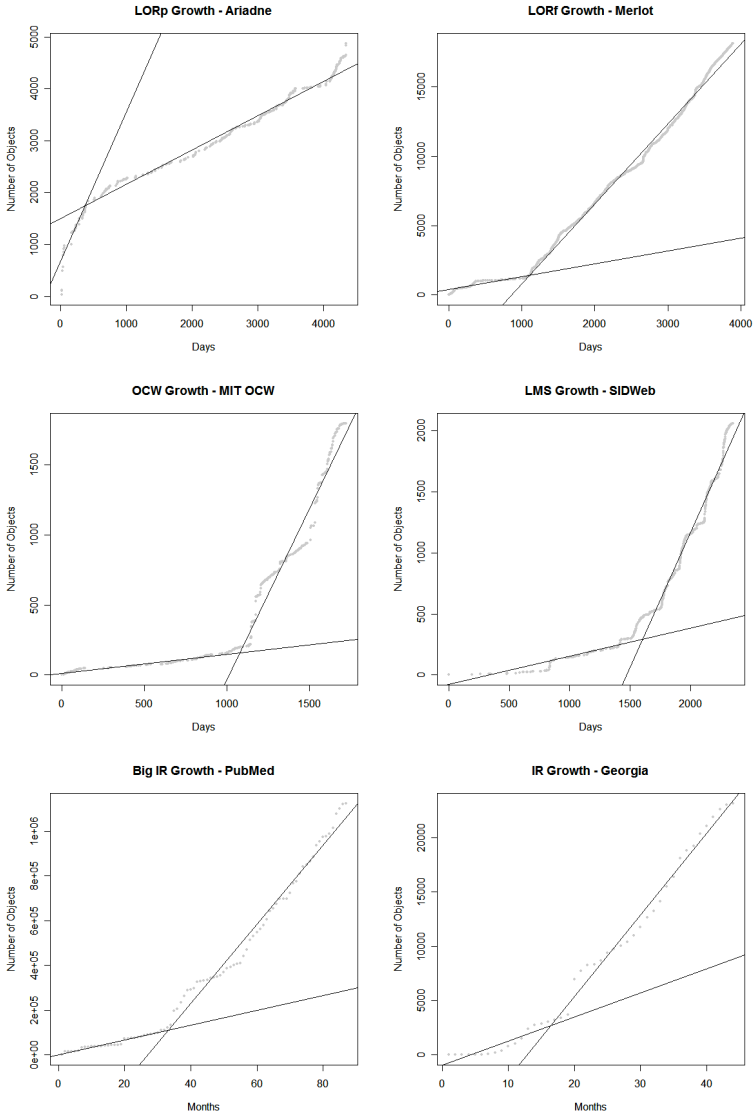


Figure 2.7: Empirical and Fitted Size Growth

Table 2.3: Results of the Growth Analysis of the Repositories

Repository	**Objects**	**Age** (y)	**AGR** (o/d)	**IGR** (o/d)	**MGR** (o/d)	**BP** (y)
LORP						
Ariadne	4,875	12.0	1.1	2.9	0.66	1.0
Connexions	5,134	7.9	1.8	0.8	2.19	2.8
Maricopa	2,221	4.2	1.4	0.9	2.32	3.0
LORF						
Intute	120,278	12.5	26.7	5.8	36	4.5
Merlot	18,110	10.8	4.6	0.9	5.8	2.8
Ferl First	3,938	6.3	1.7	5.0	1.1	1.0
OCW						
MIT OCW	1,796	4.9	1.0	0.1	2.44	2.9
OpenLearn	499	1.8	0.7	0.1	4.12	1.5
LMS						
SIDWeb	1,445	5.7	0.6	0.2	2.21	4.6
IR						
PubMed	1,124,197	7.3	431	111	591	2.8
RePEc	514,636	4.9	306	65	90	3.3
NII	179,153	5.7	88	42	151	3.6
Queensland	12,069	5.3	6.4	2.4	14	2.7
MIT	27,416	3.8	17.5	32	11	1.6
Georgia Tech	23,163	3.7	20.3	7.4	25	1.4

on the other hand, the decline is explained by the abandoning of the project. At the time of writing, this repository has been decommissioned and absorbed by another project, Excellence Gateway[25]. These two can be considered special cases, where the norm is an increase at maturity.

Another interesting finding is that BP is, in most cases, located between two to three years after the first object has been inserted into the repository. In the case of LORs, this can be seen as the time needed by the repository to reach a critical mass of objects that could attract more users or funding, and therefore, more objects. In the case of LMSs, OCWs and University IRs, this can be the time taken to "cross the chasm" [Moore, 2002] between early adopters and mainstream use inside the institution.

It is important to note that the linear trend is observed at large time scales. The short-term growth, specially for OCWs, LMS and IR is characterized by irregular "jumps". These jumps can be explained by external events such as the start of the academic semesters or the deadline for annual reviews. If we smooth these jumps

[25]QIA Excellence Gateway. http://excellence.qia.org.uk/

over a long period of time, however, the linear growth is apparent.

The main conclusion of this analysis is that linear growth in the number of objects is a sign of the lack of penetration of Learning Object technologies in educational settings. It would be expected that the amount of learning material created follows an exponential growth [Lyman and Varian, 2000]. It seems that much of this material is not published in any type of repository.

2.3.2 Contributor Base Growth

Another way to consider repository growth is to measure its contributor base at different points in time. For this analysis, we try to use the same set of data as in the previous experiment, with some exceptions. Intute, FerlFirst (LORFs) and OpenLearn (OCW) were excluded because they do not provide contributor data for their objects. To obtain the data for the IRs, the complete metadata set was harvested from the repositories. For this reason, the three biggest IR (PubMed, RePEc and National Institute of Informatics) were excluded. It was not feasible to process the large amount of information using a single computer in a reasonable amount of time. Nonetheless, the remaining repositories are representative of their respective type.

In the case that one object have several authors, only the first author is considered its contributor. For example, if the learning object metadata mentions three authors, the act of publication is only attributed to the first author. Counting the first author instead of assigned fractional counts to all the authors is a common practice in Scientometrics [Lindsey, 1980].

As can be seen in the second column of Table 2.4, the size of the current contributor base of the studied repositories (with the exception of IRs) is within one order of magnitude. The smaller contributor base is ARIADNE (166) and the largest is Merlot (1,446). It is important to note that the difference in numbers of objects of LORP and LORF cannot be explained just by the size of the contributor base. As it was hypothesized in the previous section, the difference should originate in a different rate of contributors. In the case of IRs, the user base is considerably bigger than in other types of repositories. The reason for this difference can be found in the fact that contributing to Institutional Repository is in most cases mandatory for post-graduate students [Davis and Connolly, 2007]. Publishing their thesis in the Institutional Repository is commonly a requirement. Contrarily, contribution to the other type of repositories is optional and, in the case of OCWs and LMSs, it is normally reserved for professors only.

The analysis of the average growth rate (AGR), in the third column of Table 2.4, presents an even more coherent picture. The number of authors in LORs seem to incorporate a new contributor each 2 to 5 days. In MIT OCW, with its accelerated publishing program, a new professor is added every day in average. SIDWeb, have a rate similar to LORs with a new professor using the system every 5 days, also on average. As expected from the previous analysis, IRs have a much

Table 2.4: Result of the Analysis of the Contributor Growth in the Repositories

Repository	**Contrib.**	**AGR** (c/d)	**Function**	**Growth Rate**	**BP** (y)
LORP					
Ariadne	166	0,04	Bi-Phase L.	IGR=0.02, MGR=0.06	3.5
Connexions	581	0,22	Exponential	ER=1.2 x 10-3	3.0
Maricopa	529	0,20	Bi-Phase L.	IGR=0.06, MGR=0,28	2.4
LORF					
Merlot	1,446	0.42	Bi-Phase L.	IGR=0.12, MGR=0.54	1.1
OCW					
MIT OCW	1,072	1.34	Exponential	ER= 3.7 x 10-3	2.3
LMS					
SIDWeb	584	0.21	Exponential	ER=1.8 x 10-3	3.3
IR					
Queensland	9,503	3.9	Bi-phase L.	IGR=1.6, MGR=6.9	3.8
MIT	21,028	9.7	Bi-phase L.	IGR=0.7, MGR=17	2.2
Georgia Tech	10,704	7.9	Bi-phase L.	IGR=4.4, MGR=9.7	1.7

higher AGR, counting between 4 and 10 new contributors each day. Based on these numbers, it would be an interesting experiment to open LMSs and OCWs also to under- and post-graduate student contributions.

The actual growth function was determined with the same candidate functions and fitting procedure used in the previous subsection. The results can be seen in Table 2.4, fourth and fifth columns. While the majority of repositories present a bi-linear growth, similar to the growth in the number of objects, it was surprising to find that, in the case of MIT OCW, SIDWeb and Connexions, the contributor base grows exponentially. This effect can be better visualized in Figure 2.8. Section 5 will present a model that could explain how an exponentially growing user base could generate a linear object growth.

The change from IGR to MGR is positive in all the studied repositories. In most cases, the MGR is between 2 to 4 times larger than the IGR. A notable exception is MIT IR. In that particular case, the rate increases more than 10 times, even taking in account artifacts in the MGR estimation. In the case of exponential growth (Connexions, MIT OCW and SIDWeb) the rate at which new users enter the system is always increasing. That growth is captured in the exponential rate parameter. This parameter is similar for the three repositories, with MIT OCW also presenting the most rapid growth. However, exponential growth cannot continue forever, especially in the case of MIT OCW and SIDWeb. Once most of the professors have created a course in those systems, the contributor growth rate will follow the incorporation of new faculty members, which is commonly linear over

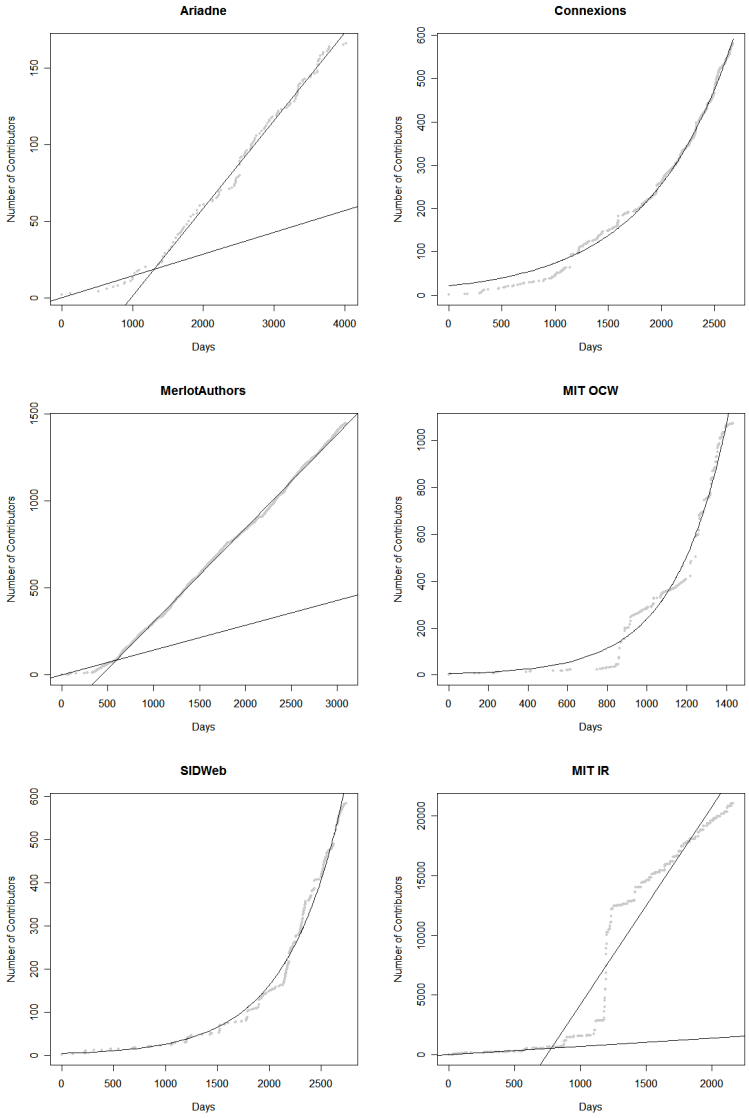


Figure 2.8: Empirical and Fitted Contributor Base Growth

large periods of time.

An interesting comparative analysis can be made between the breakpoints of the object and contributor base growth. This analysis can shed light on the

"chicken or egg" dilemma regarding repositories. This dilemma can be summarized as: does an increase in the number of objects attract more users? or does the increase in the number of contributors generate more objects? The breakpoint, in the case of Bi-phase linear growth is the point of transition between the two linear phases. In the case of exponential growth, we selected the point where the function grows faster than linear. Comparing the BP values in Tables 2.3 and 2.4. The evidence is inconclusive. Some repositories, such as Maricopa, MERLOT and MIT OCW first have an increase in the rate at which new contributors arrive and eight months later, on average, an increase in rate of growth is perceived. However, in the IRs, SIDWeb and Connexions, the contrary is true. First, the rate object growth increases and four months to one year afterward, the rate of new contributors follows. It seems that to gain insight on how the 'chicken or egg" dilemma is solved in the repositories, a deeper analysis is needed.

A final conclusion for this analysis brings hope for the establishment of Learning Object Technologies. Spotting exponential growth in the number of contributors in three repositories is maybe signaling that a transition phase between linear and exponential growth is happening. However, this change will depend on the ability of the repositories to retain the productive users. This will be ultimately defined by the engagement and fidelity that the repository could produce in their contributors.

2.4 Contribution Analysis

In this section, we analyze in more detail how contributors publish objects in the repositories. The first study will analyze how many objects are published by each contributor. The second subsection analyzes how frequently objects contributors publish objects. The third and last subsection examines the amount of time that each contributor keeps contributing objects. These analyzes will help us to gain an insight on the inner workings of the learning object publication process.

2.4.1 Contribution Distribution

To understand contributor behavior, full publication data from three LORPs (Ariadne, Connexions and Maricopa), one LORF (Merlot), one OCW site (MIT OCW), one LMS (SIDWeb) and three IRs (Queensland, MIT and Georgia Tech) was obtained. Each learning object was assigned according to the data to one contributor. If more than one contributor was listed, we counted the first author only.

The first step in this analysis is to obtain the average number of publications per contributor (AC). This value was obtained dividing the total number of objects in the repository (Table 2.3) by the number of contributors (Table 2.8) . Table 2.5 presents this value in the second column. It is interesting to note that the

average output of contributors to different kind of repositories differ substantially. The contributors to the ARIADNE repository, while few, have produced during the lifetime of the repository, more objects per capita than any of the other LORs. The results for Merlot also confirm that the bigger size of LORF is not due to a bigger, but to a more productive contributor base. The results for MIT OCW and SIDWeb show that they have a similar productivity per contributor, hinting that the publishing mechanics in LMSs and OCWs are very similar. For IRs, the only way to explain the low productivity, in what it is a scientific publication outlet, is to assume that the majority of the content is made by student thesis.

The next step in the analysis is to obtain an approximation for the distribution of the number of publication for each author. Given that the data is highly skewed to the left, the best way to present it is the size-frequency graph in logarithmic scales (Figure 2.9). This figure represents how probable (y axis) is to find a contributor that have published a certain amount of objects (x axis). Five statistical distributions were fitted against the data: Lotka, Lotka with exponential cut-off, Exponential, Log-Normal and Weibull. The parameter estimation was made with the MLE method and the Vuong test was used to find the best fitting of the competing distributions. The best-fitting distribution and their estimated parameters can be seen in the third and fourth columns of Table 2.5.

From the result of the distribution fitting, it is clear that the number of objects published per each contributor varies according to the type of repository. All LORs follow a Lotka distribution with exponential cut-off. The meaning of this cut-off is that it becomes increasingly harder to publish a large amount of objects. At low levels of output, it does not affect the common Lotka distribution, but at high levels, it reduces the probability of publishing. The effect can be seen in Figure 2.9 as a slight concavity at the tail of the distribution. The parameters for the different LORs are similar. Even the cut-off rate seems to agree with previous conclusions. For LORP the cut-off starts sooner by an order of magnitude than for LORF. The finding of this distributions means that most LOR contributors only publish one object. Even high producing individual start loosing interest after publishing many objects. Maybe one of the reasons behind this distribution is the lack of some type of incentive mechanism [Campbell, 2003].

OCW MIT and SIDWeb present a Weibull distribution. This is not surprising given that it was already found that the number of objects per course also follows that distribution and usually professors are usually in charge of one or two courses. The values of the Weibull shape parameter found for the distribution of course size (Table 2.2) are identical to the ones found in the distribution of objects per contributor (Table 2.5). On the other hand, the scale parameter is around 1.5 times bigger for contributors. This is also explained by the fact that some professors are responsible for more than one course. The finding of a weibull distribution means that for OCWs and LMSs there is an increased probability to produce a certain amount of objects. This can be seen as the strong concavity in the curve compared

with the flat Lotka. Notice that the peak of this concavity is located around the 1.5 times the average number of objects per course (30 objects). The mechanism behind this distribution is that there is an interest to produce courses with a given amount of learning objects (maybe 1 object per session).

The tail of the IRs are fitted by the pure Lotka distribution. The head of the distribution, users that have published 1 or 2 objects, have a disproportionately high value that cannot be fit by any of the tried distributions. The tails, however, have an α of around 2.50 that is consistent with previous studies [Coile, 1977] [Pao, 1986] of the distribution of scientific publications among authors. This result suggests that the publication of documents in IRs have a different mechanism than the publication of learning objects in LORs, and maybe what we are measuring in the IRs tail is a by-product of the scientific publication process.

Finally, the percentage of objects created by 20% of the users is calculated. The results are presented in the last column of Table 2.5 (C-20). From these results, it can be concluded that the LORs are affected by the Pareto inequality (20/80 rule). The concentration for OCW MIT is less unequal, with just 50% of the objects being published by the top 20% contributors. The explanation for this result is that there is a considerable proportion of users that produce between 10 and 50 objects. This group of contributors is productive enough to balance the production of the tail. In the case of IRs, an interesting effect, produced by the thesis publication can be observed. The most productive section of the contributors is located at the head of the distribution. This effect is more visible in the Queensland repository. There 20% of the most productive contributors publish around also the 20% of the material. The only way to reach this percentage is that almost all the contributors publish the same amount of documents, more concretely for this repository, one document. Georgia Tech, while also have a large proportion of one-time contributors, have a heavy-tail that can balance it out.

As can be concluded from the previous analysis, not all contributors are equal. In any population, regardless if it is Weibull or Lotka, there will always be several "classes" of users, similarly to the segmentation use to classify socioeconomic strata (As mentioned before, income is also heavy-tailed distributed). We can divide the contributing population in a large "lower class" of contributors that only publish few objects. A smaller "middle class" that publishes intermediate amount of objects and a very small "higher class" that publishes a large amount of objects. These classes arise naturally and have to be dealt with. While the publishing capacity can be increased with better tools and intuitive environments, the inherent inequality will, most probably, persist.

2.4.2 Lifetime and Publishing Rate

In this analysis, we consider the publication history of each individual contributor. Two variables will be measured: First, the lifetime of the contributor. This is the time from its first to its last publication. Second, the publishing rate. This is

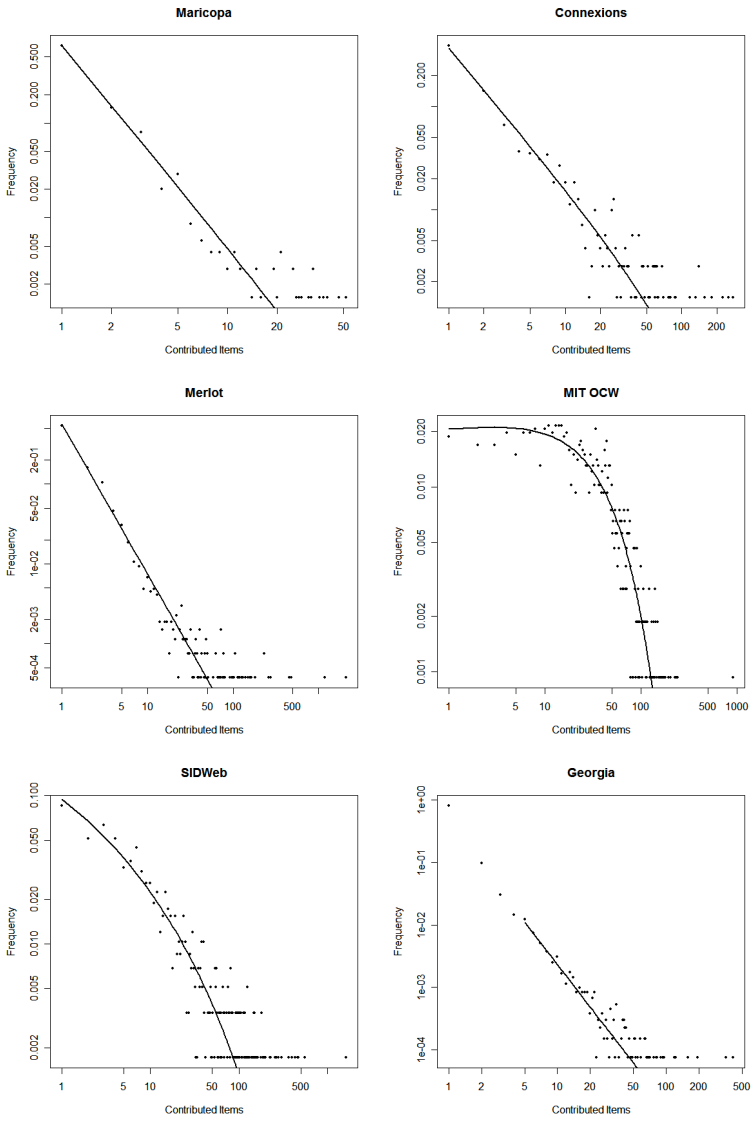


Figure 2.9: Empirical and Fitted Distribution of Number of Publications between Contributors

Table 2.5: Analysis of Distribution of Contribution

Repository	AC (o)	Distribution	Parameters	C-20 (%)
LORP				
Ariadne	29.4	Lotka exp. cut-off	$\alpha = 1.57$, $\lambda = 0.011$	75%
Connexions	8.84	Lotka exp. cut-off	$\alpha = 1.35$, $\lambda = 0.0094$	78%
Maricopa	4.20	Lotka exp. cut-off	$\alpha = 2.12$, $\lambda = 0.0067$	64%
LORF				
Merlot	12.5	Lotka exp. cut-off	$\alpha = 1.88$, $\lambda = 0.0006$	82%
OCW				
MIT OCW	39.7	Weibull	$k = 1.07$, $\lambda = 40.5$	50%
LMS				
SIDWeb	40.0	Weibull	$k = 0.52$, $\lambda = 17.14$	72%
IR				
Queensland	1.3	Lotka	$\alpha = 3.01$, $xmin = 2$	22%
MIT	1.3	Lotka	$\alpha = 2.55$. $xmin = 3$	36%
Georgia Tech	2.1	Lotka	$\alpha = 2.25$, $xmin = 5$	68%

the average number of objects published during the lifetime of the contributor. In real world terms, the lifetime can be considered as the period during which the contributor is engaged with the repository. The publishing rate, on the other hand, can be considered as proxy measurement of the talent or capacity that the contributor have to publish learning objects. To compute these values we extract the contributor information from the repositories used in the previous analysis.

In the calculation of the lifetime, we always know its beginning, but we are never sure about its end. A contributor could have published its first object two years ago and its last object one year ago. The measured lifetime will be 1 year. However, if the contributor published one more object just the day after the data was captured, its actual lifetime will be 2 years. To cope with this limitation, the lifetime of a user is only considered finished if the time from the last object insertion is at least as long as the longest period without activity between two consecutive publications. If a lifetime is not ended, it will be assigned the time interval from the first object insertion until the date of data collection.

The measurement of the rate of production also presents some difficulties. The rate of contribution could not be measured if all the objects have been published on the same day. Also, the publication of few objects in a short lifetime will produce inflated rate values. To alleviate this problem, only users whose lifetime is larger than 60 days, and that have published at least 2 objects are considered for the calculation. To prevent the bias in the distribution produced by only considering highly productive contributors, the contributors that have a lifetime shorter than 60 days are assigned the smallest production rate.

As expected, each contributor present a different lifetime. To obtain a clearer picture of how this values compare between different repositories, we calculate the Average Lifetime (ALT). Table 2.7 presents the results measured in days. The first conclusions that can be extracted from the lifetime values is that, in average, they are much smaller than the lifetime of the repositories (Table 2.3). This means that most contributors are "retired" after a period of 1 year. This conclusion holds even if the contributors "born" during the last year are removed from the calculation. However, the actual values of ALT are not related to the type of repository and do not provide much information about the distribution of the lifetime among contributors.

Table 2.6: Result of the Analysis of Publishing Rate

Repository	**APR** (o/d)	**Rate Distribution**	**Rate Parameters**
LORP			
Ariadne	0.082	Log-Normal	$\mu log = -3.25$, $\sigma log = 1.27$
Connexions	0.046	Log-Normal	$\mu log = -4.11$, $\sigma log = 1.36$
Maricopa	0.010	Log-Normal	$\mu log = -5.18$, $\sigma log = 0.95$
LORF			
Merlot	0.37	Log-Normal	$\mu log = -2.47$, $\sigma log = 1.11$
OCW			
MIT OCW	0.32	Log-Normal	$\mu log = -1.68$, $\sigma log = 1.07$
LMS			
SIDWeb	0.12	Log-Normal	$\mu log = -2.57$, $\sigma log = 0.96$
IR			
Queensland	0.17	Log-Normal	$\mu log = -4.05$, $\sigma log = 2.07$
MIT	0.24	Log-Normal	$\mu log = -7.08$, $\sigma log = 2.14$
Georgia Tech	1.55	Log-Normal	$\mu log = -1.84$, $\sigma log = 2.53$

Based on the skewed nature of the distribution of lifetime, we fit the five heavy-tailed distributions used previously. The results can be seen in Table 2.7. The distribution is clearly related to the type of repository. A comparison between

Table 2.7: Result of the Analysis of Publishing Rate. ALT is measured in days

Repository	**ALT** (d)	**LT Distribution**	**LT Parameters**
LORP			
Ariadne	514	Exponential	$\lambda = 0.0010$
Connexions	261	Exponential	$\lambda = 0.0012$
Maricopa	304	Exponential	$\lambda = 0.0012$
LORF			
Merlot	328	Exponential	$\lambda = 0.0015$
OCW			
MIT OCW	67	Weibull	$k = 1.72$, $\lambda = 325$
LMS			
SIDWeb	364	Weibull	$k = 1.21$, $\lambda = 588$
IR			
Queensland	319	Log-Normal	$\mu log = 6.01$, $\sigma log = 0.89$
MIT	120	Log-Normal	$\mu log = 3.84$, $\sigma log = 2.45$
Georgia Tech	9.3	Log-Normal	$\mu log = 3.61$, $\sigma log = 2.16$

the different lifetime distributions across several repository types can be seen in Figure 2.10.

LORs contributors have lifetimes that are distributed exponentially among the population. The λ parameter of the exponential is also similar across LORs. This similarity suggests that LORs contributors share the same type of engagement with the repository. The probability of cease publishing is proportional to the time that the contributor have been active. The result is that there are considerable amount of user with short lifetimes (less than 3 months). We can classify this behavior as engagement by novelty. As the novelty worn off, the user ceases contributing.

In the case of OCWs and LMS, the lifetime follows a Weibull distribution. Again, the shape and scale parameters share some similarity. A Weibull distribution with those parameters hints that the amount of contributors with very short lifetimes (less than 1 month) do not dominate the population. It is more common to find contributors that keep publishing after three months to one year. However, Weibull decreases rapidly after its peak, meaning that it is infrequent to find contributors with several years of publication. We describe this behavior as engagement by need. The average contributor keeps publishing until a goal is reached (for example, a course is completed and/or improved). After the goal have been reached the probability of stopping increases rapidly.

The IRs present a very different publishing behavior, denoted by the Log-

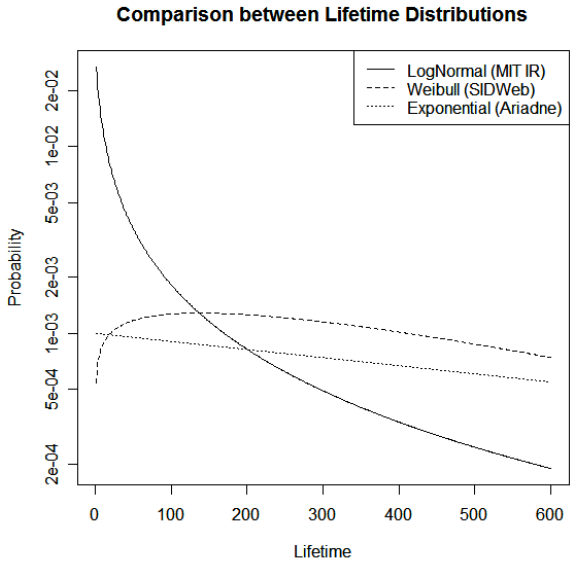


Figure 2.10: Comparison between Lifetime Distribution between Repository Types.

Normal distribution. The low σlog parameter, found in all the IRs lifetimes, means that the majority of the contributors have a very short lifetime (few weeks), with a neglectable amount having lifetimes measured in months or years. This result is consistent with the finding that most contributors in the studied IRs published just one object, probably their thesis. After this publication, and, maybe, some immediate corrections or additions, the contributors cease publishing. We describe this behavior as low engagement. The repository does not require or promote continuous submissions from the majority of users. As a response, most user lifetimes are basically instantaneous compared with the lifetime of the repository.

After analyzing the lifetime, we calculate the average publication rate (APR) for each repository. The results are presented in Table 2.7. The only clear conclusion that can be extracted from the APR is that LORP contributors publish less frequently than other types of repositories. Specially if compared with the similar LORF, MERLOT, the publication rate seems to be one order of magnitude lower. As mentioned before, the main difference between the size of LORPs and LORFs seems to radicate in a difference in productivity of the contributor base. The difference of APR between the other type of repositories is not clear.

To gain better insight on how the publication rate is distributed across the contributing population, the five previously used statistical distributions are fitted

to the data. The results of the fitting are presented in Table 2.7. Surprisingly, all the repositories shows the same distribution, Log-Normal. The main difference seems to be that the σlog parameter is around one for LORPs, LORFs, OCWs and LMSs, and around 2 for IRs. A higher σlog create a larger skewness to the left, meaning that a larger proportion of contributors is low-productive. The finding of the same distribution for all the repositories is very significant, because it means that there is no difference between the distribution of talent or capacity among the different contributor communities.

This analysis shows that the main differentiator between different types of repositories is the type of engagement that the contributors have. According to the findings, the most successful models of repository seems to be OCWs and LMSs, where most of the contributors keep publishing for longer periods of time. This result suggests again that an incentive-based publishing is the most effective form to increase the total number of learning objects available.

2.5 Modeling Learning Object Publication

Once we have measured several characteristics of the publication of learning objects, we try to formulate a model that could generate the observed results with the lowest amount of initial parameters. The objective of this model is to understand the relation between the micro-behavior (contributors publishing learning objects at a given rate during a given time) and the macro-behavior (repositories growing linearly, publication distribution having a heavy tail). This model will also enable us to answer some of the questions raised during the Introduction. Finally, this model could help us to play with the initial parameter and simulate the macro-behavior that the repository would have. For example, the model will help us to know what type of initial factors give rise to exponential growth. This model is inspired by the ideas of Huber [Huber, 2002]. Huber modelled the distribution of the amount of patents published among inventors using four variables: the Frequency (publication rate), the Career Duration (lifetime), the Poisoness and the Randomness. While we use some of his ideas, our methodology expands that model in two main ways: 1) our model is capable of generating non-Lotka distributions, and 2) the predictive scope of our model is larger, including the growth function and total size.

2.5.1 Model Definition

Our model will be based in three initial factors that can be changed to simulate different types of repository:

- Production Rate Distribution (PRD): This specifies how talent or capability is distributed among the contributor population. In our case, we found

that the Log-Normal is a good approximation for the studied repositories. However, any distribution can be set to test "what-if" scenarios.

- Lifetime Distribution (LTD): This specifies the amount of time that different contributors will be active in the repository. For the studied repositories, Exponential, Log-Normal and Weibull seem to represent different types of contributor engagement.

- Contributor Growth Function (CGF): This is a factor that, for now, cannot be predicted. Different contributor growth functions give rise, in combination with the lifetime and production rate, to different content growth functions and size distributions among repositories.

To validate this model, the initial factors are extracted from the results of the quantitative analysis and used to simulate the behavior of the publication process in those repositories. The results of this simulation are then compared with the empirical data.

While the initial factors can be formally defined (distribution functions), the process to derive the model predictions involves non-linear calculations [Huber, 2002] that makes it unfeasible to derive an exact mathematical solution (resulting distribution) that can be easily interpreted. Therefore, we use numerical computation to run our model. This model, while less formal, is very flexible to accommodate a greater range of initial factors. It is perhaps important to note that existing mathematical models proposed to explain the presence of Lotka distributions in the publication process, such as [Egghe, 2005] and [Egghe and Rousseau, 1995], do not necessarily apply to publication of learning objects. The mostly constant production rate of the contributors defy the traditional explanation that "success breeds success".

The construction of the model can be described as follows: First, the period of time, over which we want to run our model, is selected. The Contributor Growth Function (CGF) is then used to calculate the size of the contributor population at the end of that period. The next step is to create the a virtual population of contributors of the calculated size. Following, the two basic characteristics, publication rate and lifetime, are assigned to each contributor. First, a publication rate value, generated randomly from the Production Rate Distribution (PRD), is assigned to each contributor. Second, a lifetime value, generated randomly from the Lifetime Distribution (LTD) is also assigned to each contributor. Once the virtual contributors parameters has been set, each contributor is assigned a starting date. The number of contributors slots for each day is extracted from a discrete version of the CGF. Each contributor is assigned randomly to each one of those slots. If the start date plus the lifetime of a contributor exceed the final date of the simulation, the lifetime is truncated to fit inside the simulation period. Once we have created a simulated population, we proceed to run the model. A Poison process is used to simulate the discrete publication of learning objects. The

lambda variable required by the Poison process is replaced by the contributor's publication rate. The process is run for each day of the contributor's lifetime. If the Poison process is applied for each contributor, the results will be a list containing the contributors, the number of objects that she have published and the dates in which those publications take place. This simulated data is similar to the empirical data used to perform the previous quantitative analysis.

In formal terms (Equation 2.1), the random variable N, representing the number of objects published by each contributor, is equal to the multiplication of R, the random variable representing the rate of production of the contributor, multiplied by L, the random variable representing the lifetime of the contributor in the repository. Given that solving the multiplication of random variables often involves the use of the Mellin transform [Epstein, 1948] and the result is not always easily interpretable [Huber, 2002], we solve this multiplication trough computation methods. Equation 2.2 shows the resulting distribution of N. The probability of publishing k objects is the combined probability of each contributor publishing k objects. Given that the production of a contributor is considered independent of the production of any other contributor, the combination of probabilities is converted into a product for the Nc contributors. To calculate the probability with which the ith contributor publish k objects, we use the formula of the Poisson process with production rate R_i and lifetime L_i randomly extracted from their correspondent distributions. This formula calculate the probability that the contributor publishes exactly k objects during her lifetime.

$$N = R \otimes L \tag{2.1}$$

$$P(N = k) = \prod_{i=1}^{Nc} \frac{(R_i \cdot L_i)^k}{k!} e^{-R_i \cdot L_i} \tag{2.2}$$

2.5.2 Model Validation

To validate this model we compare the simulated results against the data extracted from real repositories. Three characteristics of the repository are compared: 1) distribution of the number of publications among contributors (N), 2) the shape of the content growth function (GF) and 3) the size of the repository (S). The repositories used in this evaluation are the same used in the analyzes of section 4. To perform the evaluation the initial factors were extracted from the results of the contributor growth analysis (section 3) and the production rate and lifetime analysis (section 4). This factors were feed into the model and used to run the simulation. To have a statistically meaningful comparison between the data and the output of the model, we generate 100 Monte-Carlo simulated runs for each repository.

First, we compare the distribution of publications (N) between the empirical and simulated data. To have a meaningful comparison, we estimate the parameters of the distribution of the simulated data with the same methodology used in the Contributor Analysis (section 4). As expected, each simulated data set was fitted with slightly different parameters values. However, the values were normally distributed. We apply a simple t-test to establish if it is reasonable to assume that the parameters fitted to the empirical data set belongs to the same population as the simulated parameters. If all the parameters of the empirical distribution belong to the same population as the simulated ones, it can be concluded that the empirical and simulated data sets have the same distribution. The p-value for the t-test is provided in Table 2.8 together with the mean values of the simulated parameters.

For LORs, the model is able to accurately simulate the *alpha* value for all the repositories. The *alpha* parameter basically determines the general shape of the Lotka distribution. The *rate* parameter, on the other hand, has a more subtle effect. This parameter determines the slight diminish of the probability to find very productive contributors. The model seems not consistently capable to reproduce this value. The subtle effect that determines the exact value of *rate* is most probably lost during the simplifications of the model. An example of the simulation of the Connexions repository is presented in Figure 2.11.

The *shape* of the OCW MIT Weibull distribution of publications seems to present a major challenge for the model. The almost horizontal head of the distribution cannot be accurately simulated with the current calculations. The *shape* parameter is vastly underestimated. However, the tail of the distribution is reasonably matched by the simulated values and the *scale* parameter is correctly estimated. The comparison between one simulation run and the empirical data can be seen in Figure 2.11. The model, nonetheless, can model less extreme weibull distributions, as can be seen in the estimation of the SIDWeb parameters.

The IRs publication distribution presents a steep slope as the majority of the users publish only one object. However these users are not considered in the publication rate estimation (section 4). This overestimation of the production rate cause the model to generate lower *alpha* values in the simulation (a lower *alpha* results in a less step distribution). This effect can be seen in Figure 2.11. Given the difference in the general shape, the estimation of $xmin$, while sometimes statistically significant, do not have any meaning. Through direct manipulation of the rate of production, we found that reducing the rate by a factor between 5 and 10 leads to the desired distribution. How to theoretically explain the value of that factor remains an open question.

The next step to validate the model is to compare the shape of the content growth function (GF) and the final size of the repository (S). For the GF evaluation, the daily simulated production of objects was counted across contributors. First, the count was fitted with the same methodology and functions used in sec-

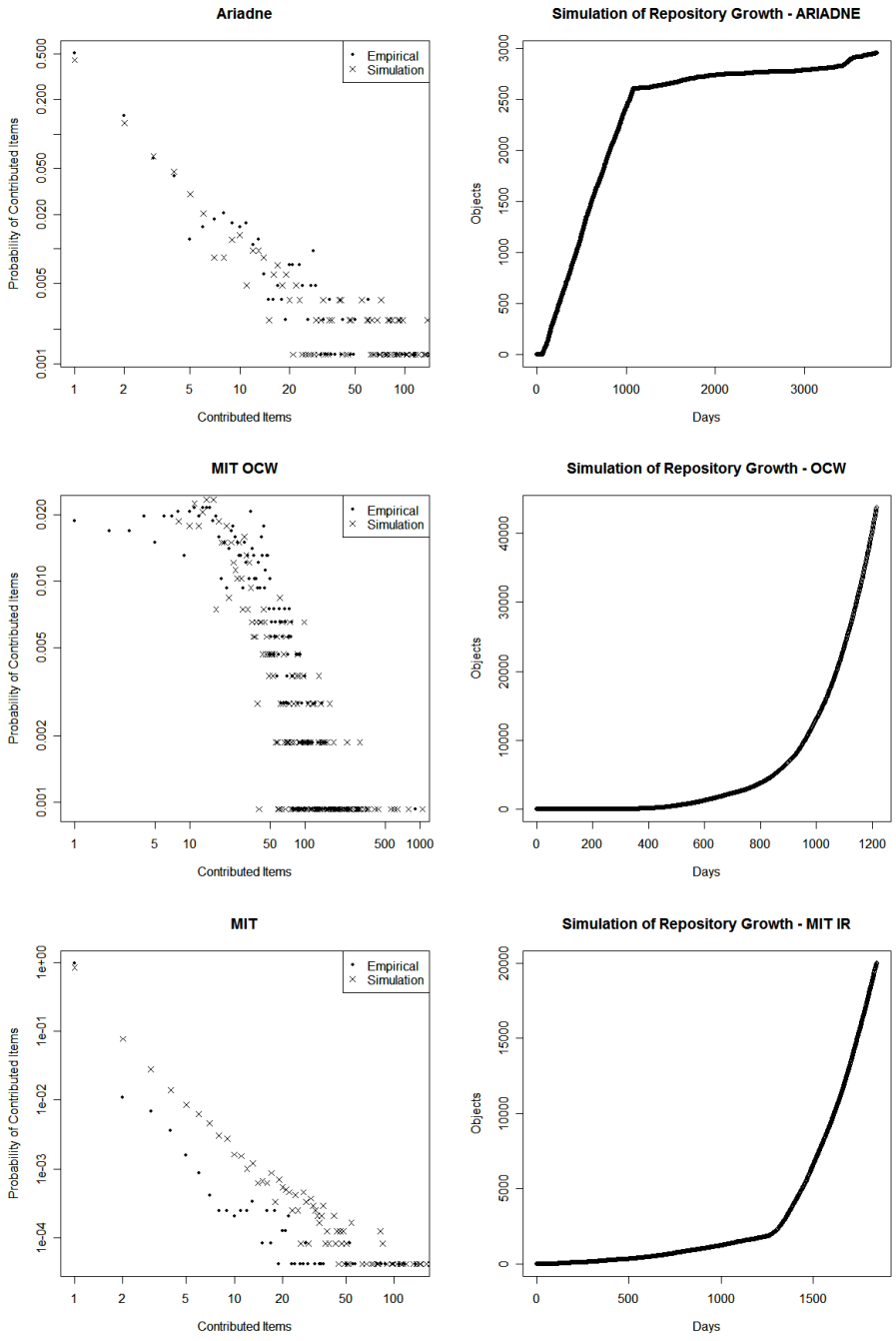


Figure 2.11: Empirical and Simulated Distributions of Publicantion and Growth Function

Table 2.8: Results of the Simulation of the Distribution of Publications

Repository	**P1**	**p P1**	**P2**	**p P2**
LORP				
Ariadne	$\alpha = 1.58$	0.60	$\lambda = 0.001$	**0.02**
Connexions	$\alpha = 1.42$	0.31	$\lambda = 0.0002$	0.07
Maricopa	$\alpha = 2.39$	0.28	$\lambda = 0.04$	0.10
LORF				
Merlot	$\alpha = 1.76$	0.34	$\lambda = 0.002$	0.13
OCW				
MIT OCW	$shape = 0.68$	**0.00**	$scale = 35$	0.22
LMS				
SIDWeb	$shape = 0.60$	0.21	$scale = 19$	0.55
IR				
Queensland	$\alpha = 3.50$	**0.00**	$xmin = 2$	**0.00**
MIT	$\alpha = 2.35$	0.18	$xmin = 2$	0.09
Georgia Tech	$\alpha = 2.01$	0.12	$xmin = 5$	1.00

tion 3. We count the times that the correct function, Bi-Phase Linear, was selected as the best-fitting alternative. For the S evaluation, we just count the total number of objects produced in each simulated data set. The distribution of the final size follows an left-skewed distribution. We use the Empirical Cumulative Density Distribution (ECDF) to calculate the chances that the empirical size came from the same population. The results of these evaluations are presented in Table 2.9.

The simulated growth functions seems to have the Bi-Phase Linear shape that was found during the analysis in section 3. When the contributor base growth function is also Bi-Phase Linear (Ariadne, Maricopa, MERLOT, Queensland, MIT and Georgia), the accuracy of the prediction is high (90% or higher). However, when an exponential contributor rate growth is involved in the calculation (Connexions, MIT OCW and SIDWeb), the identification rate decreases (60-80%). Exponential and Simple linear growth are the main winners when the Bi-Phase Linear was not selected. It is interesting to note that thanks to the variability in the lifetime, an exponential contributor growth does not necessarily means exponential growth in the number of objects. However, as the miss-interpretation rate shows it, when there is exponential growth in the number of contributors, exponential growth in the number of objects is a viable outcome. These effects can be observed in Figure 2.11 (rigth). There, a graphical representation of random simulated growth functions is presented.

The actual parameters of the Bi-Linear are not analyzed, as they varied widely from simulation to simulation. The implication of this variation is not clear. It can be that natural variation could create several types of growth from the same contributor population or that our model, in its simplicity, does not take into

Table 2.9: Simulation of the Size of the Repositories

Repository	Bi-Phase Linear	Simulation S	p S
LORP			
Ariadne	100 %	5,516	0.48
Connexions	73 %	6,052	0.50
Maricopa	100 %	3,105	0.36
LORF			
Merlot	98 %	20,389	0.61
OCW			
MIT OCW	65 %	48,320	0.52
LMS			
SIDWeb	76 %	25,443	0.20
IR			
Queensland	83 %	13,121	0.52
MIT	92 %	28,294	0.61
Georgia Tech	96 %	25,657	0.47

account some relation between lifetime and production rate that is responsible for the shape of the function. More research is needed to solve this question.

Finally, we compare the final number of produced objects when the simulations have been run for the same period of time as measured in the empirical data sets. As can be seen in Table 2.9, the size values for all the repositories were estimated correctly, even in repositories where the simulated and empirical publication distributions does not completely match (OCWs and IRs). The reason for this resilience is that the tail of the distribution (or the head, in the case of OCW) are responsible for a small fraction of the object. If the simulation can match the head (or the middle section, in the case of OCW), where most of the objects are published, the total simulated output is similar to the original repository. These results support the use of this model to calculate growth and required capacity.

2.5.3 Conclusions

The presented model makes the simple assumption that the only variables or factors that affect the analyzed characteristics of a repository are how frequently the contributors publish material (publication rate), how much time they persist in their publication efforts (lifetime) and at which rate they arrive at the repository (contributor growth function). The model combines those variables through a computational simulation that is capable of predicting other repository characteristics as the distribution of publications among contributors, the shape of the content growth function and the final size of the repository.

The model is evaluated with the data extracted from the analysis sections.

From this evaluation, it can be concluded that the simple model is capable of simulating quite well the characteristics observed in real repositories based only on the initial factor. However, the simplicity of the model can be seen when the model tries to simulate repositories with special characteristics. For example, with a larger or smaller than expected low-publishing communities. Nonetheless, the model can be used at it is to predict future growth of current repositories or to simulate repositories with characteristics not seen naturally. For example, how the publication distributions will be like if the publication rate is uniformly distributed. Improvements of these model to include special cases, as well as interactions between the factors is an interesting topic for further research.

2.6 Implication of the Results

The results of the quantitative analysis and the presented model can be used to answer the questions raised in the introduction. This section presents those answers and the implications that they have in our understanding of the learning object publication process and the technological design of repositories.

- What is the typical size of a repository? Is it related to its type?
  In general, individual learning object repositories seems to vary from hundreds to million of objects. Their average size depends of the type of repository. LORPs can be considered to have few thousand of objects. LORFs are in the order of the tens of thousands. However, those numbers are small compared with multi-institutional IRs that can count hundreds of thousands and even millions of objects. OCWs and LMSs can have from hundreds to thousand of courses.

  However, the answer to this question is not that simple. The size is not Normally distributed, meaning that the average value cannot be used to gain understanding of the whole population. It is not strange to find repositories several orders of magnitude bigger or smaller than the average. Sampling biases aside, the distribution of learning objects among repositories seems to follow a Lotka or Power Law distribution with an exponent of 1.75. The main implication of this finding is that most of the content is stored in few big repositories, with a long, but not significant tail. Administrators of a big repository would want to federate [Simon et al., 2005]] their searches with other big repositories in order to gain access to a big proportion of the available content. On the other hand, it makes more sense for small repositories to publish their metadata [Van de Sompel et al., 2004] for a big repository to harvest it in exchange for the access to their federated search. It seems, through an initial reading of this finding, that a two (or three) tiered approach mixing federation and metadata harvesting is the most efficient

way to make most of the content available to the wider audience possible using the current infrastructure.

Another implication of the current different size of repositories is that the technological solutions used by IRs (DSpace [Tansley et al., 2003], Fedora [Lagoze et al., 2006], etc.) could provide a tested architecture over which current LORs can grow in the future. IR architecture been tested with million of objects. Exponential growth aside, LORs will only start needing that capacity over the five years.

- How many learning objects are typically used in a course?
  From the analysis of section 2, the simple answer is 20. That is the amount of learning objects that, in average, are present in common aggregation of learning objects (courses) in LORs, OCWs and LMSs. However, a heavy-tail distribution, Weibull, reduces the meaning of this value. An instructor normally strives to have between 15 to 35 objects in her course. This number can be related to the number of lessons or sections of the course (Probably 1 or 2 objects per lesson). There is a considerable amount of courses that have from 1 to 20 objects and small fraction of the courses that can have more than 200 objects.

  The main implication of this finding is that if OCWs and LMSs are decomposed and converted into repositories, they can be considered very large LORPs. The fact that LMSs are a widely deploy technology [Harrington et al., 2004] and that these systems are not accessible for external visitors make us think of the learning objects present in LORs as just the "tip of the iceberg". The bigger part of learning resources is hidden behind login pages. This finding validates the effort of the OCW Consortium and OER Commons [Joyce, 2007]. If we want to create a really functioning Learning Object Economy, we must start opening the door of our LMSs.

- How repositories grow over time?
  Linearly. This is a discouraging finding. Even popular and currently active repositories grow linearly. Even if we add them all together, we will still have a faster linear, but no exponential. The main reason for this behavior is the contributor desertion. Even if the repository is able to attract contributors exponentially, it is not able to retain them long enough to feel the effect. The value equation, how the contributor benefits from contributing to the repository, is still an unsolved issue in most repositories. Several researches have suggested incentive mechanism [Downes, 2007] [Hummel et al., 2005] comparable to scientific publication, in order to provide the professor with some type of reward for their contribution.

  Another interesting result in the growth analysis was to find that all repositories went through an initialization with usually a very low growth rate.

The length of this stage varied from 1 to 3 years (shortening for more recent repositories). After this period, a more rapid expansion begins caused by (or that cause) an increase in the number of contributors joining the repository. Having knowledge of these phases could help repository administrators to not discard slow growing repositories too soon.

- What is the typical number of contributors a repository has? Is it related to its type?
  We can estimate, from the analysis in section 3, that medium LORs have a base of 500 to 1500 contributors. This number is similar also for OCWs and LMSs contributor bases. On the other hand, IRs, being targeted also to students, have contributor bases one order of magnitude bigger. The size of the contributor base, however, is not always related to the size of the repository. Merlot contributors, being outnumbered 1 to 10, produce a comparable amount of objects as MIT IR contributors. More over, The title to the most productive contributors in the study goes to the OCWs and LMSs professors (Table 2.5) with around 40 objects in average. Interesting results that correlate well with the average of 20 objects per course if each professor is in charge, in average, of two courses. This results also support the idea that LMSs are the most effective type of repository, given that they provide a clear value into the publishing step (students not asking for copies of the material, for example).

  Given the relatively small size of the communities that build repositories, it would be an interesting experiment to measure the impact that the introduction of social networks could have in the sharing of material. For example, users would be interested in knowing when a colleague in his same field has published new learning objects [Duval, 2005]. This social networks can be created explicitly (a lá Facebook) or implicitly (relationship mining) [Matsuo et al., 2006]. The deployment of these types of networks could also help to solve the lack of engagement problem.

- How the number of contributors grows over time?
  Most of them linearly, but surprisingly three of them Connexions, MIT OCW and SIDWeb, exponentially. This unexpected result, specially in SIDWeb, a run-of-the-mill LMS, is very encouraging for the future of the Learning Object Economy, because it can give rise, with the right environment, to exponential growth of content available. However, we also found that at this stage, the growth in those repositories continue linear. However, this observation can be due to the recent kick-off of exponential contributor base growth in those repositories. A follow-up study in a year period would help us to have a better perspective. Again, the finding of exponential growth in course-based repositories confirms the idea that we should strive to connect LMS as the main source of learning material.

Similarly to content growth, the number of contributors also have an initial slow stage that accelerate once the repository reaches maturity. It could not be stablished with the present study if the increase of objects cause the increase of contributors or vice versa. The "chicken and egg" dilemma deserves further analysis.

- How many learning objects a contributor publishes in average?
  As mentioned before, the average productivity of users depends on the type of repository. OCWs and LMSs contributors are at the top with a total output at 40 objects in average. For LORP it can be around 10 objects per contributors. IRs present the lowest production per contributor with 1 or 2 in average. However, heavy tail distributions, Lotka and Weibull, makes this answer a little more complicated. The problem with the average values given previously in the current situation, is that in heavy tailed distributions "there is not such thing as an average user" [Ochoa and Duval, 2008b]. As mentioned in section 4, the best way to describe the production of different contributors is to cluster them in "classes" similar to socioeconomic strata. If we adopt this approach we gain a new way to look at our results. In LORP and LORF, the repository is dominated by the higher-class. Most of the material is created by a few hyper-productive contributors. the 10% of the users could easily have produced more than half of the content of the repository. In the case of OCWs and LMS, the Weibull distribution determines that the middle-class is the real motor of the repository. The low- and high-class are comparatively small. Finally, University IRs, with Lotka with high *alpha* are dominated by the lower-class as more than 98% of the population produces just one object.

  From our analysis on publishing rate and lifetime, we can conclude that these different distributions are caused not by an inherent difference in the talent or capacity among the different communities, but by the difference in contributor engagement with the repository. It seems that the distribution of lifetime, the time that the contributor remains active, is different for this three observed repository types. In LORP and LORF, there is some time of novelty engagement that keep the contributor active at the beginning, but the chances of ceasing publication increases as more time is spent in the repository. For OCWs and LMSs, there is a goal-oriented engagement that keeps the contributor productive until her task is finished (course is fully published). In the case of IRs, there is no engagement at all. The norm is just discrete contributions. Changes on the type of engagement should have an effect not only in the distribution of publications among users, but also in the growth and size of the repository.

  In conclusion, it is very important for a repository administrator to know the composition and characteristics of her contributor base. Having a clear

view of what and who need to be incentivized is the first step before building any type of incentive plan [Berendt et al., 2003].

It is also interesting to note that these distributions are not exclusive for the publication of learning objects, but are shared by various types of user generated content (UGC) [Ochoa and Duval, 2008b] . Moreover, there is a long research history of how the Lotka law fit the process of scientific publishing [Pao, 1986]. The quantitive study of Learning Objects (or Learnometrics as we call it) can borrow substantial amount of research results from other Informetrics fields. Moreover, Informetrics could also benefit from having new sources of data in an specific domain to test the generality of their conclusions.

- Is there a model that could explain the observed distributions?
  In section 5, we propose a simple model that could explain the different publication distributions based only on three characteristics of the contributor base: production rate, lifetime and growth. Even if these three characteristics are considered independent, the model produces a good approximation to the characteristics of most of the studied repositories. This model, because of its simplicity, presents several shortcomings. However, it can be used as a first approximation to the process of learning object publishing. Maybe the most important characteristic of the proposed model is its testability. It would be easy to construct competing models and test if they predict better different characteristics of the repositories and can handle special cases that the current model can't. This testability provides a way to measure progress into our journey to understand the nature and workings of the learning object publication process.

Despite the previous answers, this analysis raises more questions than it solves. We invite the reader to check the Further Research section at the end of this disertation to share what we consider to be the most interesting new paths opened by this work.

2.7 Conclusions

The present chapter is the first quantitative analysis performed to the publication of learning objects. We have raised and answered several basic questions important for the understanding of the publication process and the design and operation of learning object repositories of several types. Based on the results of the analysis, we propose and evaluate a model based on the characteristics of the user base to offer explanation to the different distributions present at different types of repositories.

Maybe the most relevant conclusion from the quantitative analysis is that the publication process is dominated by heavy-tailed distributions and the usual

Gaussian-based statistics are not enough to gain insight on the nature of the compiled data. These distributions also provide the repositories with several characteristics not found in more normal sets. For example, difference in size or productive can span through several order of magnitude. Depending on the parameters of the distributions, it will not be unexpected that most of the content of a repository is produced but few individuals or that 99% contributor base only publish one object. The black-swan effects [Taleb, 2007] can be seen, measured and modeled in the composition of all repositories.

The provided model, although simplistic, enables us to simulate most common types of repositories to an acceptable degree. It is encouraging that a first attempt can already produce good results. Improved models can use the same validation methodology and data to test themselves and measure if they indeed improve our basic assumptions. Having a common ground over which to build assures that competitive explanations could, for the first time, be quantitatively compared in the field of Learning Object technologies.

Finally, measuring the publication process enable us to take better decisions about the architecture and infrastructure needed to support the Learning Object Economy. Moreover, measuring is our only way to test the unproven assumptions over which some of the current Learning Object technology rests.

To complement this study about the supply of learning objects, the next chapter will analyze the other side of the economy: the demand. Having a clear view of how these two process work could help Market-Makers and Policy-Makers to understand how different technologies and policies affect the Learning Object Economy.

Chapter 3

Quantitative Analysis of the Reuse of Learning Objects

3.1 Introduction

The reuse of learning resources is the *raison d'etre* of Learning Object technologies. Reusing learning objects is believed to generate economical and pedagogical advantages over the construction of learning objects from scratch [Campbell, 2003]. Creation of high quality learning objects is a time and resource consuming task [Wilhelm and Wilde, 2005]. Reusing them in many contexts helps to compensate for those creation costs. Also, learners could have access to learning materials of good quality even if those objects were produced for other contexts.

Due to the importance of reuse in the context of learning objects, it has been one of the most visited topics in Learning Object literature. Some papers concentrate on the theoretical issues that are thought to intervene in the reuse of learning material. Littlejohn discusses the more relevant aspects affecting the reuse of learning objects [Littlejohn, 2003]. McNaugth identifies conflicting cultural and educational factors involved in how teachers address the reuse of learning objects [McNaught, 2003]. Collis and Strijker compare the issues affecting reuse in educational, corporate and military contexts [Collis and Strijker, 2004]. From those papers, it is easy to conclude that reusing learning objects is a complex process affected by several aspects of the objects itself and the context they are reused. Another category of papers tries to simplify this complexity using methods to estimate the reusability of the learning objects. Sicilia and García equate the reusability of an object to its usability in a given context [Sicilia and García, 2003]. The paper presents a methodology for measuring the usability of the object. Cuadrado and Sicilia go a step further, defining measurements based on software reuse metrics to calculate the reusability of the object [Cuadrado and

Sicilia, 2005]. Recently, Zimmerman et al. present a similar approach, measuring the reusability of an object as the inverse of the number of transformations that it has to undergo to fit a specific learning context [Zimmermann et al., 2007]. However, the effectiveness of the estimations described in these papers is not evaluated or contrasted with real reuse, limiting the contribution of those metrics. Finally, a more recent collection of papers tries to measure the process of actual reuse of learning objects. Schoner et al. measure the perception of users that had to create a lesson exclusively reusing learning objects [Schoner et al., 2005]. Elliot and Sweeney, with a similar experiment, measure the time and effort saved when an object has been constructed reusing learning objects, rather than from scratch [Elliott and Sweeney, 2008]. Verbert and Duval, while estimating the effectiveness of a PowerPoint plug-in to (de-)compose slide presentations, measure the number of components reused [Verbert and Duval, 2007]. While interesting, these studies cannot be extrapolated to real reuse situations. First, the number of objects involved in the experiments was small (the tasks consisted of creating one or two composed learning objects) and second, the subjects in the experiment had been instructed to reuse as many objects as possible, which is not an accurate representation of the real world where reusing material is optional.

Despite all the research done during the last 15 years, there exists practically no quantitative information about the actual reuse of Learning Objects. Simple questions, such as what percentage of Learning Objects would be reused in a given collection, have not answers yet. Moreover, assertions, such as the inverse relation between granularity and probability of reuse [Duncan, 2003] , are taken for granted based on theoretical discussion, but have never been contrasted with real-world data. The reason for this lack of quantitative studies is simple. Most reuse occurs privately and it is published in user-password restricted systems, such as LMSs. Moreover, content creators may be reluctant to openly publish material that contains reused components, because fair-use rules that apply for the classroom setting do not apply for public publishing. For example, most of the cost of the Open Courseware projects, such as MIT OCW [Kumar et al., 2001], comes from establishing the original copyright owner of reused material and obtaining permission or rights to publish it on the web [JOCW, 2006]. This reluctance to share objects that rely on reuse leaves Learning Object researchers without access to the principal source of data. Researchers, accordingly, have pursued more productive avenues waiting for the situation to change. The lack of reuse measurement does not only affect researchers, it also has repercussions in the evaluation of Learning Object projects. For example, MIT OCW managers, in the annual reports [Carson, 2004] [Carson, 2005] to justify their existence and success to their funding sources cite the number of hits as a measure of success. While there is scientific support the implied relation between the number of hits and number of citations [Brody et al., 2005], it is not completely clear if this relation also holds for the number of hits and reuse of the objects. Funding agencies often

require explicit measurement of impact that most Learning Object related projects are not able to make.

In recent times, however, the landscape of learning object publishing has changed thanks to initiatives like Creative Commons (CC) [CreativeCommons, 2003]. In its most common form, this license not only permits, but encourages the reuse, adaptation and re-publishing of existing resources. Moreover, sites like Connexions [Baraniuk, 2007], and more recently LabSpace [McAndrew, 2006], are online applications where users can "find, mix and burn" Learning Objects. This openness finally enables the study of reuse mechanisms. Another source of recently available reuse information comes from the automatic decomposition of learning objects. The ALOCOM PowerPoint plug-in, described in [Verbert and Duval, 2007], not only decomposes slide presentations, but also detects the reuse of images, tables and even text fragments. This automatic measurement of reuse can also provide insight in the amount of reuse in real world situations.

This chapter uses this newly available information to perform a quantitative analysis of the reuse of Learning Objects of different granularities in different contexts. In order to provide a useful comparison framework, the same analysis is also applied to other forms of component reuse, such as images in encyclopedia articles, libraries in software projects and web services in web mashups. This chapter provides initial answers to the following questions:

1. What percentage of learning objects is reused?
2. Is the reuse in learning objects similar to other types of component reuse?
3. Does the granularity of a learning object affect its probability of reuse?
4. Is there a relation between the popularity of an object and its reuse?
5. What is the distribution of reuse among learning objects?
6. Is the distribution of reuse in learning objects similar to other types of component reuse?
7. What model of reuse could explain the observed distributions?

Together with the findings of chapter 2, the answers to these questions will help Market-Makers to understand how their repositories are being used. Knowing better the behavior of their Consumers can lead to an improved presentation of the content, as well as to the development of metrics to help in the selection process (Chapter 5). Understanding the percentage and distribution of reuse will also help Policy-Makers to estimate the benefit that the publication of learning material has in the facilitation of creation of new material. This knowledge is key to evaluate the importance that Learning Object Technologies have in learning.

The structure of the chapter is as follows: section 2 describes the data sources for the analysis. Section 3 presents three quantitative analysis performed over the data: 1) the amount of reused is measured, 2) the reuse is compared with the popularity of the objects and 3) the distribution of reuse among objects is analyzed. Section 4 provides a model to interpret the observed results. Section 5 presents the main implications of the result and the answers to the research questions.

3.2 Data Sources

To perform a quantitative analysis of the reuse of learning objects, this chapter uses empirical data collected from three different openly available sources. The sources were chosen to represent different reuse contexts and different object granularity.

Small Granularity: Slide Presentation Components. A group of 825 slide presentations obtained from the ARIADNE repository [Duval et al., 2001] were decomposed and checked for reuse using the ALOCOM framework [Verbert et al., 2006]. From the decomposition of the slides 47,377 unique components were obtained. A component is considered reused if it is present in more than one slide. The collection of the slide presentations was made on March 2007.

Medium Granularity: Learning Modules. The 5255 learning objects available at Connexions [Baraniuk, 2007] at the time of data collection were downloaded. Some of these objects belong to collections, a grouping of a similar granularity as a course. 317 collections are available at Connexions. A module is considered reused if it is used in more than one collection. The data were collected through Web scraping on March 2nd, 2008.

Large Granularity: Courses. The 19 engineering curricula offered by ESPOL, a technical University at Ecuador, reuse basic and intermediate courses. When a new curriculum is created, existing courses, such as Calculus and Physics, are reused. On the other hand, more advanced courses, for example Power Lines in the case of Power Engineering, are created and only used in the specific curriculum. Based on the published information[1], the 463 different courses were obtained. A course is considered reused if it is mandatory in more than one curriculum. The data were collected manually from the published curricula on March 20th, 2008.

In order to offer a reference for comparison, data from other reusable components was also obtained from openly available sites on the web. The sources were chosen to be as similar in granularity as possible to their learning object counterparts.

Small Granularity: Images in Encyclopedia Articles. A dump of the English version of the Wikipedia database[2] was used to obtain the identifier of the images used in different articles. 1,237,105 unique images were obtained. An image is

[1] ESPOL Curricula. http://www.espol.edu.ec/
[2] Wikipedia Database. http://en.wikipedia.org/

considered reused if it is included in more than one Wikipedia article. The database dump corresponded to the latest version available on March 15th, 2008.

Medium Granularity: Software Libraries. The information posted at Freshmeat[3] under the category "Software Libraries" was used to obtain a list of 2,643 software projects whose purpose is to be used in other programs. Freshmeat is a site where software projects can be posted. Each project can declare which libraries it depends on. That information was used to measure the reuse of each one of the posted libraries. A library is considered reused if it has been included as a dependency in more than one Freshmeat project. The data were collected from Freshmeat site using Web scraping on March 17th, 2008.

Large Granularity: Web Services. Programmable Web[4] compiles one of the most comprehensive lists of Mashups and Web Services available on the Web. Given that a the code of the Mashup is small compared with the code of the Web Services, the Web Service could be considered as coarse-grained in the context of the Mashup. 670 Web Services were listed in Programmable Web. A Web Service is considered reused if more than one Mashup uses it as a source of information. The data were collected from the Programmable Web site using Web scraping on March 18th, 2008.

It should be noted that we are aware of the limitations of the collected data set. The measure of reuse obtained could be not completely correct. For example, the ALOCOM reuse mechanism could not detect reuse when an image has been modified even slightly outside PowerPoint; software projects in Freshmeat could have neglected to include all the libraries they depend on. Also, the number of objects considered in the data set is relatively small compared with the number of objects contained in big repositories [Ochoa and Duval, 2008b]. This is especially true for objects of large granularity. However, we believe that the impact of these two problems is limited and that they do not affect the results of the analysis. For the first problem, the percentage of undetected reuse is bound to be low. There is not common that users edit their images outside PowerPoint and software libraries need to list their dependencies in order to easy their installation. For the second problem, the small amount of objects is compensated by the very low noise in the reuse estimation.

3.3 Quantitative Analysis

Once the data were collected, three analyses were run to measure characteristics of the reuse in those data sets. The first analysis was to obtain the percentage of objects reused in the given sets. The second analysis compared the amount of reuse of an object with its popularity within the collection. The third analysis

[3]Freshmeat Web Site. http://www.freashmeat.net
[4]Programmable Web Site. http://www.programmableweb.com

Table 3.1: Percentage of reuse in the different data sets.

Data Set	Objects	Reused	% of Reuse
Small Granularity			
Components in Slides (ALOCOM)	47,377	5,426	11.5%
Images (Wikipedia)	1,237,105	304,445	24.6%
Medium Granularity			
Modules in Courses (Connexions)	5,255	1,189	22.6%
Soft. Libraries (Freshmeat)	2,643	538	20.4%
Large Granularity			
Courses in Curricula (ESPOL)	463	92	19.9%
Web APIs (P.Web)	670	216	32.2%

studied the statistical distribution that best fitted the reuse data. A complete description of the analyses and their results are in the following subsections.

3.3.1 Amount of Reuse

First, we measure only the percentage of object that has being reused within a collection. To measure this percentage, the number of objects that have been reused was obtained for each set. This number was then compared with the total number of objects in the set. This simple measurement provides insight into the Questions 1, 2 and 3 proposed in the Introduction. Table 3.1 presents the results of this measurement for each data set.

The most interesting result from this analysis is that, in almost all the data sets, the percentage of reuse is close to 20%. This percentage is the same for Learning Object related sets and sets used for comparison. It is also maintained at different levels of granularity. However, two sets deviate from this value. The reuse of components into slides has a percentage of reuse significantly lower (11.5%). On the other hand, the reuse of Web APIs is significantly higher (32.2%). A possible interpretation for this factor is presented in section 4.

3.3.2 Popularity vs. Reuse

The objective of this analysis is to establish if the actual reuse of a learning object is linked to its relative popularity within the collection or repository (Question 4). To perform this analysis, the Connexions and Freshmeat data sets were enriched with information about the number of times that the objects have been accessed. The popularity data was obtained from Web scraping These data sets were selected for this analysis because they were the only ones with access information and have similar granularity.

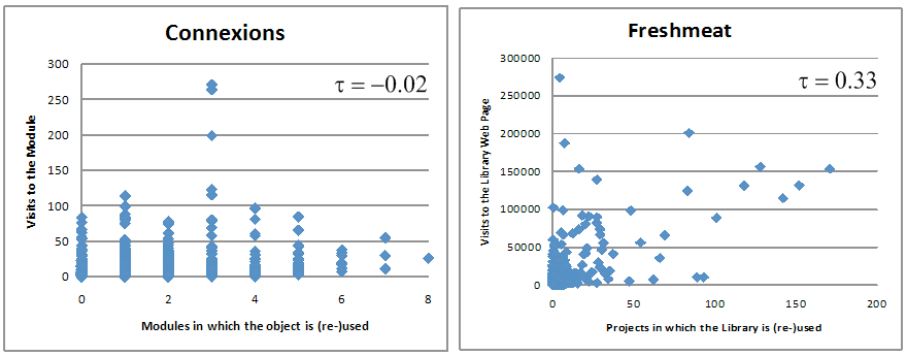


Figure 3.1: Scatter plots of the Reuse vs. Popularity in the Connexions and Freshmeat sets

The analysis consisted in obtaining the Kendall's tau correlation coefficient between the rank of the object in the reuse and popularity scales. Pearson's coefficient is not used because there is no guaranty that the values come from a bi-variate normal distribution. Also, scatter plots were created to visually analyze the relation between popularity and reuse. These graphs are presented in Figure 3.1.

The correlation coefficient tau for the Connexions set was -0.02 (0.05 significant). This value means that there is absolutely no correlation between the popularity of the object and the times that it has been reused. This lack of correlation can be easily seen in Figure 3.1 (left). For example, the most visited object has only be reused in three collections, while the most reused object (8 times) has only received 25 visits. On the other hand, the Freshmeat set obtained a tau of 0.33 (0.01 significant). This result suggests that in the case of software libraries the popularity is slightly linked with the reuse. However, there are cases that have a large popularity but have a low track of reuse. For example, the DeCSS library [Eschenfelder and Desai, 2004], normally used to break DVD encryption, has a large popularity (circa 180.000 visits) but is only used in a small set of specialized DVD players for Linux (8 projects). These results suggest that the popularity of an object cannot always be used as a proxy for its reuse. A more counter-intuitive finding that can be obtained from this result is that a high level of reuse does not imply a high popularity. It would be usually expected that an object reused in several contexts is more findable and, therefore, more visited. The measurement indicates that it is not the case. The explanation for these observations is presented in section 3.4.

3.3.3 Distribution of the Reuse

To gain more insight in the reuse process, the distribution of reuse among different objects was analyzed. The first step in this analysis was to obtain the total number of reuses for each object. The histogram of the data was plotted to obtain a first indication of the type of statistical distribution that could fit to the data. The resulting histograms were highly skewed to the left for all data sets. The left skewness is an indication of heavy tail distributions [Mitzenmacher, 2003]. The next step in the analysis was to fit 5 common heavy tail distributions (Lotka, Lotka with exponential cut-off, Log-Normal, Weibull and Yule) to the data set. The fitting was performed using Maximum Likelihood Estimation (MLE) technique. This is the most recommended technique to estimate parameters for power-law and other heavy-tailed distributions [Goldstein et al., 2004]. Then, the goodness-of-fit of the distributions were compared using the methodology suggested by Vuong [Vuong, 1989] to obtain the distribution that better fits the data.

For all the data sets the Log-normal distribution provided the best fit to the data among the 5 tested distributions. However, for the Web APIs, Software Libraries and Connexions Modules, Vuong's test is not significant and from the statistical point of view, can't be ruled out as possible fit for the data. This phenomenon is common, especially when there are few data points, given that the distributions are very similar over certain ranges [Clauset et al., 2007]. Table 3.2 presents the fitted parameters for the Log-Normal distribution for each set, as well as the significance of the Vuong test for goodness-of-fit against competing distributions. As a visual aid, Figure 3.2 presents the size-frequency plot of the data [Newman, 2005] together with the best fitted Log-Normal distribution.

The fitted parameters of the Log-Normal distributions suggest that there exists two sub-types of reuse distributions. The first group, that includes Connexions Modules, Freshmeat Libraries and Web APIs, has a shape parameter (sdlog) with a value in the vicinity of 1 and scale parameter (meanlog) near 0. The effect of these parameters can be observed in Figure 3.2 as an internal curvature. This curvature is most apparent in the Connexion Modules given the low maximum reuse. The second group, that includes ESPOL Course, Slides Components and Wikipedia Images, has a shape parameter with a value in the vicinity of 20 and a high scale parameter. These values generate a curve that has a slight external curvature. This effect is more noticeable in ESPOL Courses graph (Figure 3.2) due also to the small maximum reuse. This clustering of the data sets was not expected and can only be explained as differences in the reuse process. Further research is needed in order to explain the causes and implications of this observation.

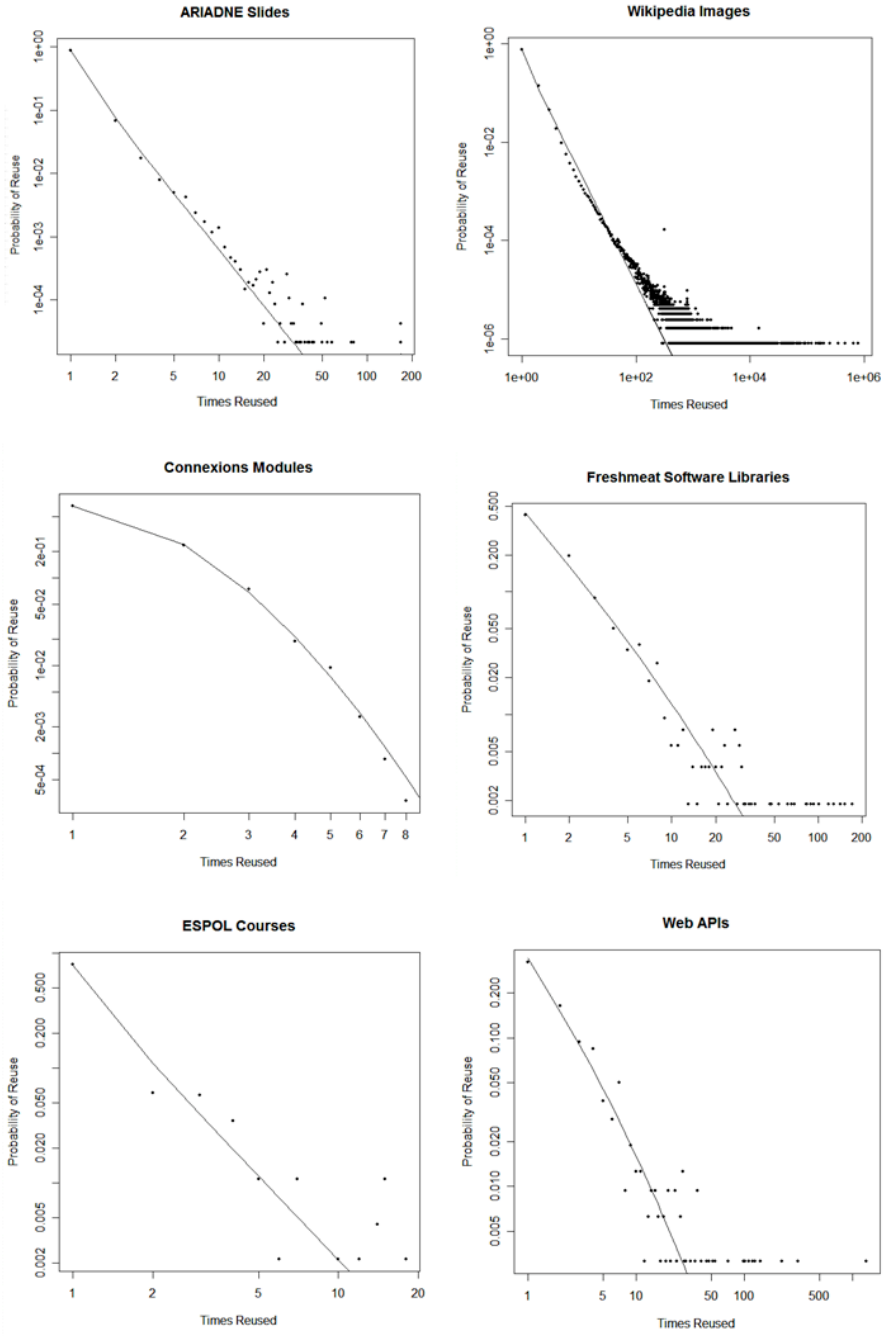


Figure 3.2: Size-Frequency graphs of the data sets (points) and the best fitting Log-Normal distribution (line)

Table 3.2: Log-Normal distribution fitted parameters for each data set and the Vuong test significace against competing distributions.

	Log-Normal		**Vuong Test (Significance)**			
Data Set	**meanlog**	**sdlog**	**Lotka**	**L-cutoff**	**Weibull**	**Yule**
Small Granularity						
ALOCOM	-725	19.5	15 (.00)	15 (.00)	20 (.00)	25 (.00)
Wikipedia	-1057	28.5	19 (.00)	19 (.00)	35 (.00)	27 (.00)
Medium Granularity						
Connexions	0.09	0.62	13 (.00)	0.7 (.48)	17 (.00)	9 (.00)
Freshmeat	-1.12	2.12	2.8 (.00)	0.1 (.91)	2.14(.03)	1.2 (0.23)
Large Granularity						
ESPOL	-490	18.8	2.4 (.02)	2.3 (.02)	5.1 (.00)	3.3 (.00)
P. Web	-0.12	2.10	2.8 (.00)	2.0 (.05)	1.5 (.13)	1.8 (.07)

3.4 Interpretation of the Results

Given the similarities found during the quantitative analysis, it is possible to create a simple model of reuse to interpret the results. This section will present such a model based on the statistical properties of the Log-Normal distribution. This model is then used to explain the results of the three previous analyses. The interpretation made by Shockley, in [Shockley, 1957], to explain the variation in patent productivity inside research laboratories inspirated this model. However, to our knowledge, this is the first time that this model has been used to explain the reuse of components.

3.4.1 Model

The proposed model will consider the actual reuse of an object or component as the result of the success of a chain of consecutive events. A similar idea has been presented theoretically by Weit et al., in [Weitl et al., 2004]. For instance, in order to reuse a learning object, the user first needs to find the object. Once found, the user evaluates through the metadata whether the object is suitable for her needs. The user then proceeds to download the selected object. The user evaluates the actual content of the object again to test if its contents suits her needs. The user adapts, if possible, the content according to her particular learning context. The user finally integrates the object with the rest of components. If all the steps in the chain are successful, the reuse takes place. If any of the steps is unsuccessful, the object is not reused.

According to probability theory [DeGroot, 1986], the probability of an event that is the consequence of a chain of consecutive sub-events, is equal to the product of the probabilities of the sub-event. Translated to our context, the probability of reuse is the product of the probability of success of each step in the reuse process. Equation 3.1 formalizes this model. In this equation, $P(R)$ is the probability of reuse, $P(E_i)$ is the probability of success of the ith event in the process. According to this equation, if the probability of one of the events is 0, the probability of reuse is also 0.

$$P(R) = \prod_{i=1}^{N} P(E_i) \tag{3.1}$$

Each step in the process could also be the result of the product of the probabilities of smaller events. For example, the probability of the user finding the object suitable for her needs could be modeled as the probability that the object is in a language she understands, that it is about the topic that she is learning and that it conforms to her learning style. Each step in the reuse process could be transformed also into the product of the probability of success of smaller steps. Equation 3.2 is the transformation of Equation 3.1 according to this idea. SE_k is the kth sub-event of the ith event. The result of this transformation is again a product of sub-events.

$$P(R) = \prod_{i=1}^{N} \prod_{k=1}^{M} P(SE_{i,k}) = \prod_{i=1}^{NM} P(SE_i) \tag{3.2}$$

3.4.2 Interpretation

The model of section 4.1 will be used to explain the results obtained in the quantitative analysis. To explain why the the distribution of the reuse among objects was Log-Normal in all the data sets, we will use the model as represented as Equation 3.2. Being the probabilities numerical quantities, we proceed to obtain the logarithm of both sides of the equation. The right hand side is converted now in the sum of the logarithms of the probability of the events (Equation 3.3). According to the Central Limit Theorem [DeGroot, 1986], the distribution of the sum of a large number (>30 for practical purposes) of random variables is Normal, regardless of the original distribution of the random variables. Therefore, if the number of sub-events (NM) is large, $log(P(R))$ is normally distributed. The variance σ^2 of the resulting normal distribution is equal to the sum of the variances of the multiplying random variables (Equation 3.4) [Montroll and Shlesinger, 1982].

$$\log(P(R)) = \sum_{i=1}^{NM} \log(P(SE_i)) \tag{3.3}$$

$$\sigma_R^2 = \sum_{i=1}^{NM} \sigma_{SE_i}^2 \tag{3.4}$$

By definition, a random variable is Log-Normal distributed if its logarithm is normally distributed. Given that we known from the previous deductions that $log(P(R))$ is normally distributed, the distribution of the reuse among objects $(P(R))$ is Log-Normal. This result explains why, even if different factors are involved in the reuse of the different data sets, all of them present a Log-Normal distribution. For this conclusion to hold, the number of factors affecting the success of reuse should be large. Theoretical studies about issues affecting reuse [Littlejohn, 2003] [McNaught, 2003] [Collis and Strijker, 2004] presented in the Introduction, suggest that this is indeed the case.

In the first analysis, the percentage of reuse of slide components is significantly lower than the percentage of reuse in other data sets. This lower percentage can be explained as the reduction of the probability of success in one of the steps that lead to reuse. According to Equation 3.1, if the probability of success of one of the steps is reduced n times, the total probability of reuse is also reduced n times. Given that, in the case of slide components, there are no search facilities and the object discovery should be manual, it can be assumed that the probability of the first event in the chain, that is finding the object, is lower. If we assume that finding an object to reuse browsing through other slides is twice as hard as using a search facility, the value of 11% is explained. A result that corroborates the interpretation that a search facility could increase the percentage of reuse can be found in [Verbert and Duval, 2007]. There, the use of a PowerPoint plug-in to search slide components triplicates the amount of objects reused. A similar line of argumentation can be followed to explain the relatively higher percentage of reuse obtained by the Web APIs set. The main factor that differentiates the creation of Mashups from Web APIs is that the same API can be used in several contexts (programming languages). If we consider that Web APIs have a slightly (0.5 times) higher probability to be adequate for the context of the user, than the other components, Equation 3.1 would offer an explanation for the higher percentage of reuse.

Finally, the conclusion that the popularity of an object does not correlate well with its reuse can be explained also with the proposed model. High popularity is an indicator that the object has been found several times, therefore, we can conclude that the probability to find it is high. However, Equation 3.1 is not a linear combination. It is enough that one of the large number of steps is unsuccessful to eliminate the possibility of reuse. For example, we consider a learning object made using Flash software. It can be popular because it conveys useful information and it is nicely designed. However, it is rarely reused because it is very difficult for a teacher to adapt it to a different context. The inverse assumption, that a highly reused object should be hihgly popular given that the user could find it through

diffent paths is also invalidated by the non-linearity of the model.

Most of the observations of section 3 can be explained by the very simple model that represents the reuse as a chain of successful events. However, the mechanisms that generate some of the results remain unknown. The similarity in the percentage of reuse in most of the measured data sets remains unexplained. While the model does not prevent that different reuse processes can generate similar amounts of reuse, it also does not explain why it is happening for most sets. Another result that defies explanation with the proposed model is the clustering of the distributions in two different groups. The reason why some of the reuse processes present a high logarithmic standard deviation (sdlog) remains an open question. These unexplained results do not invalidate the model, but suggest, as expected, that it is too simple to fully explain a complex process as the reuse of components. The development of a better model remains a very interesting topic for further research.

3.5 Implication of the Results

The results of the quantitative analysis and the model of reuse presented in previous sections can be used to answer the questions raised in the introduction. This section presents the answers to these questions and the implications that they have in our understanding of the process of reuse of learning objects. The reader should be aware that this results are preliminary because are based in a small amount of data. However, they already present interesting insights.

1. What percentage of learning objects is reused?
   The quantitative analysis seems to indicate that in common settings, the amount of learning objects reused is around 20%. While relatively low, this result is very encouraging for Learning Object supporters. It indicates that even without encouragement or the proper facilities, users do reuse a significant amount of learning materials. The multiplicative model also implicates that improving even one of the steps in the reuse chain, the others remaining equal, would improve the probability of reuse and, therefore, the amount of objects being reused. As mentioned above, Verbert and Duval, in [Verbert and Duval, 2007], empirically found that facilitating one of the steps, in this particular finding slide components, leads to a significant increase in the amount of reuse.

2. Is the amount of reuse in learning objects similar to other types of component reuse?
   The quantitative analysis suggests that the percentage of learning object reuse in a given collection or repository is similar to the percentage of reuse of other types of reusable components, such as images, software libraries and

Web APIs. This answer implies that learning objects are not intrinsically easier or harder to reuse than other types of components.

3. Does the granularity of the learning object affect its probability of reuse?
The theory of Learning Objects affirms that higher granularity leads to lower reusability. A naïve interpretation of the results contradicts this affirmation. The percentage of object reuse was similar regardless of the granularity of the object. Courses were even reused more often than slide components. Merging the theory with the empirical finding leads to a new interpretation of the role of granularity in the reuse of learning objects. This new interpretation involves also the granularity of the context of reuse as the determining factor. Objects that have a granularity immediately lower than the object being built are easier to reuse than objects with a much lower or higher granularity. For example, when building a course, it is easier to reuse whole lessons than reusing complete courses or individual images. Also, when building a curriculum, it is easier to reuse complete courses than to reuse another complete curriculum or individual lessons. Empirical support for this new interpretation can be found in [Verbert and Duval, 2007]. It was found that when building a slide presentation, the most reused component was by far individual slides. The reuse of text fragments and individual images represent just the 26% of the total reuse.

4. Is there a relation between the popularity of an object and its reuse?
The second quantitative analysis suggests that there is not a linear relationship between the popularity and the actual reuse of the object. The main implication of this result is that the success or failure of Learning Object provider cannot be measured only by the number of hits that their objects receive. If the model is used to extrapolate this result, knowing only the probability of any individual step in the reuse process is not enough to obtain an indication of the probability of actual reuse. Measuring the actual reuse, while more complicated, is the only certain way to assess the reusability of the objects.

5. What is the distribution of reuse among learning objects?
The distribution that best fits the reuse of learning objects is Log-Normal. The main implication of the finding is that the "Long Tail" effect [Anderson, 2006] applies to reuse. Few objects are reused heavily while most of the reused objects are reused just once. However, the volume of reuse in the tail is at least relatively as important as the volume of reuse in the head. According to this result, federating repositories in order to provide a wider selection of objects is a good strategy to foster reuse. Objects present in small repositories have a high probability of being reuse at least once if they are exposed to a wider universe of userss.

6. Is the distribution of reuse in learning objects similar to other types of component reuse?
   The Log-Normal distribution was the best-fit for both the Learning Object and reusable components data sets. Moreover, the parameters of the Log-Normal distribution were more similar between types than between the Learning Object sets. This similarity suggests the existence of an underlying process that governs different types of reuse. If this is the case, the research done in software and other components reuse can be used to gain insight in the reuse of learning objects and vice versa. The development of a global theory of reuse, of which our model is just a simplistic approximation, can be used to explain and improve the reuse process. A first step in this directon is presented by Markus in [Markus, 2001].

7. What model of reuse could explain the observed distributions?
   Representing the reuse as the result of a chain of successful events explains the appearance of the Log-Normal Distribution, the observed difference in the amount of reuse of Slide Presentations and the non-linear relation between popularity and reuse. A similar model has been theoretically presented before by Weit et al., in [Weitl et al., 2004]. However, this model cannot explain all the results. Also, there are still many unknowns in the definition of the model, most importantly which are the actual steps involved. This model can be used as a first tentative interpretation of reuse that could be improve, refined and even debunked with more empirical and theoretical research.

3.6 Conclusion

This chapter is the first quantitative analysis of the reuse of learning objects in real-world scenarios. Long-held ideas and beliefs about learning object reuse are tested against empirical data. The results obtained in the different analyses should force us to rethink some of those ideas. However, the analysis also shows that the theoretical and empirical developments made in other types of component reuse can be "reused" in our context to accelerate the understanding of the mechanisms behind learning object reuse.

The answers to the research questions obtained from the empirical data have several implications. Arguably the most important one is that the reuse of Learning Object is a process taking place in the real world, even without encouragement or the support of an adequate technological framework. However, it also can be concluded that the efforts made in Learning Objects Technologies to improve the reuse process through facilitating the different steps during the process can lead to increases in the amount of reuse.

Finally, the simple model presented in this chapter seems to be a good first approximation to formalize the mechanisms behind reuse. This interpretative model

enabled us to explain the findings and to draw conclusions that could be tested with further experimentation. Creating this kind of model, as well as improving the access to reuse information, would improve the testability of Learning Object research. The implications of new models could be tested and the efficiency of new systems could be compared with previous ones. This formalization in the field could help use to "get better at getting better" [Engelbart, 1995].

Together with chapter 2, these analyses provide a first look at the supply and demand behavior of the Learning Object Economy. We have found that inequality is the rule, more than the exception, in this market. Learning Objects are produced and consumed according to heavy-tail distributions and the models presented suggests that several factors interplay at the micro-level to produce the observed patterns. While much more quantitative and qualitative analysis are needed in order to obtain a full picture of the workings of the Learning Object Economy, the answers found in these two first chapters sketch a preliminary, but revealing, approximation.

The following two chapters (4 and 5) will take an engineering approach to build a set of metrics to improve the working of the process deeply related to supply and demand: labeling and selection. These metrics are evaluated as mechanisms to help the final user to interact better with learning objects.

Chapter 4

Metadata Quality Metrics for Learning Objects

In the previous two chapters, the characteristics of the publication and reuse processes of Learning Objects were measured to gain insight on how those processes work. The following three chapters, on the other hand, are focused on how the measurement of the characteristics and usage of learning object, in the form of metrics, can be used to improve those processes. This chapter will propose and evaluate metrics for metadata quality while chapter 5 will concentrate on relevance ranking metrics. Finally, chapter 6 will discuss how those metrics can be implemented in software.

4.1 Introduction

The quality of metadata instances stored in digital repositories is perceived as an important issue for their operation [Barton et al., 2003] [Beall, 2005] and interoperability [Liu et al., 2001] [Stvilia et al., 2006].

The main functionality of a digital repository, to provide access to resources, can be severely affected by the quality of the metadata. For example, a learning resource indexed with the title "Lesson 1 - Course CS20", without any description or keywords will rarely appear in a search for materials about "Introduction to Java Programming", even if the described resource is, indeed, a good introductory text to Java. The resource will just be part of the repository but will never be retrieved in relevant searches.

Secondary functions of metadata in a digital repository can also be heavily compromised by low metadata quality. For example, the metadata instance should contain enough information, so that the user can obtain a good idea of the purpose and content of the described resource without directly accessing the resource. For

example, incorrect or out-dated information about the URI of the resource could prevent the user to access the object. Also, the effectiveness of a distributed search could be degraded even if just one of the connected repositories contains mainly low quality metadata instances. Consequently, the usefulness of a digital repository is strongly correlated to the quality of the metadata that describe its resources.

Due to its importance, metadata quality assurance has always been an integral part of resource cataloging [Thomas, 1996]. Nonetheless, most implementations of digital repositories have taken a relaxed approach to metadata quality assurance. For example, these implementations rely on the assumption that metadata were created by an expert in the field or a professional cataloguer and, as such, should have an acceptable degree of quality. In reality, experts in a given field are not necessarily experts in metadata creation, and hiring professional indexers to do the cataloging of resources is usually not feasible for most repositories due to scalability issues and the costs involved.

As repositories grow (through automatic metadata generation [Cardinaels et al., 2005] or resource decomposition [Verbert et al., 2005]) and merge (through search federation [Simon et al., 2005] or metadata harvesting [Van de Sompel et al., 2004]), quality issues become more apparent. This problem has led to the adaptation of techniques developed to review physical library instances to address the quality of digital metadata. Also, new techniques that take advantage of the ability of computers to perform repetitive calculations have been developed to assure a minimum level of quality. A review of earlier work on metadata quality evaluation for digital repositories reveals these two general approaches:

- *Manual Quality Evaluation.* The majority of approaches (see Table 4.1) manually review a statistically significant sample of metadata instances against a predefined set of quality parameters, similar to sampling techniques used for quality assurance of library cataloguing [Chapman and Massey, 2002]. Human evaluations are averaged and an estimation of metadata quality in the repository is obtained. Until now, these methods are the most meaningful way to measure the metadata quality in a digital repository. However, they have three main disadvantages: 1) the manual quality estimation is only valid at sampling time. If a considerable amount of new resources is inserted in the repository, the assessment could be no longer accurate and the estimation must be redone. 2) only the average quality can be inferred with these methods. The quality of individual metadata instances can only be obtained for those instances contained in the sample. 3) obtaining the quality estimation in this way is costly. Human experts should review a number of objects that, due to the growth of repositories, is always increasing. Dushay and Hillman, in [Dushay and Hillmann, 2003], propose the use of visualization tools to help metadata experts in their task, but it is still mainly a manual activity.

Because of this last disadvantage, manual review of metadata quality is mainly a research activity with few practical implications in the functionality or performance of the digital repository.

- *Simple Statistical Quality Evaluation.* From the studies we analyzed, three follow a different approach (see Table 4.1). These studies collect statistical information from all the metadata instances in the repository to obtain an estimation of their quality. Hughes, in [Hughes, 2004], calculates simple automatic metrics (completeness, vocabulary use, etc.) at repository level for each of the repositories in the Open Language Archive [Hughes and Kamat, 2005]. Bui and Park [Bui and Park, 2006] perform a wide study in which more than one million instances were reviewed for completeness. Najjar et al. [Najjar et al., 2003] compare the metadata fields that are produced with the metadata fields that are used in searches. This comparison provides a simple estimation of the quality of the metadata in the ARIADNE [Duval et al., 2001] repository. All these studies automatically obtained a basic estimation of the quality of each individual metadata instance without the cost involved in manual quality review. However, they do not provide a similar level of "meaningfulness" as a human generated estimation. They are mainly used as "interesting" information about the repository without any other real application.

An ideal measurement of metadata quality for fast-growing repositories should have two characteristics: to be automatically calculated for each metadata instance inserted in the repository (scalability) and to provide a useful measurement of the quality (meaningfulness). None of the approaches reviewed could claim to be scalable and meaningful at the same time. Manual evaluation are meaningful but not scalable. Simple Statistics are scalable, but are not meaningful. The main contribution of this chapter is the description and evaluation of a set of metadata metrics based on the same quality parameters used by human reviewers but with the difference that they can be calculated automatically. These metrics can be used to build tools for any kind of digital repository and can provide scalable and meaningful metadata quality assurance. These kind of automated quality assurance is key to enable a true Learning Object Economy where millions of objects are published and automatically labelled throughout their lifetimes.

The structure of this chapter is as follows: A review is conducted in section 2 to select a framework to measure metadata quality. Based on the selected framework, ten quality metrics are described in section 3. Three validation studies are conducted in section 4 to evaluate 1) the degree of correlation between the proposed metrics and human quality review, 2) the discriminatory power of the metrics and 3) the effectiveness of the metrics as low quality instances filters. The implications of the findings are also discussed in detail in section 4. Section 5 describes possi-

Table 4.1: Review of different quality evaluation studies

Study	**Approach**	**# of instances**	**Main focus of evaluation**
[Greenberg et al., 2001]	Manual	11	Quality of non-expert metadata
[Shreeves et al., 2005]	Manual	140	Overall quality of instances
[Stvilia et al., 2006]	Manual	150	Identify quality problems
[Wilson, 2007]	Manual	100	Quality of non-expert metadata
[Moen et al., 1998]	Manual	80	Overall quality of instances
[Hughes, 2004]	Statistical	27,000	Completeness of instances
[Najjar et al., 2004]	Statistical	3,700	Usage of the metadata standard
[Bui and Park, 2006]	Statistical	1,040,034	Completeness of instances

ble applications of the quality metrics. The chapter closes with related work and conclusions.

4.2 Measuring Metadata Quality

Despite the wide agreement on the need to produce high quality metadata, there is less consensus on what high quality means and even less on how it should be measured. This chapter will consider quality as the measure of fitness for a task [Ede, 1995]. The tasks metadata should enable in a digital repository are to help the user to find, identify, select and obtain resources [O'Neill, 2002]. The quality of the metadata will be directly proportional to how much it facilitates those tasks.

Measurements of the quality of the metadata instance do not address the quality of the metadata schema or the set of values that fields on the schema could take (we call these sets vocabularies). These measurements should be schema-agnostic, when possible. They also do not evaluate the quality of the resources themselves. This chapter will provide metrics to estimate the quality of the information entered manually by indexers, generated automatically or a mixture of both.

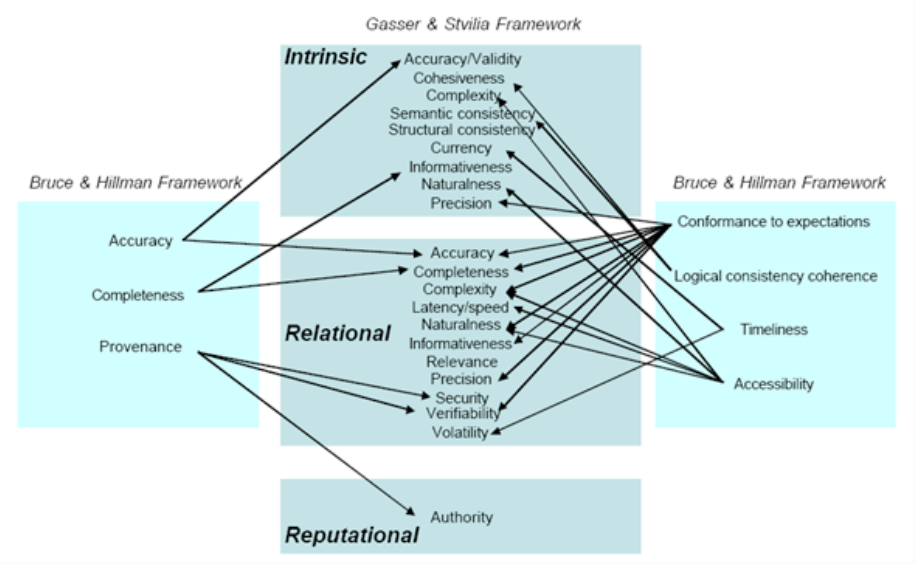
In order to reduce subjectivity in the assessment of information quality, several researchers have developed quality evaluation frameworks. These frameworks

define parameters that indicate whether information should be considered of high quality. Different frameworks vary widely in their scope and goals. Some have been inspired by the Total Quality Management paradigm [Strong et al., 1997]. Others are used in the field of text document evaluation, especially of Web documents [Zhu and Gauch, 2000]. Particularly interesting for our work, because they are focused on metadata quality, are the frameworks that have evolved from the research on library catalogs [Ede, 1995].

While no consensus has been reached on conceptual or operational definitions of metadata quality, there are three main references that could guide this kind of evaluation. We rely on these here as they summarize the recommendations made in previous information quality frameworks and eliminate redundant or overly specific quality parameters. Moen et al. [Moen et al., 1998] identify 23 quality parameters. However, some of these parameters (ease of use, ease of creation, protocols, etc) are more focused on the metadata schema standard or metadata generation tools. Given that the metrics should be schema-agnostic and measure only the quality of metadata instance, [Moen et al., 1998] is not considered as our base framework. Stvilia et al. [Stvilia et al., 2007] use most of Moen's parameters (excluding those not related with metadata quality), add several more, and group them in three dimensions of Information Quality (IQ): Intrinsic IQ, Relational/Contextual IQ and Reputational IQ. Some of the parameters (accuracy, naturalness, precision, etc) are present in more than one dimension. The Stvilia et al. framework describes 32 parameters in total. Bruce & Hillman [Bruce and Hillmann, 2004], based on previous Information Quality research, condense many of the quality parameters in order to improve their applicability. They describe seven general characteristics of metadata quality: completeness, accuracy, provenance, conformance to expectations, logical consistency and coherence, timeliness, and accessibility. A relation between the frameworks of Bruce & Hillman and Stvilia et al. is proposed in [Shreeves et al., 2005] and it is summarized in Figure 4.1.

This analysis will use the Bruce & Hillman framework because its seven parameters are easy to understand by human reviewers and also because they capture all the dimensions of quality proposed in other frameworks. The compactness will also help to operationalize the measurement of quality in a set of automatically calculated metrics. Another advantage of this choice is that this framework is deeply rooted on well-known Information Quality parameters. There exists parallel research on how to convert these parameters into metrics for quality assurance of other types of information (for instance Web Pages [Zhu and Gauch, 2000]). However, the Bruce & Hillman framework (also Stvilia et al.) are designed with a static metadata instance in mind. These frameworks are more appropriate for library purposes than for the dynamic metadata instances of digital libraries. Given that to the knowledge of the author there are no frameworks to describe the quality of dynamic metadata, Bruce & Hillman will be used as a first approach, adapting the quality characteristics when needed for the particularities of dynamic instances.

Figure 4.1: Mapping between the Bruce & Hillman and the Stvilia et al. frameworks. (Taken from [Shreeves et al., 2005])



These adaptations are presented in section 3, where the metrics are introduced.

To improve the readability of this chapter, a summary of the framework developed by Bruce & Hillman is presented. This framework defines seven parameters to measure the quality of metadata. These parameters are:

- *Completeness:* A metadata instance should describe the resource as fully as possible. Also, the metadata fields should be filled in for the majority of the resource population in order to make them useful for any kind of service. While this definition is most certainly based in static library instance view of metadata, it can be use to measure how much information is available about the resource.

- *Accuracy:* The information provided about the resource in the metadata instance should be as correct as possible. Typographical errors, as well as factual errors, affect this quality dimension. However, estimating the correctness of a value is in not always a "right"/"wrong" choice. There are metadata fields that should receive a more subjective judgement. For example, while it is easy to determine whether the file size or format are correct or not, the correctness of the title, description or difficulty of an

object has much more levels that are highly dependent of the perception of the reviewer.

- *Conformance to Expectations:* The degree to which metadata fulfills the requirements of a given community of users for a given task could be considered as a major dimension of the quality of a metadata instance. If the information stored in the metadata helps a community of practice to find, identify, select and obtain resources without a major shift in their workflow it could be considered to conform to the expectations of the community. According to the definition of quality ("fitness for purpose") used in this chapter, this is one of the most important quality characteristics.

- *Logical Consistency and Coherence:* Metadata should be consistent with standard definitions and concepts used in the domain. The information contained in the metadata should also have internal coherence, that means that all the fields describe the same resource.

- *Accessibility:* Metadata that cannot be read or understood have no value. If the metadata are meant for automated processing, for example GPS location, the main problem is physical accessibility (incompatible formats or broken links). If the metadata are meant for human consumption, for example Description, the main problem is cognitive accessibility (metadata is too difficult to understand). These two different dimensions should be combined to estimate how easy is to access and understand the information present in the metadata.

- *Timeliness:* Metadata should change whenever the described object changes (currency). Also, a complete metadata instance should be available by the time the object is inserted in the repository (lag). The lag description made by Bruce & Hillman, however, is focused in a static view of metadata. In a digital library approach, the metadata about a resource is always increasing which each new use of the resource. The lag, under this viewpoint, can be considered as the time that it takes for the metadata to describe the object well enough to find it using the search engine provided in the repository.

- *Provenance:* The source of the metadata can be another factor to determine its quality. Knowledge about who created the instance, the level of expertise of the indexer, what methodologies were followed at indexing time and what transformations the metadata has passed through, could provide insight into the quality of the instance.

For a discussion on the rationale behind these parameters, as well as for a thoughtful analysis of what "metadata quality" means, we invite the reader to consult [Bruce and Hillmann, 2004]. The following Section will present calculations (metrics) that could provide a low cost estimation of these quality parameters.

4.3 Quality Metrics for Metadata in Digital Repositories

Bruce & Hillman [Bruce and Hillmann, 2004] devised their framework to guide human reviewers. The parameters, being domain-independent, are necessarily abstract. This level of abstraction could be easily managed by metadata experts, but presents a problem for the automatic estimation of quality. This section will describe a set of calculations that work over the existing metadata information and easy-to-collect contextual data in order to "instantiate" the quality parameters into a set of quality metrics. The objective of these metrics is to provide a meaningful estimate of the quality of each metadata instance for a given community of practice in a scalable way.

The proposed metrics are standard-agnostic and can be used for a wide range of digital repositories such as digital libraries, learning object repositories or museum catalogs. These metrics are easy to implement in real environments and fast enough to be applied to each metadata instance at indexing or transformation time. An evaluation of their scalability is presented in chapter 6.

The metrics calculations are also independent of the specific community of practice being served. However, the parameters needed to initialize the calculations heavily depends of the particularities of each group of users, because quality itself is context dependent. Also, the proposed metrics are mainly designed to work over text and numbers. Given that most metadata is some form of alphanumeric value, this metrics could be applied "as is" for the majority of metadata formats currently in use. However, if multimedia information is added to the metadata record, for example, the thumbnail of an image, new approaches based on Multimedia Information Retrieval should be used to extract a similar level of information from those multimedia fields.

4.3.1 Completeness Metrics

As described in Section 2, Completeness is the degree to which the metadata instance contains all the information needed to have a comprehensive representation of the described resource. While easy to understand for static, library records, this concept is less clear for dynamic metadata instances, where new information is added each time that the resource is used. In the case of dynamic metadata, there is certain information, that, due its nature, should be present to enable the services of the digital library. For example, some digital libraries rely on the title of the object to present it in a list to the user. If the metadata do not contain a title, the quality of the metadata decrease. On the other hand, while reviews and ratings collected through the lifetime of the resource are highly valuable, their existence or not do not prevent the digital library searching facilities from present it to the user. This metric should consider the former types of metadata information

to estimate its completeness.

The most direct approach to measure completeness of a instance is to use the number of filled in metadata fields as a proxy. Each metadata standard, for example Dublin Core (DC) [DCMI, 1995] or Learning Object Metadata (LOM) [IEEE, 2002], defines a number of possible fields (15 for DC, 58 for LOM). In some cases there can be more than one instance of the fields. A basic completeness metric will be to count the number of fields in each metadata instance that contain a no-null value. In the case of multi-valued fields, the field is considered complete if at least one instance exists. Equation 4.1 expresses how this metric can be determined.

$$Qcomp = \frac{\sum_{i=1}^{N} P(i)}{N} \tag{4.1}$$

Where $P(i)$ is 1 if the ith field has a no-null value, 0 otherwise. N is the number of fields defined in the metadata standard.

The maximum value of this metric is 1 (in the case all the fields contain information) and the minimum value is 0 (an empty instance). For example, if a LOM instance has 40 fields filled in, its *Qcomp* value will be $40/58 = 0.69$.

While straightforward, the simple completeness metric does not reflect how humans measure the completeness of a instance. Not all data elements are relevant for all resources. Moreover, not all metadata elements are equally relevant to all contexts. For example, a human expert may assign a higher degree of completeness to a metadata instance that has a title, but lacks publication date than vice versa. To account for this phenomenon, a weighting factor could multiply the presence or absence of a metadata field. This factor represents the importance of the field. This weighting factor can easily be included in the calculation of the completeness metric as shown in the Equation 4.2.

$$Qwcomp = \frac{\sum_{i=1}^{N} \alpha_i * P(i)}{\sum_{i=1}^{N} \alpha_i} \tag{4.2}$$

Where α_i is the relative importance of the ith field.

The maximum value for *Qwcomp* will be 1 (all fields with importance different from 0 are filled) and a minimum value of 0 (all fields with importance different from 0 are empty).

The α should be any positive value that represent the importance (or relevance) of the metadata field for some context or task. This implies that each community of practice could have a different set of weighting factors to calculate the weighted completeness for different kinds of tasks. For example, α_i could represent the

number of times field i has been used in queries to a given repository [Najjar et al., 2004].

Consider a metadata standard that has 4 fields: Title, Description, Author Name and Publication Date. Consider also that after 5000 queries to the repository, Title has been used 5000 times, Description 2500 times, Author Name 1000 times and Publication Date 0 times, so $\alpha_1 = 5000$, $\alpha_2 = 2500$, $\alpha_3 = 1000$ and $\alpha_4 = 0$. Table 4.2 shows the *Qwcomp* calculation for different instances and its contrast with Qcomp. The presence of Publication Date is not relevant for the Qwcomp as it is never used in the queries. Title, on the other hand, is the most important field, and its sole presence corresponds to more than half the completeness value.

Table 4.2: Example of the calculation of Qwcomp for a 4-field metadata instances

instance				**Qwcomp**	**Qcomp**	**Qwcomp**
Title	*Desc.*	*Author*	*Date*			
Yes	No	No	No	(5000)/8500	0.25	0.59
No	No	Yes	Yes	(1000+0)/8500	0.50	0.12
Yes	Yes	Yes	No	(5000+2500+1000)/8500	0.75	1.0

Alternatively, when measuring the completeness for the selection task, α_i could represent the score that the ith field obtained in an experiment to measure the amount of time the user expend reading each field while selecting an appropriate resource. Obtaining the alpha values from the analysis of user interaction with the digital library has the added advantage of adapting the completeness quality estimation to the changing behavior of the user community. Each time that new importance values are generated (for example, query information is collected or usability studies are performed) a more refined estimation could be obtained.

The algorithm to calculate the Completeness metric needs only access to the metadata repository. For the Weighted Completeness, also a table containing the pre-calculated α values should be available.

4.3.2 Accuracy Metrics

Accuracy is the degree to which metadata values are "correct", i.e. how well they describe the object. For objective information like file size or document format correctness could be a binary value, either "right" or "wrong". In the case of subjective information, it is a more complex spectrum with intermediate values (e.g.: a title of a picture, or the description of the content of a document). In general, correctness and, therefore accuracy, could be considered as the semantic distance

between the information a user could extract from the metadata instance and the information the same user could obtain from the resource itself and its context. The shorter the distance, the higher the accuracy of the metadata instance.

While humans can assess with relative ease the accuracy of a metadata instance, computers require complex artificial intelligence algorithms to simulate the same level of understanding. Nevertheless, there exists accuracy metrics that are easy to calculate, proposed in quality evaluations presented in [Hughes and Kamat, 2005] and [Moen et al., 1998]. These metrics establish the number of easy-to-spot errors present in metadata instances. Typical examples of this type of errors are broken links, inaccurate technical properties of the digital resource, such as size or format, typographical errors in the text fields, to name a few.

This chapter proposes a more complex and meaningful approach to calculate the semantic difference between the metadata instance and resources that contain textual information. Using the metadata and the resource itself helps to provide a better estimation of the accuracy of the record that just counting the number of errors in the metadata. This method builds upon Vector Space model techniques used in Information Retrieval to calculate the distance between texts [Salton et al., 1975]. A multidimensional space is constructed in which each word present in the text of the original resource defines a dimension. The number of times a word appears in the text is considered as the value of the text in that word-dimension. Following those definitions, a vector is created for the text contained in the original resource and the text present in the textual fields of the metadata instance (e.g. title, description, keywords, etc.). Finally, a vector distance metric, such as the cosine distance, is applied to find the semantic distance between both texts. In Equation 4.3 the cosine distance formula is presented.

$$Qaccu = \frac{\sum_{i=1}^{N} tfresource_i * tfmetadata_i}{\sqrt{\sum_{i=1}^{N} tfresource_i^2 * \sum_{i=1}^{N} tfmetadata_i^2}} \tag{4.3}$$

Where $tfresource_i$ and $tfmetadata_i$, are the relative frequency of the ith word in the text content of both the resource and the metadata. N is the total number of different words in both texts.

The minimum value (lower quality) is 0, meaning that the two texts have no words in common. The maximum value (higher quality) is 1, meaning that all the words from one of the texts appear in the other.

Table 4.3, as an example, presents two metadata instances with an excerpt from the text from their respective described objects. The resulting *Qaccu* value is also presented for each instance. In the first example, the word "SEPHYR" appears in the metadata description, but it cannot be found in the document. The other word in the title, "METHODOLOGY", is matched against the same and similar

Table 4.3: Example of the Qaccu values for two metadata instances

	Qaccu
Resource 1	**0.56**
Metadata Text (title+description): *SEPHYR METHODOLOGY*	
Extract of Resource Text (Word document): *Methodology of Pedagogic Segmentation Extract from the doctoral thesis by Miss M. Wentland Forte entitled: "Knowledge domain modelling and conceptual orientation in a pedagogic hypertext" What is a concept? Taking it at the level of the spontaneous mental processes (unorganised and non-verbalised), we can say that we are dealing with the realm of ideas. As soon as an idea can be named, it becomes a concept..*	
Resource 2	**0.96**
Metadata Text (title+description): *Searching for the Future of Metadata - Looking in the wrong places for the wrong things? Keynote at DC2004 conference*	
Extract of Resource Text (PowerPoint presentation): *Searching for the Future of Metadata Looking in the wrong places for the wrong things? by: Wayne Hodgins wayne.hodgins@autodesk.com Looking in the Wrong Places? Searching Helping Remembering. Looking in the Right Place? Looking in the Right Place Looking where the light is!! A few words about LEARNING Vision for learning*	

words in the resource text. Given that the dimensionality of the metadata text is 2, the result is approximately 0.5. In the second example, most of the words that appear in the title and description also appear in the document itself. The result of the *Qaccu* metric approaches 1.

The first example on Table 4.3 is also a good demonstration of how this metric could fail for some especial cases in real world applications. While the word "SEPHYR" does not appear at all in the text of the document itself, the document is indeed about the "SEPHYR METHODOLOGY". To minimize the impact of this type of omissions, a method to detect synonyms or words with close semantic relation can be used. One of the most successful is Latent Semantic Analysis (LSA) [Landauer et al., 1998]. This algorithm can be used to reduce the dimensionality of the space before the distance calculation. With the reduction of dimensionality, the noise introduced by semantic similar words is reduced.

To implement this metric, the LSA algorithm needs to be trained with corpora taken from the text resources present in the repository. Afterwards, the

lower-dimensional matrices resulting from the Single Value Decomposition (SVD) [Landauer et al., 1998] calculation could be used to compute the semantic distance between any arbitrary pair of texts.

4.3.3 Conformance to Expectation Metrics

Conformance to expectations measures the degree to which the metadata instance fulfills the requirements of a given community of users for a given task. As mentioned previously, metadata in digital repositories is mainly used to find, identify, select and obtain resources. The usefulness of a metadata instance for the first three tasks (find, identify and select) depends heavily on the amount of useful (unique) information contained in the instance. instances with non-common words are easier to find. Users can differentiate resource more easily if their metadata instances are not similar. Users can make better selections if the instances provide better descriptions of the resource. A method that could measure the amount of unique information in the metadata instance can be used to estimate its conformance to the expectation of a community. The method propose in this chapter is the calculation of the information content of the metadata instance.

In Information Theory, the concept of entropy is used as a measure of the information content of a message [Shannon and Weaver, 1963]. Entropy is the negative logarithm of the probability of the message. Intuitively, the more probable a message is, the less information it carries. For example, if all the metadata instances in a repository have the field "language" set as "English", a new instance with that field set to "English" carries few information, meaning that it does not help to distinguish this particular resource from the rest. On the other hand, if the new instance has the "language" field set to "Spanish", it is highly improbable (based on the content of the repository) and this value helps to differentiate this new resource from the others.

The information content of categorical fields (those that can only take a value from a defined and finite vocabulary) can be easily calculated using the entropy method described in the previous paragraph. To obtain the Categorical Information Content for a given instance, the entropy values of the categorical fields can be averaged. This calculation is presented in Equations 4.4 and 4.5.

$$infoContent(categorical_field) = -\log(f(value)) \tag{4.4}$$

Where $f(value)$ is the relative frequency of $value$ in the categorical field for all the current instances in the repository. This relative frequency is equivalent to the probability of $value$.

$$Qcinfo = \frac{\sum_{i=1}^{N} infoContent(field_i)}{N} \tag{4.5}$$

Where N is the number of categorical fields.

Table 4.4 shows the Qcinfo calculation for some categorical fields of real metadata instances in the ARIADNE repository [Duval et al., 2001]. The Qcinfo is calculated by averaging only the entropy values of two fields: "Main Discipline" and "Difficulty Level". From these two categorical fields, Resource 1 seems to be an average instance, similar to the majority of the other instances in the repository, therefore it has a low Information Content (Qcinfo). On the other hand, Resource 2 is highly atypical, leading to a high Information Content value.

Table 4.4: Example of the calculation of Qcinfo for 2 metadata instances

Field	Value	f(value)	Entropy	Qcinfo
Resource 1				
Main Discipline	Computer Science	1220/4460	0.59	0.31
Difficulty Level	Medium	4124/4460	0.03	
Resource 2				
Main Discipline	Mechanical Engineering	314/4460	1.15	1.36
Difficulty Level	High	120/4460	1.57	

In order to normalize the Information Content value, so it will vary from a minimum of 0 (lowest quality) to a maximum of 1(highest quality), the formula in Equation 4.4 should be changed as to the one presented in Equation 4.6.

$$infoContent(categorical_field) = 1 - \frac{\log(times(value))}{\log(n)} \tag{4.6}$$

Where $times(value)$ is the number of times that the value is present in that categorical field in the whole repository. n is the total number of instances in the respository. When $times(value)$ is 0 (the value is not present in the repository), the infoContent is 1. On the other hand, if $times(value)$ is equal to n (all the instances have the same value), the infoContent is 0.

For free text fields the Information Content calculation is not as straight forward as for categorical fields. Each word in the text can be considered as a possible carrier of information. To calculate the total information content of textual fields we have to estimate the contribution of every word in each field. In the field of Information Retrieval, the importance of a word is calculated with the Term Frequency-Inverse Document Frequency (TFIDF) [Sparck Jones, 1972] [Salton and Buckley, 1988] value. The importance of a word in a document is directly proportional to how frequently that word appears in the document and inversely proportional to how frequently documents in the corpora contain that word. More concretely, the frequency in which a word appears in the document

is multiplied by the negative log of the relative frequency in which that word appears in all the documents in the corpora. This calculation could be considered as a weighted entropy measurement for each word. To get the Information Content of the text field, the TFIDF value of each word is added. The Information Content of a instance can be calculated by adding the Information Content of its text fields. Equation 4.7 provides a description of the Information Content calculation.

$$infoContent(freetext_field) = \sum_{i=1}^{N} tf(word_i) * \log\left(\frac{1}{df(word_i)}\right) \tag{4.7}$$

Where $tf(word_i)$ is the term frequency of the ith word, $df(word_i)$ is the document frequency of the ith word. N is the number of words in the field.

A common method to normalize TFIDF values is to divide the sum of the TFIDF values by the total number of words in the text. This division gives a measure of the information density. However, the Qtinfo metric attempts to estimate the total information content of the instance, not its density. A way to reduce the range of the Information Content value, while preserving the length of the text as a component, is to obtain its logarithm. Therefore, the final formula for the Qtinfo is the logarithm of the sum of the Information Content of the textual fields (Equation 4.8).

$$Qtinfo = \log\left(\sum_{i=1}^{N} infoContent(field_i)\right) \tag{4.8}$$

Where $field_i$ is the ith textual field. N is the number of textual fields in the metadata standard.

Intuitively, texts composed mainly of common words in a language (for example: “the”, “a”, “for”, etc.) and words that are common in a given repository (for example, for a learning object repository: “course”, “lesson”, “material”, etc.) carry less information to identify the resource than more specialized words. Also, longer texts contain more information than shorter texts. Table 4.5 present the calculation of the Textual Information Content for four different texts extracted from metadata instances of the ARIADNE Repository. Although Resource 1 has fewer words (17), it obtains a slightly higher value than Resource 2 (19 words). The reason for the score difference is that the words “Metadata” and “Searching” are quite common in the ARIADNE repository. However, when one of the texts has considerably more words than other the difference is clearly represented in the Information Content values. This is the case of Resource 4 (291 words) which obtain a higher Qtinfo score than Resource 3 (34 words).

The information needed to calculate Qcinfo and Qtinfo could be extracted from the target repository. Pre-calculated probabilities for the categorical fields, such as Document Frequencies (DF) for existing words, can be stored in temporal

database tables that are refreshed in periodical intervals. With this tables, the Qcinfo and Qtinfo calculation will only involve few mathematical operations.

Table 4.5: Example of the calculation of Qtinfo for text of different words and lenghts

Resource	**Text**	**infoContent**	**Qtinfo**
1	Gap Report Identified consequences of the developments of the other WPs for design of knowledge work management.	85	1.93
2	Searching for the Future of Metadata - Looking in the wrong places for the wrong things? Keynote at DC2004 conference	83	1.91
3	Traveling salesman This is a quick implementation of the traveling salesman problem, written in Java.It shows a frame with the execution of the backtracking algorithm for some citiesusage: java Practicum2 'SHORTEST'—'ANY'	162	2.20
4	Control of the transfer channel Control of the transfer channel: First step towards a competence map This deliverable is based on the performance indicators developed in D8.4. Controlling the activities via the transfer channels by performance indicators is the main issue outlined in this deliverable. The monitoring process is focussed on major activities made by visitors... [235 WORDS MORE]	1557	3.19

4.3.4 Consistency & Coherence Metrics

Consistency

The logical consistency of a metadata instance can be estimated as the degree to which it matches the metadata standard definition. There are three ways in which this consistency can be broken: 1) instances include fields not defined in the standard or do not include fields that the community sets as mandatory. 2)

Table 4.6: Recommendations for values in the LOM Standard (v.1.0)

Field 1	Field 2	Recommendation
Structure	Aggregation Level	Structure=atomic :: Aggregation Level=1
Interactivity Type	Iteractivity Level	Interactivity Type=active :: high values of Interactivity Level
Semantic Densisty	Difficulty	high values of Semantic Density :: high values of Difficulty
Resource Type	Interactivity Level	Resource type=narrative text :: Interactivity Level=expositive
Context	Typical Age Range	Context=higher education :: Age Range at least 17 years

Categorical fields, that should only contain values from a fixed list, are filled with a non sanctioned value 3) The combination of values in categorical fields is not recommended by the standard definition. In the case of isolated repositories, problems of type 1 and 2 are heavily reduced by the use of a common indexing tool. For distributed or aggregated repositories, problems of type 1 and 2 should be expected as the result of different indexing practices [Shreeves et al., 2005]. Problems of type 3 are more subtle and affect all types of repositories. They can be directly associated with violation of consistency rules at indexing time. An example of such rules is defined in the LOM Standard (v.1.0): If the value of the "Structure" field is set to "atomic", the "Aggregation Level" field should be set as "1 (Raw media)". Other Structure values could be paired with any other value of Aggregation Level except 1. Table 4.6 presents more of these rules for the LOM standard.

An estimation of the Consistency of the metadata instance should be inversely proportional to the number of problems found in the instance. Firstly, the amount of possible problems of type 1, 2 or 3 is obtained by the examination of the metadata standard and indexing rules of a community. Secondly, the number of problems present in the instance is counted. The number of problems of type 1 or 2 can be calculated with a simple validation parser. For problems of type 3, a set of "If...Then" rules could be used instead. Finally, the Consistency metric will be equal to 1 minus the average of fraction of problems found, for each type of problem. Equations 4.9 and 4.10 present the calculation for the type 3 consistency. The minimum value for the Consistency metric is 0 (all possible errors were made) and the maximum value is 1 (there were no consistency problems).

$$brokeRule_i = \begin{cases} 0; \text{ if instance complies with } i\text{th rule} \\ 1; \text{ otherwise} \end{cases} \tag{4.9}$$

$$Qcons = 1 - \frac{\sum_{i=1}^{N} brokeRule_i}{N} \tag{4.10}$$

Where N is the number of rules in the metadata standard or community of use.

Coherence

The Coherence of the instance, on the other hand, is more related to the degree to which all the fields describe the same object in a similar way. The Coherence metric can be estimated analyzing the correlation between text fields. A procedure similar to the one used in the Accuracy metric (Section 3.2) can be implemented. The semantic distance is calculated between the different free text fields. The average semantic distance is used as a measure of the coherence quality (Equations 4.11 and 4.12). This method is commonly used to establish the internal coherence of a text piece [Foltz et al., 1998]. To cope with synonyms, a LSA algorithm could be applied before the semantic distance is calculated.

$$\text{distance}(field1, field2) = \frac{\sum_{i=1}^{N} tfidf_{i,field1} * tfidf_{i,field2}}{\sqrt{\sum_{i=1}^{N} tfidf_{i,field1}^2 * \sum_{i=1}^{N} tfidf_{i,field2}^2}} \tag{4.11}$$

Where $tfidf_{i,field}$ is the Term Frequency Inverse Document Frequency of the ith word in the textual $field$. N is the total number of different words in the field 1 and 2.

$$Qcoh = \frac{\sum_{i}^{N}\sum_{j}^{N} \begin{cases} \text{distance}(field_i, field_j); \text{ if i} < \text{ j} \\ 0; \text{ otherwise} \end{cases}}{\frac{N*(N-1)}{2}} \tag{4.12}$$

Where N is the number of textual fields that describe the object.

Table 4.7 presents the calculation of the Coherence metric for the Title and Descriptions belonging to Learning Objects in the ARIADNE Repository. If the Title and Description have semantically similar words, the Qcoh is close to 1, otherwise, as in the case of Resource 2, where there are no words in common, the QCoh approaches 0.

The second example in Table 4.7 presents a possible problem of this metric in real world scenarios. While the Title and Description are completely different, they are indeed describing the same resource. This problem will make this metric not very informative for individual instances. However, the value of Qcoh can provide some information if applied to a whole repository. A low value of Qcoh for a considerable number of instances could be the signal of poor titles or descriptions.

Table 4.7: Example of Qcoh calculation for 2 metadata instances

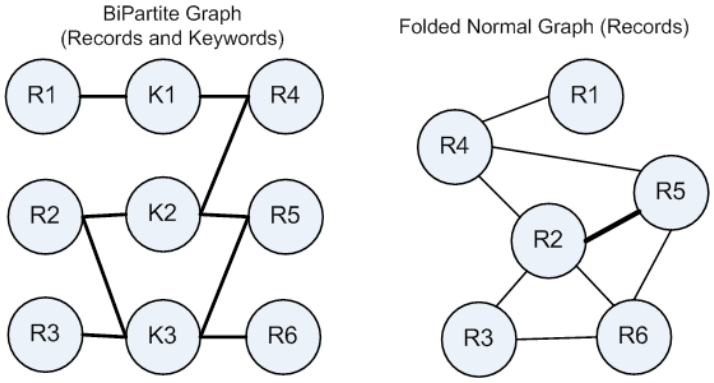
Field	**Text**	**Qcoh**
Resource 1		
Title	Infrastructure for (semi-)automatic generation of Learning Object Metadata	0.95
Description	The Month 6 deliverable for D4.1 is a "functional prototype" for an "infrastructure that supports (semi-) automatic generation of LOM metadata". As such, this deliverable consists of software: this report documents the design and status of the software. The actual software is also deposited on http://ariadne.cs.kuleuven.ac.be/amg	
Resource 2		
Title	Searching for the Future of Metadata - Looking in the wrong places for the wrong things?	0.0
Description	Keynote at DC2004 conference	

4.3.5 Accessibility Metrics

Accessibility implies the level to which a metadata instance can be found and later understood. It should not be confused with the more common meaning of accessibility, "design for all". One way to estimate the logical accessibility or "findability" of a metadata instance could be to count the number of times that the instance has been retrieved during searches. However, that value not only measures the intrinsic properties of the metadata instance, but also the capabilities of the searching tool and preferences of the users. To isolate the metadata properties, a metric should measure the potential accessibility of the object independently of the method used for its retrieval. In Network Science, the logical accessibility of a node in the network is calculated as the number of links from the node to other nodes [Newman et al., 2006]. Borrowing this idea, this work proposes the use of the linkage of a instance as an intrinsic accessibility value. A link can be explicit (for example "is-related-to" or "is-version-of" fields) or it can be implicit

(for example objects of the same author, on the same subject, etc.). An easy way to visualize how implicit linking could be calculated is to create a bipartite graph where Partition 1 contains the instances and Partition 2 contains the concept through which the linking will take place (authors, categories, etc.). Then the graph is folded over Partition 2, leaving a normal graph with linking between resources. An example of this procedure is shown in Figure 4.2.

Figure 4.2: Procedure to establish the linking between instances, based on classifying concept



The linkage metric can be calculated by adding all links pointing from or towards a instance and dividing that number by the number of links of the most connected object (Equation 4.13).

$$Qlink = \frac{links(instance_k)}{\max_{i=1}^{N}(links(instance_i))} \tag{4.13}$$

Where $links(instance)$ represent the number of pointers to or from the metadata instance. N is the number of resources in the repository.

Cognitive accessibility measures how easy it is for a user to understand the information contained in the metadata instance. Librarians measure this characteristic [Guy et al., 2004] with several simple metrics: measuring spelling errors, conformance with the vocabulary, etc. Nonetheless, they always include a human evaluation of the difficulty of the text. The difficulty assessment could be automated using one of the available readability indexes, for example the Flesch Index [McCallum and Peterson, 1982]. This metric could be applied especially to analyze long text fields of instances (e.g. description). Readability indexes in general count the number of words per sentence and the length of the words to provide a value that suggest how easy it is to read a text. For example, a description where only

acronyms or complex sentences are used will receive a lower score (lower quality) than a description where normal words and simple sentences are used.

Table 4.8: Example calculation of the Flesch Index for different texts

Resource	Description	Flesch Index
1	This deliverable is based on the performance indicators developed in D8.4. Controlling the activities via the transfer channels by performance indicators is the main issue outlined in this deliverable. The monitoring process is focussed on major activities made by visitors and registered users in the Virtual Competence Centre. By organizing and managing the community of practice we focussed in the interpretation on two aspects...	80
2	This deliverable reports on the LOMI seminars - a series of virtual seminars on Learning Objects, Metadata and Interoperability (LOMI). The basic intent of the seminars is to facilitate exchange of opinions, ideas, plans and results on the overall theme of learning objects, metadata and interoperability. .... o 03 May, 15:00-16:30 CEST o 23 May, 16:00-17:30 CEST o 07 June, 15:00-16:30 CEST o 21 June, 16:00-17:30 CEST o 05 July, 15:00-16:30 CEST There is no cost involved for the participants. The wiki at http://ariadne.cs.kuleuven.ac.be/wiki/lomi/Wiki is the real, continuously updated and thus up-to-date deliverable D4.2. ...	30
3	Analysis of future professional training needs in Europe It is the same document as D6.1 joint report on economical approaches, user needs and market requirements for technology enhanced learning already submitted by WP6.	14

Table 4.8 presents the calculation of the Flesch index for descriptions taken from descriptions of learning object metadata instances from ARIADNE. Short sentences and words lead to high values of Readability. On the other hand, long sentences, lack of punctuation, numbers and heavy use of acronyms reduce that value. The approximate maximum value of the Flesh index is 100 (easy to read

text) while the minimum approximate value is 0 (hard to read text).

The readability metric is the normalized average of the Flesh Index of all the text fields in the instance. This calculation is presented in Equation 4.14.

$$Qread = \frac{\sum_{i}^{N} Flesch(fieldtext_i)}{100 * N} \tag{4.14}$$

Where N is the number of textual fields and $Flesch()$ is the calculation of the Flesch readability index.

4.3.6 Timeliness Metrics

The timeliness in digital repositories mainly relates to the degree to which a metadata instance remains current. The currency of a metadata instance could be measured as how useful the metadata remains with the pass of time. For example, if a instance describing a resource was created 5 years ago, and users could still find, identify, select and obtain the resource correctly, the metadata can be considered current. On the other hand, if the metadata instance misleads users, because the referred resource has changed to the point where the description in the metadata differed from the resource, the metadata registry is obsolete and must be replaced.

The currency of the instance at a given time can be equated with its overall quality. Following this reasoning, the average value of previously presented metrics could be used as an estimation of the instantaneous currency of a instance (Equation 4.15). However, the instantaneous currency does not offer any information on how long the instance will continue to be current. For example, knowing that the currency of the description of a web page is high at the moment of the creation of the description does not guaranty that it will stay current after a year. Also, different objects change at different paces. A better estimation of the timeliness of a instance could be obtained measuring the rate of change of the instantaneous currency over a period of time. In more concrete terms, the timeliness of a instance will be equal to its change of average quality per unit of time (Equation 4.16). Following the example of the web page descriptor, if after a year, the currency of the instance has been reduced by half, it is logical to expect that after another year it will be degraded to one quarter of its original currency. This metric can also measure positives changes in currency, for example, if the metadata instances are constantly enriched through user tagging and usage information. In those cases, the timeliness metric could be used to estimate how much better the instance will be after a defined period.

$$Qcurr = Qavg = \frac{\sum_{i=1}^{N} \frac{(Q_i - minQ_i)}{(maxQ_i - minQ_i)}}{N} \tag{4.15}$$

Table 4.9: Example calculation of Qtime

t1-t2	Qavg(t1)	Qavg(t2)	Qtime	Qcurr in 1 year
1 year	0.8	0.5	−37.5% per year	0.31
1 year	0.5	0.8	+60% per year	1.28
1 month	0.95	0.85	−26% per month	0.22

Where Q_i is the value of the ith quality metric (for example Qcomp, Qtinfo or Qread), $minQ_i$ and $maxQ_i$ are the minimum and maximum value of the ith metric for all the instances in the repository. N is the total number of metrics considered in the calculation. Q_avg is then the average of the different quality metrics for a given instance.

$$Qtime = \frac{Qcurr_{t2} - Qcurr_{t1}}{Qcurr_{t1} * (t2 - t1)} \tag{4.16}$$

Where $t1$ is the time when the original currency ($Qcurr_{t1}$) was measured and $t2$ is the current time with is corresponding value of instantaneous currency ($Qcurr_{t2}$).

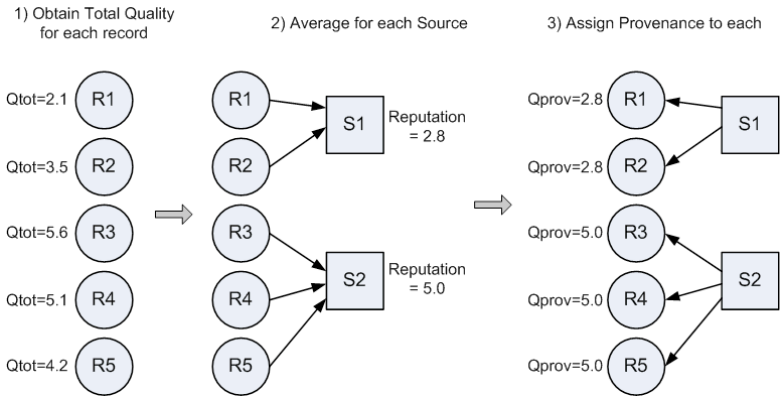
The sign of $Qtime$ indicates if the change in quality has been positive (increase in quality) or negative (decrease in quality). The absolute value represents the rate of currency change per unit of time used (years, months, days, etc.). Equation 4.17 can be used to estimate the currency ($Qcurr$) of the instance in a future time. Table 4.9 presents some example calculations of $Qtime$. The lower bound for $Qcurr$ is 0 while it does not have an upper bound (a metadata instance could always be improved). Given that we are working with rates, this formulat is identical to the one used to calculate the future value knowing the present value with compound interest.

$$Qcurr_{t3} = \left(\left(1 + Qtime_{(t2-t1)}\right)^{(t3-t2)}\right) \bullet Qcurr_{t2} \tag{4.17}$$

Where $Qtime_{(t2-t1)}$ is the calculation of the Qtime metric during the interval between $t1$ and $t2$. $t3$ is the time to which the Qcurr estimation is desired.

This metric can only be calculated if there are at least two values for Qavg taken at two different known times. In case that there are no previous measurements of $Qtime$ is 0 (no change). On the other hand, if three or more values of $Qavg$ exist, the $Qtime$ is calculated pairwise and then averaged in order to obtain a more representative value of change. Two consecutive measurements of $Qavg$ can be stored into the instance itself. Some metadata standards provide annotation fields where this information can be included. In case that the metadata standard or repository policies do not allow on-instance storage, a simple database could be implemented as part surrounding technological infrastructure.

Figure 4.3: Calculation of the Source Reputation and the Provenace of each instance (R represent the instances and S the sources)



4.3.7 Provenance Metrics

Provenance quality measures the trust that a given community has in the source of the metadata instance. For example, a metadata instance from the Library of Congress could be considered to have a higher Provenance quality than one generated in a local library. This higher level of provenance quality is not estimated by any intrinsic property of the metadata, but from the reputation that the Library of Congress has (quality assurance methods, expert staff, etc.) among the library community.

The main problem in converting the Provenance parameter into a metric is in obtaining information about the perception of the users about the metadata produced by a given source. This information can be captured explicitly, for example surveying the user about how useful the metadata has been to select the resources. The explicit collection of this type of information is bound to distract the user from her normal workflow. Given that the metrics proposed in this section should be an approximated measurement of the quality of the instance, a more scalable way to obtain the perceived quality of a source of metadata is to combine the metric values of its instances. The more straightforward way to combine those values is to first, obtain an Average Quality (Qavg) for each instance (Equation 4.15), and afterwards, average the Qavg of all the instances produced by the source (Equation 4.18). Once the quality of the source has been obtained, it is assigned to each one of its objects. This process is graphically explained in Figure 4.3.

$$Qprov = Reputation(S) = \frac{\sum_{i=1}^{N} Qavg_i}{N} \tag{4.18}$$

Where $Qavg_i$ is the Average Quality of the ith instance contributed by the source S. N is the total number of instances produced by S. The $Qprov$ of a instance is equal to the reputation of its source.

The Qprov metric can be calculated once the other quality metrics has been calculated and assigned to each instance. As it can be distilled from the calculations, each time a new instance is entered in the repository, the reputation of its source should be recalculated and the Qprov of all its instances could change. This is a desired effect given that the provenance of a source is not static. A previously good source could diminish its reputation if all its recent instances have low quality. In order to compromise between having an up-to-date value of reputation against the calculation load in the system, the recalculation could be peformed at fixed interval of time or instances inserted.

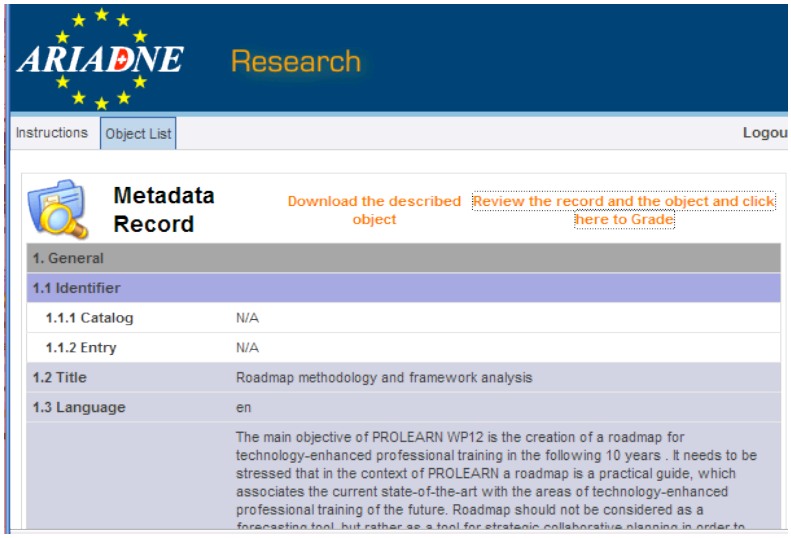
4.4 Evaluation of the Quality Metrics

Three validation studies were conducted in order to evaluate the metrics proposed in the previous Section. The first study measures the correlation between the value of the quality metrics and the quality assessment by human reviewers. The second study applies the metrics to two different sets metadata in order to establish their discriminatory power. The third study tests the metrics in a more realistic application: filtering bad quality metadata instances. These studies, along with the analysis of their results, are presented in the following subsections.

4.4.1 Quality Metrics correlation with Human-made Quality Assesment

A validation study was designed to evaluate the level of correlation between the quality metrics presented above and the quality assessment scores provided by human reviewers. During the study, several human subjects graded the quality of a set of instances sampled from the ARIADNE Learning Object repository [Duval et al., 2001]. We selected metadata instances about objects on Information Technologies that were available in English. From this universe (425 instances), we randomly selected 10 instances that were manually generated and 10 with metadata generated by an automated indexer. Each manual instance was produced by the author of the object (in this study, each metadata instance had a different author). The automatic metadata instances were produced by SAmgI [Meire et al., 2007]. The original objects, from which these metadata were automatically generated, are a set of Project deliverables that explain internal technologies of

Figure 4.4: Screen were the reviewer is presented with the metadata of the object, the option to download and to rate its quality



ARIADNE. An example of the sampled instances has been presented as examples in Section 3.

Following a common practice to reduce the subjectivity in the evaluation of the quality of metadata, we used the same evaluation framework described by Bruce & Hillman on which the metrics are based. A brief explanation of this framework can be found in Section 2. The reviewers had to grade the completeness, accuracy, provenance, conformance to expectations, consistency and coherence, timeliness and readability of the metadata instances.

The study was carried out online using a web application. After being trained in how to use the quality framework, each reviewer was presented with a list of the 20 selected objects in no specific order (automatic and manual generated instances were mixed). When the user selected an object, a representation of its IEEE LOM instance was displayed. The user then downloaded the referred object for inspection. Once the user had reviewed the metadata and the object, he was asked to provide grades in a 7-point scale (From "Extremely low quality" to "Extremely high quality") for each one of the seven parameters. A screen capture of the application can be seen in Figure 4.4.

Only participants that graded all the objects were considered in the study. The online application was available for 2 weeks. During that time, 22 participants completed successfully the review of all the 20 objects. From those 22, 17

(77%) work with metadata as part of their study/research activities; 11 (50%) were undergraduate students in their last years, 9 (41%) were postgraduate students and 2 (9%) had a Ph.D. degree. The participants belong to 3 different, and geographically distant, research & development groups. All of them had a full understanding of the nature and meaning of the examined objects and their metadata, and had a working knowledge of the evaluation framework.

Parallel to the human evaluation, an implementation of the quality metrics described earlier was applied to the same set of data that was presented to the reviewers. The metrics used in the study were:

- *Completeness metric (Qcomp):* It was implemented taking as a base the complete LOM instance, as described in Equation 4.1.
- *Weighted Completeness metric (Qwcomp):* The alphas needed in Equation 4.2 were obtained from the frequency of use of the fields in searches to the ARIADNE repository as reported in Najjar et al. [Najjar et al., 2003].
- *Accuracy metric (Qaccu):* It was calculated using Equation 4.3 to measure the semantic distance between the text extracted from the object and the title and description of the metadata instance. A LSA algorithm (SVD with S=2) was applied before obtaining the distance.
- *Categorical Information Content metric (Qcinfo):* The probability of each one of the values for different fields was extracted from all the metadata information in the ARIADNE repository. Equations 4.4 and 4.6 were used to compute the final metric.
- *Textual Information Content metric (Qtinfo):* The Inverse Document Frequency (IDF) values needed to compute Equation 4.8 were extracted from the corpora made with all the text from the instances of the ARIADNE repository.
- *Coherence metric (Qcoh):* The title and description of the LOM instances were contrasted to measure their semantic distance as described in Equation 4.12.
- *Readability metric (Qread):* Equation 4.14 was applied to text contained in the title and description of metadata instances.
- *Provenance metric (Qprov):* The Qav (Equation 4.15) was obtained from all the previous metrics (Qcomp, Qwcomp, Qaccu, Qcinfo, Qtinfo, Qcoh and Qread) for each instance. Qprov was equal to Qavg for all the manual generated instances becase they were created by different sources. In the case of the automatic generated instances, they all were assigned to the same source and they were assigned the same Qprov.

Table 4.10: Inter Class Correlation values for the rates provided by the human reviewers. 0.7 is the critical point for ICC

Quality Parameter	**ICC (average, two-way mixed)**
Completeness	0.881
Accuracy	0.847
Provenance	0.701
Conformance to Expectations	0.912
Consistency & Coherence	0.794
Timeliness	0.670
Accessibility	0.819

A limitation of the study was the constant result of some metrics. The Consistency metric (Qcons) always returned 1 because the instances did not violate any of the community or LOM rules. The Linking metric (Qlink) always returned 0 because there were no explicit nor implicit linking between the objects in the study set. Finally, the Timeliness metric (Qtime) was not calculated because there were no previous registers of the average quality (Qavg). Those metrics were excluded from the study.

Because of the inherent subjectivity in measuring quality, the first step in the analysis of the results was to estimate the reliability of the human evaluation. In this kind of study, the evaluation could be considered reliable if the variability between the grades given by different reviewers to a instance is significantly smaller than the variability between the average grades given to different objects. To estimate this difference, we use the Intra-Class Correlation (ICC) coefficient [Shrout and Fleiss, 1977] which is commonly used to measure the inter-rater reliability. We calculate the average measure of ICC using the two-way mixed model, given that all the reviewers grade the same sample of objects. In this configuration, the ICC is equivalent to another widely used reliability measure, the Cronbach's alpha.

The ICC was calculated for each one of the quality parameters. The results can be seen in Table 4.10.

The results of all the parameters, except for Timeliness, are higher than the recommended threshold of 0.7. This result suggests that reviewers provided similar quality scores and that further statistical analysis may be performed with those values. Given the near miss of the Timeliness evaluation, it will only be used to calculate the average quality score, but not in further statistical analysis. Table 4.11 presents the average value for each parameter of the human review for 6 of the 20 instances in the sample. Higher values represent higher quality.

Table 4.12 presents the metrics values for the same objects presented in Table 4.11. For all these metrics, higher values represent higher quality. While metadata

Table 4.11: Example of the average quality value assigned to 6 of the 20 sampled instances. The first 3 were obtained from manually generated metadata, the last 3 from automatic generated metadata

Parameter	**Manual**			**Automatic**		
	R1	R2	R3	R4	R5	R6
Completeness	2.59	3.86	3.14	3.27	2.14	3.27
Accuracy	3.36	4.27	3.86	3.73	3.23	3.86
Provenance	2.95	3.77	3.73	3.18	3.14	3.55
Conformance to Expectations	1.95	4.14	3.23	3.50	2.14	3.64
C & C	3.59	4.14	3.64	4.23	3.59	3.77
Timeliness	2.91	3.41	3.36	3.77	3.27	3.91
Accessibility	3.14	4.00	3.36	3.73	2.77	3.68
Average	**2.93**	**3.94**	**3.47**	**3.63**	**2.90**	**3.67**

Table 4.12: Example of the metric values assigned to 6 of the 20 sampled instances. The first 3 were obtained from manually generated metadata, the last 3 from automatic generated metadata

Metric	**Manual**			**Automatic**		
	R1	R2	R3	R4	R5	R6
Completeness (Qcomp)	0.33	0.35	0.29	0.29	0.29	0.30
Weighted Completeness (Qwcomp)	0.81	0.81	0.81	0.48	0.48	0.48
Accuracy (Qaccu)	0.96	0.93	0.97	0.97	0.99	0.98
Categorical Info Content (Qcinfo)	0.32	0.32	0.20	0.20	0.22	0.22
Textual Info Content (Qtinfo)	1.49	2.21	1.92	3.34	1.93	2.46
Coherence (Qcoh)	0.0	0.27	0.13	0.90	0.80	0.35
Readability (Qread)	32	15	40	0	30	3
Provenance (Qprov)	0.56	0.57	0.54	0.35	0.35	0.35

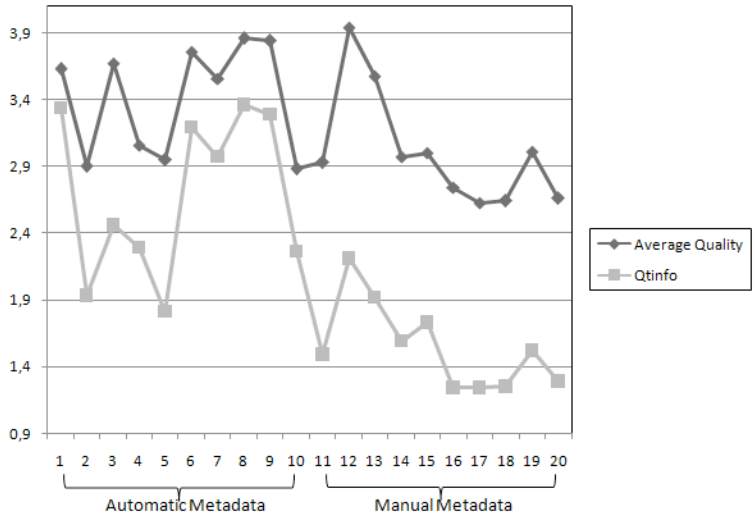
instances with high quality review present roughly higher values of the metrics, it is difficult to evaluate from these tables if the metrics are a good estimation of the manual quality review of the metadata instances. In order to provide a more appropriate evaluation of the effectiveness of the metrics, the next step in the analysis was to correlate the human quality score for each parameter with the metrics. The results are presented in Table 4.13. The main insight obtained is that, in general, the quality metrics do not correlate with their expected quality parameters as human rate them. For example, the Qcomp metric has a low and insignificant correlation with the completeness value. On the other hand, Qaccu has a slightly significant correlation with completeness. Moreover, Qtinfo correlates with all the human parameters. The default assumption with this kind of results should be to reject the hypothesis that the proposed metrics produce an estimation of the quality parameters proposed by Bruce & Hillman. Nevertheless, before the hypothesis is rejected, the unusual correlation of all the human scores with Qtinfo deserves a closer examination.

Table 4.13: Correlation between the human quality evaluation and the quality metrics. Bold font represents that the correlation is significant at the 0.01 level (2-tailed). Italic font represents that the correlation is significant at the 0.05 level (2-tailed).

	Qcomp	Qwcomp	Qaccu	Qcinfo	Qtinfo	Qcoh	Qread	Qprov
Completeness	.247	.537	*.519*	.011	**.787**	.282	.241	.152
Accuracy	-.370	-.421	*.492*	-.170	**.761**	.098	.270	.033
Conformance	-.290	-.533	.345	-.159	**.752**	.460	.191	-.022
C & C	-.393	-.453	*.470*	-.170	**.805**	-.083	.178	-.037
Accessibility	-.328	-.371	-.430	-.177	**.770**	.103	.334	.027
Provenance	-.437	-.473	.392	-.272	**.798**	.045	.397	-.101
Average	-.395	- *.457*	*.461*	-.182	**.842**	.225	.257	-.022

In a previous study by Zhu et al. [Zhu and Gauch, 2000], it was found that the Information Content of text is highly correlated to the quality of web pages as perceived by human reviewers. In this chapter (Section 4.3.4), Qtinfo measures the Information Content of the text fields of the metadata instance. A longer, more specialized text receives a higher score than a shorter, common one. Given that this value correlates highly with all the average human scores provided for each one of the quality parameters and that the ICC between reviewers was high, it can only be concluded that the human review was biased. This bias consists in rating instances with good textual fields with high values, even when that was not an indicated aspect of the framework quality parameter. Taking into account the diversity of the reviewer group, their knowledge in the field of metadata and

Figure 4.5: Comparison between the average quality score and the textual information content metric values)



that they have received instruction on how to apply the framework (and also had access to the descriptions while rating), the results suggest that non certified-expert evaluation of metadata is not a reliable method to estimate the quality of the instance in all its different dimensions.

While it can be concluded that this study is not suited to establish the "quality" of the quality metrics, it can be turned around and used to extract more information about what the reviewers took into account when rating the quality of the metadata. Firstly, a deeper analysis of the components that affect the human evaluation will be conducted. Figure 4.5 presents in the first 10 positions the objects with automated generated metadata. In the following 10, the objects that have their metadata manually generated. The average value for the human review is represented by the line at the top. The Qtinfo values are represented by the bottom line. The Qtinfo has higher values for the automatic generate learning objects. This result is expected because during the automatic generation process text segments contained in the objects are added to the description field. Manually generated instances, on the other hand, have small and sometimes not descriptive descriptions. Nevertheless, the quality value of human evaluations does not decrease as sharply for manually generated metadata. There seem to be other factors that determine the human review.

A multivariate regression analysis (Stepwise) was performed including all the

Table 4.14: Multivariate regression analysis of the quality parameters in function of the quality metrics. The Explanatory metrics specify which metrics where selected in the model (Stepwise) and their explanation power.

Parameter	**Explanatory metrics**	**Adjusted R^2**	**Std. Error**
Completeness	Qtinfo(62%) + Qcomp(22%)	0.824	0.2366
Accuracy	Qtinfo (58%)	0.555	0.3570
Conformance	Qtinfo (57%) + Qwcomp (14%)	0.681	0.4025
C & C	Qtinfo (65%) + origin (9%)	0.705	0.2162
Accessibility	Qtinfo (59%) + Qwcomp (14%)	0.702	0.2563
Provenance	Qtinfo (64%)	0.617	0.2501
Average Quality	Qtinfo (71%) + origin (10%)	0.798	0.2062

metrics and the origin of the metadata (1 for manual, 0 for automatic) to find possible explanations to the variability of each one of the parameters considered in the human review (except Timeliness). The results of the analysis are shown in Table 4.14 and explained in the following lines:

- *Completeness:* The rating behaviour for the Completeness is almost fully explained ($R^2 = 0.824$) by the addition of Qtinfo (62%) and Qcomp (22%). In other words, when assigning the value for completeness of the instance, the reviewers took into account the amount and quality of text fields and the total number of filled fields.

- *Accuracy:* The rating of Accuracy is only partially explained (58%) by the Qtinfo. The Qaccu was not relevant in the model. While textual information is good to establish the general quality of the object, textual similarity cannot explain how the reviewer rated the accuracy. Factors, not considered in the calculated metrics, seem to play a major role in the rating behaviour of the reviewers.

- *Conformance to Expectations:* Qtinfo seems to explain part (57%) of this parameter. As mentioned in Section 2.3, Qwcomp also seems to play a role in how reviewers perceive this dimension of quality. The relatively low adjusted R^2 value (0.681) suggests that there are other factors that influence the reviewers.

- *Logical Consistency and Coherence:* Again, Qtinfo explains more than half (65%) of the variability of this parameter. It is interesting to find out that the origin of the metadata also play a small role in the model (9%). This result suggests that users found manually generated instances more coherent.

- *Accessibility:* Apart from Qtinfo contribution (59%), unexpectedly from previous discussion but logical in retrospective, the presence of some fields, measured by Qwcomp, seems to affect (14%) the accessibility rate of the metadata.

- *Provenance:* This parameter can only be partially explained (61%) by the Qtinfo. Qprov was not related with the reviewers' score.

- *Average Quality:* As it can be inferred from Figure 4.5, if all the parameters are averaged, the final result could be mostly estimated (80%) by the Qtinfo metric in combination with the origin of the metadata. This is consistent with the high level of correlation of Qtinfo with the value of all the quality parameters.

A final analysis that could be performed with the results of the study will be to establish whether the origin of the metadata can be deduced from the metrics. To find out, a multivariate regression (Stepwise) is performed with the metrics as independent variables and the origin as the dependent. It was found that the origin of the data can be completely deduced (Adjusted $R^2 = 0.99$) from the values of Qwcomp (90%) and Qcinfo(10%). As was found after manual inspection of the metadata instances used in the study, manual instances provide more important fields (higher Qwcomp), while automatic instances have a low variability in their categorical values, being the same for most of the objects. As a result, the origin variable used to explain some quality parameters (Consistency & Coherence and Average Quality) could be replaced by a sum of Qwcomp and Qcinfo.

The main, serendipitous, conclusion from this study is that non-expert evaluation of metadata instances, even when guided with a multidimensional metadata quality framework, is biased toward considering metadata as content. The most measurable consequence of this bias is the application of one-dimensional assessment shortcuts (in this case, quality as amount of text) as the main factor for quality estimation. While a biased human evaluation of quality could not be used to establish how the proposed metrics correlate with the different quality parameters as described by Bruce & Hillman, it offered the opportunity to measure the usefulness of the metrics to explain the rating behavior of the reviewers. Even the origin of the metadata could be deduced from the metric values.

A side conclusion of the results is something that Web Search engines have known since their beginning, but almost all Digital Repositories user interfaces seem to neglect: Users do not process metadata [Duval and Hodgins, 2003], only text. While a considerable amount of web pages have metadata, these metadata are never shown to the user. It is only used to improve the efficiency of the search. On the other hand, most digital repositories try to present the user with the most complete metadata instance. This action is possibly detrimental [Duval and Hodgins, 2004]. Maybe a better approach will be to present the user with

just a brief text describing the resource while using the metadata to improve the precision and recall of the finding process.

4.4.2 Quality Metrics comparison between two metadata sets

In the second study, the quality metrics were applied to two different sets of metadata to evaluate their ability to discriminate key properties of the sets. The first set was composed of 4426 LOM instances corresponding to an equal number of PDF Learning Objects provided in 135 courses in Electrical Engineering and Computer Science at the MIT Open Courseware site. These metadata have been manually generated by expert catalogers in the MIT OCW team [Lubas et al., 2004]. The metadata downloading was performed on January 12nd 2008. The second set of metadata was composed by LOM instances automatically generated from the same 4426 PDFs described in the first metadata set. The metadata was generated only using the Text Content Indexer of SAmgI [Meire et al., 2007] that extracted and analyzed the text from the PDF files in order to fill the LOM fields. This setup was created in order to compare the value of the metrics for a set composed of expected good quality metadata (manual metadata created by experts) against a set of expected bad quality metadata (automated metadata only based on the text of the learning object). Also, the fact that both instances refer to the same object enables the use of statistical tools to establish whether the difference between the average metric values for both sets is significant.

This study uses the same metrics used in the previous study with three important changes. Firstly, the Qlink metric was added because both the manual and the automatic metadata instances contained keywords. These keywords were used to link the instances using the procedure proposed in the first part of section 4.3.6. A considerable amount of links (130 per object in average) were obtained. Secondly, to reduce the computational time for the 8852 instances, the SVD algorithm used to calculate the semantic distance between words was replaced by the Random Projection algorithm [Bingham and Mannila, 2001] in the calculation of Qaccu and Qcoh. The Random Projection produces similar results to SVD at a fraction of the computational time [Bingham and Mannila, 2001]. Thirdly, Qprov was not calculated because the MIT OCW metadata set does not specify the author of the metadata. Moreover, the automatic generated metadata set just have one source, thus having a constant value for Qprov. Instead of Qprov, the Qavg value was obtained for each instance.

The metrics were applied to each metadata instance in both sets. Once the values were obtained, a Paired T-Test was applied to measure whether the difference between the average values was statistically significant. The average value of metrics for each metadata set as well as the result of the Paired T-Test are reported in Table 4.15. All the metrics have a statistically significant different

Table 4.15: Metric values for the Manual and Automatic metadata sets, the correlation between the values for a same instance and the result of the comparison of means using the Paired T-Test. In bold, the highest quality average for each metric.

	Avarge Metric Value			
Metric	**Manual**	**Automatic**	**Correl.**	**Paired T-Test (2-tailed)**
Qcomp	**0.49**	0.38	0.073	t=344,df=4425,Sig=.000
Qwcomp	**0.75**	0.41	0.182	t=232,df=4425,Sig=.000
Qaccu	0.59	**0.90**	0.191	t=107,df=4425,Sig=.000
Qcinfo	**0.93**	0.16	0.142	t=432,df=4425,Sig=.000
Qtinfo	**6.14**	5.9	0.029	t=10,df=4425,Sig=.000
Qcoh	**0.40**	0.26	-0.024	t=8,df=4425,Sig=.000
Qlink	0.22	**0.24**	0.103	t=3.5,df=4425,Sig=.001
Qread	**0.26**	0.11	-0.014	t=4.5,df=4425,Sig=.000
Qavg	**0.66**	0.47	0.115	t=210,df=4425,Sig=.000

average value for the two sets. Also, the values obtained for metadata instances referencing the same learning object in the manual and automatic sets are not correlated. This independence let us discard the influence that the object itself have in the metadata quality measurement.

From the Qavg values in Table 4.15, it can be concluded that, in general, the metrics found that the manual metadata set has higher quality than the automatic metadata set. This corroborates the hypothesis raised at the setup. A closer examination of the average of each quality metric reveals more information about the differences between both sets. The Completeness (Qcomp) and Weighted Completeness (Qwcomp) metrics point that human experts filled more fields (and also more important fields) that the SamgI Text Content Indexer. This is an expected result given the limited amount of information that can be extracted by simple text analysis algorithms.

The automatic set has a better average value of the Accuracy (Qaccu) metric. This, however, does not mean that automatic metadata is more accurate that the manual one, but it is attributable to a measuring artifact. Qaccu is calculated measuring the semantic distance between the text in the metadata instance and the text in the original object. The fact that all the text in the automatic metadata instances is directly extracted from the object's text explains the high value of Qaccu for the automated metadata set.

Another expected result is that humans tend to select a richer set of categorical values than the simple automated algorithm. This is reflected in the average values of the Categorical Information Content (Qcinfo) metric. For example, where the

Learning Resource type value for all the learning object is set to "narrative text" in the automated instances, the human experts classify the same objects as "problem statement", "lecture", "questionnaire", "slide", etc. When all the objects in the set have the same value, Qcinfo tend to be low.

An interesting result from the comparison is that the Textual Information Content (Qtinfo) of both sets is high and very similar. That means that both instances, manual and automatic, contain long (and useful) descriptions. The manual ones were generated by humans; the automatic ones were obtained from text fragments of the original document. This finding implies that both metadata sets could have a similar level of performance (or quality) in learning object search engines that are based on text search in the metadata content. Also, as found in the previous studies, humans will be satisfied with the automatic metadata instances, given that they provide good text descriptions.

The Coherence (Qcoh) and Readability (Qread) metrics are higher also for the manual metadata sets. Text written by humans is bound to be easier to read and more coherent than text fragments automatically obtained from the learning object itself. Also, the coherence between the title and the description in the automatic set is expected to be low because the automatic algorithm takes the title contained in the PDF metadata as the value for the Title field. Normally this title in the PDF metadata is just the name of the file.

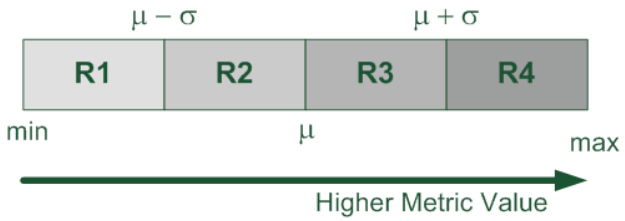
Finally, another interesting result is the almost tie in the Linkage metrics (Qlink). That result implies that the keywords manually added to the instances, and the keywords automatically generated have the same capability to link instances among them. This capability could be useful in search engine that use keywords as a way to discover new material (similar to the practice to use tags to link content).

This study comparing the quality metrics values for two different metadata sets confirms the ability of the metrics to measure quality characteristics in the instances. Differences expected from a-priori knowledge of the origin of the datasets were discovered as differences in the quality metrics values. Also, the study served to test the feasibility of applying the quality metrics to a relatively large set of instances.

4.4.3 Quality Metrics as automatic low quality filter

A final study was setup to test the metrics in a more realistic task. This task is to automatically filter or identify low quality instances inside a collection. It is expected that lower quality instances get lower metric values. To test this hypothesis, human reviewers were asked to select the lowest quality instance (according to different quality dimensions) from a given set of instances. These instances belong to different ranges of the corresponding metric value. At the end the human selections were compared with the metric value to establish whether the ones with the lowest values were selected as the worst instances.

Figure 4.6: Range explanation. 4 ranges were selected from the quality metric value to indicate 4 groups (R1, R2, R3 and R4) of increasing metric value



The instances for this study were selected from the manual and automatic sets used in the previous study. Four ranges were created for each metric value. The ranges were delimited by the mean, the mean minus one standard deviation and the mean plus one standard deviation. Figure 4.6 represents graphically these ranges. R1 represents the instances with the lowest metric value, while R4 contains the instances with the highest metric value.

All the metrics (Qcomp, Qwcomp, Qaccu, Qcinfo, Qcoh, Qread, Qavg) used in the previous study with exception of Qlink were considered also for this study. Qlink was removed because humans cannot evaluate how connected a instance without access to the whole repository. Nonetheless Qlink was considered for the calculation of Qavg.

For each metric 10 comparisons were generated. Five comparisons were drawn from the manual metadata set and five from the automated metadata set. Each comparison contains four instances. Each of those four instances was selected randomly from a different range of the metric. Once generated, each comparison has a instance from each one of the four ranges. Each comparison was presented to four human reviewers. The four reviewers were assigned to each comparison in alternate order to balance the effect of subjective review. Eight reviewers participated in the study. All of them were graduated research assistants that work with metadata as part of their research. When presented with a comparison, the reviewer had to select the lowest quality instance according to a given directive, different for each metric. The directives for each metric are presented in Table 4.16.

Once the results of the comparisons were collected, the first step was to determine whether there was consistency in the selections performed by the reviewers. An object was consistently selected if at least three of the four reviewers had selected it. Table 4.17 shows the consistency percentage for each one of the metrics. Given that for all the metrics obtained more than 70% of consistency (only in three or less comparisons there was not a majority for a given instance) it can be

Table 4.16: Directives given to reviewers to select the lowest quality instance according to each metric

Metric	Directive
Completeness (Qcomp)	Select the instance that present less information
Weighted Completeness (Qwcomp)	Select the instance that present less useful information
Accuracy (Qaccu)	Select the less accurate instance (original object supplied)
Categorical Info Content (Qcinfo)	Select the less descriptive instance
Textual Info Content (Qtinfo)	Select the less descriptive instance
Coherence (Qcoh)	Select the instance with less internal coherence
Readability (Qread)	Select the instance that is less readable
Total (Qavg)	Select the lowest quality instance

concluded that the noise present due to different criteria of each reviewer was low and the results could be used as a good approximation of what human reviewers would select as low quality instances.

The next step in the result analysis was to compare the selections performed by the reviewers against the value of each metric. Figure 4.7 presents the percentage of times that an object in each range was selected as the lower quality instance for each one of the metrics. Three metrics (Qcomp, Qwcomp and Qtinfo) seem to agree with the human selection. The majority of the reviewers selections for these metrics took place in the R1 range. A decreasing number of selections can still be seen in R2 and R3. No instances with a high value (R4) in those metrics were selected by any of the human reviewers. On the other hand, Qaccu, Qcoh and Qread do not seem to correlate well with human selections. A non-conclusive distribution could be seen across the four ranges. This is an indication that the metrics are not measuring the same quality characteristics as humans interpret from the given directives. An exceptional case is Qcinfo. Here, there is a clear preference of the reviewers for the R2 range. A deeper analysis of the metrics and human selection suggests that those instances in R1 do miss several categorical fields while the ones in R2 have those fields, but are filled with very common values. It seems that Human reviewers seem do not take into account missing values when evaluating the descriptive power of the instance. Finally, the combination of the metrics, Qavg, seems to be well related to what human reviewers consider as the general quality of the instance.

As a final analysis, Table 4.18 presents the effectiveness percentage (percentage

Figure 4.7: Distribution of human selection of lowest quality instances among Ranges of the Quality Metrics. R1 are the lowest metric values and R4 are the highest metric values.

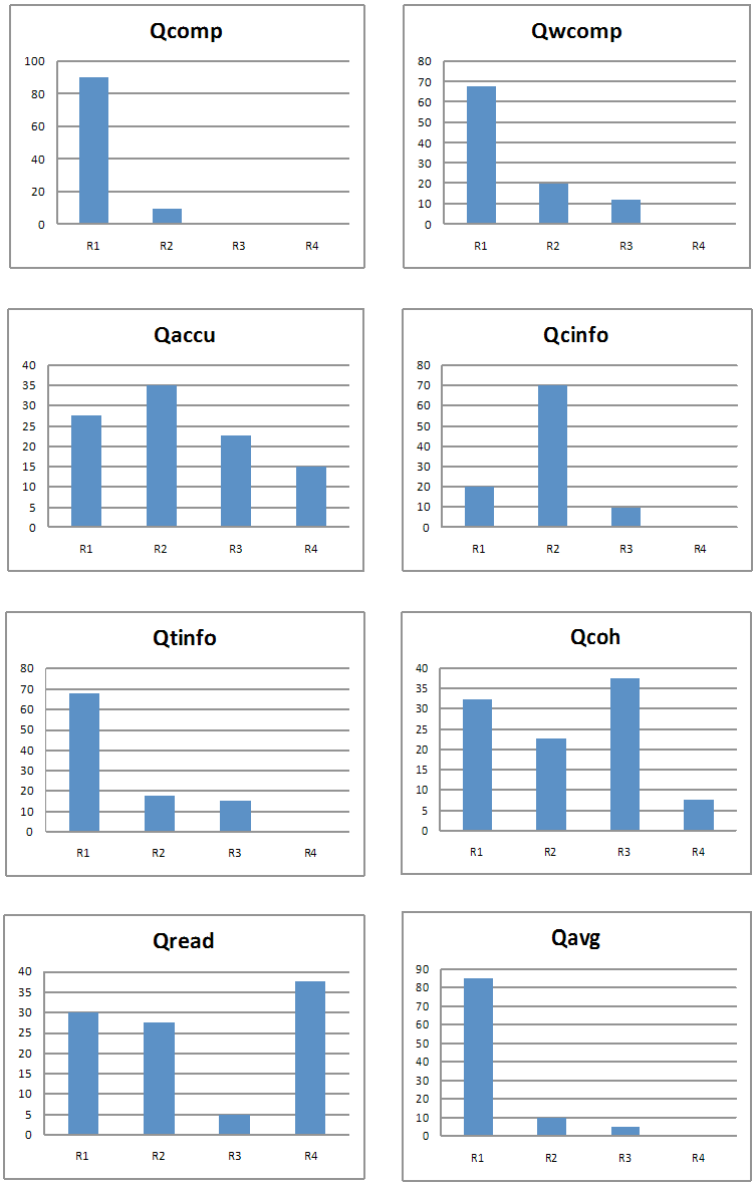


Table 4.17: Consistency percentage for each Comparison Sets

Comparison Set	Consistency of Human Reviewers
Completeness (Qcomp)	100%
Weighted Completeness (Qwcomp)	90%
Accuracy (Qaccu)	70%
Categorical Info Content (Qcinfo)	70%
Textual Info Content (Qtinfo)	100%
Coherence (Qcoh)	70%
Readability (Qread)	80%
Total (Qavg)	90%

of times that a value in the R1 range won the reviewers vote in a comparison). This value amounts to the percentage of times that the metric would have agreed with the human selection if it was meant to be an automatic filter to discard low quality instances. The most important finding in this analysis is that the Qavg metric (the combination of all other metrics) would have flagged 9 out of 10 instances selected as the lowest quality by the majority of human reviewers. Table 4.18 also presents the effectiveness percentage considering only the manual and the automated set. From these values, it can be concluded that the source of the metadata does not affect the effectiveness of the metrics.

The results of this study strongly suggest that some of the metrics (Qcomp, Qwcomp and Qtinfo), as well as the combination of all the metrics (Qavg), can be used to build an automated quality filter system for metadata instances. This system could also take the form of a metadata expert assistant that flag the most problematic instances to guide cleaning or enrichment processes. This type of system is presented as an application of the metrics in the next Section.

4.4.4 Studies Conclusions

From the three validation studies performed to evaluate the quality metrics, several conclusions could be drawn:

- Human reviewers tend to agree when evaluating the quality of metadata. However, it is no so clear which dimensions of quality they evaluate, even if they are guided by a framework as in the first study or guidelines as in the third one. From the results of the first study it seems that when confronted with the metadata, the reviewers evaluated it as content.

- Some metrics correlate well with human reviews while others seems to be completely orthogonal. From all the proposed metrics, the Textual Infor-

Table 4.18: Effectiveness percentage for each metric. This indicate the percentage of times that the metric agreed with the human most voted instance. It also presents the percentage disaggregated for the Manual and Automated Metadata sets.

	Effectiveness		
Metric	**General**	**Manual**	**Automated**
Completeness (Qcomp)	90%	100%	80%
Weighted Completeness (Qwcomp)	70%	80%	60%
Accuracy (Qaccu)	30%	20%	40%
Categorical Info Content (Qcinfo)	20%	40%	0%
Textual Info Content (Qtinfo)	80%	80%	80%
Coherence (Qcoh)	40%	40%	40%
Readability (Qread)	30%	20%	40%
Total (Qavg)	90%	100%	80%

mation Content (Qtinfo) seems to be a good approximation of the human perceived quality of a instance (the metadata as content effect). In a surprising result, given that half of the metrics did not correlate with human evaluation, Qavg, the combination of all the metrics does seem to also capture the elements taken in account by reviewers when accessing the general quality of the objects.

- There are quality characteristics that human reviewers are not able to evaluate. The variability of the categorical values or the level of connection of the instances, where the reviewer needs to have information about the whole universe of instances, are specially difficult to evaluate manually. In this sense, the quality metrics, even the ones that did not correlate well with the human evaluation, were able to measure characteristics related to the quality of the two different metadata sets in the second study.
- The usefulness of the combination of the proposed quality metrics in at least one practical application, low quality metadata filtering, was strongly suggested by the results of the third study. This set of metrics is indeed a step forward the automatic evaluation of metadata quality in digital repositories.

4.5 Implementation and Applications of Metadata Quality Metrics

The most important aspect of the metrics proposed in this work is that they can be automatically calculated from the metadata present in the repository and

the digital objects being described. The result of the metrics can then be used in tools that generate metadata (manually or automatically) to provide an automatic quality estimation of each metadata instance that is produced. Also, the value of the metrics for a whole repository, or federation of repositories, can be used in quality assurance applications that allow an administrator to identify quality problems in order to take corrective actions.

The metrics can be used by applications that generate metadata (to provide a quality control over each produced instance), applications that analyze the indexing behavior of metadata producers or applications that search for low quality instances in order to correct them. Some examples of that type of applications that can benefit from the quality metrics include:

- *Automatic validation and correction of metadata.* While previous research suggests that automatic metadata generation has a similar quality level as human generated metadata [Meire et al., 2007], the main objection against automatic generation of metadata is how to provide it with some degree of quality assurance [Ochoa et al., 2005]. Metadata extraction mechanisms work most of the time, but sometimes they produce useless instances. Without quality assurance, those low quality instances will be mixed with the whole repository, decreasing its overall value. Manually reviewing the output of an automatic generator is an unfeasible task. The metadata quality metrics proposed in this chapter could be used to implement an automatic evaluator of metadata that can flag low quality instances. For example, instances that do not contain a meaningful description or whose title is not coherent with the description can be flagged before they are inserted into the repository. On the other hand, if the automatic evaluator of metadata is run over human generated metadata, it could guide an automatic generator of metadata to improve the content of low quality instances. For example, metadata instances that lack a description could be improved with an automatic summary created by an automatic generator from the textual content of the resource.

- *Visualization of repository wide-quality.* The metrics values can be used to create visualizations of the repository in order to gain a better understanding of the distribution of the quality problems. For example, a treemap visualization [Bederson et al., 2002] could be used to find answers to different questions: Which authors or sources of metadata cause quality problems? How has the quality of the repository evolved over time? Which is the most critical problem of the metadata in the repository?, etc. An example of such visualization is shown in Figure 4.8. The treemap represents the structure of the ARIADNE repository. The global repository contains several local repositories and different authors publish metadata in their local repository. The boxes represent the set of learning objects metadata instances published

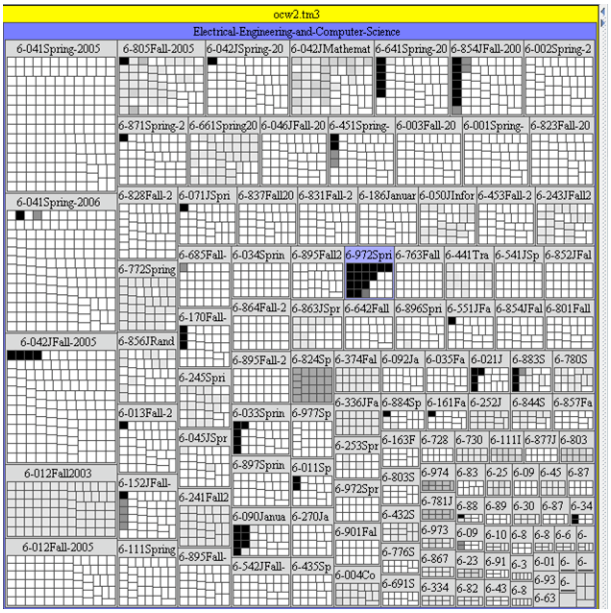
by a given author. The color of the boxes represents the average of the Qtinfo metric score of that set of instances. The color scale goes from red/dark grey (low quality) to yellow (medium quality), to green/light grey (high quality). This visualization helps to easily spot authors that provide good textual descriptions to their objects. Figure 4.9 shows the same type of visualization, but indicating incomplete instances (Qwcomp) in the human generated metadata from MIT OCW used during the studies. In this case, finding the incomplete instances would have been difficult without the help of the visualization tool.

Figure 4.8: Visualization of the Textual Information Content of the ARIADNE Repository. Red (dark) boxes indicate authors that produce low quality descriptions.



- *(Automatic) Selection of repositories for federated search.* If the repositories belonging to a federation publish their results for the quality metrics, that information can be used by federated search engines to automatically select repositories with a quality similar or superior to the local repository. Also, depending on the task to perform, the engine could choose to return only instances that have a good textual description of the resource. An initial implementation of this kind of application has already been devised by Hughes [Hughes, 2004] to provide a "star-ranking" for repositories of the Open Language Archive but based mostly on completeness metrics.

Figure 4.9: Visualization of the Completeness of the Manual Metadata set extracted from MIT OCW. Dark boxes represents instances that are incomplete.



4.6 Related Work

As shown in Section 1 and 2, there is extensive conceptual research in Information Quality and more specifically Metadata Information Quality. On the other hand, automatic calculation of metrics to estimate quality of metadata is much rarer. To our knowledge, only Stvila et al. in [Stvilia, 2006] and [Stvilia et al., 2006] seriously address the issue of multidimensional metadata quality estimation based on automatic calculations. Their metrics are also based on a 9 quality parameter framework: Intrinsic Precision, Intrinsic Redundancy, Intrinsic Semantic Consistency, Intrinsic Structural Consistency, Relational Accuracy, Relational Completeness, Relational Semantic Consistency, Relational Structural Consistency and Relational Verifiability. In [Stvilia, 2006], Stvila presents 13 quality metrics. While most of them (11) are simple counts of errors or defects (for example: # of broken links, # of words not recognized by MS word dictionary over the total number of words, etc.), the remaining two, Information Noise and Kullback - Leibler Divergence have some relation with our Qtinfo and Qcinfo metrics respectively. Due to the lack of reference quality information, Stvila was not able to evaluate his metrics directly. What he found is that the metrics correlate with

a-priori knowledge of two different sources of metadata (similar to what have been done in the second study of this chapter). A comparison analysis over a common set of metadata could be an interesting subject for further research.

4.7 Conclusions

Although quality of metadata for digital repositories is a very difficult concept to measure as a whole, when divided into more concrete parameters, as the ones proposed by several quality frameworks, quality can be operationalized in the form of quality metrics. These metrics, while simple to calculate, could be effective estimators of quality. In this work, some of the proposed metrics, especially Textual Information Content metric and the combination of metrics (Qavg), were able to explain the quality rating behavior of human reviewers, discriminate between different sets of metadata and even automatically flag low quality instances as good as any human reviewer.

The development of quality metrics will enable metadata quality researchers to not only obtain snapshots of the quality of a repository, but also to constantly monitor its evolution and how different events affect it without the need to run costly human-involving studies. This could lead to the creation of innovative applications based on metadata quality that would improve the final user experience.

The proposed metrics are not presented as an optimal solution to the problem of automatic evaluation of quality, but they can be used as a baseline against which new, better, metrics could be compared. While a lot more research and experimentation in metadata quality metrics is needed, this chapter shows that automatic quality assurance based on metrics is possible. Moreover, automatic evaluations have to be provided in order to sustain the increase in the metadata production. That is the only way for current digital repositories to avoid degradation of their functionality.

Following with the idea of using the information about the learning object to produce useful metrics, the next chapter will present a set of relevance ranking metrics that can estimate the quality of the learning object itself for a given user and context.

Chapter 5

Relevance Ranking Metrics for Learning Objects

5.1 Introduction

In a broad definition, learning objects are any digital document with that can be used for learning. Learning Object Repositories (LOR) exist to enable sharing of such resources [McGreal, 2004]. To be included in a repository, learning objects are described by a metadata record usually provided at publishing time. All current LORs provide or are coupled with some sort of search facility.

In the early stages of Learning Object deployment, these repositories where isolated and only contained a small number of learning objects [Neven and Duval, 2002]. The search facility usually provided users with an electronic form where they could select the values for their desired learning object. For example, through the early ARIADNE Search and Indexation tool [Duval et al., 2001] a user could select "English" as the language of the object, "Databases" as the sub-discipline and "Slide" as learning result type. The search engine then compared the values entered in the query with the values stored in the metadata of all objects and returned those which complied with those criteria. While initially this approach seems appropriate to find relevant learning objects, experience shows that it presents three main problems: 1) Common users (i.e. non metadata experts) found this query approach too difficult and even "overwhelming" [Najjar et al., 2005]. The cognitive load required to express their information need into the metadata standard used in the repository was too high. Metadata standards are useful as a way to interchange information between repositories, but not as a end-user query interface. 2) In order for this approach to work, the metadata fields entered by the indexers need to correspond with the metadata fields used by the searchers. A usability study by Najjar [Najjar et al., 2004] found that this is usually not the

case. And finally, 3) The high precision of this approach often leads to a low recall [Sokvitne, 2000]. Being small repositories, most searches produced no results, discouraging the users.

Given these problems with the metadata-based search, most repositories provided a "Simple Search" approach, based on the success of text-based retrieval exemplified by Web Search engines [Chu and Rosenthal, 1996]. In this approach, users only need to express their information needs in the form of keywords or query terms. The learning object search engine then compared those keywords with the text contained in the metadata, returning all the objects that contained the same words. This approach solved the three problems of metadata based search: The searchers express their queries as a sequence of keywords, the metadata completeness was not as important as before as the query terms could be matched with any field or even the text of the object and finally, the recall of the query's results increased. This approach seemed the solution for small repositories. However, working with small, isolated repositories also meant that an important percentage of users did not find what they were looking for because no relevant object was present in the repository [Najjar et al., 2005].

Current research in the Learning Object community has produced technologies and tools that solve the scarcity problem. Technologies like SQI [Simon et al., 2005] and OAI-PMH [Van de Sompel et al., 2004], enable search over several repositories simultaneously. Another technology, ALOCOM [Verbert et al., 2005], decomposes complex learning objects into smaller components that are easier to reuse. Finally, automatic generation of metadata based on contextual information [Ochoa et al., 2005] allows the conversion of the learning content of Learning Management Systems (LMS), into metadata-annotated Learning Objects ready to be stored into a LOR. Although these technologies are solving the scarcity problem, they are creating an inverse problem, namely, abundance of choice [Duval, 2005] . The user is not longer able to review several pages of results in order to manually pick the relevant objects. The bias of the search engines towards recall only exacerbates this problem. The final result is that even if a very relevant object is present in the result list, the user still could not find it, again, reducing the perceived usefulness of LORs.

While doing a stricter filtering of results (increasing precision at expense of recall) could solve the oversupply problem, it could also lead again to the initial problem of scarcity. A proven solution for this problem is ranking or ordering the result list based on its relevance. In this way, it does not matter how long the list is, because the most relevant results will be at the top and the user could manually review them. As almost all search engines use this method, searchers are not only used to work with these sorted lists of results, but expect them [Kirsch, 1998]. To help the user to find relevant learning objects, Duval in [Duval, 2005] proposed the creation of LearnRank, a ranking function to define the relevance of learning objects similarly to how PageRank [Page et al., 1998] defines the relevance of web

pages. Also, in a related paper [Ochoa and Duval, 2006c], the authors explore how Contextualized Attention Metadata [Najjar et al., 2006], that is data obtained from the interaction of the users with the system, could be mined to obtain meaningful information about the relevance of a specific learning object for a specific user and context. The present chapter provides important progress in this direction, proposing and testing a set of multi-dimensional relevance ranking metrics. These metrics use external sources of information in addition to what is explicitly stated in the user query to provide a more meaningful relevance ranking than current query-matching implementations. The development of these metrics address three main questions: 1) What does means relevance in the context of Learning Objects? 2) How to convert the multi-dimensional relevance concept into numerical values that could be used for sorting? and 3) Can the proposed metrics outperform current generic ranking practices in LOR search?

The proposal and evaluation of these metrics have the purpose of providing the Learning Object Consumers with smarter tools to select learning objects in a economy of abundance where there are more relevant objects than the user is able to review. Having a set of effective ranking metrics is a technical requirement before learning objects can be commonly used in mainstream learning.

The structure of this chapter is as follows: Section 2 analyzes the current state of Learning Object Ranking. Section 3 discusses different dimensions of the relevance concept and how they translate to the context of Learning Object search. These relevance dimensions are used as guidelines in Section 4 to propose and compare a set of metrics that can rank a list of learning objects based on usage and contextual information. Section 5 presents different mechanism in which these metrics could be combined in a unique rank value. To obtain a rough estimate of the benefit that these metrics could have in a real implementation, an exploratory study, where the metrics are compared against human relevance rankings and existing ranking methods, is presented in Section 6. This study also tests the efficacy of the metrics combination.

5.2 Current Status of Learning Object Ranking

Current LOR search interfaces already provide ranking functionalities based on generic ranking strategies. In this section, we present three categories of those strategies found in practice and in the literature. The advantages and disadvantages of each one are discussed. Finally, the profile of the ideal ranking strategy for learning objects is contrasted with these approaches.

5.2.1 Ranking based on Human Review

Some LORs sort their results mainly by peer evaluation. In the case of MERLOT [Nesbit et al., 2002] for instance, a special group of expert users have the ability

to review the objects and grade the Content Quality, Effectiveness and Ease of Use. The average of these three grades is considered the rating for the object. Peer-reviewers also provide an explanation about the decisions behind the grade. The main advantage of this system is that it provides the searching user with a meaningful evaluation of the overall quality of the object and the possible scenarios where it could be useful. However, there are two main disadvantages of this approach. First, this is a very laborious, manual process. Not surprisingly, in the case of MERLOT for instance, only 10% of the objects contained in the repository have been peer-reviewed [Zemsky and Massy, 2004] (A more recent value of 12.3% was obtained through direct observation). That means that, even if an object is relevant for a user and it is returned in the result list, but it belongs to the 90% not peer review material, the user will probably not find it, as it will be hidden in a deep result page. Even worse, an object that received a low score in the review will still be ranked higher than any other not-rated object regardless of its quality. A solution tried by MERLOT to this problem is to allow users to comment on and rate the objects. While this helps to increase the number of rated objects (circa 25%), it is still difficult to reach a majority of objects. Also, the user reviews are less detailed than the peer review, providing less help to the searcher.

The second disadvantage is that a human measurement of quality is static and does not adapt to different users' needs. For example, searching for "databases" in Merlot will present in the first results highly rated resources for educational databases of content. This answer, while useful for users that are searching other repositories of learning materials, will not help the user that is looking for learning resources about relational databases, such as MySQL or Oracle. A similar approach is taken by Vargo et al. [Vargo et al., 2003]. They use the data generated by user's evaluation of the quality of the learning object to sort. Users measure the LO quality using the Learning Object Review Instrument, a set of nine quality parameters which the learning object should meet. The main drawback of this ranking approach is the previously mentioned problem of the lack of scalability of user review.

In summary, using manual review of learning objects for ranking suffers from the same problem of indexing by experts or librarians: humans do not scale [Weibel, 2005]. While normally highly meaningful for the end user, this approach will break apart in a projected ecosystem where millions of objects are created every day. Another perceived problem of manual review ranking is that it cannot be easily adapted to different users or contexts. The recall of the top-k elements in the result list tends to be low, as relevant objects will not be evaluated. The precision also varies depending on whether the context of the human evaluator was or not the same as that of the searcher.

5.2.2 Ranking based on Text Similarity

A completely different approach is followed by other repositories. We will take SMETE [Agogino, 1999] as an example. It relies on content-based calculations in order to assign a relevance value to all the objects returned in a search. In the case of SMETE, it calculates the similarity between the query terms and the text contained in the metadata of the learning objects. It use some variation of vector-space algorithms [Salton and Buckley, 1988]. This algorithm creates a vector for the documents and the query where each dimension is a word. The magnitude of the document or query vector in each word is the frequency of the word in the query divided by its frequency in the whole repository. This is similar to the algorithms used for basic text information retrieval [Salton and McGill, 1986] and early web search engines [Stata et al., 2000]. Other examples are presented in the work of Chellappa [Chellappa, 2004] that summarizes the methodology followed by several repositories: adapting full-text search approaches to rank the learning objects based only on the similarity between the query terms and the text fields of the metadata record. This approach, using simple text-based metrics, has the advantage that can be computed easily for each one of the objects in the list. Nonetheless, it presents two main disadvantages. First, the amount of text normally contained in the learning object metadata is low. This leads to equal values for several objects and underperformance compared to the use of the same algorithm against a full-text index. Second, the order of the final list reflects how many times the query words appear on the metadata fields of the object, but it does not transmit to the user any notion about the quality/relevance of the object itself. Fine-grained, but very relevant differences between learning objects, for example targeted age-group or educational context, are very difficult to be captured by any of the current text analysis / clustering mechanisms. In order to obtain a more meaningful ranking more contextual information is needed. For example, all current Web Search engines make use of ranking algorithms heavily based on the analysis of the Web network [Page et al., 1998] or click-through information [Joachims and Radlinski, 2007] in order to improve the sorting of the result list.

In summary, using the distance between the text in the metadata and the query terms to rank learning objects, specially using advanced approaches that deals with synonyms words such as Latent Semantic Analysis [Landauer et al., 1998], leads to high recall. However, using only a text-based approach reduces the precision of the top-k results. The lack of additional information to address learning specific aspects of the relevance ranking makes uncertain that objects in top positions correlate well with the real information need of the learner.

5.2.3 Ranking based on User Profile

Olmedilla [Olmedilla, 2007] propose to compare topics provided in the user profile with the classification of the learning object. The closer the values of the profile and the object are in the taxonomy, the higher the relevance of the object. From a theoretical point of view, this approach should lead to a high precision in the top-k. However, in practice, this approach presents several handicaps: 1) Users need to explicitly select their interest from a taxonomy before performing the search. 2) It could only be applied to objects that have been classified with the same taxonomy as the one presented to the users.

Personalized ranking based on user profile has, for instance, been implemented in the HCD-Online tool. The personalized ranking of this tool was evaluated by Law et al. [Law et al., 2006]. The results of the evaluation show that the text-based ranking (based on a Lucene [Hatcher and Gospodnetic, 2004] index) outperforms any of the personalized rankings. In a similar work, Dolog et al. [Dolog et al., 2004] propose a rule-based personalization based on the semantic description of both the user profile and the learning object. The main disadvantage of this approach is that it requires very rich manual metadata annotation of both the user and the object in order to work.

In summary, while the idea of using the profile to personalize the search results works in similar environments [Pitkow et al., 2002] [Sugiyama et al., 2004] , the way in which it is implemented could lead to unwanted results. Manually generated user profiles usually do not capture the real information need of the user [Quiroga and Mostafa, 1999] as this need is always changing depending on the context of the task that she is performing. The implicit learning of users profiles based on their interactions with the system seems to adapt better to changes in the needs of the user.

5.2.4 Current Approaches vs. Ideal Approach

All current approaches, namely, manually rating the objects, using only document information or asking the user to provide a profile, have serious disadvantages. The first is not scalable, the second does not carry enough insight on the quality (and thus relevance) of the object and the third does not integrate well into the normal workflow of the user. To enable a new generation of more useful search facilities for learning objects, an ideal ranking approach should take into account human generated information (to be meaningful). It should be possible to calculate its value automatically, no matter the size of the repository (to be scalable) and should not require conscious intervention from the user (to be transparent).

Other communities of practice, for example web search and journal impact, have already developed strategies that approximate the ideal approach. The PageRank [Page et al., 1998] metric used to rank web pages is scalable. It is routinely automatically recalculated for more than 11 billion pages [Gulli and Sig-

norini, 2005]. It is also meaningful as usually the top results are the most relevant pages for a query. And it is also transparent because it uses the information stored in the structure of the already existing Web graph to obtain its value. In the field of Scientometrics, the Impact Factor [Garfield, 1994] metric calculates the relevance of a scientific journal in a given field. It is automatically calculated from more than 12 million citations, its results are perceived as useful and editors do not need to provide any additional information than the one already contained in the published papers. These two examples prove that the implementation of ranking metrics that are scalable, meaningful and transparent to the final user is feasible and desirable. Following the ideas that lead to the development of these examples, this chapter proposes and evaluates metrics to automatically calculate the relevance of learning objects based on usage and contextual information generated by the interaction of the end user with learning object tools.

5.3 Relevance Ranking of Learning Objects

The first step to build metrics to rank learning objects by relevance is to understand what "relevance" means in the context of a Learning Object search. Borlund [Borlund, 2003], after an extensive review of previous research on the definition of relevance for Information Retrieval, concludes that relevance is a multidimensional concept with no single measurement mechanism. Borlund defines four independent types of relevance:

1. System or Algorithmic relevance which represents how well the query and the object match.

2. Topical relevance that represents the relation between an object and the real-world topic of which the query is just a representation.

3. Pertinence, Cognitive or Personal relevance which represents the relation between the information object and the information need, as perceived by the user.

4. Situational relevance that represents the relation between the object and the work task that generated the information need.

To operationalize these abstract relevance dimensions, they need to be interpreted for the specific domain of learning object search. Duval in [Duval, 2005] defined nine "quality in context" or relevance characteristics of learning objects. These nine characteristics can be disaggregated into eleven atomic characteristics. In this chapter, we map those characteristics to the dimensions proposed by Borlund. Table 5.1 presents those characteristics, the dimension to which they were mapped and a brief explanation of their meaning. For a deeper discussion on the rationale behind those characteristics we refer to [Duval, 2005].

Table 5.1: Map of Duval's "quality in context" characteristics into Borlund's relevance dimensions.

Topical Relevance	
Learning Goal	What the learner wants to learn
Personal Relevance	
Learning Motivation	Why the learner wants to learn
Culture	Cultural bias of the learner
Language	Languages understood by the learner
Educational Level	Age and learning background
Accesibility	Auditory, visual and motor skills
Situational Relevance	
Learning Setting	Activities surronding the learning object
Learning Time	Time available to study the learning object
Time of Learning	Time of the day when the learning takes place
Geo. Learning Space	Place where the learning takes place
Learning Space	Physical and technological conditions and limitations

Because the relevance characteristics of learning objects deal with the information need of the user and her preferences and context, but not on how the query is formulated, they do not map to the algorithmic dimension of relevance.

Those relevance characteristics that are related to "what" the learner wants to learn are mapped into the Topical Relevance dimension. Therefore, the only relevance characteristic mapped in this dimension is the learning goal. For example, if a learner is looking for materials about the concept of inheritance in Object Oriented Programming, the topical relevance of the object is related to how useful the object has been to learners studying courses related to Object Oriented Programming.

The relevance characteristics that are intrinsic to the learner and do not change with place and only slowly with time are mapped into the Personal Relevance dimension. Inside this group are the motivation, culture, language, educational level and accessibility needs. Further elaborating the previous example, we can imagine that the same learner feels more comfortable with objects in Spanish (her mother tongue) and is more motivated by visual information. The learner will find a slide presentation with graphics and descriptions in Spanish more relevant than a text document in English about the same subject, even if both have been successfully used to explain inheritance in Object Oriented Programming courses.

Finally, those relevance characteristics that deal with conditions and limitations that depend on the learning task, as well as the device, location and time, are mapped into the Situational Relevance dimension. Continuing the example, if the learner is doing his learning while commuting on the train on a mobile device, she will find more relevant for that context material that could be formatted to the

limited screen size.

The information to estimate these relevance dimensions is not only contained in the query parameters and the learning object metadata, but also in records of historical usage and the context where the query takes place. It is assumed that this information is available to the relevance ranker. This could seem unrealistic for classical Learning Object search, where the users, usually anonymous, perform their queries directly to the LOR through a web interface and the only information available are the query terms. On the other hand, new implementations of LMSs, or plugins for old implementations such as Moodle [Broisin et al., 2005] and BlackBoard [Vandepitte et al., 2003], as well as plugins for authoring environments [Verbert and Duval, 2007] enable the capture of information by providing logged-in users with learning objects search capabilities as part of user workflow during the creation and consultation of courses and lessons. Moreover, the development of Contextualized Attention Metadata [Najjar et al., 2006] to log the user interactions with different tools in a common format will help with the collection and simplify the analysis of usage and contextual information.

While this interpretation of the relevance concept is exemplified in a traditional or academic learning environment, it is at least as valid in less structured or informal settings such as corporate training or in-situ learning, given that the environments used to assist such learning also store information about the user and the context where the search takes place. This information takes the form of personal profiles, preferences, problem descriptions, previous and required competence.

A ranking mechanism that could measure some combination of the abovementioned learning relevance characteristics should provide the user with meaningfully ordered learning objects in the result list. The next section will propose pragmatic metrics that estimate those characteristics based on usage and contextual information in order to create a set of multidimensional relevance ranking metrics for learning objects. While not every characteristic is considered, at least two metrics are proposed for each relevance dimension.

5.4 Ranking Metrics for Learning Objects

To enable learning object search tools to estimate the relevance characteristics described in the previous section, those characteristics should be operationalized as ranking metrics and calculated automatically. The metrics proposed here are inspired on methods currently used to rank other types of objects, for example books [Linden et al., 2003], scientific journals [Garfield, 1994], TV programs [Pigeau et al., 2003], and so forth. They are adapted to be calculable from the information available from the usage and context of learning objects. These metrics, while not proposed as a complete or optimal way to compute the real relevance of a learning object for a given user and task, are a first step to set a strong baseline implementation with which the effectiveness and efficiency of more advanced

learning-specific metrics can be compared.

For all the following metrics it is assumed that, in response to a query, the learning object search engine applies first a filtering mechanism to return a list of objects that satisfy the query. This mechanism could be based on metadata filtering (returning only the objects that agree with any or all the metadata values used in the query) or query-proximity (returning only objects whose metadata contains any or all the query terms). The metric value is then calculated for all the objects in the result list and those values are used to sort it.

The following metrics are grouped according to the Relevance Dimension (Table 5.1) that they estimate. There are at least two metrics for each dimension describing different methods in which that relevance can be calculated from different information sources. Each metric is described below by: 1) the raw data it requires and 2) the algorithm to convert that data into concrete ranking values. Also for each metric: 1) an example is provided to illustrate its calculation and 2) methods to bootstrap the calculation of the metric in a real environment are discussed.

At the end of this section, the metrics are compared and a selection table is provided according to the desired relevance dimension and the information availability.

5.4.1 Topical Relevance Ranking Metrics.

Metrics to estimate the Topical Relevance should establish which objects are more related to what a given user wants to learn. The first step in the calculation of this type of metric is to estimate what is the topic that interests the user. The second step is to establish the topic to which each learning object in the result list belongs. There are several ways in which the first part, the topic that interests the user can be obtained: the query terms used, the course from which the search was generated and the previous interactions of the user with the system [Chi et al., 2001] [Agichtein et al., 2006]. For the second part, establishing the topicality of the objects, the information can be obtained from the classification in the learning object metadata, from the topical preference of previous learners that have used the object or the topic of courses that the object belongs to. Once the topical need of the user and the topic described by the object are obtained, the Topical Relevance metric is calculated as the distance between the two. The following subsections describe three possible Topical Metrics based on different sources of information.

Basic Topical Relevance Metric (BT)

This metric makes two naïve assumptions. The first assumption is that the topic needed by the user is fully expressed in the query. The second assumption is that

each object is relevant to just one topic. As a consequence of these two assumptions, the degree of relevance of an object to the topic can be easily estimated as the relevance of the object to that specific query. That relevance is calculated by counting the number of times the object has been previously selected from the result list when the same (or similar) query terms have been used. Defining NQ as the total number of similar queries of which the system keeps record, BT relevance metric is the sum of the times that the object has been selected in any of those queries (Equation 5.2). This metric is an adaptation of the Impact Factor metric [Garfield, 1994] in which the relevance of a journal in a field is calculated by simply counting the number of references to papers in that journal during a given period of time.

$$selected(o,q) = \begin{cases} 1, & \text{if } o \text{ clicked in } q \quad (5.1a) \\ 0, & \text{otherwise} \quad (5.1b) \end{cases}$$

$$BT(o,q) = \sum_{i=1}^{NQ} distance(q,q_i) * selected(o,q_i) \quad (5.2)$$

In Equations 5.1 and 5.2, o represents the learning object to be ranked, q is the query perfomed by the user. q_i is the representation of the ith previous query. The distance between q and q_i can be seen as the similarity between two queries. This similarity can be calculated either as the semantic distances between the query terms (for example their distance in WordNet [Budanitsky and Hirst, 2001]) or the number of objects that both queries have returned in common. NQ is the total number of queries.

Example: We assume that the query history of the search engine consists of queries Q_A, Q_B and Q_C. In Q_A, objects O_1 and O_2 were selected, in Q_B objects O_2 and O_3 and in Q_C, objects O_1 and O_2. A new query is performed, Q, and objects O_1, O_2, O_3 and O_4 are present in the result list. The distance between Q and Q_A is 1 (both are the same query), between Q and Q_B is 0.8 (they are similar queries), between Q and Q_C is 0 (not related queries). The BT metric value of O_1 is equal to $1 * 1 + 0.8 * 0 + 0 * 1 = 1$; for O_2 is 1.8; for O_3 is 0.8 and for O_4 is 0. The order of the final result list ranked by BT would be (O_2, O_1, O_3, O_4).

Data and Initialization: In order to calculate this metric, the search engine needs to log the selections made for each query. If no information is available, the metric assigns the value 0 to all objects, basically not affecting the final rank. When information, in the form of user selections, starts entering the system, the BT rank starts to boost previous selected objects higher in the result list. One way to avoid this initial training phase is to provide query-object pairs given by experts or obtained from information logged in previous versions of the search engine.

Course-Similarity Topical Relevance Ranking (CST)

In the context of formal learning objects, the course in which the object will be reused can be directly used as the topic of the query. Objects that are used in similar courses should be ranked higher in the list. The main problem to calculate this metric is to establish which courses are similar. A very common way to establish this relationship is described by SimRank [Jeh and Widom, 2002], an algorithm that analyzes the object-to-object relationships to measure the similarity between those objects. In this metric, the relation graph is established between courses and learning objects. Two courses are considered similar if they have a predefined percentage of learning objects in common. This relationship can be calculated constructing a 2-partite graph where courses are linked to objects published in them. This graph is folded over the object partition leaving a graph representing the existing relationships and strengths between courses. The number of objects shared between two courses, represented in this new graph as the number of links between them, determines the strength of the relationship. A graphical representation of these procedure can be seen in Figure 5.1. The ranking metric is then calculated counting the number of times that a learning object in the list has been used in the universe of courses (Equation 5.5). This metric is similar to the calculation made by e-commerce sites such as Amazon [Linden et al., 2003] where additionally to the current item, other items are recommended based on their probability of being bought together.

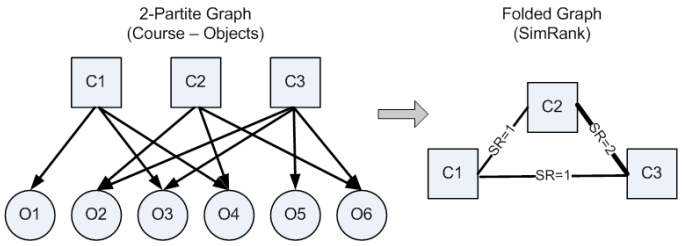


Figure 5.1: Calculation of the SimRank between courses for the Course-Similarity Topical Relevance Ranking (CST)

$$present(o, c) = \begin{cases} 1, & \text{if } o \in c \quad (5.3a) \\ 0, & \text{otherwhise} \quad (5.3b) \end{cases}$$

$$SimRank(c_1, c_2) = \sum_{i=1}^{NO} present(o_i, c_1) * present(o_i, c_2) \quad (5.4)$$

$$CST(o,c) = \sum_{i=1}^{NC} SimRank(c, c_i) * present(o, c_i) \tag{5.5}$$

In Equations 5.3, 5.4 and 5.5, o represents the learning object to be ranked, c is the course where it will be inserted or used. c_i is ith course present in the system. NC is the total number of courses and NO is the total number of objects.

Example (Figure 5.1): We assume that 3 courses are registered in the system C_1, C_2 and C_3. Objects O_1, O_3 and O_4 are used in C_1, objects O_2, O_4 and O_6 in C_2 and objects O_2, O_3, O_5 and O_6 in C_3. The SimRank between C_1 and C_2 is 1, between C_1 and C_3 is 1 and between C_2 and C_3 is 2. A query is performed from C_2 and in the result list are the objects O_1, O_3 and O_5. The CST value for O_1 is $1 * 1 + 2 * 0 = 1$, for O_3 is 3, for O_5 is 2. The order of the final result list ranked by CST would be (O_3, O_5, O_1).

Data and Initialization: To apply the CST, the search engine should have access to the information from one or several Learning Management System, such as Moodle or Blackboard, where learning objects are being searched and inserted. First it needs to create a graph with the current courses and the objects that they use in order to calculate the SimRank between courses. Second, it needs to obtain, along with the query terms, the course where the query was performed. In a system without this information, the CST will return 0, leaving unaffected the rank of the results. When the first results of insertion are obtained from the LMS, the CST could start to calculate course similarities and therefore ranking for the already used objects. This metric could be bootstrapped from the information already content in common LMSs or Open Courseware initiatives [Downes, 2007].

Internal Topical Relevance Ranking (IT)

If there is no usage information available, but there exists a linkage between objects and courses, the Basic Topical Relevance Rank can be refined using an adaptation of the the HITS algorithm [Kleinberg, 1999] proposed to rank web pages. This algorithm states the existence of hubs, pages that mostly point to other useful pages, and authorities, pages with comprehensive information about a subject. The algorithm presumes that a good hub is a document that points to many good authorities, and a good authority is a document that many good hubs point to. In the context of learning objects, courses can be considered as hubs and learning objects as authorities. To calculate the metric, a 2-partite graph is created with each object in the list linked to its containing courses. The hub value of each course is then calculated as the number of in-bound links that it has. A graphical representation can be seen in Figure 5.2. Finally, the rank of each object is calculated as the sum of the hub value of the courses where it has been used (Equation 5.6).

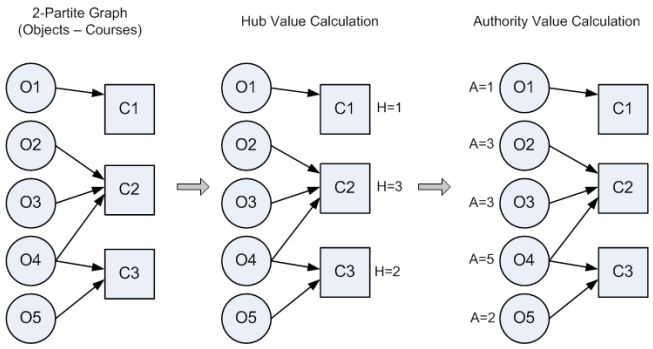


Figure 5.2: Calculation of Internal Topical Relevance Ranking (IT)

$$IT(o) = authority(o) = \sum_{i=1}^{N} degree(c_i) \mid c_i \text{ includes } o \tag{5.6}$$

In Equation 5.6, o represents the learning object to be ranked, c_i represent the ith course where o has been used. N is the total number of courses where o has been used.

Example (Figure 5.2): We assume that in response to a query, objects O_1, O_2, O_3, O_4 and O_5 are returned. From the information stored in the system, we know that O_1 is used in course C_1, O_2, O_3 and O_4 in C_2 and O_4 and O_5 in C_3. The hub value of C_1 (its degree in the graph) is 1, of C_2 is 3 and of C_3 is 2. The IT metric for O_1 is 1, the hub value of C_1. For O_2 and O_3 the value is 2, the hub value of C_2. For O_4 IT is the sum of the hub values of C_2 and C_3, 5. For O_5, it is 2. The order of the final result list ranked by IT would be (O_4, O_2, O_3, O_5, O_1).

Data and Initialization: The calculation of IT needs information from Learning Management Systems. Similarly to CST, IT uses the relationship between courses and objects. On the other hand, IT does not need information about the course at query time, so it can be used in anonymous web searches. Course-Object relationship can be extracted from existing LMS that contribute objects to the LOR and used as bootstrapping data for this metric. An alternative calculation of this metric can use User-Object relationships in case that LMS information is not available.

5.4.2 Personal Relevance Ranking Metrics

As discussed in Section 3, the Personal Relevance metrics should try to establish the learning preferences of the user and compare them with the characteristics of the learning objects in the result list. The most difficult part in these metrics is to

obtain transparently an accurate representation of the personal preferences. The richest source of information about these preferences is the attention metadata that could be collected from the user [Wasfi, 1999]. There are different ways in which this metadata could be used to determine a profile for each user or similarity between users. For example, [Mobasher et al., 2000] presents some strategies to build user profiles from Web access data and [Pampalk et al., 2005] discuss the generation of playlists based on user skipping behavior. The second step in this metric calculation is to obtain the characteristics of the objects. If metadata is present, this process is vastly simplified, because there already exists a description of the characteristics of the object. However, if metadata is not complete or inaccurate, contextual and usage information can be used to automatically generate the desired metadata values [Meire et al., 2007]. The following subsections present the calculation of two possible Personal Relevance metrics for learning objects.

Basic Personal Relevance Ranking (BP)

The easiest and least intrusive way to generate user preference information is to analyze the characteristics of the learning objects they have used previously. First, for a given user, a set of the relative frequencies for the different metadata field values present in their objects is obtained (Equations 5.7 and 5.8). In these equations, $val(o, f)$ represents the value of the field f in the object o. The frequencies for each metadata field are calculated counting the number times that a given value is present in the given field in the metadata. For example, if a user has accessed 30 objects, from which 20 had "Spanish" as language and 10 had "English", the relative frequency set for the field "Language" will be (es=0.66 , en=0.33). This calculation can be easily performed for each of the categorical fields (fields that can only take a value from a fixed vocabulary). Other types of fields (numerical and free text) can be also used in this calculation if they are "categorized". For example, the numerical field "Duration" that contain the estimated time to review the object, can be transformed into categorical clustering the duration values in meaningful buckets: (0-5min, 5-30min, 30min-1hour, 1-2hours, more than 2hours). For text fields, keywords present in a predefined thesaurus could be extracted. An example of this technique is presented in [Medelyan and Witten, 2006].

Once the frequencies are obtained they can be compared with the metadata values of the objects in the result list. If the value present in the user preference set is also present in the object, the object receives a boost in its rank equal to the relative frequency of the value. This procedure is repeated for all the values present in the preference set and the NF selected fields of the metadata standard (Equation 5.9). This metric is similar to that used for automatically recording TV programs in Personal Video Recorders [Pigeau et al., 2003]. The metadata of the programs watched by the user, such as genre, actors, director and so forth, is averaged and compared against the metadata of new programs to select which ones will be recorded.

$$cont(o, f, v) = \begin{cases} 1, & \text{if } val(o, f) = v \quad (5.7a) \\ 0, & \text{otherwise} \quad (5.7b) \end{cases}$$

$$freq(u, f, v) = \frac{1}{N} \sum_{i=1}^{N} cont(o_i, f, v) \mid o_i \text{ used by } u \quad (5.8)$$

$$BP(o, u) = \sum_{i=1}^{NF} freq(u, f_i, val(o, f_i)) \mid f_i \text{ present in } o \quad (5.9)$$

In Equation 5.7, o represents the learning object to be ranked, f represents a field in the metadata standard and v is a value that the f field could take. Additionally, in Equation 5.8, u is the user, o_i is the ith object previously used by u and N is the total number of those objects. In Equation 5.9, f_i is the ith field considered for the calculation of the metric and NF the total number of those fields.

Example: We assume that a given learner has previously used 3 objects: O_1, O_2 and O_3. O_1 is a Computer Science-related slide presentation in English. O_2 is a Computer Science-related slide presentation in Spanish. O_3 is a Math-related text document in Spanish. If the previously mentioned technique is used to create the profile of the learner, the result will be learner = [Classification (ComputerScience=0.67, Math=0.33), LearningResourceType (slide=0.67, narrative text=0.33), Language (en=0.33, es=0.67)] . The learner performs a query and in the result list are the objects O_4, O_5 and O_6. O_4 is a Computer Science-related text document in English, O_5 is a Math-related figure in Dutch and O_6 is a Computer Science-related slide presentation in Spanish. The BP value for O_4 is $0.67 * 1 + 0.33 * 1 + 0.33 * 1 = 1.33$. For O_5, it is 0.33. For O_6 it is 1.66. The order of the final result list ranked by BP would be (O_6, O_4, O_5).

Data and Initialization: The BP metric requires the metadata information about the objects previously selected by the users. This Identifier of the user and objects can be obtained from the logs of the search engine (given that the user is logged in at the moment of the search). Once the identifier is know, the metadata can be obtained from the LOR. A profile for each user can be created off-line and updated regularly. To bootstrap this metric, the contextual information of the user can be transformed into first profile. For example, if the user is registered in a LMS we will have information about his major and educational level. Also, information collected at the registration phase could also be used to estimate user age and preferred language.

User-Similarity Personal Relevance Ranking (USP)

The Basic Personal Relevance Metric relies heavily on the metadata of the learning object in order to be effective. But metadata is not always complete or reliable

[Sicilia et al., 2005]. A more robust strategy to rank objects according to personal preferences is to find the number of times similar users have reused the objects in the result list. To find similar users we can apply the SimRank algorithm, previously used to obtain the CST metric. A 2-partite graph contains the objects linked to the users who have reused them. The graph is folded over the object partition and a relationship between the users is obtained. The relationship graph is used to calculate the USP metric, as in Equation 5.11. The final calculation is performed adding the number of times similar users have reused the object. This kind of metric is that used, for example, by Last.fm and other music recommenders [Upendra, 1994] who present new songs based on what similar users are listening to; similarity is defined in this context as the number of shared songs in their playlists.

$$hasReused(o,u) = \begin{cases} 1, & \text{if } o \text{ used by } u \quad (5.10a) \\ 0, & \text{otherwise} \quad (5.10b) \end{cases}$$

$$UST(u,o) = \sum_{i=1}^{NU} SimRank(u,u_i) * hasReused(o,u_i) \quad (5.11)$$

In Equations 5.10 and 5.11, o represents the learning object to be ranked, u is the user that performed the query. u_i is the representation of the ith user. NU is the total number of users.

Example: We assume that there are four users registered in the system: U_1, U_2, U_3 and U_4. User U_1 has previously downloaded objects O_1, O_2 and O_3, user U_2, objects O_2, O_3 and O_5, user U_3, objects O_2, O_5 and O_6, user U_4, objects O_5, O_6. User U_1 performs a query and objects O_4, O_5 and O_6 are present in the result list. The SimRank between U_1 and U_2 is 2, between U_1 and U_3 is 1 and between U_1 and U_4 is 0. The USP metric for O_4 is $2 * 0 + 1 * 0 + 0 * 0 = 0$, for O_5 it is 3 and for O_6 it is 1. The order of the final result list ranked by USP would be (O_5, O_6, O_4).

Data and Initialization: The USP metric uses the User-Object relationships. These relationships can be obtained from the logging information from search engines (if the user is logged in during their interactions with the learning objects). The USP does not need metadata information about the learning objects and can work over repositories that do not store a rich metadata description. If no data is available, the metric returns 0 for all objects, not affecting the final ranking. To bootstrap this metric when there is not previous User-Object relationships information, the User-Course and Course-Object relationship obtainable from LMS systems could be used.

5.4.3 Situational Relevance Ranking Metrics

The Situational Relevance metrics try to estimate the relevance of the object in the result list to the specific task caused the search. In the learning object environment, this relevance is related to the learning environment in which the object will be used as well as the time, space and technological constraints that are imposed by the context where the learning will take place. Contextual information is needed in order to establish the nature of the task and its environment. When some description of the context is extracted from this information, it can be used to rank the objects. Again, these characteristics could be extracted from the object metadata or information already captured about the previous usage of the objects. The following subsections present two alternative methods to calculate Situational Relevance metrics.

Basic Situational Relevance Ranking (BS)

In formal learning contexts, the description of the course, lesson or activity in which the object will be inserted is a source of contextual information. Such information is usually written by the instructor to indicate to the students what the course, lesson or activity will be about. Keywords can be extracted from these texts and used to calculate a ranking metric based on the similarity between the keyword list and the content of the textual fields of the metadata record. To perform this calculation the similarity is defined as the cosine distance between the TF-IDF vector of contextual keywords and the TF-IDF vector of words in the text fields of the metadata of the object in the result list Equation 5.12. This procedure is base on the vector space model for information retrieval [Sparck Jones, 1972] [Salton and Buckley, 1988]. One parallel application of this type of metric has been developed by Yahoo for the Y!Q service [Kraft et al., 2005], that can perform contextualized searches based on the content of a web page in which the search box is located.

$$BS(o,t) = \frac{\sum_{i=1}^{M} tv_i * ov_i}{\sqrt{\sum_{i=1}^{M} tv_i^2 \cdot \sum_{i=1}^{M} ov_i^2}} \tag{5.12}$$

In Equation 5.12, o represents the learning object to be ranked, c is the course where the object will be used. tv_i is the ith component of the TF-IDF vector representing the keywords extracted from the course description, ov_i is the ith component of the TF-IDF vector representing the text in the object description. M is the dimensionality of the vector space (number of different words).

Example: We assume that an instructor creates a new lesson inside an LMS with the following description “Introduction to Inheritance in Java”. The instruc-

tor then searches for learning objects using the term "inheritance". The result list is populated with 3 objects. O_1 has as description "Introduction to Object-Oriented languages: Inheritance", O_2 has "Java Inheritance" and O_3 has "Introduction to Inheritance". The universe of words, extracted from the description of the objects would be ("introduction", "inheritance", "java", "object-oriented", "languages"). The TF-IDF vector for the terms in the lesson description is then (1/2, 1/3, 1/1, 0/1, 0/1). For the description in object O_1 the vector is (1/2, 0/3, 0/1, 1/1, 1/1). For O_2, it is (0/2, 1/3, 1/1, 0/1, 0/1). For O_3, it is (1/2, 1/3, 0/1, 0/1, 0/1). The cosine distance between the vector of the lesson description and O_1 is $(0.5*0.5+0.33*0+1*0+0*1+0*1)/\sqrt{(0.5^2+0.33^2+1^2)*(0.5^2+1^2+1^2)} = 0.14$. For O_2, it is 0.90 and for O_3, it is 0.51. The order of the final result list ranked by BS would be (O_2, O_3, O_1).

Data and Initialization: To calculate the BS metric, the only information needed is the text available in the context and the object metadata. The text information of the context should be provided at query time. The information needed to bootstrap this metric is a corpus with the text available in the objects metadata to provide the value of the IDF (Inverse Document Frequency) of each word.

Context Similarity Situational Relevance Ranking (CSS)

A fair representation of the kind of objects that are relevant in a given context can be obtained from objects that already have been used under similar conditions. For example, if we considered the case where the course represent the context, objects already present in the course are a good representation of what is relevant in that context . Similarly to the calculation of the BP metric, the N objects contained in the course are "averaged" to create a set of relative frequencies for different fields of the learning object metadata record (Equation 5.13). This set of frequencies is then compared with the objects in the result list. The relative frequencies of the values present in the object's metadata are added to compute the final rank value (Equation 5.14). This method can be seen as creating a different user profile for each context (in this case seen as course) where the learner is involved. This method can be applied to more complexes description of context. For example, if query is used during the morning, a frequency profile can be obtained from objects that the learner has used during similar hours in the morning. That "time of the day" profile can latter used to rank the result list using the same approach presented above. Other contextual descriptors that can be used are place, type of task, access device, and so forth.

$$freq(c, f, v) = \frac{1}{N} \sum_{i=1}^{N} cont(o_i, f, v) \mid o_i \text{ included in } c \tag{5.13}$$

$$CCS(o,c) = \sum_{i=1}^{NF} freq(c, f_i, val(o, f_i)) \mid f_i \text{ present in } o \tag{5.14}$$

In Equations 5.13 and 5.14, o represents the learning object to be ranked, c is the course where the object will be used. o_i is the ith object contained in the course c. f represent a field in the metadata standard and v is a value that the f field could take. $val(o, f)$ return the value of the field f in the object o. f_i is the ith field considered for the calculation of the metric and NF the total number of those fields. $cont(o, f, v)$ is presented in Equation 5.7.

Example: We assume that a learner issues a query from course C. Course C has three objects O_1, O_2 and O_3. O_1 is flash animation whose duration is between 0 to 5 minutes and is for higher education. O_2 is another flash animation whose duration is between 5 to 10 minutes and is for higher education. O_3 is a video of 20 minutes also targeted to higher education. The profile for that specific course will be $C = [LearningResourceType(animation = 0.67, video = 0.33), Duration(0 - 5min = 0.33, 5-10min = 0.33, 10-30min = 0.33), Context(higher-education = 1)]$. The result list contains the following objects: O_4, a text document with estimated learning time of 1 hour for higher education; O_5 a video whose duration is between 0 to 5 minutes targeted to primary education and O_6 a flash animation whose duration is between 10 to 30 minutes, targeted to higher education. The CSS value for O_4 is $0 * 0 + 0 * 0 + 1 * 1 = 1$. For O_5, it is 0.66. For O_6, it is 2. The order of the final result list ranked by CSS would be (O_6, O_4, O_5).

Data and Initialization: The CSS metric depends on the contextual information that can be captured during previous interactions of the user with the learning objects, as well as, during query time. The most basic context that can be obtained from a LMS is the course from which the user submitted the query. Also, using the course as the context facilitates to capture of information about previous objects used in the same context, helping in the bootstrapping of the metric. Nevertheless, more advanced context definitions are allowed to calculate variations of this metric, at the cost of a more detailed logging of user actions.

5.4.4 Ranking Metrics Comparison

Different metrics estimate different relevance dimensions and consume different types of raw data. Therefore, not all the metrics can or should be implemented in different environments. For example, if the searching environment does not include information from a LMS or similar system, some of the metrics (CST and CSS) for example, could not be calculated. This subsection present a comparison between the proposed relevance ranking metrics based on their related relevance characteristics and the origin of the data needed for their calculation.

Table 5.2 presents the correspondence of the ranking metrics with the Relevance or Quality Characteristics presented in Section 3. It can be clearly seen

that each metric only covers a small percentage of the characteristics. Also their correspondence has different levels. A metric can correspond strongly with some characteristics and weakly with others. For example, CST, by its definition, can be used to estimate the Learning Goal of the user. However, given that it is based on the similarity of courses, it is also correlated in a lower level with the Learning Setting (Similar courses sometimes use similar learning approaches), and, even a weaker level, with the Language (courses sharing the same objects usually are in the same language) and the Cultural characteristics (the decision to chose similar material could be related to the cultural context). Another interesting example is BT. This metric is based on the metadata of the object that a learner has previously used. While it is designed to estimate the Personal Relevance, the presence on the metadata of contextual related fields like Duration, Interactivity Type and Technical Requirements also correlate it with some of the Contextual Relevance characteristics. Some metrics need to be adapted to address different Contextual Relevance characteristics. For example, CSS can be calculated from different type of contextual information to be used to estimate the relevance for different Learning Settings, Times or Spaces.

Table 5.2 also shows that the proposed metrics as a whole correspond with most of the relevance characteristics. However, Learner Motivation and the Contextual characteristics are not well covered. This is a reminder that the proposed metrics are not a comprehensive set, but a first formal proposal of multidimensional relevance ranking metrics for learning objects.

The implementation of the metrics in real systems is bound to be dependent on the availability of the raw data for their calculation. Table 5.3 present a summary of the data needed for the different metrics. It is important to note that some data is required at query time (QT) , for example the user identification in the Personal Relevance metrics. Other information is needed for Off-line calculations (OL), for example the similarity between queries used in BT metric can be pre-calculated from the Query-Object Relationship. As expected, all metrics rely in usage and contextual information provided by a LMS or the capture of Contextualized Attention Metadata (CAM). If only information from a LMS is available, the best metrics to cover most of the relevance dimensions will be CST, IT and BS. If the system is not connected to a LMS, but has CAM from the users, then BT, BP and USP are the most appropriate metrics.

All the metrics need some sort of off-line calculation. Even BT, which only use words in the context and words in the text of the object, needs to have an index with the frequency of different words in order to be calculated. Any system that implements those metrics is bound to provide some kind of temporal storage. Moreover, depending on the scale of data collection, the solutions for data storage and processing could be the principal concern in the metric calculation system [Brin and Page, 1998].

In summary, the different origin and target of the ranking metrics makes them

Table 5.2: Correspondence of the Ranking Metrics with the Quality Characteristics and Relevance Dimensions. S = Strong, M = Medium, W = Weak, A = After Adaptation

	BT	CST	IT	BP	USP	BS	CSS
Topical Relevance							
Learning Goal	S	S	M	M	M	M	M
Personal Relevance							
Motivation					W		
Culture	W	W			S		
Language	W	W		S	S		M
Level				S	S	W	M
Accesibility				S	S		
Situational Relevance							
Learning Setting		M		W		S	S
Learning Time				M			A
Time of Learning							A
Geo. Space							A
Learning Space				W			A

strong when they are seen as a group, but weak if they are taken alone. That is the reason why metrics in real world search engines are combined in order to provide a final rank. Next section will discuss different methods to combine all the proposed ranking metrics into a unique LearnRank estimation.

5.5 Learning to (Learn)Rank

In order to be useful, the different metrics should be combined to produce a unique ranking value that could be easily used to order result lists. This combination of metrics is not a trivial task. A series of workshops, titled "Learning to Rank" [Joachims et al., 2007], has been conducted in recent years to discuss and analyze different methods to combine metric values to produce a final, optimal ranker. All these methods share a similar approach: 1) Obtain human generated values of relevance (explicitly or implicitly) for different result lists. 2) Calculate the metrics for the same objects. 3) Use the metrics values as input and the human generated relevance values as output to train some machine learning algorithm. And 4) use the resulting trained machine learning model as the final ranking metric.

The most basic approach to learn a ranking function based on numerical metrics is multivariable linear regression [Herbrich et al., 2000]. In this approach, the human ranking is considered the dependent variable and all the ranking metrics the independent variables. The coefficients that produce the best fit of the learned

Table 5.3: Source Data needed to calculate the Ranking Metrics. QT = Query Time, OL = Off-Line

	BT	CST	IT	BP	USP	BS	CSS
Search Engine							
Query	QT						
Repository							
Metadata				QT		QT	
Content							
Learning Management System							
User ID				QT	QT		
Course ID		QT					QT
Course Information						QT	
Course-Object Rel.		OL	OL				OL
Contextualized Attention Metadata							
Query-Object Rel.	OL				OL		
User-Object Rel.				OL			
Advanced Context							OL

function against the human generated relevance values are estimated. The final function takes the form of a linear combination of the metrics. While simple, the main problem with this approach is that it over-constrains the problem to solve. We want to learn the order in which objects should be ranked, not the actual value assigned to each object [Richardson et al., 2006]. More advanced approaches do not use the numerical value of human relevance estimation as the target for learning, but only the relative position on the human generate rank. The machine learning algorithm is trained to rank the objects in the same order as a human would do, without caring about the individual rank values. These approaches have been shown to be much more effective [Yan and Hauptmann, 2006].

To generate LearnRank, a metric that combines the different relevance metrics to rank learning objects, this chapter will use one of the order-based learning strategies. The selected algorithm was RankNet [Richardson et al., 2006]. The selection was based on the effectiveness of this algorithm [Raykar et al., 2007], as well as, its commercial success (It is the rank learn algorithm behind MSN Search). RankNet uses a Neural Network to learn the optimal ranking based on the value of the original metrics. The training is conducted using pairs of results ranked with respect to each other. The Neural Net is trained to produce the smallest amount of error with respect to the training pairs (cost function). For example, if it is known that R1 should be ranked higher than R2 and the Neural Network output indicates that LearnRank(R2) is higher than LearnRank(R1), a corrective coefficient is backpropagated to adjust the weight of the neurons of

the Net. The best set of coefficients is selected based on the ones that produce the lower difference between the calculated and human based ordering. More details about the properties of RankNet algorithm are presented in [Richardson et al., 2006]. The main advantage of using a learning mechanism based on relative relevance is that the human generated relevance data needed to learn and improve the ranking can be automatically extracted from the interactions of the users with the search (or recommendation) tool. It has been demonstrated that users review the result list from the first to the last item in order [Joachims and Radlinski, 2007]. Therefore, the position of the object selected gives information about the relative relevance of it against previous objects. For example, if a user confronted with a result list selects the third object only, it means that she considered it of higher relevance than the first and the second objects. That information could be converted to relative relevance pairs that could be fed into the RankNet algorithm in order to improve the ranking. The next time that the user is confronted with the same result list, the third object should be in a higher position.

5.6 Validation Study

In order to evaluate the potential impact that the proposed metrics could have on the relevance ranking of learning object searches, an exploratory study has been performed. This study consisted of a study in which subjects were asked to simulate the creation of a lesson inside a LMS. The subjects were required to quantify the relevance of a list of top-10 learning objects, ranked using the default text-based TF-IDF metric provided in Lucene [Hatcher and Gospodnetic, 2004]. They also had to select from the list objects they considered appropriate for the lesson. The TF-IDF metric is compared with the subject's ranking to create a base line performance score. The proposed basic metrics for each one of the relevance dimensions, as well as the best-fitting linear combination and trained RankNet are then used to reorder the list. Finally, the re-ordered lists are compared against the human generated rank.

5.6.1 Study Setup

Ten users, eight professors and two research assistants from the Computer Science field, were required to create ten lessons related to different computer science concepts presented in Table 5.4. In each lesson, the subjects were required to write a brief description of the lesson for hypothetical students. The subject was then presented with a list of ten objects. These objects were obtained from a LOR containing all PDF learning objects currently available in the MIT OCW website[1] (34,640 objects). The objects belong to all the majors taught at MIT, not only to Computer Science. This LOR was queried with a different query

[1]MIT OCW. http://ocw.mit.edu

phrase for each lesson, as listed in Table 5.4. The title, description and keyword fields were text-matched with the query terms. The top-10 objects of each result list were used in the study. The subject then graded the relevance of each object to the lesson, to which end they used a 7-value scale, from "Not relevant at all" to "Extremely Relevant". Moreover, subjects were required to select the objects they would include in the lesson. The data recollection was conducted using a Web application[2].

The initial rank of the objects was performed by the Lucene ranking algorithm, which is based on vector space retrieval [Hatcher and Gospodnetic, 2004]. This algorithm can be considered a good representation of current algorithmic relevance ranking. The basic topical relevance metric (BT) was calculated counting the number of times each object was selected to be included in the lesson. The selection of each subject was left out for his individual relevance evaluation. The basic personal relevance metric (BP) was calculated using historical information about the objects which subjects had published in their LMS courses. Three fields were captured: main discipline classification, document type and context level. These fields were selected on the basis of information available in the LOM record of the MIT OCW learning objects and the metadata available from objects previously published by the participants. The specific course of previous objects was not taken into account because the participants not necessarily teach the study topics. The basic situational relevance ranking (BS) captured the text fed by the subjects into the description of the lesson. Any stopwords were eliminated and the resulting keywords were used to expand the query made to Lucene. The revised 10 objects were extracted then from the new result list.

Once the values of the metrics were calculated, they were combined. In order to have into the combination a metric from each one of the relevance dimensions, the relevance score provided by Lucene was used as an estimate of the Algorithmic relevance. Two methods were used to obtain the combination of the metrics: First the assigned human relevance was used to compute the coefficients of the linear combination of the metrics through multivariable linear regression. This combination will be referenced as "Linear Combination". Second, the relative relevance pairs, also generated as result of the human ranking, were used to train a two-layer RankNet with 10 hidden neurons. The values of the different metrics were used as the input of the neural net. This combination will be referenced as "RankNet Combination".

In order to avoid over-fitting the combined metrics, the training-testing was conducted using a 10-fold approach. The human generated rank data was divided in 10 sets according to the reviewer to which it belonged. The learning algorithm was trained with 9 of the sets and tested in the remaining set. The results reported in this study are the ones obtained in the test phase.

Once all the metrics and the two combinations were calculated, they were

[2]On-line study application. user:test, password:test. http://ariadne.cti.espol.edu.ec/Ranking

Table 5.4: Task performed during the study and their corresponding query phrase

#	**Lesson to Create**	**Query Phrase**
1	Inheritance in object oriented languages	inheritance
2	Algorithmic complexity	complexity
3	Introduce the concept of computer networks	networks
4	Introduce Human Computer Interaction concept	human computer interaction
5	Explain tree structures	trees
6	Xml markup	xml
7	Introduce the concept of operating system	operating system
8	Explain the artifical neural networks	neural networks
9	How to normalize database tables	normalization
10	Explain routing of packages in computer networks	routing

compared against the manual rank performed by the human reviewers. In order to measure the difference between the manual rank and each of the automated ranks, a variation of the Kendall tau metric [Fagin et al., 2004] which deal with ties in the rank was used. This metric measures the distance between two permutations and is proportional to the number of swaps needed to convert one list into the other using bubble sort. If two ranks are identical, the Kendall tau is equal to 0, if they are in inverse order, Kendall tau is equal to 1.

5.6.2 Results

Only 12% of the objects presented to the users were manually ranked "Very Relevant"(5), "Highly Relevant"(6) or "Extremely Relevant"(7). This implies that pure algorithmic relevance ranking does a mediocre job at providing relevant results to the user in the top-10 positions of the result list, especially if the repository contains a large amount of objects in different topics. Some searches, for example "human computer interaction" return almost only "Not Relevant at All" results, even if in the repository there were material for courses about Interface Design and Human Centered Computing. This was due to the fact that several unrelated objects in the test repository contained the words "human", "computer" and "interaction".

The Kendall tau distance between the Base Rank (based on the Lucene algorithmic relevance metric) and the human ranking has a mean value of 0.4 for all

the searches. For query terms that are common to appear in other disciplines besides Computer Science such as "trees"(5) and "human computer interaction"(6) it borders 0.5. This means that there is no relation between the relevance given by the automatic ranker and the human review. For example, the Lucene algorithm considered of high relevance objects about the biological evolution of natural trees. However, for very specific Computer Science query terms, such as "xml"(6) and "operating systems"(7) it provides a lower value, 0.3, implying a stronger correlation between manual and automatic ranks. This tau values are consistent with the low quality of the retrieval.

If the top-10 results provided by Lucene are reordered using the basic metrics, topic relevance metric (BT) provides the best ranking, with an improvement of 31% over the Lucene ranking. The situational relevance ranking (BS) provide an improvement of 21%. The less performing metric was the personal relevance (BP), but it alone still produces an improvement of 16% over the base line ranking. If the metrics are combined, the Rank Net combination produces a much better result that the Linear Combination and any of the individual metrics with an improvement of 51% over the Lucene ranking. The Linear Combination, on the other hand, produces a result (22%) comparable with the ones of the individual metrics. The summary of the results, as well as their statistical significance can be seen in Table 5.5. Figure 5.3 and 5.4 show the disaggregated tau values for each one of the queries for the individual and combined metrics.

Table 5.5: Average distances between the manual ranking and the calculated metrics and the average improvement over the Base Rank

Ranking Metric	**Kendal τ**	**Improv.**	**Paired T-Test (df=99)**
Lucene score	0.4046	–	
Basic Topical	0.2790	31.0%	t=9.50, p=0.000
Basic Personal	0.3392	16.2%	t=2.93, p=0.004
Basic Situational	0.3183	21.3%	t=6,42, p=0.000
Linear Combination	0.3139	22.4%	t=7.88, p=0.000
RankNet Combination	**0.1963**	**51.5**%	t=5.71, p=0.000

5.6.3 Discussion of the Results

The basic topic relevance metric (BT) provides the best correlation with manual ranking of the individual metrics. It was the metric most directly related to human choice, as normally highly relevant items were selected for inclusion in the lessons. It performed better than the base-line ranking in all the searches. However, this result has been affected by the fact that all the subjects participating in the study belong to the same field. In a real situation a lower performance is expected as noise is present in the data used to calculate this metric. This noise comes from

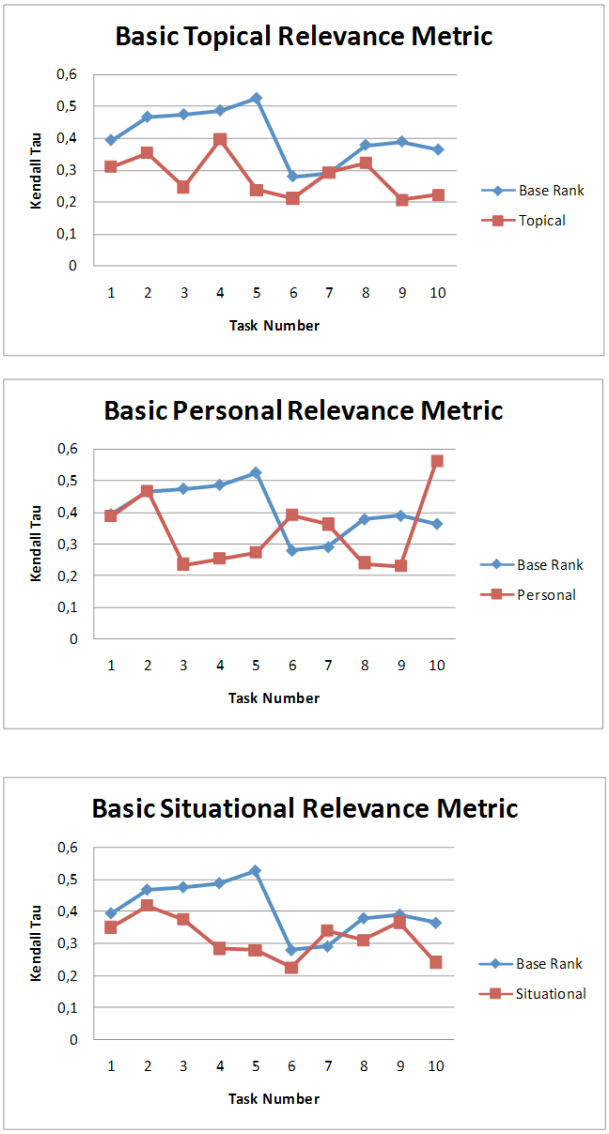


Figure 5.3: Results of the Kendall tau distance from the manual ranking of the individual metrics

unrelated searches using similar query terms. For example, if Biology professor were also using the same search engine, it is expected that the query term “tree”

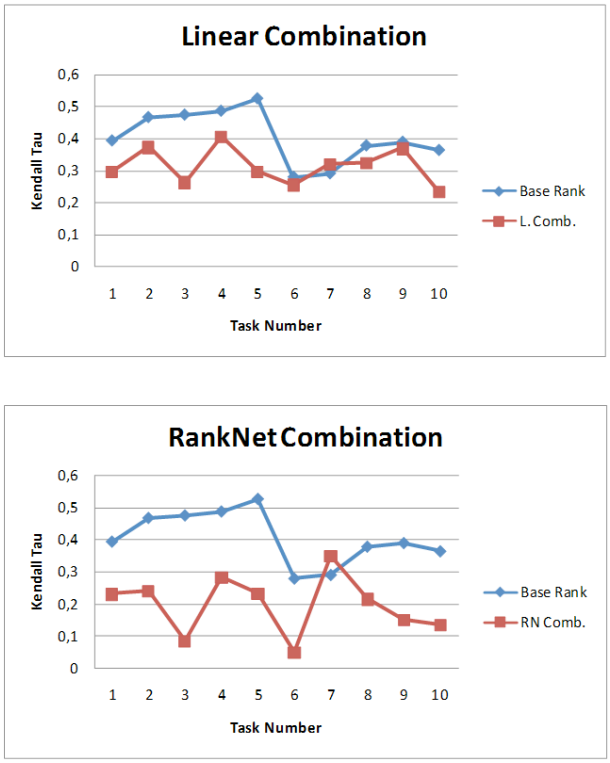


Figure 5.4: Results of the Kendall tau distance from the manual ranking of the combined metrics

produce two different patterns of selections. This problem can be solved applying a more advanced Topical Relevance, such as CST.

The basic personal relevance metric (BP) presented some problems in certain queries. This can be explained as errors or unexpected values in the metadata records of the objects. While the object was relevant for a given lesson, metadata values do not always match user preferences. For example, in search number 10 ("routing"), the topical classification of the objects was "Electrical Engineering", different from the "Computer Science" value that all the subjects had in their profile. Another case which exemplifies this problem was present in search number 1 ("inheritance"). The objects found more relevant came from a Programming course of the Civil Engineering department. This value was different from the value present in subject's profile. This problem could be address measuring the distance between different metadata values instead of the current Boolean comparison.

An interesting method to measure this distance using classification ontologies is proposed by Olmedilla in [Olmedilla, 2007].

Basic situational relevance metric (BS) provided an improvement over the base-line rank in all but one search. It performed better for ambiguous query terms (note search number 4 and 5) while almost not affecting the performance of very specific query terms (searches 6 and 7). This result was expected given similar studies on query expansion using contextual descriptions [Kraft et al., 2006].

By far the best performance was obtained by the RankNet Combination of the metrics. It outperformed the base-line ranking and all the other rankings in most of the searches. However, given that it is still a combination of the metrics, it is bound to under-perform individual metrics in specific situations, especially when all the metrics provide a very similar ranking. The clearest example of this effect is the tau value obtained for the "operating system" (case 7). In this case, all the metrics provide a similar tau value. The neural network does not have sufficient input information to produce a better ranking. The Linear Combination, on the other hand, behaves like an average between the individual metrics. It is better than the base-line and the BP, but it is worse than BT. The use of this linear regression is not recommended to learn the ranking function.

In conclusion, the combination of the ranking using RankNet provides a significant increase in the performance of the ranking compared with the base-line Rank (Lucene text-based ranking). These results suggest that a full-fledged implementation of these metrics in a real environment, learning from the continuing interaction of the users with the system, will lead to a meaningful, scalable and transparent way to rank learning objects according to their relevance for a given query, topic, user and context.

5.6.4 Study Limitations

Given its exploratory nature, the study has several limitations that are being taken into account. The most important of these are:

- *Reordering of the same objects.* Only objects present in top-k results of the algorithmic relevance search were used. The main reason for this choice was to limit the amount of manual relevance ranking needed. This limitation, nonetheless, does not affect the result of the evaluation for two reasons. First, the evaluation only compares relative ordering and not absolute relevance score. The inclusion of an additional element in the second position does not alter the fact that the first was better ranked than the third. Second, the bias introduced works against the proposed metrics as they were not able to bring more relevant results from the post-10 objects. Given that the results show that the metrics outperfomed the base-line rank, the elimination of this bias will only reinforce the conclusion.

- *Use of partial rankings.* The subjects were not required to provide a total ranking (that is to fully order the objects from most relevant to least relevant). A more relaxed approach where two or more objects in the list could have the same relevance value was preferred, as humans tend to evaluate relevance in a fuzzier way. This choice complicates the analysis of the results as several ties were present (a variation of the most commmon Kendal tau metric was used), but it did not force the subjects to artificially rank one object over another.

- *Limited subject variety.* All the subjects were selected from the same field and had similar teaching styles (their selection was based in the fact that all of them belong to a group applying the constructivist approach [Fosnot, 1996] in their classes). While this homogeneity boost the result of the basic topical metric because of the absence of noise in the data, it can be seen also as the result of applying a filtering based on user topic preference before calculating BT. Future evaluation in a real system should work with a multidisciplinary sample.

5.7 Conclusion

The main contribution of this chapter is the development and evaluation of a set of metrics related to different dimensions of learning object relevance. The conclusions of this chapter can be summarized in the following points:

- Information about the usage of the learning objects, as well as the context where this use took place, can be converted into a set of automatically calculable metrics related to all the dimensions of relevance proposed by Borlund [Borlund, 2003] and Duval [Duval, 2005]. This information can be obtained implicitly from the interaction of the user with the system.

- The evaluation of the metrics through an exploratory study concludes that all the proposed basic metrics outperformed the ranking based on pure text-based approach. This study shows that the most common of the current ranking methods is far from optimal, and the addition of even simple metrics could benefit the relevance of the results for LOR's users.

- The use of methods to learn ranking functions, for example RankNet, leads to a significant improvement of more than 50% over the base-line ranking. This result is very encouraging for the development of ranking metrics for learning objects, given that this improvement was reached with only 4 metrics as contributors to the ranking function.

The metrics proposed here have the characteristics needed by the theoretical LearnRank. The very nature of the presented metrics and their combination

makes them scalable. They consume information implicitly collected through attention metadata, making them transparent for the end user. Finally, the results of the study suggest that they are good estimators of the human perception of the relevance of the learning object, making them at least more meaningful than text-based algorithms. Even if not proposed as an optimal solution, these metrics could be used to improve current LORs. More important for this field of research, these metrics could be considered the new base-line against which new, more advanced metrics could be compared.

The previous and the current chapters present two sets of metric calculation oriented to improve the labelling and selection stages during the Learning Object life cycle. Having automated calculation that can convert the vast amount of information related to the learning object, its use and context, is the most straight-forward way to create smarter tools that will facilitate the use of learning objects in mainstream learning. Next chapter presents how these metrics can be efficiently implemented in a Service Oriented Architecture and used to improve existing Learning Object tools.

Chapter 6

Metrics Service Architecture and Use Cases

6.1 Introduction

The points of interaction between users and the concept known as Learning Object Economy [Campbell, 2003] are the tools that they use to create, publish, find, adapt, reuse and manage learning objects. These tools, however, are very immature [Dodani, 2002] [Duval and Hodgins, 2003] [Ochoa, 2005] compared with their counterparts in other fields. We can cite several examples: Authors of Web pages or scientific papers do not need to manually index their work in order to make it findable on the web. In the case of learning objects, most of the current implementations of publishing tools require the user to fill long electronic forms [Duval and Hodgins, 2003] in order to make their content available. Web search engines usually let us find relevant material among billions of web pages [Gulli and Signorini, 2005]. Learning object search engines, however, are not enough to sort limited amount of results returned by federated queries. There is no readily available equivalent in the learning object technologies for the Amazon's book recommending feature [Linden et al., 2003]. The lack of maturity in the end-user tools cause a low level of adoption of learning object technologies among instructors and learners.

In order to improve the adoption of learning object technologies, smarter and friendlier end-user tools must be developed. These tools should capitalize the vast amount of information that is present in the learning object metadata and other sources as context and usage. To be exploitable, that information should be automatically measured and processed to extract deep knowledge of the characteristics, relations, usefulness, behavior and recommended usage of individual learning objects, as well as, complete learning object repositories. In the previous chapters,

we presented automatic quantitative measurements (metrics) for learning objects. As mentioned during the introduction, one of the objective of the metrics is to improve the performance and usability of the current learning object tools. For example, the number of times an object is reused in a similar context seems to be a good predictor of the relevance of a learning object to a given user. This calculation can be used inside the ranking algorithm of a learning object search engine to improve the relevance of the first page of the result set presented to the user.

The final step to implement the metrics is to design a software architecture to provide access to the required data, storage for intermediate and historical results and mechanisms to be queried about the metric values by interested applications. This chapter proposes such architecture to compute and integrate the metrics into current Learning Object tools. The structure of this chapter is as follows: Section 2 presents a software architecture based on Services. Section 3 presents how the metrics are calculated, as well as their scalability. Section 4 discusses several use cases where the metric service is being integrated to improve tools been deployed in real applications. The chapter closes with conclusions of the implementation experiences. Additionally to this chapter, Appendices A and B describe the Metrics Services Interfaces.

6.2 Service Oriented Architecture

Any type of implementation of the proposed metrics should interact with at least three main types of systems: Attention Metadata Repositories [Najjar et al., 2006], Learning Object Repositories and End-User Tools (Learning Management Systems, Learning Activity Management Systems [Dalziel, 2003], authoring tools, search engines, etc.). The interdependences with these systems present several challenges in the instantiation of the metrics:

1. The metrics must be accessed by several, heterogeneous systems: The cost of implementing the metrics in different languages and platforms could be prohibitive. However, current learning object tools and repositories are very heterogeneous in their implementation. An example could be a PHP based LMS, such as Moodle [Cole and Foster, 2007], that recommends objects available in a Java-based LOR, such as ARIADNE [Duval et al., 2001] and the information about the user actions is stored in log files accessible through a Perl interface. While this scenario could seem extreme, it is actually very common.

2. The metrics can be added, changed or become obsolete at a fast rate: Giving the immature status of the metrics, it would be expected that new metrics are to be added continuously. Also, the way in which the metrics are calculated could be vastly optimized. Obsolete metrics will be removed from the system

to increase the efficiency of the calculation. All these changes should have a minimum or zero impact on the other systems that use or are used by the metrics.

3. Running the metrics requires vast amount of storage and processing power: It would not be expected for each LMS implementation to have a copy of the metric calculation running locally. The resources needed to process large amounts of attention metadata, compute the required metrics and store intermediate and historical results are not neglectable. The metrics calculation should be shared between several systems in order to be cost effective.

4. Different tools need different services: The service implementation should be modular given that different types of tools require different kind of services. For example, an authoring tools is maybe interested in measuring the quality of the metadata provided by the user. LMSs, on the other hand, could be more interested in metrics for recommendation. LORs could be interested in both metadata quality and ranking metrics. Moreover, different system could be interested in just one or two specific metrics out of all the offered metrics.

5. Cost of integrating them into existing tools should be minimal: Ideally, integrating the metrics into existing tools should be as easy and fast as possible. If using the metrics require extensive rewrite of the code base of the host tool, the chances of being adopted decrease.

A proven approach to cope with the mentioned challenges are Service Oriented Architectures (SOA) [Erl, 2004]. In SOA, the desired functionality is provided to different applications in the form of "services". Services are small modules that provide a defined functionality through a previously agreed interface. A common implementation of SOA are so called "Web Services". Web Services are not more than services that can be called through the Internet with a defined set of protocols. Web Services have the advantage that they can be remotely deployed and could be accessible by various systems simultaneously. Also, the sets of protocols needed to communicate with Web Services (SOAP [Curbera et al., 2002], XML-RPC [Laurent et al., 2001], REST [zur Muehlen et al., 2005]) are widely available in all major platforms. This specific characteristic helps us to solve our challenge number one. Any tool, regardless of the language or platform in which it is implemented, could access Web Services. The information hiding generated by the use of a defined interface, lets us face challenge number two. The actual implementation of the metrics can be changed without any effect on the calling applications beside a faster or better response. Also, new metrics could be added as new services or as new options inside existing services with the only condition that the interface is maintained. In the case of obsolete metrics, the mapping to new, improved versions can be made transparently by the service

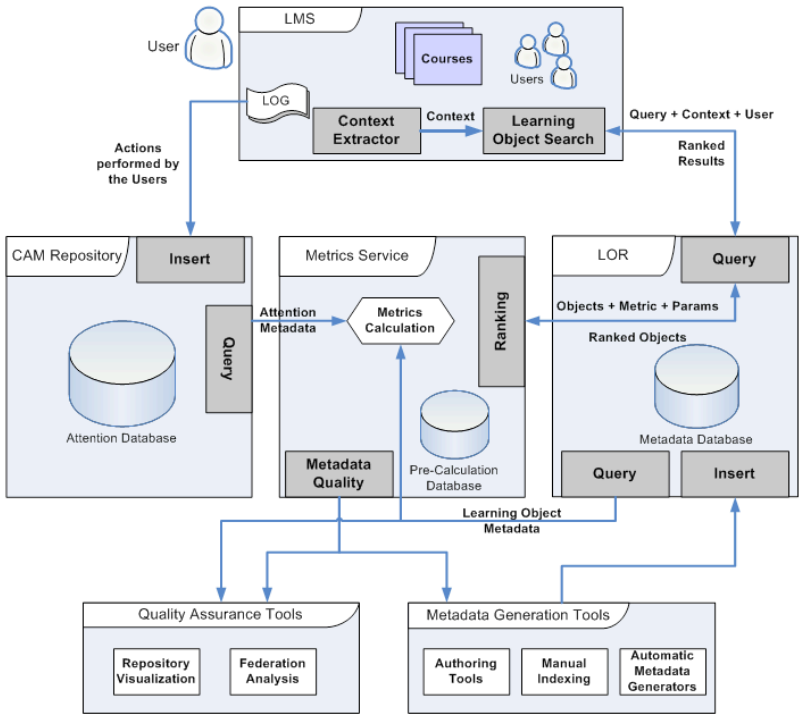


Figure 6.1: Architecture for Metrics Services

without any change at the calling application. Using Web Services also enables the sharing of the metrics between several tools. This feature addresses our challenge number three. Given that the metrics do not need to be installed in any specific tool, they can be deployed on a Web Server and they can be accessible for any application that has Internet access. Finally, adding code to call a Web Service inside existing applications is usually less cumbersome than installing and using a library [Alonso et al., 2004]. The major changes in the code will be related to how to use the metric values to improve the functionality of the tool, rather than how to get those values. These characteristics help us directly in our fourth challenge. Due to their advantages, Web Services are a common choice to generate interoperability and encapsulate behavior in the design of interoperable learning tools [Liu et al., 2003b].

Based on the previous analysis, we present an architecture for the Metrics Services (Figure 6.1). The flow of information in this architecture begins with the End-User tool (here exemplified by a LMS) where users (teachers and learner)

search and interact with learning objects. All the actions performed by the users in the End-Tool are then logged into a Contextual Attention Metadata (CAM) repository (described in detail in [Najjar et al., 2006]). The LMS also interacts with a Learning Object Repository, submitting queries and receiving learning objects. The LOR, in order to provide relevance ranked results, calls the Ranking Metric Service providing the parameters needed to the desired metric calculation. The LOR would receive a list with object identifiers ordered according to the required metrics. Other tools (exemplified in the graph by the Quality Assurance and the Metadata Generation Tools) would connect to the Metadata Quality Metrics Service to obtain information of one or more metadata instances just created or stored in the LOR. Given that all the different components (with exception of the End-User tools, such as LMS, Authoring Tools and LOR Administrators) provide a Web Service interface, the coupling between them is low and they can be interchanged for interface-compliant replacements without affecting the workings of the system as a whole.

One advantage of this architecture is that several End-User tools could share the same Metrics Service. Moreover, being also services, the LOR and the CAM Repository could have different implementations behind the same interface. For example, instead of the centralized LOR, the Query interface could give access to a federated network of repositories. In the same way, the actual implementation of the CAM repository could be based on the LMS logs or on capturing the users actions in their browsers. The most standard interface to access LORs is defined by the SQI standard [Simon et al., 2005]. The format that CAM could take is defined in [Najjar et al., 2006]. In a similar way, we propose two service interfaces: Metadata Quality Metrics Interface (MeQuMI) and the Ranking Metrics Interface (RaMI). These interfaces are presented in Appendices A and B.

The proposed architecture and interfaces can be used by tools implementors to make use of the Metrics Service. However, it does not describe how the metrics should be calculated. The next section describes the inner workings of a first implementation of the Metrics Service.

6.3 Implementation of the Metric Service

One of the main pre-requisites for the metrics is that they should be scalable. That means that the implemented Metric Service should be able to provide the metric value in a reasonable amount of time even if the metric calculation involves a large amount of objects. This section explains how this goal was reached for each one of the proposed metrics.

6.3.1 Metadata Quality Metrics

In chapter 4, 11 metrics to estimate the quality of a metadata instance are proposed. These metrics can be calculated for existing instances in the repository through the Repository or Subset level calls or for a new instance through the Instance level calls (Appendix A). In this subsection, we present how these metrics are calculated from raw data in an scalable way.

- Completeness (Qcomp): To calculate the Qcomp value of a metadata instance, the presence or absence of value for each metadata field is counted. The final count is divided by the number of possible fields in the metadata profile. The source information is taken from the contents of the metadata instance. The output is a number indicating the percentage of filled fields. The time to calculate this metric is proportional to the number of instances being evaluated (n) and the number of fields on the metadata description (F). Being O(nF), the computation time grows only linearly with the number of instances and, therefore, the computation is scalable for large values of n or F. In the case of Repository or Subset Levels, the value of Qcomp can be calculated off-line for all the instances present in the repository. This information can be stored for each instance in a local pre-calculation database. With this technique, the response time for the Repository and Subset calls can be reduced to O(1). In the case of Instance level, the instance is not known apriori and no off-line calculation is possible. The minimum response time in this case is O(n), considering a fixed number of metadata fields.

- Weighted Completeness (Qwcomp): The calculation of this metric is very similar to Qcomp. However, Qwcomp needs an external source of data: the importance of each field. As mentioned in chapter 4, this information can be explicitly (experts) or implicitly (data mining) collected. The response time for the Repository and Subset level is O(1) if off-line calculation is used. In the case of Instance level, the calculation is always O(n). The same discussion to justify the scalability of Qcomp can be applied to Weighted Completeness.

- Accuracy(Qaccu): To calculate Qaccu, two sources of information are needed: the metadata instance and the textual content of the document. To initialize Qaccu, a document-word matrix with the relative frequencies of different words in the documents present in the repository is built. A scalable LSA algorithm, such as [Bingham and Mannila, 2001], is then applied to reduce the length of this vector to reduce the noise produced by synonymous words. This operation can be executed in O(N), with N equal to the number of metadata instances in the Repository. Updating this matrix, however, requires the recalculation of the whole matrix. This reduced matrix is used to calculate the TFIDF cosine distance [Salton and Buckley, 1988] between

the text extracted from the metadata instance and the described document. This operation takes O(n), where n is the number of objects in the call to the metric. If the metric is pre-calculated, the calls to this metrics take O(1). Initially, the calculation of Qaccu can only be obtained at Repository and Subset levels. At Instance level, because the lack of information about the object, this metric is not available.

- Categorical Fields Information (Qcinfo): To calculate the Qcinfo, the Metric Service needs access to all the metadata instances stored in the repository, as well as the metadata of the analyzed instances. To calculate the entropy of each possible value in the categorical fields, the number of times that a specific value appears in the metadata stored in the repository is counted. The relative frequency of the count can be regarded as the probability of the value to be present in the metadata repository. Once the entropy for each field is stored, the Qcinfo can be obtained from the Information Content of each field in the analyzed instance. The initialization of the database with the entropy values is an operation dependant on the number of instances already present in the repository (N). Updating the database is a single operation for each new metadata instance. Calculating the Qcinfo, once the entropy values are available, is only dependant on the number of instances involved in the call (n). The Qcinfo metric can also be pre-calculated for the Repository and Subset calls, reducing the response time to near O(1). In the case of calls at the Instance level, the common response time is O(n).

- Textual Fields Information (Qtinfo): The first requirement to calculate Qtinfo is to obtain the document-word matrix from the text fields in the metadata instances. This matrix is used to calculate the IDF value of each word. The IDF value can be considered as the entropy value of that word. The application of a LSA algorithm reduces the dimensionality of the matrix and compensates the noise introduced by semantically related words. The build of this matrix require O(N) operations. However, due to the use of the LSA algorithm, the update of the matrix requires its complete recalculation. Once the document-word matrix is created, the final calculation of the Qtinfo is only dependant of the number of words in the text (W) and the number of instances involved in the call (n). In practice, given that the number of words in the text varies according to a Normal distribution, the calculation time is O(n). As in previous metric, the Qtinfo can be pre-calculated at Repository and Subset levels, reducing the response time to O(1). At Instance level, the calculation is done at call-time, therefore the response time is also O(n).

- Consistency (Qcons): To compute Qcons, the metadata instance has to be checked against a set of metadata rules. Given a fixed set of rules, the calculation time is O(n), with n being the number of instances involved in the service call. The pre-calculation of these metrics at Repository and

Subset level could reduce the response time to O(1). In the case of Instance level, the response time remains O(n).

- Textual Coherence (Qcoh): The calculation of textual coherence also needs to calculate a document-word matrix of the textual fields in the metadata instances. This matrix is also pre-calculated for the Qtinfo metric, being reusable for Qcoh calculation. With the document-word matrix already present, the remaining calculation consists in measuring the semantic distance between each pair of textual fields required. If we describe the number of fields as F and the number of words per field as W, the calculation time of this metric is O(F^2Wn). If we considered F and W as fixed over a repository, the practical calculation time is O(n). Pre-calculation of this metric for Repository and Subset levels, reduces the response time to O(1). At the Instance level, no pre-calculation is possible, leaving the response time at O(n).

- Readability (Qread): The calculation of the readability metric does not require any initialization. The application of the Flesch metric is proportional to the number of words in the text and the number of instances where the metric is calculated. Given that the number of words in the text follows a Normal distribution, it can be concluded that the response time of this metric is O(n). This metric can be pre-calculated for the Repository and Subset levels.

- Keyword Linkage (Qlink): To calculate keyword linkage, first an inverted index is created where each keyword points to the metadata instances that include it. The time to construct the invert index is proportional to the number of keywords and the total number of instances in the repository. If we considered that the number of keywords per instance is Normally distributed, the practical calculation time is O(N). The second step in the calculation is to count the number of metadata instances that share 1 or more keywords with the current instance. The number of links between the N metadata instances in the repository has an upper limit of O(N^2) (complete graph). However, it is expected that the actual number of links between elements is considerably lower. If the linkage is pre-calculated for the Repository and Subset level, the response time for the call is O(1). In the case of Instance level call, the response time is O(n*N).

- Currency (Qcurr):The calculation of this metric involves first the calculation and storage of the average value of the previous metrics (Qavg). Due to its nature, it can only be calculated at the Repository and Subset level. The computation time of this metric is always dependant of the number of instances in the call (O(n)). If this metric is pre-calculated, the response time of the call should be near O(1).

Metric	Initialization	Update	Call
Qcomp	0	0	O(n)
Qwcomp	0	0	O(n)
Qaccu	O(N)	O(N)	O(n)
Qcinfo	O(N)	O(1)	O(n)
Qtinfo	O(N)	O(N)	O(n)
Qcons	0	0	O(n)
Qcoh	O(N)	O(N)	O(n)
Qread	0	0	O(n)
Qlink	O(N^2)	O(N)	O(n*N)
Qcurr	O(N)	O(1)	O(n)
Qprov	O(N)	O(1)	O(n)

Table 6.1: Scalability of the Metadata Quality Metrics

- Provenance (Qprov): This calculation uses information from the metadata instance only. Similarly to the Keyword Linkage (Qlink) metric, an inverted index is created with the different contributors pointing to the metadata instances that she has created. The creation of this inverted index can be done in O(N) time. The next step is to obtain the average metric value (Qavg) for all the instances pointed by a given contributor. The average of the Qavg of the individual metrics is then assigned to the contributor. Finally the average quality metric value for each contributor is transferred as Qprov to their metadata instances. The calculation time of this process has an upper value proportional to the number of contributors (C) multiplied by the number of metadata instances in the repository (N). In practice, given that instances can only be produced by one contributor, the calculation time is O(N). Due to its nature, this metric can be easily pre-calculated reducing the response time at Repository and Subset level to O(1). The response time for Instance level is proportional to the number of instances in the call, O(n).

6.3.2 Ranking Metrics

In chapter 5, seven metrics to estimate the relevance of learning objects are proposed. These metrics are calculated from usage and contextual information captured through some sort of Attention Metadata. These metrics can be calculated at two levels: Repository and Result List (Appendix B). In this subsection, we present how those metrics are calculated in the first version of the Metric Service and how they can be scaled.

- Basic Topical (BT): In this metric the information about previous queries

and click-through on those queries are used to establish the relevance of the different objects. The first step in the calculation of this metric is to construct an inverted index where each query points to tuples consisting on the identification of the object clicked during that query and a timestamp. The time to initialize this index is proportional to the number of queries (Q) and the number of objects clicked in a query (C). If we considered the number of clicks per query as Normally distributed, the initialization time will be O(Q). New information can be introduced into the system in O(1) time. Once the index is built, the calculation of the BT metric at Repository and Result List level is proportional to the number objects clicked in a query (q) and the number of objects involved in the call (n). This O(nq) calculation can be reduced to O(n) if the time limit is used in the call to the metric. The time limit filters all results from the calculation. In the case of a Repository level call, the values can be pre-calculated and stored. This reduce the response time of this call to near O(1).

- Course-Similarity Topical (CST): The calculation of this metric first needs to establish the similarity between courses. To establish these similarities, the number of shared learning objects is used. An inverted index is built with the existing objects pointing to the courses where they are included. The time upper limit to built this index is proportional to the total number of Learning Objects (N) and the total number of courses (C). In practice, however, the initialization time tends to be close to O(N) because just a small fraction of courses share objects. The next step to initialize this calculation is to create a table where each course point to a tuple consisting in the id of a related course and the weight of that relation. The upper limit of time needed to build this table is also O(NC), with a practical value near O(N). The final initialization step is to build a table where each course points to their contained objects. The cost of this operation is O(C) if we considered that the number of objects per course do not increase through time. Adding new data to the three tables is the sum of the O(1)+O(N)+O(1) or simplifying O(N). The calculation of the metric at Repository level has a cost of O(1) if we consider that the fraction of related courses is low. In the case of Result List level call, the response time is proportional to the number of objects in the result list (n).

- Internal Topical (IT): This metric can only be calculated at Result List level. The initialization of this metric needs the construction of the same inverted index where objects point to their contained courses built for CST. Once this index is available, the calculation of this metric only needs to count the number of objects sharing the same parent courses. The calculation time is O(n), where n is the number of objects in the result list.

- Basic Personal (BP): The initialization of the BT metric needs information

about the objects clicked by each user. The metadata instance of those objects is used to construct a profile for each user. The building of this profile depends on the total number of users (U) and the number of interaction per user (I). The O(UI) time required to build the profiles cannot be reduced. However, this initialization only needs to take place once. Any further increase on information has a cost of O(1) per interaction. Once the profile is built, the response time for Repository and Result List levels is O(n), where n is the total number of objects in the repository or the number of objects in the result list. The Repository level calculation time can be reduced to O(1) if the BP value is pre-calculated and stored for each user and object in the result list through a table.

- User-Similarity Personal (USP): The initialization and calculation of this metric is very similar to CST. The only difference is that the similarity between users is extracted form the Attention Metadata repository in the form of objects clicked or used by each user. The initialization time will be O(N+U), the update cost is also O(N) and the calculation time is O(n) where n is the number of objects included in the call.

- Basic Situational (BS): The calculation of this metric used the TFIDF calculation used in the Lucene library [Hatcher and Gospodnetic, 2004]. It does not have any additional initialization cost to the one already used to create the inverted index used to search for learning objects. The calculation time is only dependant of the number of objects in the repository, roughly O(N).

- Context-Similarity Situational (CSS): This metric builds a profile for each course base on the objects that it already contains. To initialize the profile for each course, the metadata instances of all the objects that it contains are used. This initialization has a cost proportional to the number of courses (C) and the number of Objects per course (O). If we consider that the number of objects per course remains relatively constant, the initialization time is nearly O(C). Adding a new object to a course is a simple operation with cost O(1). Once the profile is built, the calculation is only dependant of the number of objects in the call (n). For Result List level, the minimum response time is O(n). On the other hand, for Repository Level, where pre-calculation can be used, the response time is near O(1).

6.3.3 Scalability

The scalability of the metrics can be measured a three points: initialization, update and call. At initialization time, the data structures needed for the calculation of the metrics are obtained from the raw data. At update time, new information arrive and the data structures to calculate the metrics need to be updated. At

Metric	Initialization	Update	Call
BT	O(Q)	O(1)	O(n)
CST	O(N+C)	O(N)	O(n)
IT	O(N)	O(1)	O(n)
BP	O(U*I)	O(1)	O(n)
USP	O(N+U)	O(N)	O(n)
BS	O(N)	O(1)	O(n)
CSS	O(C)	O(1)	O(n)

Table 6.2: Scalability of the Relevance Ranking Metrics

call time, the client system requires the metric value for a group of instances or objects.

The most important scalability point, because it has to be executed on-line, is call time. The rest, initialization and update, can be executed off-line and their results can be stored. As can be seen in Tables 6.1 and 6.2, most metric calls only involve O(n) calculations. That means that it only grows linearly with the number of objects in the call and it is independent of the number of objects stored in the repository. This independence makes the metrics scalable for any size of repository. Only the calculation of the Qlink metric is also dependent of N, the size of the repository.

At initialization size, most of the metrics depend linearly of the size of the repository, N. In specialized metrics, such as BT and CST, the initialization time depends linearly on the number of queries (Q) or the number of courses (C), instead of N. In other ranking metrics, such as CST and UST, the initialization time is linearly dependant of the number of objects in the repository plus other repository variables, such as the number of courses (C) or users (U). Only in Qlink and BP, the initialization time grows faster than linear. Being an operation done off-line, the linear nature of the calculation is not critical, but can be a problem in very large systems.

To get the full picture of the time of pre-calculation of these metrics, the update time should also be considered. In most metrics, the cost of adding one more item of information to the pre-calculation data structures is just O(1). In those cases, more importantly in BP, it means that the initialization time is not significant given that any new data arriving to system can be added to the pre-calculated data structures with few operations that do not depend of the current size of those structures. In the case that the pre-calculation time is O(N), the update cannot be done continually but in batches when a certain amount of new information have arrived to the system. For the metadata quality metrics Qaccu, Qtinfo and Qcoh, the data structure being pre-calculated is the word-document matrix. The constant update of this structure is not critical to the correct calculation of the metric. In other cases, such as Qlink, CST and UST, an updated structure is

fundamental to provide an updated result.

From this analysis it can be concluded that only Qlink present serious scalability problems. A larger repository would result in longer response times for the call to obtain the value of the metric. Also the initialization of the metric and the update of the pre-calculated data structures could take a potential increasing amount of time as the repository grows. The calculation of this metric should be analyzed to make it independent of repository size. One way to do this is to limit the maximum number of related instances. The rest of the initial metric implementation can be used as they are in real, large systems.

6.4 Uses Cases

The most important validation of the effectiveness of the metrics can only come from their application in large scale, real-world Learning Object tools. Due to the time involved in this kind of studies, this evaluation is not part of the current disertation. However, this section present how the metrics are being initially applied to four different Learning Object systems to improve or expand their functionality.

6.4.1 OER Commons Metadata

OER Commons [Joyce, 2007] is a portal that provides access to several repositories of Open Educational Resources. The sources of these resources vary from Open Courseware initiatives, such as MIT [1] and Johns Hopkins [2], to online LORs, such as Connexions [3] or Wikiversity [4]. OER Commons harvest the content of those sources via OAI-PMH protocol [Van de Sompel et al., 2004]. If needed, the harvested metadata is manually enriched. Finally, the metadata is inserted in a centralized repository to serve the search engine provided at the portal. IEEE LOM [IEEE, 2002] is used as the metadata standard inside this repository.

The quality of the metadata arriving at OER Commons vary greatly from source to source. The metadata from some sources need little or no work. On the other hand, for some sources, most of the metadata values should be added at OER Commons. Moreover, several of the repositories only provide metadata based on the DC standard. This standard does not provide all the richness of LOM for educational resources. Until now, the enrichment process was feasible as the number of objects accessible to OER Commons was slowly growing. However, the success of OER Commons also means that more repositories are willing to share their resources with them. The amount of new objects being accessible from the portal is increasing rapidly and soon it will surpass the enrichment capacity.

[1] MIT Open Courseware. http://ocw.mit.edu
[2] Johns Hopkins Open Courseware. http://ocw.jhsph.edu
[3] Connexions Site. http://www.cnx.org
[4] Wikiversity Site. http://www.wikiversity.org



Figure 6.2: Visualization of the Qtinfo metric of the OER Commons Harvested Metadata

One of the first strategies considered by OER Commons to cope with the increasing number of objects is to focus the enrichment process in those objects or repositories that need it the most. However, manually reviewing all the metadata instances is an unfeasible option. To solve this problem, they are evaluating the use of the metadata quality metrics proposed in chapter 4, to provide an initial quality filter to direct their enrichment efforts. The metrics selected for this evaluation are Qcomp, Qwcomp, Qcinfo, Qtinfo and Qread.

The selected metrics are calculated through the Metric Service for all the metadata instances present in the OER Commons repository. The values of the metrics are then presented in a treemap visualization in order to provide the enrichment team with the information about the lower quality metadata instances and repositories. An example of this visualization for the Textual Information Content (Qtinfo) metric is presented in Figure 6.2.

6.4.2 MELT Project

The MELT (Metadata Ecology for Learning and Teaching)[5] Project is a pan-European project focused on enriching the metadata from several repositories of learning objects. The goal of this enrichment is to enable teachers and learners to quickly and easily find the specific learning materials they need. The project has three enrichment strategies:

- Some MELT content will be enriched with metadata by expert or trained indexers
- Teachers will be provided with folksonomy [Gruber and Gruber, 2007] and social tagging tools so that they can add their own metadata to MELT content they have used
- New frameworks for automatic metadata generation will be used to enrich MELT content

Regardless of the enrichment strategy, the MELT project needs some way to measure it. Giving the large amount of objects (circa 100,000) involved in the first phases of the project, the most scalable solution was to use the Metadata Quality Metrics Services to compare the metrics values before and after the enrichment of the content. Figure 6.3 explain how the Metrics Services has been included inside the MELT Architecture.

The values of the metrics before and after the enrichment are obtained for each metadata instance. If the difference is small, inexistant or negative, the system alert the administrators of a possible problem in the enrichment process. The values of the metrics are also presented in a control Dashboard that enable the administrator to assess the progress of the project.

6.4.3 MACE Project

The MACE (Metadata for Architectural Contents in Europe)[6] Project aims to improve architectural education, by integrating and connecting vast amounts of content from diverse repositories, including past European projects and existing architectural design communities. Given the visual nature of the architectural content, the main focus of the project is how to render that information available to the community. The main strategy to improve the findability of the objects is to use the user interaction with the content to suggest or recommend the material to possibly interested users. In other words, the MACE project looks to present the most relevant object to each type of user. Our Relevance Ranking Metrics are helping MACE to fulfill this goal.

[5]MELT Site. http://www.melt-project.eu
[6]MACE Site. http://www.mace-project.eu

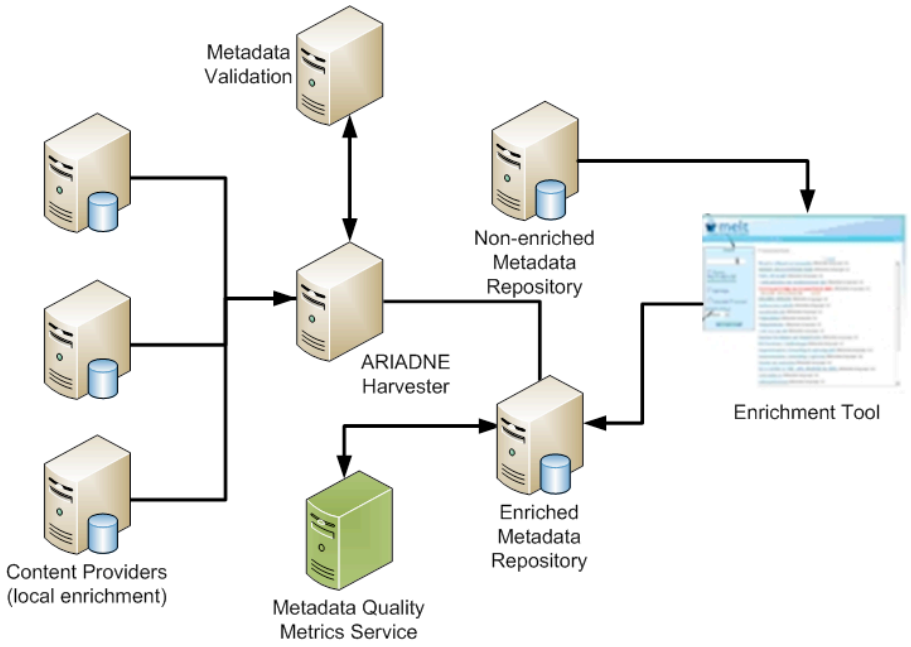


Figure 6.3: Architecture of the MELT Project including Metadata Quality Metrics Service

The MACE infrastructure captures the interaction of the users with the content and stores that data into an Attention Metadata repository. The Metrics Services are used to calculate useful General ranking metrics that can be used to recommend MACE users with material relevant material. The main ranking metrics used in this project are BT and BP. Figure 6.4 presents how the Metrics Service is integrated in the MACE technological architecture.

The main use of the relevance ranking metrics is to present, along the search interface, a recommendation box presenting the most recommended objects for the searching user (in the case that the user is logged-in) or the most relevant objects in the repository in case that the user remains anonymous.

6.4.4 Ariadne Finder

The ARIADNE Finder is the search tool used to gain access to the new architecture of the ARIADNE Repository. The ARIADNE Finder tries to improve the findability of the most relevant material. All users interactions inside the ARIADNE Finder are logged into a Attention Metadata Repository. This information is used by the Relevance Ranking Metrics to improve the ordering of the result

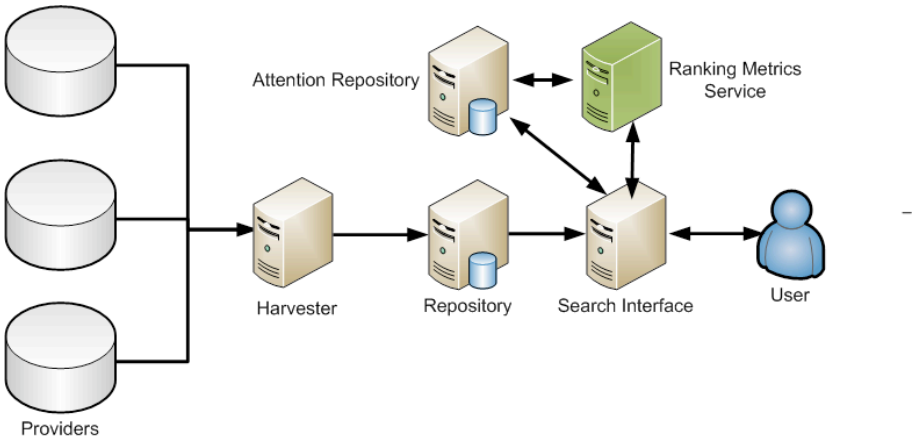


Figure 6.4: Architecture of the MACE Project including Ranking Metrics Service

list obtained from the ARIADNE Repository.

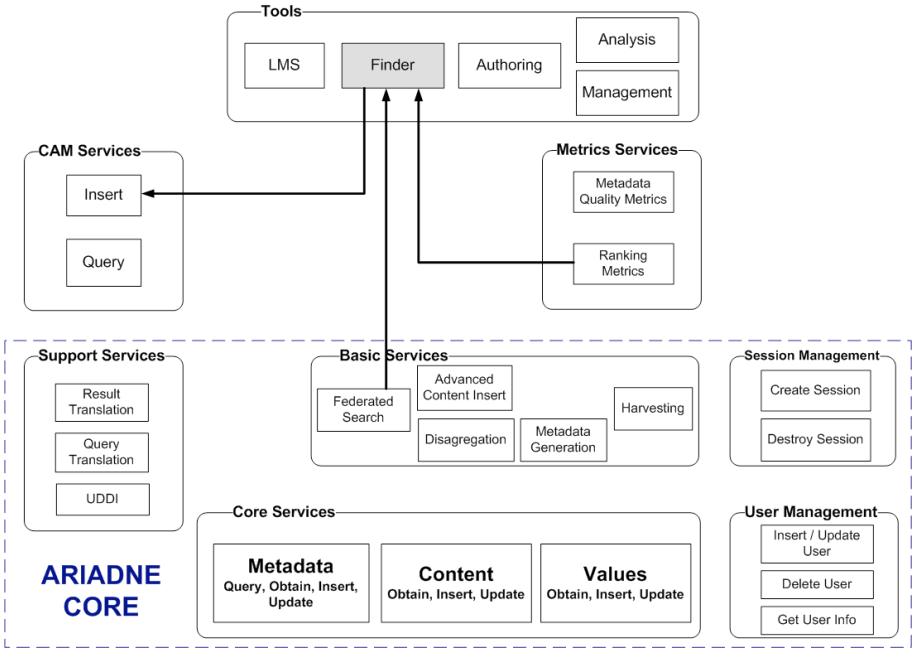


Figure 6.5: Finder and ARIADNE Next Architecture

Figure 6.5 presents the ARIADNE-NEXT Architecture that includes the Finder, the Repository, the Contextualized Attention Metadata Repository and the Metric Service. The current implementation of ARIADNE-NEXT uses the BT, BP and USP metrics to improve the ordering in the Finder. Figure 6.6 presents an example of the Finder interface with the ordering provided by the Metric Services.



Figure 6.6: Ariadne Finder interface with sorted results

6.4.5 Early Feedback

Even if the use in the Metrics Services in current tools is still in an early stage, the services provided are enabling increased functionality in the test applications. This functionality will be difficult to provide by any other method. For example, quality control of the metadata in OER Commons and MELT will not be scalable if the traditional human based review technique is used. Similarly, the ranking functions provided for MACE and ARIADNE Finder provide a more meaningful order than using just text-matching approaches.

The interest of these projects in the provided Metrics Services validates, if not the current implementation, the need to calculate metrics that could improve the usability and effectiveness of current Learning Object tools. We are confident that the research and development of new, more sophisticated, metrics could have a direct impact in the acceptance of these tools into the mainstream learning community.

6.5 Conclusions

This chapter has presented how the proposed metrics can be converted in executable code and used inside existing architectures to improve the usability of end-user Learning Object tools. A key factor to facilitate the use of the metrics inside those architectures is the proposed Service Oriented Architecture that enable the rapid deployment of new types of metrics without changing the host application. The adaptability of this Metric Service is empirically tested in four large scale projects.

An interesting finding at implementation time is that all but one of the proposed metrics are scalable and can be used regardless the size of the learning object repository or federation of repositories. If we add this result to the findings on chapters 4 and 5 about the meaningfulness of the metrics, the main criteria and requirement set for the development of the metrics are reached.

Finally, the service interfaces (MeQuMI and RaMI), defined in Appendices A and B, provide a way to build different, but compatible Metric Services. This compatibility opportunity is important to bootstrap an ecology of different metrics and implementation that could be used and exchanged between different tools and architectures.

Chapter 7

Conclusions

In this dissertation, Informetric studies has been performed to data about Learning Object Publishing and Reuse, finding interesting properties of these processes. Also, several metrics to calculate the Quality of the Metadata and the Relevance of the Learning Object have been developed and successfully evaluated. This chapter concludes the dissertation with a summary of the main contributions of this research and their possible impact on the field of Learning Object Technologies. Also, interesting open questions raised, but not addressed, in this dissertation are presented as possible paths to continue research on Learnometrics.

7.1 Main Contributions

The main contribution of this body of research is to, for the first time, measure the characteristic of the supply and demand processes of the Learning Object Economy [Campbell, 2003]. Moreover, useful metrics are extracted from the data generated by those processes in order to improve the workings of this Economy.

The following subsections present a detailed account of the principal findings and implications of this dissertation.

7.1.1 Publication of Learning Objects

Chapter 2 presents an Informetric study of the publication of Learning Objects in different types of repositories. The main original contributions obtained from this study are:

- The amount of learning objects in a repository is distributed according to an inverse power law ($1.5 < \alpha < 2.0$). This mean that there are few ($< 20\%$) big repositories that have the majority ($> 80\%$) of objects. A two tier

architecture based on search federation for the top layer, and harvesting for lower layers seems to be an ideal solution to interconnect those repositories.

- The number of objects in Learning Object Repositories grows linearly. Two phases are easily identified in the lifetime of any repository. A initial phase with usually slow growth, and a mature phase with an usually increased growth. The duration of the initial phase seems to be around 1 to 3 years.

- There is a strong "Participation Inequality" in the publication of Learning Objects. While all the studied repositories followed a heavy-tailed distribution, each type of repository follow a different type of heavy-tailed distribution. Learning Object Repositories (LORP) and Learning Object Referatories (LORF) seem to follow an inverse power law with exponential cut-off ($1.5 < \alpha < 2.2$). Under this regime, the majority of the content of the repository is published by few super-productive individuals. Open Courseware (OCW) sites and Learning Management Systems (LMS) follow a Weibull distribution ($0.5 < scale < 1.0$). In this case, most of the content on the repository is published by "middle class" users that contribute around the average. For Institutional Repositories (IR), the distribution of objects per contributor follows a steep inverse power law ($2.2 < \alpha < 3$) where the majority of the content is published by the "low class", users that contribute 1 or 2 objects.

- The main difference between repository types is how the lifetime of the contributor is distributed. For OCWs and LMSs, the lifetime of the majority of users is longer than for LORPs and LORFs. In the case of IRs, most contributors only contribute once. The distribution and duration of the lifetime seems to be the result of the "value proposition" of a repository to its contributors.

Chapter 2 also proposes a model to explain the characteristics of the publication process based on the interaction between the publication rate and lifetime among contributors, as well as the number of contributor growth. This simple model is able to successfully simulate the distribution of objects among contributors, as well as the repository growth for most of the studied repositories. This model is a first step to quantitatively understand the variables that influence the publication process.

The main findings of this study shows that in the Publication of Learning Objects there is no such thing as an average user or repository. The presence of several heavy-tailed distributions, controlling different aspects of the project, makes the common way to analyze quantities trough their mean and standard deviation useless. Informetrics has, since its beginning [Lotka, 1926], dealt with heavy-tailed processes. Learnometrics, Informetrics applied to Learning Objects, provide useful knowledge about the characteristics of the Learning Object Publication Process.

7.1.2 Reuse of Learning Objects

Chapter 3 presents a study of the reuse of Learning Objects from different sources and granularities. The main results and findings of this study are:

- The percentage of objects that are reused from a given collection seems to be around 20%. This reuse occurs without any special encouragement or special technological support. This percentage seems to be the same for objects of different type of collections, such as images, software libraries and Web Services.
- The percentage of reuse seems to be similar for learning objects of different granularities. This finding is very important, specially given that the reusability paradox [Wiley et al., 2004] predicts that smaller objects should be more reusable. This result should at least provoke the re-examination of this belief. We already proposed an alternative explanation that takes into account the difference in granularity between the object being reused and the object being built to establish the reusability of the object in that context.
- The reuse of an object is not linearly correlated with its popularity. While popularity statistically affects reuse, the number of times an object has been accessed cannot be used as a proxy to establish how reusable the object is. The reverse of this relation is also surprising. Even if an object is present in several higher-granularity objects, it is not necessarily a popular object.
- The number of reuses per object is distributed Log-Normally. This heavy-tailed distribution causes that very few objects are being reused a lot, while the majority of objects is once or never reused. The distribution is the same for different types of collections, such as images, software libraries and Web services.

Chapter 3 also presents a statistical model, based on the Log-Normal distribution of reuse, to explain the observation in the data sets. This model explains the way that the different factors affecting reuse should combine in order to produce a successful reuse. While this model does not provide which are the factors involved, it is a first step to understand how the process of reusing learning objects works.

The detection of a heavy-tailed distribution (Log-Normal) in the number of reuses per object also confirms that Informetric studies are more suited to unveil the inner workings of the Reuse of Learning Objects.

7.1.3 Quality of Learning Object Metadata

Chapter 4 proposes and evaluate a group of metrics to estimate the Quality of the Metadata of Learning Objects. The evaluation was performed through a series of

experiments with human reviewers and metadata sets. The main contributions of this chapter are:

- Three of the proposed metrics (Completeness, Weighted Completeness and Textual Information Content) correlate well with human reviews. The most successful quality metric is the Textual Information Content (Qtinfo). The rest of the evaluated metrics (Categorical Information Content, Accuracy, Coherence and Readability) seems to be completely orthogonal to the quality value assigned by human reviewers. However, the not-correlating metrics seem to measure real characteristic of the metadata that humans are not able to assess.

- The average value of the proposed metrics (Qavg) is a good predictor of low quality records. This metric can be used to implement an automatic quality filter. This filter can be coupled with an automatic generator of metadata, such as SAmgI [Meire et al., 2007], in order to quickly create useful metadata from vast amounts of existing learning objects.

- As discussed in Chapter 6, the proposed metadata quality metrics (except Qlink) are scalable to large amounts of metadata instances.

Maybe the best evaluation of the usefulness of the metrics is their use in real projects to determine the characteristics and quality of large metadata sets. Chapter 6 discusses two use cases where the metrics are applied to help in the evaluation of learning object metadata quality.

The main contribution of this chapter is not the set of proposed metrics, as we expect them to be soon improved and superseded. The main contribution is the successful evaluation of metrics as a meaningful, scalable and transparent way to determine the quality of metadata instances inside digital repositories. We expect that this work will give rise to more research in this area.

7.1.4 Relevance of Learning Objects

Chapter 5 proposes and evaluates a group of metrics to estimate the Relevance of Learning Objects for a given user in a given context. The evaluation was performed through an experiments that compared the ranking generated by human reviewers and the one generated by individual metrics and different combinations. The main contributions of this chapter are:

- The evaluation of the metrics shows that the contextual and usage information can be converted into a meaningful set of relevance ranking metrics. All the proposed basic metrics outperformed the ranking based on the most common text-based approach.

- The combination of the metrics through the use learning methods, such as RankNet [Richardson et al., 2006], provides a significant improvement of more than 50% over the text-based ranking. This results is specially remarkable as only very basic ranking metrics were used as support of the final ranking function.
- Chapter 6 discusses how all the proposed metrics can be scaled to large repositories. Also, the metrics can be transparently calculated from implicitly extracted information.

These metrics are being used into real projects to provide ranking metrics inside learning object search engines (Chapter 6).

As in the case of the metadata quality metrics, the proposed metrics are not presented as an optimal solution, but as an initial step. The goal of these metrics and their future successors is to convert contextual and usage information into a meaningful order personalized for a specific user in a specific situation.

7.2 Further Research

As a first exploration of Learnometrics, this dissertation, through its studies, raises more questions than it answers. Ample opportunities for further research are provided as the field of Learnometrics unfolds. The following is a list of what we consider are the most interesting and urgent research questions seeking for answers and explanations.

7.2.1 Quantitative Studies

The quantitive analyses of the processes involved into the Learning Object Economy have received little research attention. The studies performed in this dissertation about the Publication and Reuse processes, while revealing, only scratch the surface of the knowledge that can be obtained about the whole Learning Object lifecycle through careful measuring. A sample of what we considered the most interesting open research issues that can be addressed through quantitative analyses, follows:

- Estimating the amount of existing learning objects. The on-line available learning objects are just the "tip of the iceberg". Estimating the total amount of learning objects in existence, while could be considered at first sight a inconsequential research question, could lead to a better understanding of the problem of scale that Learning Object Technologies are facing.
- Effect of openness. Another interesting question is whether repositories with open publication, such as Connexions or Merlot, are more efficient or productive that closed projects, such as Intute or MIT OCW, in the long run.

- Relation between scientific and teaching output. Given that learning objects and scientific publications usually are produced by the same set of individuals, it would be interesting to find out the relation between both rates of publication. Having this information could help us to design efficient and fair incentive programs to improve the quantity or quality of the teaching output.

- How to integrate LMSs. One of the interesting findings of the publication analysis is that LMSs seem to be the best environment for learning object publication. However, traditionally those are isolated silos of information. How to intercommunicate between them and share their contents, not only at the technical level, but also, and more importantly, at the social, legal and administrative level should be one of the main challenges of our field.

- The "chicken or egg" dilemma of LORs. From the analyses in this dissertation is not clear whether the increase in the number of objects in a repository attracts more users or vice versa. Knowing the actual cause-effect order can help administrators to bootstrap faster their repositories.

- Effect of granularity on reuse. A re-evaluation of the role of the granularity in object reusability is due in the light of the results presented in Chapter 3. The existence or not of the reuse paradox should be put to test.

- Percentage of reuse. It will be interesting to find out whether the 20% value found for the percentage of reuse of learning objects and other reusable components has a deeper explanation or it is just a coincidence caused by the repositories used in this study.

- Steps in the reuse process. The proposed model for the reuse process only determines how the different factors are combined to produce successful reuse. However, nothing is said about what are the specific factors for the chain that lead to the reuse of a learning object. Controlled experiments or observations can be performed in order to measure the impact of different factors while the other are maintained equal.

- Creation process. There are almost no quantitative studies about the process of learning object creation. Arguably, the most important variables to analyze in this process are time, learning object type and length. Also the comparative advantages of different creation methods, such as individual, collaborative, successive improvements, etc., deserve further attention.

- Retaining process. A much greater indication of the quality of a learning object can be extracted to the understanding of which characteristics are important during the retaining of a learning object for further use. Interesting questions that should be answered are: How many learning objects

are retained in a course? What are the more important factors affecting the retaining process? What is the half-life of a learning object?

Answering these questions through quantitative analyses will increase our understanding of how the Learning Objects Economy works. This understanding can help us to create the right environment for this economy to flourish and provide its predicted benefits.

7.2.2 Metrics for Learning Objects

The main use that we can give to the information extracted from the analysis of the data created at the different processes of the Learning Object Economy is the creation of metrics to improve the tools used in those processes. This dissertation presented an initial set of metrics to estimate the metadata quality and relevance of learning objects. These metrics are far from optimal. Further research can take these metrics as a base to build better, more refined versions. We list some open and interesting research topics not addressed in this dissertation:

- *New metadata quality frameworks oriented to automatic processing* Current metadata quality frameworks are deeply rooted in traditional, analog metadata. This metadata was meant to be consumed by humans and thus the quality characteristics considered in the frameworks were the ones that humans found important. Now, the metadata is mainly consumed and processed by automated software systems. It could be created or modified with each human interaction (corrections, annotations, tags, reviews, etc.) with the system. While it preserves some relation with its analog counterpart, digital metadata could not be measured with the same standards. New quality frameworks oriented to digital metadata and automatic processing should be developed.

- *Establishing a common data set.* Borrowing the idea that the TREC conference [Harman, 1993] initiated and in order to provide a better "measurement" of the quality of different metrics, the quality metrics should be applied to a known set of test metadata instances with an established and known value for different dimensions of quality. This is especially important to provide common ground to metrics proposed by different researchers. When applied to a common set, the prediction power of the metrics could be objectively compared and progress could be measured.

- *Federated Search.* How should the relevance metric calculation be adapted to environments in which only the top-k objects of each repository are known? Moreover, how should the ranking made by different LORs be aggregated?

- *Underlying theory.* What are the deeper pedagogical or cognitive reasons which explain the success or failure of different metrics?

The main task left for further work is to execute large empirical studies with full implementations of the metrics in real environments. Once there is enough data collected, the user interaction with the system and the progress of the different metrics could be analyzed to shed light on these questions. We also hope that other researchers start proposing improvements to these initial approaches.

7.3 Final Words

The field of Learning Object Technologies, and Technology Enhanced Learning (TEL) in general, has the potential to solve one of the most important challenges of our time: enable everyone to learn anything, anytime, anywhere. However, if we look back at more than 50 years of research in Technology Enhanced Learning, it is not clear where we are in terms of reaching our goal and wether we are, indeed, moving forward.

The pace at which technology and new ideas evolve have created a rapid, even exponential, rate of change. This rapid change, together with the natural difficulty to measure the impact of technology in something as complex as learning have lead to a field with abundance of new, good ideas and scarcity of evaluation studies. This lack of evaluation has resulted into the duplication of efforts and a sense of no "ground truth" or "basic theory" of TEL. This dissertation was an attempt to stop, look back and measure, if not the impact, at least the status of a small fraction of TEL, Learning Object Technologies, in the real world.

During the research journey that lead to the different chapters of this dissertation, many surprises were found. The apparent inexistence of the reuse paradox, the two phase linear growth of repositories or the ineffective metadata quality assessment of humans are clear reminders that even bright theoretical discussions do not compensate the lack of experimentation and measurement. Both theoretical and empirical studies should go hand in hand in order to advance the status of the field.

Finally, this dissertation is an invitation to other researchers in the field to apply Informetric techniques to measure, understand and apply in their tools the vast amount of information generated by the usage of Technology Enhanced Learning systems. This new field of research, that we called Learnometrics, promises to provide deep insight into how instructors and learners are making use of the technology. Only this kind of understanding could help us to be sure that we are moving forward in our quest to provide great learning experiences for anyone, anytime, anyplace.

Bibliography

[Agichtein et al., 2006] Agichtein, E., Brill, E., Dumais, S., and Ragno, R. (2006). Learning user interaction models for predicting web search result preferences. In Dumais, S., Efthimiadis, E. N., Hawking, D., and Jarvelin, K., editors, *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, New York, NY, USA. ACM Press.

[Agogino, 1999] Agogino, A. (1999). Visions for a digital library for science, mathematics, engineering and technology education (SMETE). In *Proceedings of the fourth ACM conference on Digital Libraries*, pages 205–206, New York, NY, USA. ACM Press.

[Akaike, 1976] Akaike, H. (1976). An information criterion (AIC). *Math Sci*, 14(153):5–9.

[Almind and Ingwersen, 1997] Almind, T. C. and Ingwersen, P. (1997). Informetric analyses on the world wide web: methodological approaches to webometrics. *Journal of Documentation*, 53(4):404–426.

[Alonso et al., 2004] Alonso, G., Casati, F., Kuno, H., and Machiraju, V. (2004). *Web Services: Concepts, Architectures and Applications*. Springer Verlag.

[Amory, 2005] Amory, A. (2005). Learning objects: Just say no! In Kommers, P. and Richards, G., editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005*, pages 1539–1544, Chesapeake, VA. AACE.

[Anderson, 2006] Anderson, C. (2006). *The long tail*. Hyperion.

[Baraniuk, 2007] Baraniuk, R. G. (2007). *Opening Up Education: The Collective Advancement of Education through Open Technology, Open Content, and Open Knowledge*, chapter Challenges and Opportunities for the Open Education Movement: A Connexions Case Study, pages 116–132. MIT Press.

[Barton et al., 2003] Barton, J., Currier, S., and Hey, J. M. N. (2003). Building quality assurance into metadata creation: an analysis based on the learning objects and e-prints communities of practice. In Sutton, S., Greenberg, J., and Tennis, J., editors, *Proceedings 2003 Dublin Core Conference: Supporting Communities of Discourse and Practice - Metadata Research and Applications*, pages 39–48, Seattle, Washington.

[Beall, 2005] Beall, J. (2005). Metadata and data quality problems in the digital library. *JoDI: Journal of Digital Information*, 6(3):20.

[Bederson et al., 2002] Bederson, B. B., Shneiderman, B., and Wattenberg, M. (2002). Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Trans. Graph.*, 21(4):833–854.

[Berendt et al., 2003] Berendt, B., Brenstein, E., Li, Y., and Wendland, B. (2003). Marketing for participation: How can Electronic Dissertation Services win authors? In *Proceedings of ETD 2003: Next Steps–Electronic Theses and Dissertations Worlwide*, pages 156–161.

[Bingham and Mannila, 2001] Bingham, E. and Mannila, H. (2001). Random projection in dimensionality reduction: applications to image and text data. In Provost, F. and Srikant, R., editors, *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250, New York, NY, USA. ACM Press.

[Bohl et al., 2002] Bohl, O., Scheuhase, J., Sengler, R., and Winand, U. (2002). The sharable content object reference model (scorm)- a critical review. In Werner, B., editor, *Proceedings of the International Conference on Computers in Education 2002*, pages 950–951. IEEE Computer Sociecty.

[Bookstein, 1997] Bookstein, A. (1997). Informetric distributions. ill. ambiguity and randomness. *Journal of the American Society for Information Science and Technology*, 48(1):2–10.

[Borlund, 2003] Borlund, P. (2003). The concept of relevance in IR. *Journal of the American Society for Information Science and Technology*, 54(10):913–925.

[Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117.

[Broadus, 1987] Broadus, R. (1987). Toward a definition of bibliometrics. *Scientometrics*, 12(5):373–379.

[Brody et al., 2005] Brody, T., Harnad, S., and Carr, L. (2005). Earlier web usage statistics as predictors of later citation impact. *Journal of the American Association for Information Science and Technology*, 57(8):1060–1072.

[Broisin et al., 2005] Broisin, J., Vidal, P., Meire, M., and Duval, E. (2005). Bridging the gap between learning management systems and learning object repositories: Exploiting learning context information. In *AICT-SAPIR-ELETE '05*, pages 478–483, Washington, DC, USA. IEEE Computer Society.

[Bruce and Hillmann, 2004] Bruce, T. R. and Hillmann, D. (2004). *Metadata in Practice*, chapter The continuum of metadata quality: defining, expressing, exploiting, pages 238–256. ALA Editions, Chicago, IL.

[Budanitsky and Hirst, 2001] Budanitsky, A. and Hirst, G. (2001). Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics*, pages 29–34, Pittsburgh, PA. NAACL.

[Bui and Park, 2006] Bui, Y. and Park, J.-r. (2006). An assessment of metadata quality: A case study of the national science digital library metadata repository. In Moukdad, H., editor, *Proceedings of CAIS/ACSI 2006 Information Science Revisited: Approaches to Innovation*, page 13.

[Burrell, 1992] Burrell, Q. L. (1992). The gini index and the leimkuhler curve for bibliometric processes. *Inf. Process. Manage.*, 28(1):19–33.

[Campbell, 2003] Campbell, L. (2003). *Reusing Online Resources: A Sustainable Approach to E-Learning*, chapter Engaging with the learning object economy, pages 35–45. Kogan Page Ltd.

[Cardinaels, 2007] Cardinaels, K. (2007). *A dynamic learning object life cycle and its implications for automatic metadata generation.* PhD thesis, Katholieke Universiteit Leuven.

[Cardinaels et al., 2005] Cardinaels, K., Meire, M., and Duval, E. (2005). Automating metadata generation: the simple indexing interface. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 548–556, New York, NY, USA. ACM Press.

[Carr and Brody, 2007] Carr, L. and Brody, T. (2007). Size isn't everything: sustainable repositories as evidenced by sustainable deposit profiles. *D-Lib Magazine*, 13(7/8):1082–9873.

[Carson, 2004] Carson, S. (2004). MIT OpenCourseWare Program Evaluation Findings Report. Technical report, MIT.

[Carson, 2005] Carson, S. (2005). Program Evaluation Findings Report MIT OpenCourseWare. Technical report, MIT.

[Chapman and Massey, 2002] Chapman, A. and Massey, O. (2002). A catalogue quality audit tool. *Library Management*, 23(6-7):314–324.

[Chellappa, 2004] Chellappa, V. (2004). *Content-Based Searching with Relevance Ranking for Learning Objects*. PhD thesis, University of Kansas.

[Chi et al., 2001] Chi, E. H., Pirolli, P., Chen, K., and Pitkow, J. (2001). Using information scent to model user information needs and actions and the web. In Jacko, J. and Sears, A., editors, *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 490–497, New York, NY, USA. ACM Press.

[Chitwood et al., 2000] Chitwood, K., May, C., Bunnow, D., and Langan, T. (2000). *The Instructional Use of Learning Objects: Online Version*, chapter Battle stories from the field: Wisconsin online resource center learning objects project, pages 2001–2011. Agency for Instructional Technology.

[Chu and Rosenthal, 1996] Chu, H. and Rosenthal, M. (1996). Search engines for the world wide web: A comparative study and evaluation methodology. In Hardin, S., editor, *Proceedings of the 59th Annual Meeting of the American Society for Information Science*, volume 33, pages 127–135, Baltimore, MD. Softbound.

[Clauset et al., 2007] Clauset, A., Shalizi, C., and Newman, M. (2007). Power-law distributions in empirical data. 26 pages. Arxiv preprint arXiv:0706.1062.

[Coile, 1977] Coile, R. (1977). Lotka's frequency distribution of scientific productivity. *Journal of the American Society for Information Science*, 28(6):366–370.

[Cole and Foster, 2007] Cole, J. and Foster, H. (2007). *Using Moodle: Teaching with the Popular Open Source Course Management System*. O'Reilly Media, Inc.

[Collis and Strijker, 2004] Collis, B. and Strijker, A. (2004). Technology and human issues in reusing learning objects. *Journal of Interactive Media in Education*, 4:1–32.

[CreativeCommons, 2003] CreativeCommons (2003). Creative commons licenses. [web document: http://www.creativecommons.org].

[Cuadrado and Sicilia, 2005] Cuadrado, J. and Sicilia, M. (2005). Learning objects reusability metrics: Some ideas from software engineering. In Grout, V., Oram, D., and Picking, R., editors, *Proceedings of the International Conference on Internet Technologies and Applications ITA 2005*, page 5, Wreham (UK). North East Wales Institute.

[Curbera et al., 2002] Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., and Weerawarana, S. (2002). *Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI.* IEEE Computer Society.

[Dalziel, 2002] Dalziel, J. (2002). Reflections on the COLIS (Collaborative Online Learning and Information Systems) Demonstrator project and the "Learning Object Lifecycle". In Williamson, A., Gunn, C., Young, A., and Clear, T., editors, *Winds of Changing in the Sea of Learning, Proceedings of the 19th Annual Conference of the Australian Society for Computers in Tertiary Education (ASCILITE)*, pages 159–166, Auckland, New Zealand. UNITEC Institute of Technology.

[Dalziel, 2003] Dalziel, J. (2003). Implementing Learning Design: The Learning Activity Management System (LAMS). In Crisp, G., Thiele, D., Scholten, I., Barker, S., and Baron, J., editors, *Interact Integrate Impact: Proceedings of the 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education*, pages 1–10, Adelaide, Australia. UNITEC Institute of Technology.

[Davis and Connolly, 2007] Davis, P. and Connolly, M. (2007). Institutional repositories. *D-Lib Magazine*, 13(3/4):1082–1101.

[DCMI, 1995] DCMI (1995). Dublin Core Metadata Innitiative, http://dublincore.org, retrieved 2/04/2007.

[DeGroot, 1986] DeGroot, M. (1986). *Probability and statistics*. Addison-Wesley Boston.

[Dodani, 2002] Dodani, M. (2002). The Dark Side of Object Learning: Learning Objects. *Journal of Object Technology*, 1(5):37–42.

[Dolog et al., 2004] Dolog, P., Henze, N., Nejdl, W., and Sintek, M. (2004). Personalization in distributed e-learning environments. In Najork, M. and Wills, C., editors, *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 170–179, New York, NY, USA. ACM Press.

[Dong and Agogino, 2001] Dong, A. and Agogino, A. (2001). Design principles for the information architecture of SMET education digital library. In Fox, E. A. and Borgman, C. L., editors, *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, pages 314–321, New York, NY, USA. ACM Press.

[Downes, 2004] Downes, S. (2004). *The Learning Marketplace: Meaning, Metadata and Content Syndication in the Learning Object Economy.* Downes, Stephen, Moncton, New Brunswick.

[Downes, 2005] Downes, S. (2005). E-learning 2.0. *eLearn Magazine*, 10:5.

[Downes, 2007] Downes, S. (2007). Models for sustainable open educational resources. *Interdisciplinary journal of knowledge and learning objects*, 3:29–44.

[Duncan, 2003] Duncan, C. (2003). *Reusing Online Resources: A Sustainable Approach to eLearning*, chapter Granularisation, pages 12–19. Kogan Page Ltd.

[Dushay and Hillmann, 2003] Dushay, N. and Hillmann, D. (2003). Analyzing metadata for effective use and re-use. In Sutton, S., Greenberg, J., and Tennis, J., editors, *DCMI Metadata Conference and Workshop*, page 10, Seattle, USA. Dublin Core Metadata Initiative.

[Duval, 2004] Duval, E. (2004). We're on the road to... . In Cantoni, L. and McLoughlin, C., editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004*, pages 3–8, Lugano, Switzerland. AACE.

[Duval, 2005] Duval, E. (2005). *Policy and Innovation in Education - Quality Criteria*, chapter LearnRank: the Real Quality Measure for Learning Materials, pages 457–463. European Schoolnet.

[Duval and Hodgins, 2003] Duval, E. and Hodgins, W. (2003). A LOM Research Agenda. In Chen, Y.-F. R., Kovcs, L., and Lawrence, S., editors, *Proceedings of the WWW2003: Twelfth International World Wide Web Conference*, pages 20–24, Budapest, Hungary. ACM Press.

[Duval and Hodgins, 2004] Duval, E. and Hodgins, W. (2004). Making Metadata go away: Hiding everything but the benefits. In *Proceedings of the DCMI 2004 conference*, pages 29–35, Shanghai, China. Dublin Core Metadata Initiative.

[Duval et al., 2007] Duval, E., Ternier, S., Wolpers, M., Najjar, J., Vandeputte, B., Verbert, K., Klerkx, J., Meire, M., and Ochoa, X. (2007). Open metadata for open educational resources in an open infrastructure. In McAndrew, P. and Watts, J., editors, *Proceedings of the OpenLearn2007 conference: researching open content in education*, pages 36–38, Milton Keynes, UK. Open University.

[Duval et al., 2001] Duval, E., Warkentyne, K., Haenni, F., Forte, E., Cardinaels, K., Verhoeven, B., Van Durm, R., Hendrikx, K., Forte, M., Ebel, N., et al. (2001). The ariadne knowledge pool system. *Communications of the ACM*, 44(5):72–78.

[Ede, 1995] Ede, S. (1995). Fitness for purpose: The future evolution of bibliographic records and their delivery. *Catalogue & Index*, 116:1–3.

[Egghe, 2005] Egghe, L. (2005). The power of power laws and an interpretation of lotkaian informetric systems as self-similar fractals. *Journal of the American Society for Information Science and Technology*, 56(7):669–675.

[Egghe and Rousseau, 1995] Egghe, L. and Rousseau, R. (1995). Generalized success-breeds-success principle leading to time-dependent informetric distributions. *Journal of the American Society for Information Science*, 46(6):426–445.

[Egghe and Rousseau, 2006] Egghe, L. and Rousseau, R. (2006). Systems without low-productive sources. *Information Processing and Management*, 42(6):1428–1441.

[Elliott and Sweeney, 2008] Elliott, K. and Sweeney, K. (2008). Quantifying the reuse of learning objects. *Australasian Journal of Educational Technology*, 24(2):137–142.

[Engelbart, 1995] Engelbart, D. (1995). Toward Augmenting the Human Intellect and Boosting our Collective IQ. *Communications of the ACM*, 38(8):3033.

[Epstein, 1948] Epstein, B. (1948). Some applications of the Mellin transform in statistics. *The Annals of Mathematical Statistics*, 19(3):370–379.

[Erl, 2004] Erl, T. (2004). *Service-oriented architecture*. Prentice Hall PTR, Upper Saddle River, NJ.

[Eschenfelder and Desai, 2004] Eschenfelder, K. and Desai, A. (2004). Software as Protest: The Unexpected Resiliency of US-Based DeCSS Posting and Linking. *The Information Society*, 20(2):101–116.

[Fagin et al., 2004] Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., and Vee, E. (2004). Comparing and aggregating rankings with ties. In Beeri, C., editor, *PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 47–58, New York, NY, USA. ACM Press.

[Foltz et al., 1998] Foltz, P. W., Kintsch, W., and Landauer, T. K. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, 25:285–307.

[Fosnot, 1996] Fosnot, C. (1996). *Constructivism. Theory, Perspectives, and Practice.* Teachers College Press, New York, NY.

[Friesen, 2004] Friesen, N. (2004). *Online Education Using Learning Objects*, chapter Three Objections to Learning Objects and E-learning Standards, pages 59–70. Routledge.

[Garfield, 1994] Garfield, E. (1994). The impact factor. *Current Contents*, 25(20):3–7.

[Goldstein et al., 2004] Goldstein, M., Morris, S., and Yen, G. (2004). Problems with fitting to the power-law distribution. *The European Physical Journal B-Condensed Matter*, 41(2):255–258.

[Goth, 2005] Goth, G. (2005). In brief: colleges taking file-sharing into their own hands. *Distributed Systems Online, IEEE*, 6(5):3.

[Greenberg et al., 2001] Greenberg, J., Pattuelli, M. C., Parsia, B., and Robertson, W. D. (2001). Author-generated dublin core metadata for web resources: A baseline study in an organization. In Oyama, K. and Gotoda, H., editors, *DC '01: Proceedings of the International Conference on Dublin Core and Metadata Applications 2001*, pages 38–46. National Institute of Informatics.

[Gruber and Gruber, 2007] Gruber, T. and Gruber, T. (2007). Ontology of folksonomy: A mash-up of apples and oranges. *International Journal on Semantic Web and Information Systems*, 3(1):1–11.

[Gulli and Signorini, 2005] Gulli, A. and Signorini, A. (2005). The indexable web is more than 11.5 billion pages. In Douglis, F. and Raghavan, P., editors, *International World Wide Web Conference*, pages 902–903, New York, NY. ACM Press.

[Guth and Kppen, 2002] Guth, S. and Kppen, E. (2002). Electronic rights enforcement for learning media. In Petrushin, V., Kommers, P., Kinshuk, and Galeev, I., editors, *Proceedings of the IEEE International Conference on Advances Learning Technologies (ICALT) 2002*, pages 496–501, Kazan, Tartastan (Russia).

[Guy et al., 2004] Guy, M., Powell, A., and Day, M. (2004). Improving the quality of metadata in eprint archives. *Ariadne*, 38:5.

[Harman, 1993] Harman, D. (1993). Overview of the first TREC conference. In Korfhage, R., Rasmussen, E. M., and Willett, P., editors, *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 36–47, New York, NY, USA. ACM Press.

[Harrington et al., 2004] Harrington, C., Gordon, S., and Schibik, T. (2004). Course management system utilization and implications for practice: A national survey of department chairpersons. *Online Journal of Distance Learning Administration*, 7(4):13.

[Hatcher and Gospodnetic, 2004] Hatcher, E. and Gospodnetic, O. (2004). *Lucene in Action (In Action series)*. Manning Publications Co., Greenwich, CT, USA.

[Herbrich et al., 2000] Herbrich, R., Graepel, T., and Obermayer, K. (2000). *Large Margin Rank Boundaries for Ordinal Regression*, chapter Large Margin Rank Boundaries for Ordinal Regression, pages 115–132. MIT Press.

[Hirsch, 2005] Hirsch, J. (2005). An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences*, 102(46):16569–16572.

[Hodgins, 2002] Hodgins, H. (2002). *The Instructional Use of Learning Objects*, chapter The future of learning objects, pages 281–298. Agency for Instructional Technology.

[Hood and Wilson, 2001] Hood, W. and Wilson, C. (2001). The literature of bibliometrics, scientometrics, and informetrics. *Scientometrics*, 52(2):291–314.

[Huber, 2002] Huber, J. (2002). A new model that generates lotka's law. *Journal of the American Society for Information Science and Technology*, 53(3):209–219.

[Hughes, 2004] Hughes, B. (2004). Metadata quality evaluation: Experience from the open language archives community. In Chen, Z., Chen, H., Miao, Q., Fu, Y., Fox, E., and Lim, E., editors, *Digital Libraries: International Collaboration and Cross-Fertilization: Proceedings of the 7th International Conference on Asian Digital Libraries, ICADL 2004*, pages 320–329, Shangay, China. Springer Verlag.

[Hughes and Kamat, 2005] Hughes, B. and Kamat, A. (2005). A metadata search engine for digital language archives. *D-Lib Magazine*, 11(2):6.

[Hummel et al., 2005] Hummel, H., Burgos, D., Tattersall, C., Brouns, F., Kurvers, H., and Koper, R. (2005). Encouraging contributions in learning networks using incentive mechanisms. *Journal of Computer Assisted Learning*, 21(5):355–365.

[IEEE, 2002] IEEE (2002). IEEE 1484.12.1 Standard: Learning Object Metadata, http://ltsc.ieee.org/wg12/par1484-12-1.html, retrieved 2/04/2007.

[Jacsó, 2001] Jacsó, P. (2001). A deficiency in the algorithm for calculating the impact factor of scholarly journals: The journal impact factor. *Cortex*, 37(4):590–594.

[Jeh and Widom, 2002] Jeh, G. and Widom, J. (2002). Simrank: a measure of structural-context similarity. In Hand, D., Keim, D., and Ng, R., editors, *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543, New York, NY. ACM Press.

[Joachims et al., 2007] Joachims, T., Li, H., Liu, T.-Y., and Zhai, C. (2007). Learning to rank for information retrieval (lr4ir 2007). *SIGIR Forum*, 41(2):58–62.

[Joachims and Radlinski, 2007] Joachims, T. and Radlinski, F. (2007). Search engines that learn from implicit feedback. *Computer*, 40(8):34–40.

[JOCW, 2006] JOCW (2006). A case study in open educational resources production and use in higher education. Technical report, Japan Open Courseware Consortium.

[Johnson, 2003] Johnson, L. (2003). Elusive vision: Challenges impeding the learning object economy. Technical report, Macromedia.

[Joyce, 2007] Joyce, A. (2007). OECD Study of OER: Forum Report. Technical report, UNESCO.

[Kirsch, 1998] Kirsch, S. (1998). Infoseek's experiences searching the internet. *SIGIR Forum*, 32(2):3–7.

[Kleinberg, 1999] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632.

[Kraft et al., 2006] Kraft, R., Chang, C. C., Maghoul, F., and Kumar, R. (2006). Searching with context. In Goble, C. and Dahlin, M., editors, *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 477–486, New York, NY. ACM Press.

[Kraft et al., 2005] Kraft, R., Maghoul, F., and Chang, C. C. (2005). Y!q: contextual search at the point of inspiration. In Douglis, F. and Raghavan, P., editors, *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 816–823, New York, NY. ACM Press.

[Kumar et al., 2001] Kumar, M., Merriman, J., and Long, P. (2001). New horizons: Building "open" frameworks for education. *EDUCAUSE Review*, 36(6):80–81.

[Lagoze et al., 2006] Lagoze, C., Payette, S., Shin, E., and Wilper, C. (2006). Fedora: an architecture for complex objects and their relationships. *International Journal on Digital Libraries*, 6(2):124–138.

[Landauer et al., 1998] Landauer, T., Foltz, P., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284.

[Laurent et al., 2001] Laurent, S., Johnston, J., and Dumbill, E. (2001). *Programming Web Services with XML-RPC*. O'Reilly Media, Inc.

[Law et al., 2006] Law, E. L.-C., Klobucar, T., and Pipan, M. (2006). User effect in evaluating personalized information retrieval systems. In Nejdl, W. and Tochterman, K., editors, *Proceedings of the First European Conference on*

Technology Enhanced Learning, ECTEL 2006, pages 257–271, Create, Greece. Springer Berlin / Heidelberg.

[Liber, 2005] Liber, O. (2005). Learning objects: conditions for viability. *Journal of Computer Assisted Learning*, 21(5):366–373.

[Linden et al., 2003] Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80.

[Lindsey, 1980] Lindsey, D. (1980). Production and citation measures in the sociology of science: The problem of multiple authorship. *Social Studies of Science*, 10(2):145–162.

[Littlejohn, 2003] Littlejohn, A. (2003). *Reusing Online Resources: A Sustainable Approach to E-Learning*, chapter Issues in reusing online resources, pages 1–7. Kogan Page Ltd.

[Liu et al., 2003a] Liu, Q., Safavi-Naini, R., and Sheppard, N. (2003a). Digital rights management for content distribution. In Johnson, C., Montague, P., and Steketee, C., editors, *Conferences in Research and Practice in Information Technology Series: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003*, volume 34, pages 49–58, Adelaide, Australia. Australian Computer Society, Inc.

[Liu et al., 2005] Liu, Q., Yang, Z., Yan, K., Jin, J., and Deng, W. (2005). Research on drm-enabled learning objects model. In Srimani, P., editor, *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*, volume 2, pages 772–773. IEEE Computer Society Washington, DC, USA.

[Liu et al., 2003b] Liu, X., El Saddik, A., and Georganas, N. (2003b). An implementable architecture of an e-learning system. In Olivier, G., Pierre, S., and Sood, V. K., editors, *Proceedings of the Canadian Conference on Electrical and Computer Engineering, 2003.*, volume 2, pages 717 – 720.

[Liu et al., 2001] Liu, X., Maly, K., Zubair, M., and Nelson, M. L. (2001). Arc - an oai service provider for digital library federation. *D-Lib Magazine*, 7(4):12.

[Lotka, 1926] Lotka, A. (1926). The frequency distribution of scientific productivity. *Journal of the Washington Academy of Sciences*, 16(12):317–323.

[Lubas et al., 2004] Lubas, R., Wolfe, R., and Fleischman, M. (2004). Creating metadata practices for MIT's OpenCourseWare Project. *Library Hi Tech*, 22(2):138–143.

[Lyman and Varian, 2000] Lyman, P. and Varian, H. R. (2000). How much information? *Journal of Electronic Publishing*, 6(2):8.

[Malloy and Hanley, 2001] Malloy, T. and Hanley, G. (2001). MERLOT: A faculty-focused Web site of educational resources. *Behavior Research Methods, Instruments, & Computers*, 33(2):274–276.

[Malloy et al., 2002] Malloy, T., Jensen, G., Regan, A., and Reddick, M. (2002). Open courseware and shared knowledge in higher education. *Behavior Research Methods, Instruments, & Computers*, 34(2):200–203.

[Markus, 2001] Markus, M. (2001). Toward a theory of knowledge reuse: Types of knowledge reuse situations and factors in reuse success. *Journal of Management Information Systems*, 18(1):57–93.

[Massey, 1951] Massey, F. (1951). The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78.

[Matsuo et al., 2006] Matsuo, Y., Mori, J., Hamasaki, M., Ishida, K., Nishimura, T., Takeda, H., Hasida, K., and Ishizuka, M. (2006). Polyphonet: an advanced social network extraction system from the web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 397–406, New York, NY, USA. ACM.

[McAndrew, 2006] McAndrew, P. (2006). Motivations for OpenLearn: The Open Universitys open content initiative. Technical report, OECD Open Educational Resources.

[McCallum and Peterson, 1982] McCallum, D. R. and Peterson, J. L. (1982). Computer-based readability indexes. In Burns, W. J. and Ward, D. L., editors, *ACM 82: Proceedings of the ACM '82 conference*, pages 44–48, New York, NY, USA. ACM Press.

[McGreal, 2004] McGreal, R. (2004). Learning objects: A practical definition. *International Journal of Instructional Technology & Distance Learning*, 1(9):9.

[McGreal, 2007] McGreal, R. (2007). A typology of learning object repositories. [pre-print]. Retrieved December 19, 2007 from http://hdl.handle.net/2149/1078.

[McNaught, 2003] McNaught, C. (2003). *Reusing Online Resources: A Sustainable Approach to E-Learning*, chapter Identifying the complexity of factors in the sharing and reuse of resources, pages 199–210. Kogan Page Ltd.

[Medelyan and Witten, 2006] Medelyan, O. and Witten, I. (2006). Thesaurus based automatic keyphrase indexing. In Nelson, M. L. and Marshall, C. C., editors, *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 296–297, New York, NY. ACM Press.

[Meire et al., 2007] Meire, M., Ochoa, X., and Duval, E. (2007). Samgi: Automatic metadata generation v2.0. In Seale, C. M. . J., editor, *Proceedings of the ED-MEDIA 2007 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 1195-1204, Chesapeake, VA. AACE.

[Mitzenmacher, 2003] Mitzenmacher, M. (2003). A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2):226–251.

[Mobasher et al., 2000] Mobasher, B., Cooley, R., and Srivastava, J. (2000). Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151.

[Moen et al., 1998] Moen, W. E., Stewart, E. L., and McClure, C. R. (1998). Assessing Metadata Quality: Findings and Methodological Considerations from an Evaluation of the U.S. Government Information Locator Service (GILS). In Smith, T. R., editor, *ADL '98: Proceedings of the Advances in Digital Libraries Conference*, pages 246–255, Washington, DC, USA. IEEE Computer Society.

[Montroll and Shlesinger, 1982] Montroll, E. and Shlesinger, M. (1982). On 1/f noise and other distributions with long tails. *Proceedings of the National Academy of Sciences of the United States of America*, 79(10):3380–3383.

[Moore, 2002] Moore, G. (2002). *Crossing the Chasm: Marketing and Selling Disruptive Products to Mainstream Customers*. Collins.

[Najjar et al., 2005] Najjar, J., Klerkx, J., Vuorikari, R., and Duval, E. (2005). Finding appropriate learning objects: An empirical evaluation. In Rauber, A., Christodoulakis, S., and Tjoa, A. M., editors, *Proceedings of : 9th European Conference on Research and Advanced Technology for Digital Libraries. ECDL 2005*, volume 3652 of *Lecture Notes in Computer Science*, pages 323–335, Vienna, Austria. Springer Verlag.

[Najjar et al., 2003] Najjar, J., Ternier, S., and Duval, E. (2003). The actual use of metadata in ariadne: an empirical analysis. In Duval, E., editor, *Proceedings of the 3rd Annual ARIADNE Conference*, pages 1–6. ARIADNE Foundation.

[Najjar et al., 2004] Najjar, J., Ternier, S., and Duval, E. (2004). User behavior in learning objects repositories: An empirical analysis. In McLoughlin, L. C. . C., editor, *Proceedings of the ED-MEDIA 2004 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 4373–4378, Chesapeake, VA. AACE.

[Najjar et al., 2006] Najjar, J., Wolpers, M., and Duval, E. (2006). Attention metadata: Collection and management. In Goble, C. and Dahlin, M., editors, *Proceedings of the 15th international conference on World Wide Web, workshop on Logging Traces of Web Activity*, page 4, Edinburgh, Scotland. IEEE.

[Nesbit et al., 2002] Nesbit, J., Belfer, K., and Vargo, J. (2002). A convergent participation model for evaluation of learning objects. *Canadian Journal of Learning and Technology*, 28(3):105–120.

[Neven and Duval, 2002] Neven, F. and Duval, E. (2002). Reusable learning objects: a survey of lom-based repositories. In Muhlhauser, M., Ross, K., and Dimitrova, N., editors, *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 291–294, New York, NY. ACM Press.

[Newman, 2005] Newman, M. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46(5):323–351.

[Newman et al., 2006] Newman, M., Watts, D., and Barabsi, A.-L. (2006). *The Structure and Dynamics of Networks*. Princeton University Press.

[Ochoa, 2005] Ochoa, X. (2005). Learning Object Repositories are Useful, but are they Usable. In Nuno Guimares, P. T. I., editor, *Proceedings of IADIS International Conference Applied Computing*, pages 138–144, Algarve, Portugal. IADIS.

[Ochoa et al., 2005] Ochoa, X., Cardinaels, K., Meire, M., and Duval, E. (2005). Frameworks for the automatic indexation of learning management systems content into learning object repositories. In Kommers, P. and Richards, G., editors, *Proceedings of the ED-MEDIA 2005 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 1407–1414, Chesapeake, VA. AACE.

[Ochoa and Duval, 2006a] Ochoa, X. and Duval, E. (2006a). Quality metrics for learning object metadata. In Pearson, E. and Bohman, P., editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006*, pages 1004–1011, Chesapeake, VA. AACE.

[Ochoa and Duval, 2006b] Ochoa, X. and Duval, E. (2006b). Towards automatic evaluation of learning object metadata quality. In Embley, D. W., Oliv, A., and Ram, S., editors, *Advances in Conceptual Modeling - Theory and Practice: Proceedings of the 25th ER Conference*, volume 4231 of *Lecture Notes in Computer Science*, pages 372–381, Tucson, AZ. Springer Berlin / Heidelberg.

[Ochoa and Duval, 2006c] Ochoa, X. and Duval, E. (2006c). Use of contextualized attention metadata for ranking and recommending learning objects. In Duval, E., Najjar, J., and Wolpers, M., editors, *CAMA '06: Proceedings of the 1st international workshop on Contextualized attention metadata*, pages 9–16, New York, NY. ACM Press.

[Ochoa and Duval, 2007a] Ochoa, X. and Duval, E. (2007a). Relevance ranking metrics for learning objects. In Duval, E., Klamma, R., and Wolpers, M., editors, *Creating New Learning Experiences on a Global Scale: Proceedings of the Second European Conference on Technology Enhanced Learning*, volume 4753 of *Lecture Notes in Computer Science*, pages 262–276, Crete, Greece. Springer Verlang.

[Ochoa and Duval, 2007b] Ochoa, X. and Duval, E. (2007b). Relevance ranking of learning objects based on usage and contextual information. In Alvarez, L., editor, *Proceedings of the Second Latin American Conference on Learning Objects*, pages 149–156, Santiago, Chile. LACLO.

[Ochoa and Duval, 2008a] Ochoa, X. and Duval, E. (2008a). Quantitative analysis of learning object repositories. In *Proceedings of the ED-MEDIA 2008 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 6031–6048, Chesapeake, VA. AACE.

[Ochoa and Duval, 2008b] Ochoa, X. and Duval, E. (2008b). Quantitative analysis of user-generated content on the web. In De Roure, D. and Hall, W., editors, *Proceedings of the First International Workshop on Understanding Web Evolution (WebEvolve2008)*, pages 19–26, Beijing, China. Web Science Research Initiative. ISBN: 978 085432885 7.

[Oliver, 2005] Oliver, R. (2005). Ten more years of educational technologies in education: How far have we travelled. *Australian Educational Computing*, 20(1):18–23.

[Olmedilla, 2007] Olmedilla, D. (2007). *Realizing Interoperability of E-Learning Repositories*. PhD thesis, Universidad Autónoma de Madrid, Madrid, Spain. Grade "Summa Cum Laude".

[O'Neill, 2002] O'Neill, E. T. (2002). FRBR: Functional Requirements for Bibliographic Records; Application of the entity-relationship model to Humphry Clinker. *Library Resources & Technical Services*, 46(4):150–159.

[Page et al., 1998] Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project.

[Pampalk et al., 2005] Pampalk, E., Pohle, T., and Widmer, G. (2005). Dynamic playlist generation based on skipping behavior. In Crawford, T. and Sandler, M., editors, *Proceedings of ISMIR 2005 Sixth International Conference on Music Information Retrieval*, pages 634–637, London, UK.

[Pao, 1986] Pao, M. (1986). An empirical examination of Lotka's Law. *Journal of the American Society for Information Science*, 37(1):26–33.

[Pigeau et al., 2003] Pigeau, A., Raschia, G., Gelgon, M., Mouaddib, N., and Saint-Paul, R. (2003). A fuzzy linguistic summarization technique for tv recommender systems. In Nasraoui, O., Frigui, H., and Keller, J. M., editors, *Proceedings of the IEEE Int. Conf. of Fuzzy Systems (FUZZ-IEEE'2003)*, volume 1, pages 743–748, St-Louis, USA. IEEE.

[Pitkow et al., 2002] Pitkow, J., Schütze, H., Cass, T., Turnbull, D., Edmonds, A., and Adar, E. (2002). Personalized search. *Communications of the ACM*, 45(9):50–55.

[Polsani, 2003] Polsani, P. (2003). Use and abuse of reusable learning objects. *Journal of Digital Information*, 3(4):2003–02.

[Price, 1976] Price, D. (1976). A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science*, 27(5-6):292–306.

[Quiroga and Mostafa, 1999] Quiroga, L. M. and Mostafa, J. (1999). Empirical evaluation of explicit versus implicit acquisition of user profiles in information filtering systems. In Rowe, N. and Fox, E. A., editors, *DL '99: Proceedings of the fourth ACM conference on Digital libraries*, pages 238–239, New York, NY. ACM Press.

[Raykar et al., 2007] Raykar, V., Duraiswami, R., and Krishnapuram, B. (2007). A fast algorithm for learning large scale preference relations. In Meila, M. and Shen, X., editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 2007)*, volume 2, pages 388–395, San Juan, Puerto Rico.

[Rehak and Mason, 2003] Rehak, D. and Mason, R. (2003). *Reusing online resources: A sustainable approach to e-learning*, chapter Keeping the learning in learning objects, pages 20–34. Kogan Page Ltd.

[Richards et al., 2002] Richards, G., McGreal, R., Hatala, M., and Friesen, N. (2002). The evolution of learning object repository technologies: Portals for on-line objects for learning. *Journal of Distance Education*, 17(3):67–79.

[Richardson et al., 2006] Richardson, M., Prakash, A., and Brill, E. (2006). Beyond pagerank: machine learning for static ranking. In Goble, C. and Dahlin, M., editors, *Proceedings of the 15th international conference on World Wide Web*, pages 707–715, New York, NY. ACM Press.

[Rousseau, 1988] Rousseau, R. (1988). Lotkas law and its leimkuhler representation. *Library Science with a Slant to Documentation Studies*, 25:150–178.

[Rousseau, 1997] Rousseau, R. (1997). Sitations: an exploratory study. *Cybermetrics*, 1(1):7.

[Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, 24(5):513–523.

[Salton and McGill, 1986] Salton, G. and McGill, M. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY.

[Salton et al., 1975] Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

[Schoner et al., 2005] Schoner, V., Buzza, D., Harrigan, K., and Strampel, K. (2005). Learning objects in use:liteassessment for field studies. *J. Online Learning Teaching*, 1(1):18.

[Shannon and Weaver, 1963] Shannon, C. and Weaver, W. (1963). *The Mathematical Theory of Communication*. University of Illinois Press.

[Shockley, 1957] Shockley, W. (1957). On the statistics of individual variations of productivity in research laboratories. *Proceedings of the IRE*, 45(3):279–290.

[Shreeves et al., 2005] Shreeves, S. L., Knutson, E. M., Stvilia, B., Palmer, C. L., Twidale, M. B., and Cole, T. W. (2005). Is "quality" metadata "shareable" metadata? the implications of local metadata practices for federated collections. In Thompson, H. A., editor, *Currents And Convergence: Navigating the Rivers of Change: Proceedings of the Twelfth National Conference of the Association of College and Research Libraries*, pages 223–237, Minneapolis, USA. ALA.

[Shrout and Fleiss, 1977] Shrout, P. and Fleiss, J. (1977). Intraclass correlations: uses in assessing rater reliability. *Psychol Bull*, 86:420–428.

[Sicilia and García, 2003] Sicilia, M. and García, E. (2003). On the concepts of usability and reusability of learning objects. *International Review of Research in Open and Distance Learning*, 4(2):11.

[Sicilia et al., 2005] Sicilia, M., Garcia, E., Pages, C., and Martinez, J. (2005). Complete metadata records in learning object repositories: some evidence and requirements. *International Journal of Learning Technology*, 1(4):411–424.

[Simon et al., 2005] Simon, B., Massart, D., van Assche, F., Ternier, S., Duval, E., Brantner, S., Olmedilla, D., and Miklos, Z. (2005). A simple query interface for interoperable learning repositories. In Olmedilla, D., Saito, N., and Simon, B., editors, *Proceedings of the 1st Workshop on Interoperability of Web-based Educational Systems*, pages 11–18, Chiba, Japan. CEUR.

[Sokvitne, 2000] Sokvitne, L. (2000). An evaluation of the effectiveness of current dublin core metadata for retrieval. In *Proceedings of VALA (Libraries, Technology and the Future) Biennial Conference*, page 15, Victoria, Australia. Victorian Association for Library Automation Inc.

[Sparck Jones, 1972] Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

[Stata et al., 2000] Stata, R., Bharat, K., and Maghoul, F. (2000). The term vector database: fast access to indexing terms for web pages. *Computer Networks*, 33(1-6):247–255.

[Strong et al., 1997] Strong, D. M., Lee, Y. W., and Wang, R. Y. (1997). Data quality in context. *Communications of the ACM*, 40(5):103–110.

[Stvilia, 2006] Stvilia, B. (2006). *Measuring information quality*. PhD thesis, University of Illinois at Urbana - Champaign, Urbana, IL.

[Stvilia et al., 2006] Stvilia, B., Gasser, L., and Twidale, M. (2006). *Information quality management: theory and applications*, chapter Metadata quality problems in federated collections, pages 154–18. Idea Group, Hershey, PA.

[Stvilia et al., 2007] Stvilia, B., Gasser, L., and Twidale, M. (2007). A framework for information quality assessment. *Journal of the American Society for Information Science and Technology*, 58(12):1720–1733.

[Sugiyama et al., 2004] Sugiyama, K., Hatano, K., and Yoshikawa, M. (2004). Adaptive web search based on user profile constructed without any effort from users. In Najork, M. and Wills, C., editors, *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 675–684, New York, NY. ACM Press.

[Taleb, 2007] Taleb, N. (2007). *The black swan: the impact of the highly improbable*. New York: Random House.

[Tansley et al., 2003] Tansley, R., Bass, M., Stuve, D., Branschofsky, M., Chudnov, D., McClellan, G., and Smith, M. (2003). The DSpace institutional digital repository system: current functionality. In *Proceedings of the 2003 Joint Conference on Digital Libraries*, pages 87–97. IEEE.

[Ternier, 2008] Ternier, S. (2008). *Standards based Interoperability for Searching in and Publishing to Learning Object Repositories*. PhD thesis, Katholieke Universiteit Leuven.

[Thomas, 1996] Thomas, S. E. (1996). Quality in bibliographic control. *Library Trends*, 44(3):491–505.

[Toffler, 1981] Toffler, A. (1981). *The Third Wave*, chapter The rise of the prosumer, pages 265–288. Bantam Books.

[Upendra, 1994] Upendra, S. (1994). Social information filtering for music recommendation. Master's thesis, Massachusetts Institute of Technology.

[Van de Sompel et al., 2004] Van de Sompel, H., Nelson, M., Lagoze, C., and Warner, S. (2004). Resource Harvesting within the OAI-PMH Framework. *D-Lib Magazine*, 10(12):1082–9873.

[Vandepitte et al., 2003] Vandepitte, P., Van Rentergem, L., Duval, E., Ternier, S., and Neven, F. (2003). Bridging an lcms and an lms: a blackboard building block for the ariadne knowledge pool system. In McNaught, D. L. . C., editor, *Proceedings of ED-MEDIA 2003 World Conference on Educational Multimedia, Hypermedia, and Telecommunications*, pages 423–424, Chesapeake, VA. AACE.

[Vargo et al., 2003] Vargo, J., Nesbit, J. C., Belfer, K., and Archambault, A. (2003). Learning object evaluation: Computer-mediated collaboration and inter-rater reliability. *Iternational Journal of Computers and Applications*, 25:198–205.

[Verbert and Duval, 2007] Verbert, K. and Duval, E. (2007). Evaluating the ALOCOM Approach for Scalable Content Repurposing. In Duval, E., Klamma, R., and Wolpers, M., editors, *Creating New Learning Experiences on a Global Scale: Proceedings of the Second European Conference on Technology Enhanced Learning*, volume 4753, pages 364–377, Crete, Greece. Springer.

[Verbert et al., 2006] Verbert, K., Duval, E., Meire, M., Jovanovic, J., and Gasevic, D. (2006). Ontology-Based Learning Content Repurposing: The ALOCoM Framework. *International Journal on E-Learning*, 5(1):67–74.

[Verbert et al., 2005] Verbert, K., Jovanovic, J., Gasevic, D., and Duval, E. (2005). Repurposing learning object components. In Meersman, R., Tari, Z., and Herrero, P., editors, *On the Move to Meaningful Internet Systems 2005: OTM Workshops*, volume 3762 of *Lecture Notes in Computer Science*, pages 1169–1178, Agia Napa, Cyprus. Springer Berlin / Heidelberg.

[Vuong, 1989] Vuong, Q. (1989). Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica*, 57(2):307–333.

[Wasfi, 1999] Wasfi, A. M. A. (1999). Collecting user access patterns for building user profiles and collaborative filtering. In Maybury, M., Szekely, P., and Thomas, C. G., editors, *IUI '99: Proceedings of the 4th international conference on Intelligent user interfaces*, pages 57–64, New York, NY. ACM, Press.

[Weibel, 2005] Weibel, S. (2005). Border crossings: Reflections on a decade of metadata consensus building. *D-Lib Magazine*, 11(7/8):6.

[Weibel and Koch, 2000] Weibel, S. and Koch, T. (2000). The Dublin Core metadata initiative: Mission, current activities, and future directions. *D-Lib Magazine*, 6(12):9.

[Weitl et al., 2004] Weitl, F., Kammerl, R., and Gstl, M. (2004). Context aware reuse of learning resources. In Cantoni, L. and McLoughlin, C., editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004*, pages 2119–2126, Lugano, Switzerland. AACE.

[Wiley, 2002] Wiley, D. (2002). *The Instructional Use of Learning Objects*, chapter Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy, pages 571–577. Agency for Instructional Technology.

[Wiley et al., 2004] Wiley, D., Waters, S., Dawson, D., Lambert, B., Barclay, M., Wade, D., et al. (2004). Overcoming the limitations of learning objects. *Journal of Educational Multimedia and Hypermedia*, 13(4):507–521.

[Wilhelm and Wilde, 2005] Wilhelm, P. and Wilde, R. (2005). Developing a university course for online delivery based on learning objects: from ideals to compromises. *Open Learning: The Journal of Open and Distance Learning*, 20(1):65–81.

[Wilson, 2007] Wilson, A. J. (2007). Toward releasing the metadata bottleneck - a baseline evaluation of contributor-supplied metadata. *Library Resources & Technical Services*, 51(1):16–28.

[Wold and Whittle, 1957] Wold, H. and Whittle, P. (1957). A model explaining the Pareto distribution of wealth. *Econometrica*, 25(4):591–5.

[Yan and Hauptmann, 2006] Yan, R. and Hauptmann, A. (2006). Efficient margin-based rank learning algorithms for information retrieval. In Leow, W.-K., Lew, M. S., Chua, T.-S., Ma, W.-Y., Chaisorn, L., and Bakker, E. M., editors, *Proceedings of the International Conference on Image and Video Retrieval (CIVR)*, number 4071 in Lecture Notes in Computer Science, pages 113–122, Tempe, AZ. Springer Berlin / Heidelberg.

[Zemsky and Massy, 2004] Zemsky, R. and Massy, W. (2004). Thwarted innovation: What happened to e-learning and why. Technical report, University of Pennsylvania and Thomson Corporation.

[Zhu and Gauch, 2000] Zhu, X. and Gauch, S. (2000). Incorporating quality metrics in centralized/distributed information retrieval on the world wide web. In Yannakoudakis, E., Leong, N. J. B. M.-K., and Ingwersen, P., editors, *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 288–295, New York, NY. ACM Press.

[Zimmermann et al., 2007] Zimmermann, B., Meyer, M., Rensing, C., and Steinmetz, R. (2007). Improving retrieval of reusable learning resources by estimating adaptation effort. In Massart, D., Colin, J.-N., and Assche, F. V., editors, *Proceedings of the First International Workshop on Learning Object Discovery & Exchange*, volume 311, page 8, Crete, Greece. CEUR.

[zur Muehlen et al., 2005] zur Muehlen, M., Nickerson, J., and Swenson, K. (2005). Developing web services choreography standardsthe case of rest vs. soap. *Decision Support Systems*, 40(1):9–29.

Appendix A

Metadata Quality Metrics Interface

The Metadata Quality Metrics Service should be able to accommodate metrics that work at three different levels: Repository, Subset and Single Instance. At Repository Level, the metrics calculation provide average values for the whole pool of metadata accessible from the Metric Service, usually a whole LOR. This calls could help to have a constant indication of the "health" of the metadata in the repository. At Subset level, the user specify, through a query, the subset of records that wants to analyze. These calls could be used to analyze the quality of records published by a given contributor or during a specific period of time. At Single Instance Level, the metrics should be calculated for an explicit instance which identification or complete content is provided together with the call to the metric. This level is more useful to check the quality of new metadata instances or to analyze in more detail the existing metadata. For each of these levels, the Metadata Quality Metrics Interface (MeQuMi) provides a specific set of calls. This appendix provides a technical description of those call sets and their results for each level.

A.1 Metadata Quality Metrics Description

This family of calls returns an XML structure that describes the quality metrics that can be applied at the specified level, their description, extreme and average values and to what fields they can be applied individually. The call for these functions is different depending on the level to which they operate.

Table A.1: Repository Level Metadata Quality Metrics Description

Method Name	repositoryGetMQMetricsDescriptions
Return Type	String
Parameters	None
Faults	GENERAL_METRICS_ERROR

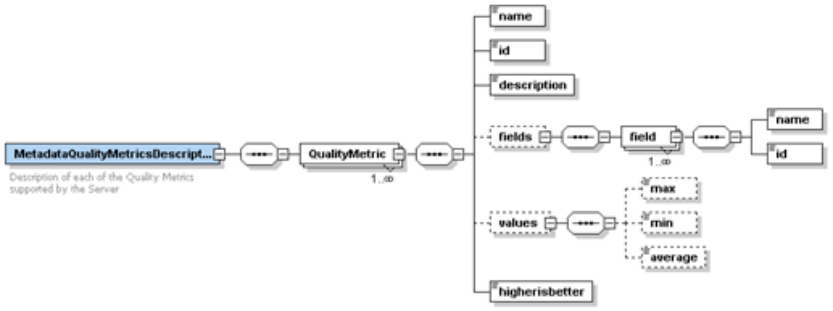


Figure A.1: Schema of the result of GetMQMetricsDescriptions

A.1.1 Repository Level

Table A.1 describes the call to this function. The call returns an XML string that contains information about which are the metrics that can be calculated at repository level, the identification string to call them, a description of the calculation that is performed, the fields where it can be independently applied and the maximum and minimum value and what level of quality they represent. The structure of the returning XML can be seen in Figure A.1. An example of this XML response can be found in Listing A.1.

Listing A.1: Returned XML from GetMQMetricsDescriptions

```
<?xml version="1.0" encoding="UTF-8"?>
<MetadataQualityMetricsDescriptions>
 <QualityMetric>
  <name>Completeness</name>
  <id>completeness</id>
  <description>
   Calculate the percentage of the fields of the
   metadata standard that have been filled
  </description>
  <values>
   <max>100</max>
   <min>0</min>
```

```
    <average>58.2</average>
   </values>
   <higherisbetter>true</higherisbetter>
  </QualityMetric>
  <QualityMetric>
   <name>Textual Information Content</name>
   <id>tinfo</id>
   <description>
    Measure the amount and importance of the
    text stored in free text fields
   </description>
   <fields>
    <field>
     <name>General:Title</name>
     <id>title</id>
    </field>
    <field>
     <name>General:Description</name>
     <id>description</id>
    </field>
   </fields>
   <values>
    <min>0</min>
    <average>14.8</average>
   </values>
   <higherisbetter>true</higherisbetter>
  </QualityMetric>
</MetadataQualityMetricsDescriptions>
```

A.1.2 Subset Level

Table A.2 describes the call to this function. This call returns which metrics are available at Subset level, the identification string to call them, a description of the calculation that is performed, the fields where it can be independently applied and the maximum, minimum and average value and what level of quality they represent. The result is codified in XML and returned in a String. The schema of this XML is the same as described in the repositoryGetMQMetricsDescriptions call (Figure A.1).

A.1.3 Instance Level

Table A.3 describes the call to this function. This call returns which metrics are available at Single Instance level, the identification string to call them, a description of the calculation that is performed, the fields where it can be independently

Table A.2: Subset Metadata Quality Metrics Description

Method Name	subsetGetMQMetricsDescriptions
Return Type	String
Parameters	None
Faults	GENERAL_METRICS_ERROR

Table A.3: Instance Metadata Qualiy Metrics Description

Method Name	instanceGetMQMetricsDescriptions
Return Type	String
Parameters	None
Faults	GENERAL_METRICS_ERROR

applied and the maximum, minimum and average value and what level of quality they represent. The result is codified in XML and returned in a String. The schema of this XML is the same as described in the repositoryGetMQMetricsDescriptions call (Figure A.1).

A.2 Calculate All Metadata Quality Metrics

This family of calls calculate all the available metrics for the group of metadata instances specified by the level of the call. The response to this call is an XML structure that return the value of the calculated metrics.

A.2.1 Repository Level

Table A.4 describes the call to this function. This function calculates all the available quality metrics for all metadata instances in the repository and averages their value metric-wise. The result provides an XML encoded in a String. The schema of this XML is provided graphically at Figure A.2. An example response can be seen in Listing A.2.

Table A.4: Repository Level Calculate All Metadata Quality Metrics

Method Name	repositoryGetAllMQMetricsValues
Return Type	String
Parameters	None
Faults	GENERAL_METRICS_ERROR EMPTY_REPOSITORY

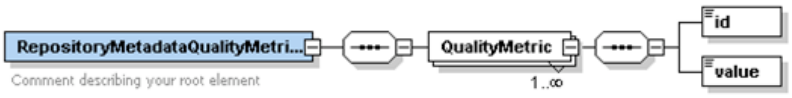


Figure A.2: Schema of the result of repositoryGetAllMQMetricsValues

Listing A.2: Returned XML from repositoryGetAllMQMetricsValues

```
<?xml version="1.0" encoding="UTF-8"?>
<RepositoryMetadataQualityMetrics>
 <QualityMetric>
  <id>completeness</id>
  <value>50.8</value>
 </QualityMetric>
 <QualityMetric>
  <id>wcompleteness</id>
  <value>80.2</value>
 </QualityMetric>
 <QualityMetric>
  <id>ninfo</id>
  <value>35</value>
 </QualityMetric>
 <QualityMetric>
  <id>tinfo</id>
  <value>32</value>
 </QualityMetric>
</RepositoryMetadataQualityMetrics>
```

A.2.2 Subset Level

Table A.5 describes the call to this function. This call calculates all the available quality metrics for the subset of metadata instances resulting from applying the specified query to the repository. The query is expressed as ProLearn Query Language (PLQL)[1]. The quality metrics values are averaged metric-wise among all the instances in the subset. An example call to this function is presented in Listing A.4. The result is provided as an XML encoded in a String. The structure of this XML is graphically presented in Figure A.3. An example of a possible response is presented in Listing A.4.

Listing A.3: Returned XML from GetAllMQMetricsValues

```
String query = "lom.general.title.string='test'";
```

[1]PLQL. http://ariadne.cs.kuleuven.be/lomi/index.php/QueryLanguages_stable

Table A.5: Subset Level Calculate All Metadata Quality Metrics

Method Name	subsetGetAllMQMetricsValues
Return Type	String
Parameters	String query
Faults	GENERAL_METRICS_ERROR EMPTY_SUBSET INVALID_QUERY

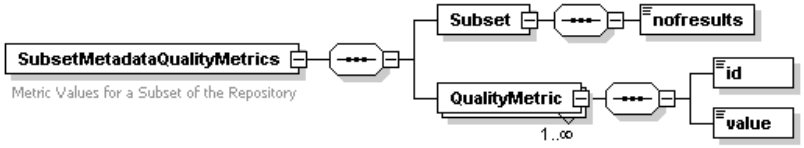


Figure A.3: Schema of the result of SubsetGetAllMQMetricsValues

```
String results = subsetGetAllMQMetricsValues(query);
```

Listing A.4: Example response to subsetGetAllMQMetricsValues

```
<?xml version="1.0" encoding="UTF-8"?>
<SubsetMetadataQualityMetrics>
<Subset>
  <nofresults>75</id>
 </Subset>
<QualityMetric>
  <id>completeness</id>
  <value>50.8</value>
 </QualityMetric>
 <QualityMetric>
  <id>wcompleteness</id>
  <value>80.2</value>
 </QualityMetric>
 <QualityMetric>
  <id>ninfo</id>
  <value>35</value>
 </QualityMetric>
 <QualityMetric>
  <id>tinfo</id>
  <value>32</value>
 </QualityMetric>
</SubsetMetadataQualityMetrics>
```

Table A.6: Instance Level Calculate All Metadata Quality Metrics

Method Name	instanceGetAllMQMetricsValues
Return Type	String
Parameters	String[] instancesId String[] instances
Faults	GENERAL_METRICS_ERROR NOT_VALID_RECORD MISMATCH_PARAMETERS

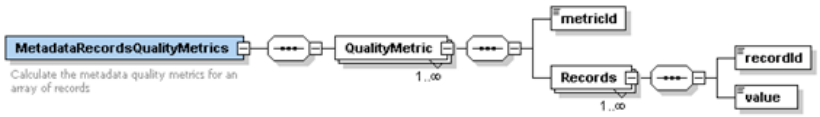


Figure A.4: Schema of the result of instanceGetAllMQMetricsValues

A.2.3 Instance Level

This call calculates all the available quality metrics for all metadata instances supplied in the instance parameter array. A corresponding identifier should also be provided in the instancesId array. The metadata instance should be represented in its XML format. Table A.6 presents the details of the call to this function. An example call to this function is presented in Listing A.5. The returned XML is defined by the Schema depicted in Figure A.4. An example response for this call is presented in Listing A.6.

Listing A.5: Call to instanceGetAllMQMetricsValues

```
String instance1= "<?xml><c>Instance 1</c>"
String instance2= "<?xml><c>Instance 2</c>"
Strring id1 = "one"
String id2  = "two"
String result = instanceGetAllMQMetricsValues
   ([id1,id2],[instance1,instance2])
```

Listing A.6: Example response to instanceGetAllMQMetricsValues

```
<?xml version="1.0" encoding="UTF-8"?>
<MetadataRecordsQualityMetrics>
 <QualityMetric>
  <metricId>completeness</metricId>
  <Records>
   <recordId>one</recordId>
   <value>50</value>
```

```
  </Records>
  <Records>
   <recordId>two</recordId>
   <value>60</value>
  </Records>
 </QualityMetric>
 <QualityMetric>
  <metricId>wcompleteness</metricId>
  <Records>
   <recordId>one</recordId>
   <value>75</value>
  </Records>
  <Records>
   <recordId>two</recordId>
   <value>75</value>
  </Records>
 </QualityMetric>
 <QualityMetric>
  <metricId>tinfo</metricId>
  <Records>
   <recordId>one</recordId>
   <value>10</value>
  </Records>
  <Records>
   <recordId>two</recordId>
   <value>5</value>
  </Records>
 </QualityMetric>
</MetadataRecordsQualityMetrics>
```

A.3 Calculate Selected Metadata Quality Metrics

This family of calls calculates the specified metrics for different groups of metadata instances according to the level of the call. A parameter containing the identifier for the desired metrics is passed to the call of this function.

A.3.1 Repository Level

Table A.7 describes the call to this function. This function calculates the specified metadata quality metrics for all metadata instances in the repository and averages their value metric-wise. An example of the call to this function can be seen in Listing A.7. The response provides an XML encoded in a String. The schema of this XML is the same that the one used for the response of the repositoryGetAllMQMetricsValues presented in Figure A.2.

Table A.7: Repository Level Calculate Metadata Quality Metrics

Method Name	repositoryGetMQMetricsValues
Return Type	String
Parameters	String[] metricsId
Faults	GENERAL_METRICS_ERROR METRIC_NOT_SUPPORTED

Table A.8: Subset Level Calculate Metadata Quality Metrics

Method Name	subsetGetMQMetricsValues
Return Type	String
Parameters	String query String[] metricsId
Faults	GENERAL_METRICS_ERROR METRIC_NOT_SUPPORTED EMPTY_SUBSET INVALID_QUERY

Listing A.7: Call to repositoryGetMQMetricsValues

```
String result =
 repositoryGetQualityMetricsValues(["completeness","tinfo"]);
```

A.3.2 Subset Level

Table A.8 describes the call to this function. This function calculates the specified metadata quality metrics for the subset of metadata instances resulting from applying the specified query to the repository. The query is expressed as ProLearn Query Language (PLQL). An example of the call to this function can be seen in Listing A.8. The response provides an XML encoded in a String. The schema of this XML is the same that the one used for the response of the subsetGetAllMQMetricsValues presented in Figure A.3.

Listing A.8: Call to subsetGetMQMetricsValues

```
String query = "lom.general.title.string='test'";
String result =
 subsetGetQualityMetricsValues(query,["completeness","tinfo"]);
```

A.3.3 Instance Level

Table A.9 describes the call to this function. This function calculates the specified metadata quality metrics for all metadata instances supplied in the instance pa-

Table A.9: Instance Level Calculate Metadata Quality Metrics

Method Name	instanceGetMQMetricsValues
Return Type	String
Parameters	String[] instancesId String[] instances String[] metricsId
Faults	GENERAL_METRICS_ERROR METRIC_NOT_SUPPORTED NOT_VALID_RECORD MISMATCH_PARAMETERS

rameter array. An example of the call to this function can be seen in Listing A.9. The response provides an XML encoded in a String. The schema of this XML is the same that the one used for the response of the instanceGetAllMQMetricsValues presented in Figure A.4.

Listing A.9: Call to instanceGetMQMetricsValues

```
instance1= "<xml?><c>Instance 1</c>"
instance2= "<xml?><c>Instance 2</c>"
id1="one"
id2="two"
String result = recordsGetQualityMetricValues
     ([id1,id2],[instance1,instance2],["completeness","tinfo"])
```

A.4 Calculate a Metadata Quality Metric for Selected Fields

This family of calls calculates one of the quality metrics for a set of specific field in different groups of metadata instances according to the level of the call. A parameter containing the identifier for the desired metric. as well as an array with the list of fields for which the metric can be calculated, are always passed to the function call.

A.4.1 Repository Level

Table A.10 describes the call to this function. This function calculates the specified metadata quality metric for the specified metadata fields for all metadata instances in the repository and averages its value per-field. An example of the call to this function can be seen in Listing A.10. The result provides an XML encoded in

Table A.10: Repository Level Calculate Metadata Quality Metric for Fields

Method Name	repositoryGetMQMetricValuePerField
Return Type	String
Parameters	String metricId String[] fieldsId
Faults	GENERAL_METRICS_ERROR METRIC_NOT_SUPPORTED FIELD_NOT_SUPPORTED



Figure A.5: Schema of the result of repositoryGetMQMetricValuePerField

a String. The schema of this XML is provided graphically at Figure A.5. An example response can be seen in Listing A.12.

Listing A.10: Call to repositoryGetMQMetricValuePerField

```
String result = repositoryGetQualityMetricValuePerField
 ("completeness",["title","aggregationLevel","description"])
```

Listing A.11: Call to repositoryGetMQMetricValuePerField

```
<?xml version="1.0" encoding="UTF-8"?>
<RepositoryMetadataQualityperField>
 <QualityMetric>
  <id>completeness</id>
  <fields>
   <field>
    <id>title</id>
    <value>0</value>
   </field>
   <field>
    <id>aggregationLevel</id>
    <value>100</value>
   </field>
   <field>
    <id>description</id>
    <value>100</value>
   </field>
  </fields>
```

Table A.11: Subset Level Calculate Metadata Quality Metric for Fields

Method Name	subsetGetMQMetricValuePerField
Return Type	String
Parameters	String query String metricId String[] fieldsId
Faults	GENERAL_METRICS_ERROR METRIC_NOT_SUPPORTED FIELD_NOT_SUPPORTED EMPTY_SUBSET INVALID_QUERY

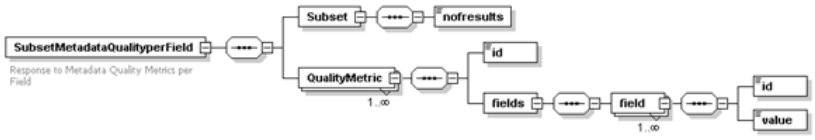


Figure A.6: Schema of the result of subsetGetMQMetricValuePerField

```
 </QualityMetric>
</RepositoryMetadataQualityperField>
```

A.4.2 Subset Level

Table A.11 describes the call to this function. This function calculates the specified metadata quality metric for the specified metadata fields for the subset of metadata instances resulting from applying the specified query to the repository. The query is expressed as ProLearn Query Language (PLQL). An example of the call to this function can be seen in Listing A.12. The result provides an XML encoded in a String. The schema of this XML is provided graphically at Figure A.6. An example response can be seen in Listing A.13.

Listing A.12: Call to subsetGetMQMetricValuePerField

```
String query = "lom.general.title.string='test'";
String result = subsetGetQualityMetricValuePerField
 (query,"completeness",["title","aggregationLevel","description"]);
```

Listing A.13: Call to subsetGetMQMetricValuePerField

```
<?xml version="1.0" encoding="UTF-8"?>
<SubsetMetadataQualityperField>
```

```
 <Subset>
  <nofresults>45</nofresults>
 </Subset>
<QualityMetric>
  <id>completeness</id>
  <fields>
   <field>
    <id>title</id>
    <value>0</value>
   </field>
   <field>
    <id>aggregationLevel</id>
    <value>100</value>
   </field>
   <field>
    <id>description</id>
    <value>100</value>
   </field>
  </fields>
 </QualityMetric>
</SubsetMetadataQualityperField>
```

A.4.3 Instance Level

Table A.12 describes the call to this function. This function calculates the specified metadata quality metric for the specified metadata fieldsfor all metadata instances supplied in the instance parameter array. An example of the call to this function can be seen in Listing A.14. The result provides an XML encoded in a String. The schema of this XML is provided graphically at Figure A.7. An example response can be seen in Listing A.15.

Listing A.14: Call to instanceGetMQMetricValuePerField

```
instance1= "<xml?><c>Instance 1</c>"
instance2= "<xml?><c>Instance 2</c>"
id1="one"
id2="two"
String result = recordsGetQualityMetricValuebyperField
   ([id1,id2],[instance1,instance2],"completeness",
   ["title","description"])
```

Listing A.15: Call to instanceGetMQMetricValuePerField

```
<?xml version="1.0" encoding="UTF-8"?>
<InstanceMetadataQualityperField>
 <QualityMetric>
```

Table A.12: Instance Level Calculate Metadata Quality Metric for Fields

Method Name	instanceGetMQMetricValuePerField
Return Type	String
Parameters	String[] instancesId String[] instances String metricId String[] fieldsId
Faults	GENERAL_METRICS_ERROR METRIC_NOT_SUPPORTED FIELD_NOT_SUPPORTED NOT_VALID_RECORD MISMATCH_PARAMETERS

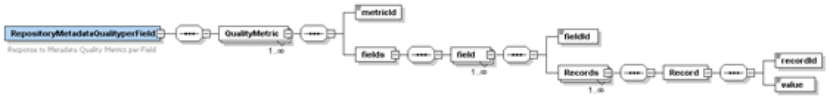


Figure A.7: Schema of the result of instanceGetMQMetricValuePerField

```
<metricId>completeness</metricId>
<fields>
 <field>
  <fieldId>title</fieldId>
  <Records>
   <Record>
   <recordId>one</recordId>
    <value>100</value>
   </Record>
   <Record>
   <recordId>two</recordId>
    <value>100</value>
   </Record>
  </Records>
 </field>
 <field>
  <fieldId>description</fieldId>
  <Records>
   <Record>
    <recordId>one</recordId>
    <value>100</value>
   </Record>
   <Record>
```

```
        <recordId>two</recordId>
        <value>0</value>
       </Record>
      </Records>
     </field>
    </fields>
   </QualityMetric>
</InstanceQualityperField>
```

Appendix B

Ranking Metrics Interface

The Ranking Metrics Service should provide their answer in response to calls made at two different lables:Global and Result List. At Global Level, the metrics calculation provide average values for the whole pool of learning objects accessible from the Metric Service, usually a whole LOR. The metrics at this level can be used to obtain important information about the repository, for example what are the most downloaded objects in the last week or what objects can be recommended to a given user. At Result List level, a list of objects is provided in the call. For these two levels, the Ranking Metrics Service Interface (RaMi) provides a specific set of calls. This appendix provides a technical description of those call sets and their results for each level.

B.1 Ranking Metrics Description

This family of function returns an XML structure that describes the ranking metrics that can be applied at different levels, their description and the parameters needed for its calculation. The call for these functions is different depending on the level to which they operate.

B.1.1 Global Level

This function returns which are the metrics available, the string to call them, a description of the calculation that is performed and the parameters needed for their calculation. The call to this function is described in Table B.1. This information is codified in XML and returned in a String. The visual representation is available at B.1. An example of the response to this call can be seen in Listing B.1.

Listing B.1: Returned XML from globalGetRankMetricsDescriptions

Table B.1: Global Level Ranking Metrics Description

Method Name	globalGetRankMetricsDescriptions
Return Type	String
Parameters	None
Faults	GENERAL_METRICS_ERROR

```
<?xml version="1.0" encoding="UTF-8"?>
<RankingMetricsDescription>
 <RankingMetric>
  <metricName>Search Popularity</metricName>
  <metricId>spop</metricId>
  <metricDescription>
Calculate how much times an object has
been downloaded from a Search
</metricDescription>
 </RankingMetric>
 <RankingMetric>
  <metricName>Query Popularity</metricName>
  <metricId>qpop</metricId>
  <metricDescription>
Calculate how much times an object has
been downloaded from a Search that include the Query Terms
</metricDescription>
  <parameters>
   <parameter>
    <paramName>Query terms</paramName>
    <paramId>query</paramId>
    <description>
Query parameter to which we want to find
the most downloaded objects
</description>
    <type>String</type>
   </parameter>
  </parameters>
 </RankingMetric>
</RankingMetricsDescription>
```

B.1.2 Result List Level

Table B.2 describe the call to this function. This function returns which are the metrics available at the Result List level, the string to call them, a description of the calculation that is performed and the parameters needed for their calculation. The result information is codified in XML and returned in a String. The schema

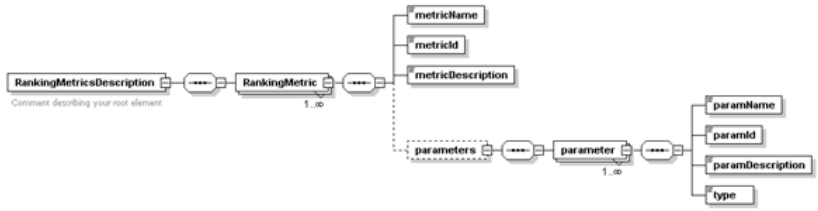


Figure B.1: Schema of the result of globalGetRankMetricsDescriptions

is the same as the one used for the globalGetRankMetricsDescriptions function.

Table B.2: Result List Level Ranking Metrics Description

Method Name	resultGetRankMetricsDescriptions
Return Type	String
Parameters	None
Faults	GENERAL_METRICS_ERROR

B.2 Calculate Ranking Metrics

This family of functions returns a list of objects sorted according to a specified metric. The calls for these functions are different depending on the level to which they operate.

B.2.1 Global Level

The call to this function is described in Table B.3. This function returns the top-k objects rank according to the specified metric (metricId). The parameters for the metrics can be found analyzing the result of the call to globalGetRankMetricsDescriptions function. A time period also can be specified according to Table B.4. An example call to this function can be seen in Listing B.2. The response to this call is an XML String that is format according to the schema described in Figure B.2. An example response is presented in Listing B.3.

Listing B.2: Call to globalGetRankingMetricValues

```
An example of a call to this function is:
String result =
 globalGetRankMetricValue(10,"qpop",["physics"],3)
```

Table B.3: Global Level Get Ranking Metric Value

Method Name	globalGetRankingMetricValues
Return Type	String
Parameters	Integer k String metricId String[] params Integer timePeriod
Faults	GENERAL_METRICS_ERROR INVALID_METRIC WRONG_PARAMETERS INVALID_TIME_PERIOD

Table B.4: Codes for Time Period

Value	Meaning
1	1 day
2	1 week
3	1 month
4	1 year
5	Since recorde hisoty

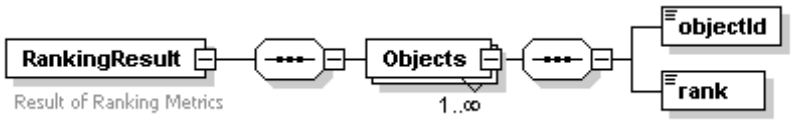


Figure B.2: Schema of the result of globalGetRankingMetricValues

Listing B.3: Result to call to globalGetRankingMetricValues

```
<?xml version="1.0" encoding="UTF-8"?>
<RankingResult>
 <Objects>
  <objectId>http://www.ariadne-eu.org/SW-124</objectId>
  <rank>0.91</rank>
 </Objects>
 <Objects>
  <objectId>http://www.ariadne-eu.org/LKP-452</objectId>
  <rank>0.90</rank>
 </Objects>
 <Objects>
  <objectId>http://www.lornet.org/12345</objectId>
```

```
  <rank>0.89</rank>
 </Objects>
 <Objects>
  <objectId>http://www.eun.org/finland/a33523</objectId>
  <rank>0.88</rank>
 </Objects>
 <Objects>
  <objectId>http://www.merlot.org/ML025</objectId>
  <rank>0.87</rank>
 </Objects>
</RankingResult>
```

B.2.2 Result List Level

Table B.5 presents the structure of the call to this funciton. This function returns an ordered list with the objectsId provided, ranked according to the specified metric (metricId). The parameters for the metrics can be found analyzing the result of the call to resultListGetRankingMetricsDescriptions function. An example call to this function can be seen in Listing B.4. The return information is codified in an XML and returned in a String. The schema is the same as the one used for the globalGetRankMetricsValues function.

Table B.5: Result List Level Get Ranking Metric Value

Method Name	resultGetRankingMetricValues
Return Type	String
Parameters	String[] objectsId String metricId String[] params
Faults	GENERAL_METRICS_ERROR INVALID_METRIC WRONG_PARAMETERS

Listing B.4: Call to resultGetRankingMetricValues

```
String [] objectsId =
  ["http://www.ariadne-eu.org/SW-124",
   "http://www.ariadne-eu.org/LKP-452",
   "http://www.lornet.org/12345"];
String result =
 resultGetRankMetricValue(objectsId,"spop",null)
```

Publications

1. Ochoa, X. Learning Object Repositories are Useful, but are they Usable? In Nuno Gimaraes, P.T.I., editor, *Proceedings of IADIS International Conference on Applied Computing*, pages 138-144, Algarve, Portugal, 2005.

2. Ochoa, X., Cardinaels, K., Meire, M. and Duval, E. Frameworks for the Automatic Indexation of Learning Management Systems Content into Learning Object Repositories. In Kommmers, P. and Richards, G., editors, *Proceedings of the ED-MEDIA 2005 World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005*, pages 1407-1414, AACE, 2005.

3. Ochoa, X. and Duval, E. Quality Metrics for Learning Object Metadata. In Pearson, E. and Bohman, P., editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006*, pages 1004-1011, AACE, 2006.

4. Ochoa, X., Ternier, S., Parra, G. and Duval, E. A Context-Aware Service Oriented Framework for Finding, Recommending and Inserting Learning Objects. In Nejdl, W. and Tochtermann, K., editors, *Innovative Approaches for Learning and Knowledge Sharing, European Conference on Technology Enhanced Learning, EC-TEL 2006*, LNCS 4227, pages 697-702, Springer Berlin / Heidelberg, 2006.

5. Ochoa, X, and Duval, E. Towards Automatic Evaluation of Learning Object Metadata Quality. In Embley, D. Olivie, A. and Ram, S., editors, *Advances in Conceptual Modeling - Theory and Practice, Advances in Conceptual Modeling - Theory and Practice: Proceedings of the 25th ER Conference*, LNCS 4231, pages 372-381, Springer Berlin / Heidelberg, 2006.

6. Ochoa, X and Duval, E. Use of contextualized attention metadata for ranking and recommending learning objects. In Duval, E., Najjar, J. and Wolpers, M., editors, *Proceedings of the 1st international workshop on Contextualized attention metadata*, pages 9-16, ACM Press, 2006.

7. Meire, M, Ochoa, X. and Duval, E. SAmgI: Automatic Metadata Generation v2.0. In Montgomerie, C and Seale, J., editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007*, pages 1995-1204, AACE, 2007.

8. Ochoa, X. and Duval E. Relevance Ranking of Learning Objects based on Usage and Contextual Information, In Alvarez, L., editor, *Proceedings of the Second Latin American Conference on Learning Objects*, pages 149-156, LACLO, 2007.

9. Ochoa, X and Duval, E. Relevance Ranking Metrics for Learning Objects. In Duval, E., Klamma, R. and Wolpers, M., editors,*Creating New Learning Experiences on a Global Scale, Creating New Learning Experiences on a Global Scale: Proceedings of the Second European Conference on Technology Enhanced Learning*, LNCS 4753, pages 262-276, Springer Berlin / Heidelberg, 2007.

10. Ochoa, X and Duval E. Quantitative Analysis of User-Generated Content on the Web. In De Roure, D and Hall, W., editors,*Proceedings of the First International Workshop on Understanding Web Evolution (WebEvolve2008)* pages 19-26, WSRI, 2008.

11. Ochoa, X and Duval, E. Quantitative Analysis of Learning Object Repositories. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008*, pages 6031-6048, AACE, 2008

12. Duval, E., Ternier, S., Wolpers, M., Najjar, J., Vandeputte, B., Verbert, K., Klerkx, J., Meire, M. and Ochoa, X. Open metadata for open educational resources in an open infrastructure. In McAndrew, P. and Watts, J., editors, *Proceedings of the OpenLearn2007 conference: researching open content in education*, pages 36-38, Open University, 2008.

13. Verbert, K., Ochoa, X. and Duval, E. The ALOCOM Framework: Towards Scalable Content Reuse. *Journal of Digital Information*, 9(26):24, 2008

14. Ochoa, X. and Duval, E. Relevance Ranking Metrics for Learning Objects, *IEEE Transactions on Learning Technologies*, 1(1):15, IEEE, 2008. In Print.