

Servicing the Federation: the Case for Metadata Harvesting

Fabio Simeoni

Centre for Digital Library Research (CDLR),
Strathclyde University
Livingstone Tower, 26 Richmond Street, Glasgow G1 1XH, UK
`fabio.simeoni@cis.strath.ac.uk`

Abstract. The paper presents a comparative analysis of data harvesting and distributed computing as complementary models of service delivery within large-scale federated digital libraries.

Informed by requirements of flexibility and scalability of federated services, the analysis focuses on the identification and assessment of model *invariants*. In particular, it abstracts over application domains, services, and protocol implementations.

The analytical evidence produced shows that the harvesting model offers stronger guarantees of satisfying the identified requirements. In addition, it suggests a first characterisation of services based on their suitability to either model and thus indicates how they could be integrated in the context of a single federated digital library.

1 Introduction

As digital libraries grow to accommodate more resources and users, their architectures embrace distribution and, in the process, discover the observables of the *federation*: a widely dispersed and loosely coupled system of cooperating but otherwise mutually autonomous parties.

1.1 Federated Digital Libraries

Federated digital libraries, or *FDLs*, are the subject of increasing development efforts across the globe: from subject-based and sector-based international initiatives – such as the Open Language Archive Community initiative [1] – to grand, cross-sectoral, and nationally-scoped initiatives which account for a large part of the current development and research efforts within the field – including the JISC’s Information Environment in the UK [2], the SURF’s Digital Academic Repository in Netherlands (DARE) [3], the ARIIC’s Information Infrastructure in Australia [4], the NSF’s National Digital Library for Science Education (NSDL) [5, 6] and the Networked Computer Science Technical Research Library (NCSTR) [7] in the US, and the Deutsche Initiative für Netzwerkinformation (DINI) [9] in Germany.

Admittedly, distribution is not a necessary implication of scope, and large-scale resource sharing may still rely on a centralised design. This is, for example, the approach adopted by the learning object community in UK for the in-progress development of the nation-wide JORUM repository [10]. Exceptions to the federated approach, however, are best interpreted as interim and exploratory solutions intended to mitigate the challenges of interoperability whilst fostering the formation of large communities of users. It is then anticipated that the cost of adequately serving such communities requires the organisational and technical support of a distributed infrastructure of local administrative domains.

In the absence of centralised content, the identity and *raison d'être* of a FDL lie exclusively in its service provision layer. It is through their services that FDLs hope to improve over the ubiquitously deployed and extremely popular services of another, globally distributed, and yet largely unmanaged federation, namely the World Wide Web. The goal is clear: by reflecting the needs and leveraging the means of comparatively smaller and more cohesive communities, FDLs set out to challenge the scope and accuracy of existing Web services, primarily search engines. The strategy is also clear: to build federated services against structured descriptions of resources, that is *metadata*, rather than the resources themselves. The underlying assumption – to date unqualified and largely untested – is that a structured approach will fare better than content-based or link-based analysis. Given the predominant implementation strategy, it is indeed suggestive to think of FDLs as ‘mini-webs’, more focused, homogeneous, and thus potentially functional subsets of the HTTP-based Web on top which they are conceptually and technically layered.

Service provision is also where FDLs meet most directly the challenges of interoperability. A federated service faces the heterogeneity of tools, policies, means, and largely purpose which derives from the foundational assumption of autonomy across participating parties. From a technical perspective, it must be able to accommodate significant variations in metadata syntax, semantics and exchange protocols. From an organisational perspective, it must also account for often dramatic variations in resource allocation, technical know-how, and local and community-wide agendas. Further, a federated service is expected to meet the qualitative requirements which its users normally associate with the provision of Web services, and to do so as the parties, resources, and users in the FDL scale up towards largely unknown bounds.

1.2 Distributed Computing and Data Harvesting

Informed by the core requirements of *flexibility* and *scalability*, this paper looks into technical models for the provision of federated services. To limit an otherwise prohibitive scope, it ignores issues related to metadata quality and metadata semantics and focuses instead on models of service delivery in the presence of distribution.

In its most generic form, the problem of service delivery is one of computing over widely distributed data and, as such, it admits either one of two comple-

mentary solutions. In *distributed computing*, the computation (i.e. the service) is distributed along with the remote data (i.e. the resource metadata), while in *data harvesting* the data is first gathered and then computed over locally.

Until recently, the distributed computing model has received most of the theoretical and practical attention, both within and outside the field. Its use for resource discovery, in particular, has been standardised and widely tested within the library community through, respectively, specifications and implementations of the Z39.50 protocol [11]. More modern, lightweight, and web-oriented interpretations of the model – most noticeably the SDLIP [14], SRW/SRU [13], and SQI [15] protocols – are also becoming increasingly popular.

At least in principle, the harvesting model is also familiar within the field. In diverse, domain-specific, and often implicit guises, it can be recognised as the approach underlying many physical union catalogues and all web-based search engines. First proposed and indeed named in the context of scalable architectures for Web-wide search services [21], harvesting can now count on an application-independent specification which has become the standard de-facto for a rapidly increasing number of implementations, namely the OAI-PMH protocol of the Open Archive Initiatives [16].

While both models are well represented in the field, early experimental evidence (e.g. [19],[20]) suggests that the harvesting model offers stronger guarantees to meet the service requirements of flexibility and scalability. The FDL initiatives mentioned in Section 1.1 vary substantially in terms of scope, architectural detail, and ultimately design philosophy; nonetheless, they have all chosen harvesting as the preferred model for the delivery of their services. One, the NCSTRL initiative, has recently undergone a phase of redeployment to replace its mechanisms for distributed computing with mechanisms for data harvesting [8].

1.3 Motivations and Outline

In the light of such extensive support, it is perhaps surprising that a high-level, comprehensive, and principled case for metadata harvesting within FDLs has not yet found, to the best of the author’s knowledge, a dedicated place in the literature. Granted, terse references to the ‘simplicity’ and sometimes ‘efficiency’ of metadata harvesting are nearly ubiquitous in publications related to the OAI-PMH protocol (e.g. [17],[18]). Similarly, complementary problems of ‘complexity’, ‘poor performance’, and ‘limited interoperability’ of Z39.50 have been repeatedly flagged some time before the advent of harvesting, most noticeably in relation to virtual and physical union catalogues (e.g. [22],[23]). Finally, some design considerations on the applicability of the two models, again predating the OAI specifications, may be discovered in service-specific consultancy reports (e.g. [24]).

Partly, the goal of this paper is to collect, expand, contextualise, and instantiate the arguments that have been produced so far. Even when the sparse

analytical evidence is collated, however, it is unclear whether the identified properties are accidents of specific protocol implementations and services, or whether they can be considered as *invariants* of the models underlying those protocols. Subsequently, it remains difficult to characterise the application domains and services which suit one model rather than the other and thus support decision makers in their choice of service delivery protocols.

In an attempt to fill this gap, the paper presents a comparative analysis of the two models which is independent of the application domains in which they are used, the services which adopt them, and the protocols which implement them. In particular, the paper seeks answers to questions like ‘if the OAI-PMH is to be preferred over Z39.50 for a given federated service, is it also more indicated than SRW/SRU for the same service?’ and ‘what services are better accommodated by the OAI-PMH and which ones suit instead a Z39.50-based or a SRW/SRU-based approach?’, and again ‘how can the two models coexist in the context of a single FDL?’.

Presented in Section 2, the analysis is carried out in five steps. Section 2.1 contextualises the general requirement of flexibility to the case of service delivery and shows how it can be approximated by the simplicity of delivery models. Section 2.2 discusses the degrees of complexity of the two models under examinations, Section 2.3 illustrates the manifestations of such complexity within an FDL, and Section 2.4 outlines the potential for scalability associated with the models. Section 2.5 considers their limitations in terms of functionality and how these limitations may inform a characterisation of services in relation to their suitability to either model. Finally, Section 3 draws some conclusions and relates service delivery models to other aspects of interoperability.

2 Analysis

One way of capturing the complementary nature of data harvesting and distributed computing is by noticing that while the former localises service provision within the FDL, the latter spreads it across all the federated parties. This Section shows how this simple observation bears profound consequences in terms of both flexibility and scalability of federated services.

2.1 Flexibility as Simplicity

In any deployment scenario, the technical and organisational costs associated with the complexity of a given solution – whether a metadata model, a service delivery model, or a service delivery protocol – must be carefully measured against the gain in functionality that justify them and the heterogeneity of the community that must absorb them [28]. The price of misjudgements is a partitioning of the community intended for that solution.

In principle, any given degree of complexity identifies a sub-community of the initially intended one, potentially excluding: (i) members who cannot sustain the

solution or do not want to in response to functionality deemed unnecessary (*the solution is too complex*), or (ii) members who desired and could have sustained a higher degree of functionality (*the solution is too simple*). If the *community of adoption* does not have or assume significance with respect to the one initially targeted, the solution fails and tends to be progressively abandoned. At best, the solution is re-purposed within a narrower scope, and the problem for which it was originally conceived remains an open one. This is, for example, the case of the Dublin Core metadata model, which was originally intended for resource description over the unmanaged Web and it is now re-purposed within more disciplined FDLs.

Undoubtedly, the diversity of organisational structures remains a primary observable within FDLs and thus the simplicity of solutions intended for FDLs is to be treasured above the functionality they can offer. When it comes to service delivery, in particular, the simplicity of a model translates into a measure of its flexibility. Simply put, a flexible model for the delivery of federated service should present a ‘low barrier’ to the interoperability of federated parties.

2.2 The Causes of Complexity

Notice now that distributed computing requires that *each* federated party participate of the implementation, deployment, and maintenance of *all* the federated services its metadata contributes to. In contrast, harvesting requires only that federated parties be able to disclose the metadata they hold, a task which is in general much simpler than service provision and, most importantly, one which offer more *resilience* across different federated services.

Consider, for example, a federated service for resource discovery. In a distributed computing interpretation of the service, federated parties must be able, at the very least, to parse, translate, and execute all the queries submitted to the service by its users. In addition, the service requires that the parties return query results in a format the service is willing to accept, and thus that parties be potentially engaged in data transformation tasks. Depending on the service functionality, the service may also require that parties perform additional functions, such as management of the result set (e.g. filtering, ordering, browsing, providing statistics, etc).

Different are the demands parties must satisfy with a harvesting interpretation of the same service. Besides the potential data transformation tasks which are necessary in any data exchange scenario, federated parties are required at most to recognise and execute a small and fixed number of simple queries to scope the disclosure of their metadata. In particular, they are *not* expected to parse and interpret the expression of a full-fledged and potentially complex query language.

The simplicity of disclosure over full service provision should not be considered in the limited context of single service, as it is normally done. Rather, it should be viewed in the common assumption that federated parties will contribute to more than one service within the FDL, where different services may

offer: (i) different functions (e.g. resource discovery, citation linking, metadata enhancement, current awareness, etc.), or (ii) specialise similar functions to the needs of different sub-communities within a single FDL (e.g. cross-community resource discovery versus learning object or eprints discovery), or (iii) simply compete on the basis of additional added value services (e.g. user interfaces, service customisation, etc).

In a ‘multi-service’ scenario, the additional complexity of the distributed computing approach leaves more room for variations across services and thus place higher costs on the ‘mobility’ of federated parties across different services. When moving across different resource discovery services, for example, a federated party may need to process different query languages and perform different result management functions as well as carry out different data transformation tasks. In contrast, only the costs associated with the latter may be faced by a federated party which simply discloses its metadata. For example, a party that discloses simple Dublin Core metadata for resource discovery will face no additional costs when ‘moving’ to another DC-based discovery service and in fact to any other service which relies on the same metadata format. Even when the party does have to translate its own metadata into other formats than DC (e.g. IEEE LOM), the availability of a FDL-wide syntactic interoperability solution – normally one based on the XML standard – implies that the costs are incremental rather than *ex novo*.

2.3 The Costs of Complexity

Once the complexity of the distributed model has been ascertained, one may consider the effects of that complexity within the FDL. Obviously, complexity raises implementation costs and thus tends to limit the number of available implementations to those produced by resourceful parties and commercial vendors. Even when free implementations are made available, the tight coupling between the functions of any delivery model and the metadata back-end of individual parties makes off-the-shelf reuse an elusive goal and does not eliminate the need of installation, customisation, and maintenance tasks.

Another way in which complexity undermines interoperability is by increasing the possibility of incomplete or erroneous specifications whilst reducing their understandability. In particular, complex protocol specifications are prone to unstable releases, problems of backward compatibility, and mutually inconsistent implementations.

Most importantly, complexity amplifies almost invariably problems of *semantic interoperability* within the model [25]. Full service provision, in particular, multiplies the requirements of semantic alignment between federated parties and thus is more prone to breaking interoperability through inconsistent implementations of the model. For example, the lack of interoperability between z39.50 targets caused by differences in mappings of search attributes onto local database indexes, extraction and normalisation algorithms for search keys, and stopwords handling is well documented in the literature (e.g.[22]).

To avoid the problem, services may make a degenerate, almost ‘syntactic’ use of the model [12], which is suitable only for high-level meta-services not oriented to the end-user (e.g. server implementation browsing). Alternatively, they may restrict their scope to all the federated parties which comply with some community-specific instantiation of the model. Instantiations may concern the query language, the format of the metadata, or the support for optional functionality, and may be approached in a number of ways, including profiling [29] and MOP-based expansion and refinement [14]. Normal practice is then to mandate support for a minimal instantiation to support the implementation of federated services against the greatest common denominator of the implementations deployed at the federated parties (e.g. [26],[33],[27]).

Clearly, the harvesting model is not immune to the interference of semantics with service deliver and thus does not obviate the need for a ‘spectrum of interoperability’ solutions within the FDL [6]. By limiting such interference to a profiling of metadata formats (e.g. [?],[?]), however, harvesting simplifies the organisational aspects of the profiling process whilst maximising the scope of the community which adopts the profile within the FDL.

2.4 Scalability

Section 2.2 and Section 2.3 have shown that an approach to interoperability based on the harvesting model promises to contain service deployment costs within FDLs. The model, however, is also beneficial for service implementation, for it delivers all the good properties which are normally associated with local computations.

With harvesting, in particular, the diverse capabilities of federated parties and the observables of the network may be factored out real-time interactions with the end-users and be faced instead off-line, possibly through flexibly configurable processes [16]. Latency-inducing factors associated with slow, congested, or simply unavailable connections have virtually no impact on the *reliability* and *responsiveness* with which a service interfaces its users.

In contrast, a service distributed across the FDL is intimately dependent on the federated parties and the underlying network, and thus tends to be constrained by the performance of the ‘weakest’ party and the fluctuations of the available bandwidth. The fact that parties and network are in principle required to sustain the full service load (e.g. all the user queries submitted to a discovery service) cannot but worsen the situation. Experimental evidence indicates that the performance of basic implementations of distributed discovery services tend to rapidly decrease as the number of participating parties grows beyond 10-15 [30].

Admittedly, manual or automated clustering techniques [31], proxy-based solutions [14], and replication strategies [7] may help to more equally distribute the service load across the FDL. However, the pragmatic and intellectual costs of scaling these approaches against the number and capabilities of participating parties are largely unclear but promise to raise significantly the overall costs of

the FDL infrastructure. Significantly, advanced implementations of distributed services have been so far confined to the prototypal domain.

If reliability is largely related to distribution, performance may be also influenced by network-independent requirements. With distributed computing, the costs of pre-processing the metadata received from participating parties before presenting it to the users must also be accommodated in real-time. The resulting penalties discourage or severely limit the possibilities of metadata translation, de-duplication, versioning, and enhancement which are so important in the diverse environment of the FDL. It is hard, for example, to imagine distributed services with consolidation capabilities which go beyond straightforward identifier-based de-duplication [22]. Again, the harvesting model allows to hide these inherently difficult and computationally intensive processes away from the users and, by doing so, paves the way for a family of middleware services which remain instead elusive in the distributed computing scenario.

Of course, the harvesting model raises its own scalability issues. A federated service based on harvesting operates on a centralised copy of the remotely distributed metadata and thus may rapidly become large in response to the number and growth of participating collections. However, the costs associated with local scalability are relatively lower when compared with those raised by network-based solutions. Equally important, the technical processes required for local scalability are well understood and require opportunistic intervention on variables which are entirely under the control of service implementors (e.g. memory, disks, processors, local networks) [22].

Clearly, more experience is needed to identify the limits of the harvesting approach beyond the positive results of early experimental services [19]. However, it may be argued that no realistic degree of scalability can be predicated on soaring costs. In this sense, the very existence of comprehensive and long-established physical union catalogues (e.g. [24], [32]) and Web search engines suggests that, whatever may be the precise limits of harvesting, these may be approached at relatively contained costs.

2.5 Functionality

In the light of the principles presented in Section 2.1 and the advantages attributed to harvesting over distributed computing in Section 2.2 and Section 2.3, it is interesting to observe that neither model enables more functionality than the other within the FDL.

At first, this statement may appear controversial for – by exposing the functionality of specific services – even the most streamlined server-side implementations of the distributed computing model (e.g. SRW/SRU) are more expressive than any server-side implementation of the harvesting model. Indeed, the advantages associated with the simplicity of harvesting are ultimately predicated on this argument. However, from a broader, service-oriented perspective – and thus from a client-side perspective – the situation is quite different.

In a strict computational sense, the harvesting model enables more expressive federated services than are possible under the distributed computing model. The reasons for this are largely those discussed in Section 2.4, and relate to the limited possibilities of metadata pre-processing which are allowed under the distributed computing model. Not only does this apply to processes which are theoretically possible but pragmatically unfeasible under that model (e.g. format translation). It also applies to processes – such as advanced consolidation and standard ranking algorithms [22]– which require the totality of the remotely distributed metadata and cannot rely solely on the responses of participating parties to individual service transactions (e.g. queries). Put another way, not all computations can be distributed across the disjoint union of participating metadata collections.

The harvesting model, on the other hand, relies on a mono-directional information flow from data providers to service providers and is thus bound to the subclass of service architectures which can be gracefully accommodated within this assumption. Whenever the intended functionality requires information to flow in the opposite direction or in both directions – and thus relies on a different distribution of roles between communicating parties – harvesting loses much of its appeal.

One case which defeats the harvesting approach is when data exhibits an extremely dynamic nature. As an example in the classic library domain, consider the needs of union catalogues which wish to offer circulation data along with bibliographic data. Here, harvesting is not an effective solution for the harvesting rates required by the dynamicity of circulation data would prove so intensive to essentially reintroduce the network as a real-time observable of service provision.

Similarly, harvesting has little to offer for the implementation of a local interface to a remote service (e.g. a local Z39.50 interface to an existing discovery service), even if the latter had facilities in place to offer its data for third-party harvesting. Here, the local interface is best viewed as an extension of the remote service and no clear distinction between data and service provision can be made. In particular, local harvesting of the remote data would simply reintroduce deployment costs which have been already absorbed within the FDL. In contrast, a two-party dialog is an ideal and indeed prototypical application scenario for the distributed computing model [12] and one in which the problems of inter-party interoperability discussed in Section 2.2 simply do not arise.

There are services, accordingly, which are – in any practical sense – outside the scope of the harvesting model and yet may play an important role within the FDL. It should be noted, however, that such services rely on strong agreements between communicating parties which can only be expected within tightly-coupled subsets of the FDL. Put another way, these services operate *within* the FDL but do not belong to the category of truly federated services.

3 Conclusions

Data harvesting and distributed computing may both serve as models of service delivery in the context of large-scale federated digital libraries.

Harvesting clearly separates the concerns and responsibilities of data providers from those of service providers, while distributed computing views data provision and service provision as inherently overlapping processes. In particular, harvesting induces a 2-phase view of service delivery which distinguishes the aspects related to communication – which involve both service and data providers – from those that relate to service-specific implementation – which instead concern only service providers. In contrast, distributed computing collapses communication and service-specific implementation within a single protocol of interaction.

That communication between service and data providers may take place in conceptual isolation from service-specific implementation is beneficial to data providers, for it shifts the costs of their participation where they are expected to be affordable, at the service providers. Vice versa, service implementation benefits from abstracting over communication, for it can deliver all the good properties normally associated with local and off-line computations.

For these reasons, the harvesting model offers stronger guarantees to meet requirements of flexibility and scalability of federated services. In contrast, the distributed computing model offers complementary support for services that operate within more cohesive subsets of the federated library.

To conclude, it is worth noticing that the harvesting model offers little help with semantic issues of metadata interoperability: successfully exchanged metadata must still be uniformly understood. In particular, the model alone cannot guarantee a uniform implementation of federated services against metadata modelled according to different models, formats, profiles, and standards. The model abstracts over the complexity of the metadata which may be harvested within sub-communities of the federated library and thus reflects a bipartite conceptual model which helps to more clearly separate, and thus tackle, different pieces of the interoperability jigsaw.

References

1. Simmons, G., Bird, S.: The Open Language Archives Community: An Infrastructure for Distributed Archiving of Language Resources. *Literary and Linguistic Computing*, 18(2) 117–128 (2003)
2. Joint Information Systems Committee (JISC): Investing in the Future: Developing an Online Information Environment. <http://www.jisc.ac.uk/ie> (2004)
3. Stichting SURF: DARE: Specifications for a Networked Repository for Dutch University. <http://www.surf.nl/download/ReportSpecs3.0.pdf> (2003)
4. Australian Government, Department of Education, Science, and Training: Information Infrastructure Outcomes of <http://www.dest.gov.au/highered/research/outcomes2003.htm> (2003)

5. Lagoze, C., Hoehn, W., Arms, W., Allan, J. et al.: Core Services in the Architecture of the National Digital Library for Science Education (NDSL). Cornell University, Ithaca, arXiv Report, cs.DL/0201025, <http://arxiv.org/abs/cs.DL/0201025> (2002)
6. Arms, W., Hillmann, D., Lagoze, C. et al.: A Spectrum of Interoperability: The Site for Science Prototype for the NDSL. *D-Lib Magazine*, 8(2) (2002)
7. Davis, J.R., Lagoze, C.: NCSTRL: Design and Deployment of a Globally Distributed Digital Library. *Journal of the American Society for Information Science (JASIS)* 51(3) 273–280 (2000)
8. Anan, H., Liu, X., Maly, K. et al.: Preservation and Transition of NCSTRL Using an OAI-Based Architecture. *Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries* 181–182 (2002)
9. Deutsche Initiative für Netzwerkinformation (DINI). <http://www.dini.de>. (2003)
10. The JISC Learning Materials Repository Service: JORUM Scoping and Technical Appraisal Study. <http://www.jorum.ac.uk> (2003)
11. Z39.50 Maintenance Agency: Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. <http://www.niso.org/standards/resources/Z39-50-2003.pdf> (2003)
12. Lynch, C.: The Z39.50 Information Retrieval Standard: Part I: A Strategic View of Its Past, Present, and Future. *D-Lib Magazine* (1997)
13. Z39.50 International: Next Generation: SRW - Search/Retrieval Web Service. <http://www.loc.gov/z3950/agency/zing/srw/spec-index.html> (2003)
14. Paepcke, A., Brandriff, R., Janee, G. et al.: Search Middleware and the Simple Digital Library Interoperability Protocol. *D-Lib Magazine*, 6(3) (2000)
15. CEN/ISSS: The Simple Query Interface (SQI) specification. <http://nm.wu-wien.ac.at/e-learning/inter/sqi/sqi.pdf> (2003)
16. The Open Archives Initiative: The Open Archives Initiative Protocol for Metadata Harvesting (2.0). <http://www.openarchives.org/OAI/openarchivesprotocol.html> (2003)
17. Lagoze, C., Van de Sompel, H.: The Open Archives Initiative: Building a low-barrier interoperability framework. *Proceedings of the First ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'01)* (2001)
18. Lynch, C.: Metadata Harvesting and the Open Archives Initiative. *Bi-monthly Report of the Association of Research Libraries (ARL)* 217 (2001)
19. Van de Sompel, H., Krichel, T., Nelson, M.J.: The UPS Prototype: An Experimental End-User Service across E-Print Archives *D-Lib Magazine*, 6(2) (2000)
20. Halbert, M., Kaczmarek, J., Hagedorn, K.: Finding from the Mellon Metadata Harvesting Initiative. *Proceedings of the 7th European Conference on Digital Libraries (ECDL)*, (2003)
21. Bowman, C.M., Danzig, P.B., Hardy, D.R. et al.: Harvest: A Scalable, Customizable, Discovery and Access System). Technical Report TR CU-CS-732-94, Department of Computer Science, University of Colorado-Boulder (1994)
22. Lynch, C.: Building the Infrastructure of Resource Sharing: Union Catalogues, Distributed Search, and Cross-Database Linkage. *Library Trends*, 45(3) 448-461 (1997)
23. Husby, O.: Real and Virtual Union Catalogue. *6th International Seminar of the Czech and Slovak Library Information Network (CASLIN)*, <http://www.caslin.cz:7777/caslin99/a2.htm> (1999)
24. Crossnet Systems Ltd.: SCURL Feasibility Study To Investigate Potential Applications Strategic Implications of Z39.50 Technology on the COPAC Service. <http://www.curl.ac.uk/projects/z3950.pdf> (1998)
25. Paepcke, A. et al.: Interoperability for Digital Libraries Worldwide. *Communications of the ACM*, 41(4) 33–42 (1998)

26. Z39.50 International: Next Generation: CQL - Common Query Language (1.1). <http://www.loc.gov/z3950/agency/zing/cql/>, (2004)
27. Nilsson, M., Sibersky, W.: RDF Query Exchange Language (QEL) - Concepts, Semantics, and RDF Syntax. (2003)
28. Arms, W.: Digital Libraries. *Digital Libraries and Electronic Publishing*. Cambridge, Ma.:MIT Press (2000)
29. Z39.50 Maintenance Agency: Z39.50 Profiles. <http://www.loc.gov/z3950/agency/profiles/profiles.html> (2004)
30. Davenby, J.: Aiming at quality and coverage combined: blending physical and virtual union catalogues. *Online Information Review*, 26(5) 326–334 (2002)
31. Nicholson, D.: Clumping towards a UK National Catalogue? *Ariadne*, 22 (1999)
32. Online Computer Library Centre (OCLC), <http://www.oclc.org/> (2004)
33. Reddy, S., Lowry, D., Reddy, S. et al.: DAV Searching and Locating, Internet Draft. *IETF* (1999)