



Resource List Management Systems

Analysis of Talislist, ReadingList Direct, and Bookworm*

Nicole Harris

Table of Contents:

1. Introduction	2
2. Product Overview	3
2.1 Talislist <www.talis.com/products/lms/talislist.htm>	3
2.2 ReadingListDirect <www.readinglistdirect.co.uk>	3
2.3 Bookworm <bookworm.lboro.ac.uk/>	4
3. Functionality Analysis.....	5
3.1 Evaluation Criteria	5
3.2 Analysis against DELIVER requirements	6
3.3 Summary Analysis	12
4. Technical Requirements and Information	13
4.1 TalisList	13
4.2 Reading List Direct	13
4.3 Bookworm.....	13
5. Financial Information.....	14
5.1 Talislist.....	14
5.2 ReadingListDirect.....	14
5.3 Bookworm.....	14
6. Summary Comments	16
6.1 Talislist Comments	16
6.2 ReadingListDirect Comments.....	18
6.3 Bookworm Comments	20
7. Recommendations.....	21

* Please note that the Loughborough System is referred to as 'Bookworm' for ease of reference. Bookworm is the name of the server that the Loughborough Reading List System is managed on at Loughborough. As of 06 May 2003, the team have been informed that Loughborough will use the acronym LORLS (Loughborough Online Reading List System) to refer to the system.



1. Introduction

As part of the second phase of the DELIVER project, the team has undertaken an evaluation of three distinctive resource list management software packages to investigate potential solutions to requirements identified in the first phase of the project (User Needs Analysis). Full documentation of the DELIVER UNA and the recommendations made are available on the project web site <<http://www.angel.ac.uk/DELIVER/>>.

Although this evaluation uses criteria identified by the partner institutions of the DELIVER project, the general overview and introduction to the software packages will be useful to a wider audience interested in purchasing resource list solutions. In selecting packages to evaluate, the team attempted to identify three distinctive approaches to the issues surrounding resource list management:

1. Talislist – a solution offered by a company that has significant experience in the management of library resources.
2. ReadingListDirect – an independent company offering different approaches to both hosting and pricing of resource lists.
3. Bookworm – an open source development from Loughborough University.

The limited timescale available to the project did not allow for a more in-depth look at all the possible resource list management solutions, but it was felt that these three options offered a fair representation of the different approaches available.

A fourth option remains open to the team in the form of in-house development of a resource list management system if a suitable solution cannot be identified. The general consensus of the project team and the partner institutions at the end of the first phase of the project was that development time could be better used in addressing 'join-up' requirements between any system purchased and existing institutional systems in the form of middleware development. These middleware developments can then be made openly available to the JISC community for reuse.

The two partner institutions in the DELIVER project are of different size, structure and focus. As with any procurement decision, it is possible that different options may offer the best solutions to the institutions. Selection of different systems may, indeed, support the project technical development as the middleware solutions can then be tested against different systems.



2. Product Overview

2.1 Talislist <www.talis.com/products/lms/talislist.htm>

Talis Information Ltd supplies Talislist. It sells software services and solutions to libraries in the UK and Ireland. Talis is the supplier of the current Library Management System software at DMU (and is an associate partner in the DELIVER Project). Talis Information Ltd evolved from BLCMP Library Services Ltd, which was first established in 1969 as a joint venture between libraries in Birmingham. The Birmingham Libraries Co-operative Mechanisation Project was created with government assistance to realise the benefits of a shared approach to library computing. Talislist is a bespoke resource tool that uses the Web and Java programming to enable libraries to deliver resources for teaching and learning. It provides categorised access to both print and electronic materials from a variety of sources both inside and outside the library. Talislist is designed to work optimally in conjunction with Talis library systems. It can also work with other library management systems using non-proprietary interfaces and protocols, but assistance from the supplier is required for configuration and installation.

The key features of the system are:

- ◆ The list authoring tools, which allow users to both cut and paste and grab references from a wide variety of sources. The tools also allow the lists to be structured and displayed in a variety of ways.
- ◆ The student side of the list that permits different views to be seen. It is also possible to link directly to electronic resources and the library catalogue, which means that they can see references exactly as they appear in the OPAC.
- ◆ The workflow monitoring tools ensure that resources are checked by the library before they are published to the university community and to restrict editing of lists to those with responsibility for resources.
- ◆ The system can track which courses do not have any resources associated with them and so prompt the writing of “missing” lists.

2.2 ReadingListDirect <www.readinglistdirect.co.uk>.

ReadingListDirect, from Sentient, centrally hosts resource lists on behalf of institutions. The product is described as having four components:

1. The reading list component. This is the main tool, which allows academics to create resource lists by searching a variety of targets.
2. The resource component. This allows students to decide how best to obtain a resource by allowing them to view availability in their own library, other consortium libraries, or to view purchase options.
3. The library component. This allows the library to view aggregated data concerning the resources selected, and to estimate demand by seeing how many courses recommend the book to how many students.
4. The publication component. This is a subscription service that alerts academics to books of interest in their field, and allows them to request appraisal copies and comment on the quality of resources.

The system is LMS and VLE independent, and supports OpenURL, Z39.50 and Athens. The business model for the ReadingListDirect service depends heavily on on-screen advertising to users, and 'click-through' revenues from book purchases made by students. This has the advantage of reducing direct subscription costs paid by an institution for the service, but also a corresponding opportunity cost, in reducing the potential for the institution itself to collect such revenues. This may also raise ethical issues for some institutions.

2.3 Bookworm <bookworm.lboro.ac.uk/>

The Loughborough University Reading List System (Bookworm) began as an in-house project for the University to address requirements identified by the Loughborough University Teaching and Learning Committee. The original specification asked that resource lists be made available online, linked to module specifications.

The main cited aims of the system are to:

- ◆ Enable taught course students to view resource lists (including lecturer's annotations) from a variety of sources e.g. Module specifications, lecturer's web pages, Library catalogue (OPAC), etc.
- ◆ Enable users to check availability of material on resource lists against the Library catalogue.
- ◆ Avoid duplication of effort by academic departments and the Library in maintaining separate resource lists.

Bookworm is an open source system, using Linux, perl, MySQL and Apache. It is publically available under the General Public License at: <<http://bookworm.lboro.ac.uk/distribution.html>>. The system is still maintained by the team at Loughborough, with version 4 currently in progress. Planned enhancements are listed on the system website <<http://bookworm.lboro.ac.uk/readinglists-todo.html>>.

In the first year of its operation at Loughborough University, temporary staff were employed to key in the 857 resource lists submitted by academic departments into the system. Thereafter responsibility for updating the resource lists on the system was transferred to the academic departments. Currently the system holds some 1,700 active resource lists (including sub-lists) at Loughborough.

Please note that the Loughborough System is referred to as 'Bookworm' for ease of reference. Bookworm is the name of the server that the Loughborough Reading List System is managed on at Loughborough. As of 06 May 2003, the team have been informed that Loughborough will use the acronym LORLS (Loughborough Online Reading List System) to refer to the system.



3. Functionality Analysis

3.1 Evaluation Criteria

The criteria used to evaluate the three software options in this report are derived from the recommendations resulting from the User Needs Analysis carried out on behalf of the DELIVER project. Various groups of institutional stakeholders were approached as part of this study and asked to give opinions and insights into current use of Virtual Learning Environments and library resources. Particular focus was given to the current use and provision of resource lists by all stakeholders. The UNA developed the following four lists of recommendations.

1. Recommendations for resource list management.
2. Recommendations for developing institutional Virtual Learning Environments (VLEs).
3. Recommendations for using ANGEL solutions within the DELIVER project.
4. Recommendations for individual institutions (cultural change).

The recommendations identified by users for resource list use and provision were then utilised to analyse three distinctive software packages that offered resource list solutions. The evaluation process consisted of live demonstrations to the project team, individual research and direct communication with each of the software providers.

The systems are analysed against each of the 21 recommendations that had been identified by stakeholders and developed by the project team into firm requirements. Each criteria is rated using the following scale:

- ◆ Yes - it demonstrably provides the function;
- ◆ No - it does not provide the function;
- ◆ Potential – it may provide some of the function, particularly with further effort by the team.

Additional comments have been collated from all of the team members and other institutional representatives present at the demonstrations.

3.2 Analysis against DELIVER requirements

1. Define resource list metadata for resource lists to carry all DELIVER functions.

It would be expected that the system would use metadata relevant to resource lists, such as Dublin Core or Z39.50. The project notes that there is no existing definition of resource list metadata, although other projects in the programme are pursuing this concept.

a. Define sufficient information for ordering.

Talislist	Yes.
ReadingListDirect	Yes.
Bookworm	Yes.

b. Allow for ordering parts of books or journals.

Talislist	Yes – as long as it is included in a target repository.
ReadingListDirect	Yes.
Bookworm	Yes, section B of the 'add a title' form allows parts of items to be ordered.

c. Identify preferred ranking system (e.g. background, essential, purchase etc) and link to ordering information.

Talislist	Yes.
ReadingListDirect	Yes.
Bookworm	Yes, but only 'key text' and 'recommended for purchase' options.

d. To allow ordering of items the system needs to know the number of students reliant on it.

Talislist	No. Note that at DMU there is no exact formula for number of books bought per student. There is an estimate that changes on a number of factors.
ReadingListDirect	Yes, RLD is very strong on this point, and provides aggregated numbers of students recommended texts across courses.
Bookworm	Potential. Academic can suggest to the library how many texts they desire. Not an ideal interpretation.

e. Allow for inclusion of abstracts in display.

Talislist	Yes.
ReadingListDirect	Potential. Academics can add notes.
Bookworm	Potential. Academics can add notes, but abstracts are not automatically included in the citation.

2. Present information 'normalised' to defined display format.

a. Unpack abbreviations.

Talislist	No. The information is displayed as it is written or harvested, there is no automated tool that converts abbreviations into full text.
ReadingListDirect	Yes. All references are fully cited.
Bookworm	Potential. Journals pulled from the library catalogue will have the full and correct citation. Journal entries added by 'hand' by the academics would run the risk of having abbreviated entries.

b. Use established citation methods.

Talislist	Yes. Uses Harvard citation.
ReadingListDirect	Yes. Offers a variety of different citation methods. RLD is currently looking at using Endnote for both citations and import.
Bookworm	Yes.

3. Highlight new order requirements for library staff.

One of the most time consuming tasks that staff do is checking to see if material is already available in the library. As there is no way of knowing how and where a lecturer may have changed a list, this process must be repeated every year. It is important that the new system highlights new materials, and also repeats requirements for existing stock to cut this process.

Talislist	No. Talis confirmed that this function was not available. There are no alerts to changes to the list. It is possible to list courses that do not have resource lists.
ReadingListDirect	Yes. The library component offers this functionality.
Bookworm	Yes. The system automatically pulls entries into a database provided by the system. This may be restrictive to institutions that already have their own database for handling resource list orders.

4. Allow students to comment on resources.

This facility will allow students to pass comments both on the suitability of resources, and on resource availability and any other problems with resource use. This may be more suitably dealt with by the VLE.

Talislist	No. There is no facility for feedback or comments.
ReadingListDirect	No.
Bookworm	No.

5. The system should allow for importing of existing electronic resource lists at institutions.

At the partner institutions, some resource lists are already held in electronic format. The ability to import these into the system would provide an excellent starting point for introducing academics to the system ('your list is already available, please update and format').

Talislist	Yes. This is possible but is dependent on the data source. An XML format has been specified for import of existing resource lists. Existing lists would need to be converted to this format for import.
ReadingListDirect	No.

Bookworm	No. Loughborough University hired individuals to hand-enter all the lists.
----------	--

6. The system should allow items and lists to be hidden from student view when not currently in use.

Not all courses are run from year to year. Academics may also choose to create slightly different lists, without wishing to discard an old list. These lists should be 'hidden' from student view, and re-activated when required.

Talislist	Yes. It is possible to hide resources by using timed access controls. Authors of lists (academics), and owners of resources (librarians), can control access to resources. Owners can edit and change lists, but not usually add lists. Permissions information is included in the summary comments.
ReadingListDirect	Yes. All lists have 'timed' availability.
Bookworm	Yes. Lists can be either published or not-published.

7. Allow for importing of Management Information System [QLS at DMU] data.
- a. New courses (create new list).

Talislist	Yes. This could be channelled through the VLE or read as XML data. Talis have provided an XML mapping document.
ReadingListDirect	No, although willing to investigate.
Bookworm	No.

- b. Student numbers, course codes, course titles, academic responsible.

Talislist	Yes. As above.
ReadingListDirect	No. As above.
Bookworm	No.

8. All modules should have a resource list on the system.

Modules that do not produce a resource list should be registered as such on the system. This will allow library staff to have a clear understanding of how many course resource lists are missing, and how many do not produce a list.

Talislist	Yes. Lists with no resources are 'flagged'.
ReadingListDirect	Potential. Although likely that librarians would need to take responsibility for creating new lists.
Bookworm	Potential. Although likely that librarians would need to take responsibility for creating new lists.

9. The system should allow for directing students to book purchase options, either with Amazon or the local bookshop.

Talislist	Yes. An internet bookshop can be included.
ReadingListDirect	Yes.
Bookworm	Potential. Will flag books as 'for purchase' but does not currently link. Potential for in-house development of links, which would allow institutions

	to benefit from click-through revenue.
--	--

10. The system should support a variety of interface routes into the system, from VLEs and course web pages.

Talislist	Yes. URLs can be included in VLE / web interfaces at course or module list level.
ReadingListDirect	Yes. Demonstrations are available from the company.
Bookworm	Yes.

11. The system should allow for linking to journals at the article level, and support for maintaining these links as static URLs.

Talislist	Potential. The system will take users to the cover or content pages of an e-journal. There is a problem with sites that require users to log-in, which would be a problem that ANGEL may be able to address.
ReadingListDirect	Potential. The system does allow for parts of books and journals to be cited, but does not automatically link. This functionality would have to be progressed through use of ANGEL software.
Bookworm	Potential. The system does allow for parts of books and journals to be cited, but does not automatically link. This functionality would have to be progressed through use of ANGEL software.

12. The system should allow resource lists to be divided into sub-lists.

This would typically support division into semesters or weeks, but should allow academics to define their own headings for sub-lists. This will allow for lists to be divided by subject headings, or material applicability.

Talislist	Yes. The list can be sub-divided in several ways. The lists can have names determined by the users.
ReadingListDirect	Yes. The list can be sub-divided in several ways. The lists can have names determined by the users.
Bookworm	Yes. This is achieved through use of sub-headings and re-ordering list entries.

13. The system should include an alerting function that highlights updates for academic staff, library staff and students.

Talislist	No. There are no alerting functions. Changes can be seen by searching the system and through use of flags.
ReadingListDirect	Yes. The system can alert academics when their list is due to expire.
Bookworm	No.

14. Resource information should be drawn from a variety of targets including: exam paper, e-course pack, electronic resource databases and catalogue information.

The system should also allow for searching of other catalogues through agreements such as the M25 or M1/M69 consortium.

Talislist	Yes. Requires installation of harvesting tools.
-----------	---

ReadingListDirect	Yes. RLD has well defined searching capabilities for alternative library catalogues.
Bookworm	No. This system currently only pulls information automatically from the library catalogue. It does allow academics to cite books not in the catalogue.

15. The system should make academics aware of all available resources.

This is most likely to be as drop-down menus that allow academics to select resource type.

Talislist	Yes. A range of targets can be added to the 'shopping' menu. Each must be searched separately.
ReadingListDirect	No.
Bookworm	No.

16. The system should allow for use of concepts already established in institutions through both student display and resource discovery interfaces.

a. Access icons, material format icons.

Talislist	No.
ReadingListDirect	No.
Bookworm	No.

b. Institutional branding.

Talislist	Yes.
ReadingListDirect	No.
Bookworm	Yes.

17. The system should allow for authorisations to be given to certain groups of users.

This will prevent list creators from adding restricted access / cost associated resources to resource lists. This could include e-course packs, off prints, restricted databases etc.

Talislist	Yes.
ReadingListDirect	Yes.
Bookworm	Yes. The system can define a range of 'owner' types.

18. Users should be able to view 'live' library catalogue data about the resources to establish availability.

Talislist	Yes. The system links directly to the institutional catalogue.
ReadingListDirect	Yes. The system links directly to the institutional catalogue, and other institutional catalogues.
Bookworm	Yes. The system links directly to the institutional catalogue.

19. The process of adding a web resource should be a simple cut and paste type of task.

Talislist	Yes. This is a simple process that requires installation of bookmark tools.
-----------	---



	Users would need to download the tools on each PC they wished to edit lists from.
ReadingListDirect	Potential. The academic would literally have to cut and paste the URL, there are no tools to support web resource discovery and addition.
Bookworm	Potential. The academic would literally have to cut and paste the URL, there are no tools to support web resource discovery and addition.

20. The system should allow for resources to be associated with supporting information / resources.

This could be another resource held by the institution, a publisher’s website, or other online information.

Talislist	Yes.
ReadingListDirect	Yes.
Bookworm	Yes, through the addition of notes.

21. The system should allow academics to add comments to resources.

This will give lecturers the opportunity to explain the relevance of the resource to the course and give any instructions to the students regarding resources.

Talislist	Yes. Pre-text and post-text boxes allow extra details about the resources to be added, using different colour text.
ReadingListDirect	Yes, through the addition of notes.
Bookworm	Yes, through the addition of notes.



3.3 Summary Analysis

Requirement# (TRS#)	Requirement	Talislist	RLD	Bookworm
RLM1a (101)	Define resource list metadata for ordering	Yes	Yes	Yes
RLM1b (103)	Allow for ordering parts of books	Yes	Yes	Yes
RLM1c (110)	Identify preferred ranking system (background,essential, purchase etc.)	Yes	Yes	Yes
RLM1d (207)	Include student number information	No	Yes	Potential
RLM1e (304)	Include abstract information in display	Yes	Potential	Potential
RLM2a (102)	Unpack abbreviations used for journals	No	Yes	Potential
RLM2b (305/309)	Use established citation methods	Yes	Yes	Yes
RLM3 (105)	Highlight new order requirments for library staff	No	Yes	Yes
RLM4 (112)	Allow students to comment on resources	No	No	No
RLM5 (113)	Allow for importing of existing electronic resource lists	Yes	No	No
RLM6 (201)	Allow for lists to be hidden from student view	Yes	Yes	Yes
RLM7a (113)	Allow for importing of MIS data (new courses)	Yes	No	No
RLM7b (113)	Allow for importing of MIS data (numbers, codes, titles, owner)	Yes	No	No
RLM8 (204)	All modules should have a resource list on system	Yes	Potential	Potential
RLM9 (206)	Ability to direct students to book purchase options	Yes	Yes	Potential
RLM10 (210)	Should support a variety of interface routes	Yes	Yes	Yes
RLM11 (306)	Allow for linking to journals at article level	Potential	Potential	Potential
RLM12 (513/514)	Allow for lists to be divided into sublists	Yes	Yes	Yes
RLM13 (308)	Include an alerting system	No	Yes	No
RLM14 (104)	Allow for resource information to be drawn from a variety of targets	Yes	Yes	No
RLM15 (312)	Should make academics aware of all resources available	Yes	No	No
RLM16a (504)	Should allow for use of access icons for individual items	No	No	No
RLM16b (504)	Should allow for institutional branding	Yes	No	Yes
RLM17 (604)	Should allow for authorisation to be given to different groups of users	Yes	Yes	Yes
RLM18 (605)	Should allow users to view 'live' catalogue information	Yes	Yes	Yes
RLM19 (608)	The process of adding a web resource should be very simple	Yes	Potential	Potential
RLM20 (310)	Allow for resources to be associated with supporting information	Yes	Yes	Yes
RLM21 (112)	Allow for academics to add comments to resources	Yes	Yes	Yes

4. Technical Requirements and Information

4.1 TalisList

Talislist requires a Sybase database and a server running either Apache or Tomcat on Unix installed locally. The cost of this will be in addition to the costs charged for the Talislist product.

Talislist needs to be configured so that it can work with the local library management system. Talis already has experience at working with several of the major systems, and will work with institutions to configure additional options as part of the set-up process.

The list of targets included in the Talislist 'shopping list' is configured by Talis. This list will need to be identified and agreed as part of the set-up service.

The work carried out by Talis to set-up and install the service is charged to the institution.

4.2 Reading List Direct

The RLD system runs as a centrally hosted service for institutions so there are no local technical requirements.

The configuration to enable searching of local library management systems is carried out by RLD and is included in the overall cost of the product. RLD already has the capability to carry out targeted searches for a large number of UK HE and FE institutions.

As the service is hosted centrally, any work to include additional services alongside the core functionality provided by RLD will need to be discussed and progressed with the providers.

4.3 Bookworm

Bookworm uses Perl software running on Linux. The specifications for the server used at Loughborough University are: Red Hat 6.2 - 600MHz Pentium III CPU, 256MB of RAM and a 12GB IDE hard disc.

Data is held in a MySQL database. Integration with the library system is via Z39.50 (for bibliographic data only). Various free software libraries are required: the perl CGI, DBI and DBD::MySQL, Net::Z3950 and MARC modules, and the yaz Z39.50 C++ library.

The team at Loughborough are keen to continue developing the system and regularly release new versions of the software. Any institutions wishing to adapt or add functionality to the Bookworm system may find it useful to consult the Loughborough team.

5. Financial Information

The following details represent an outline guide of the potential cost for each of the systems. A careful analysis of the lifecycle costs of each system should be carried out by any institution wishing to implement one of these options.

This information is based on information provided to the DELIVER partner institutions. Any institution wishing to adopt a system should discuss cost issues directly with the supplier.

5.1 Talislist

- Initial Cost.

Talis charge a one-off set up cost of £10,000 for an existing Talis site, and approximately £15,000 for non-Talis users. In addition, site-specific installation is likely to cost institutions an additional £2,000.

Institutions will also need to purchase a host environment for the system, which is likely to be a standard Unix / Apache / Tomcat server with a cost of around £5,000. A Sybase RDBM license will also need to be purchased.

- Annual Fee.

Talis charge Talislist users 20% of set up cost as annual support fee (£2,000 - £3,000).

- Additional Costs.

The local costs for the institution, including input required by IT staff, general system administration, and product-specific training have not been assessed.

5.2 ReadingListDirect.

- Initial Cost.

ReadingListDirect currently charge a one-off set up fee of £500 per institution, which is scheduled to rise to £3000 on the 1st April 2003. As the system is hosted off site, there are no additional hardware or software costs.

- Annual Fee.

An annual license fee is charged, based on the number of students at the institution. This ranges from £1,000 - £2,000. Fixed price contracts of up to five years are available.

- Additional Costs.

The local costs for the institution, including input, general system administration, and product-specific training have not been assessed.

5.3 Bookworm.

- Initial Cost.

Bookworm is freely available as open-source software. All initial costs will be related to in-house IT and administrative staff time to set the system up.

Institutions will additionally need to purchase a server for the system. This is likely to be a standard Linux server with a cost of around £5,000.



- *Annual Fee.*

Annual fees will be in terms of staff maintenance time in-house.

- *Additional Costs.*

The local costs for the institution, including input required by IT staff, general system administration, and product-specific training have not been assessed.

6. Summary Comments

6.1 Talislist Comments

- Strengths

The clear strength of Talislist is in resource discovery and list compilation. The 'Shopping Tools', that allow list compilers to add bookmarks to their browsers for seamless addition of materials, are a very good example of seamless functionality and user interface design. The tools are available for both Netscape and IE browsers, and the IE version installed effortlessly during the demonstration at DMU. As long as users are careful to read the instructions provided, this process would be extremely beneficial.

Although there is no specific interface to the institutional MIS for import of course and staff IDs, batch import of XML records defining these should be possible. A specification of this process has been provided by Talis.

The resource lists are controlled using a series of permissions. These allow the various types of user right - of - access to different functions. The lowest level of permission is public, which is equivalent to the student level and allows the list to be viewed and printed. The highest level of permission is library staff. This gives access to all the functions within the system. In particular, it allows the "giving of permissions" to other users. These permissions functions mean that users cannot alter lists without the typical owner of the majority of the resources – namely the library, knowing and agreeing to them.

Straightforward links are used from VLE interfaces into Talislist, to resource lists for a specific lecture or for a larger element of a course. Resource lists can be defined at a total of 9 hierarchical levels, 3 each within Course, Module and Topic. Levels can be used to define such categories as "Essential", "Recommended" etc.

- Weaknesses

The Talislist product is designed to work with libraries that have the Talis library management system. It was agreed in the demonstration on January 30th that they have less experience of working with other computerised library catalogues and that there may be some issues with installing and maintaining the system with other products. This was reflected in the purchase price and the likely annual maintenance needs being four days rather than two.

Talislist suffers from the (common, and rather fundamental) problem that, although a URL can (probably) be generated easily to link into a WebOPAC page for an individual item (showing shelving location and status, etc), extracting this information for a list of items (e.g. all items on the same resource list, to print for use as a 'shopping list' when visiting the library) is difficult (impossible?) from any LMS (except Prism, the Talis LMS product) because there are no independent standards (or even emerging standards, known to Talis) for such interchange. The Talis Prism WebOPAC makes such access possible from a non-Talis external system, by embedding XML data (to a proprietary, but available DTD) about items in such HTML pages. It is not known which other LMS vendors include such a technique to make 'scraping' more reliable than from readable-formatted HTML. The JISC 7/02 "41" project at the University of Ulster is exploring this as mentioned in recent Ariadne article by Steve Richardson and Andy Powell.

Talislist (in common with ReadingListDirect) has not addressed the problem of 'deep-linking' to e-resources, to find/include an indirect URL that could dynamically find the 'appropriate copy' for each user, with side-benefits such as usage-logging. This confirms the DELIVER decision to focus the development of ANGEL tools on addressing this problem.



Talislist is mainly dependent on the local library catalogue (therefore, of items already held by the library) for bibliographic reference finding. The list of “Entry Shopping” targets could be extended to include some wider catalogue, such as OCLC WorldSearch. Searching for bibliographic references, from an incomplete author or title spec, across several of the defined targets is not possible. This means that list compilers may have to repeat searches to find the desired resource (i.e. once in the library catalogue, then externally if not found).

The admin interface for library staff doesn't show in which target a resource has been found. It doesn't show aggregated numbers of students (on all known courses) for which an item has been recommended (although this information could be extracted by a report, entries/changes are not time-stamped, so changes would be hard to detect). There is no alerting mechanism for library staff of changes to lists or demand for items to be ordered.

Icons for editing within the system were very small and not very intuitive. The interface makes no claims to meet accessibility standards. It is likely, however, that a company such as Talis will be aware of the requirement on HE institutions to meet accessibility guidelines and will hopefully address these issues in time.

Authentication of users with better-than-public access to Talislist is via internally maintained usernames/passwords; there is no inter-operability as yet with LDAP or other AAA systems.

- Summary

Talislist has obvious benefits as a system for existing Talis users, especially with the continued development with (and within) the Prism system. Talislist also offers useful tools for non-Talis users. The resource discovery ‘shopping’ tools are a very attractive feature of this system.

The major drawback to the system is poor support for library staff requirements. Talislist does not automatically support searching for new collection items. Although a ‘books in print’ catalogue could be added to the list of search targets, it is not a standard feature. The system also does little to support library processing of resource lists in terms of identification of new material, statistics on number of student recommendations per book etc.

An institution wishing to adopt this system would need to weigh up the importance of the resource discovery / list building tools against the absence of library support.

6.2 ReadingListDirect Comments.

- Strengths

The strengths of ReadingListDirect are associated with its 'knowledge' of book collections. The fact that academics can easily find and include in lists books that are not in library catalogues is useful for the academics, and for the librarians who do not then need to instigate a separate process for new items. RLD also automatically searches all the available targets, so that the list compiler is not expected to know where the target information is held or repeat multiple searches to find references.

The fact that students are guided to a variety of resource locations is another strength of RLD. The system can be configured to suggest a variety of consortium libraries, or local libraries that a student may wish to search for availability. This actively encourages students to consider all the options available to them.

Lists have a time period attached to them, which is a useful feature if annual updates are required. List owners are automatically notified of the need to update the lists by e-mail. Lists can also be made dormant, if a course does not run in a particular year, and compilers have a lot of flexibility when it comes to sorting lists into sub lists.

The Library component provides good statistical information to librarians concerning resource recommendations, and RLD is the only one of three systems looked at that is able to provide aggregated information concerning the total number of students that have been recommended a text across all courses. This is essential for ensuring the correct number of texts are available institution-wide.

The ReadingListDirect interface is badged as 'Bobby Approved'. For more information on Bobby and accessibility issues visit <www.cast.org/bobby>.

- Weaknesses

Two clear potential disadvantages to RLD are the fact that the system is hosted off-campus, and the use of advertising. Off-campus hosting relieves the need to provide server capacity in-house, but takes away control of availability (and down-time). If the company ceased to operate, the service would be completely withdrawn.

Use of advertising allows RLD to offer extremely competitive start-up costs, but will be undesirable for many individuals and institutions. The space that the adverts take on-screen may also effect integration with other systems, such as VLEs.

The use of an off-site service means that RLD has a predetermined look and feel, and is not institutionally branded. The layout of lists is neat, but compressed and links are not always distinguishable from plain text.

As mentioned in the Talislist comments, RLD also suffers from the inability to produce a printable list of precise resource locations to carry to the library.

- Summary

The strength of RLD lies in its ability to search a vast range of targets in one search, identifying locally held copies, partner institution copies and bibliographic details for items to be added to



collections. The system also provides strong aggregated statistical data for librarians that clearly identifies the number of students each resource supports.

The business model adopted by RLD is likely to be a key factor for procurement decisions. Off-site hosting, use of advertising and the click-through revenue model may be areas of concern for institutions wishing to adopt RLD. It is worth noting that both the advertising and book purchase recommendations are sensitively handled by RLD and are clearly secondary to the display of reading lists and promotion of in-house availability of resources.

6.3 Bookworm Comments.

- Strengths

As an open source system, Bookworm has a clear strength in the lack of initial expenditure on system purchase. There will naturally be a cost in terms of staff time to set the system up in-house, without the support that is often supplied by commercial vendors.

Bookworm is a particularly useful tool for librarians, supported by a database that processes list information. This includes a nightly alerting system that will alert the correct librarian of updates to a list. Such a system will clearly support and promote ongoing updating of resource lists, as well as the traditional annual update.

Bookworm offers a completely plain interface, which can be seen as an advantage and a disadvantage. It allows individual institutions the opportunity to adapt the system and look to suit needs, and allows a greater amount of flexibility in the way that the lists are displayed. This may be a disadvantage to some list compilers as some knowledge of html is required for effective presentation of lists. It does allow institutions to appropriately brand the system and make sure that institutional policy regarding accessibility is followed.

Use of Z39.50 should ensure that the system will operate with any library catalogue system.

- Weaknesses

Bookworm is essentially a tool for creating lists of items that are already in the library catalogue. Information about resources not included in this target have to be entered by hand by the academic and cannot then be centrally supported. This is an essential feature for the resource list system, and this functionality would therefore need to be provided by an alternative system or in-house development.

The plain interface offers limited functions to the list compiler, although users have the opportunity to redisplay the list for convenient printing (although not including library location information).

- Summary

Bookworm has an immediate advantage in terms of price as it does not have the set-up and annual fees that are charged by the commercial products. Any institution considering an open source solution would need to carefully identify all of the potential staff time cost for all of the systems to make a true cost comparison.

The nature of Bookworm makes it a good potential system for further development, or integration with additional tools.

Bookworm provides good functionality for library staff by supporting the system with a strong back-end database. The resource discovery tools are, however, very limited. The time that staff would need to take to cite references not held in the library catalogue may be off-putting.



7. Recommendations.

The three solutions evaluated are very different from each other, despite them all addressing a common set of broad requirements. Talislist is a 'traditional' commercial software product, installed and supported by a vendor, but hosted and supported (directly) by the institution/library. Bookworm is an open source software package, relying more heavily on the availability of local support and maintenance expertise within the institution/library. ReadingListDirect is a managed service, rather than a software product, with consequently reduced technical support costs for the institution/library, but also risks associated with the outsourcing of such an important institutional function.

All three solutions have strengths, weaknesses, costs, benefits and risks, which this report has attempted to set out in a way that is primarily intended to facilitate decisions by the two partner institutions in the DELIVER Project about which, if any, to use for the purposes of the Project. A group of appropriate senior staff with service and budget responsibilities at LSE and De Montfort should each make separate and independent decisions.

The Project will then work with the solution selected by each partner, to integrate them with other elements of required functionality, using and further developing software tools built by the JISC ANGEL Project <http://www.angel.ac.uk/>. If the two partners make the same choice, there will be benefits in terms of more development work that can be completed by DELIVER, within limited time and resources. If two different solutions are selected, the Project will benefit from the opportunities for wider inter-operability testing of the open source components it develops. JISC funding for DELIVER will not be used to cover purchase or support costs of the chosen solution(s).

Subject to the in-depth evaluation-in-use (including assessment of ongoing operating costs) that the DELIVER Project will allow, until 31st July 2003, it is expected that each institution will then make a further decision on whether to continue using the selected reading list management solution, having secured long-term institutional funding to sustain future operating costs and staffing requirements.,

This report has been published by the DELIVER Project, with the prior agreement of ReadingListDirect and Talislist, because it may also be of use to other HE and FE institutions attempting to make similar decisions. Feedback on the content of this report is welcomed to the general contact address for the ANGEL projects group: info@angel.ac.uk.