Appendix A: Functional Requirements for Electronic Resource Management[1]

Ivy Anderson, Sharon E. Farb, Adam Chandler, Timothy Jewell, Kimberly Parker, Angela Riggio, and Nathan D. M. Robertson

Introduction

The goal envisioned in these requirements is a system that supports management of the information and workflows necessary to efficiently select, evaluate, acquire, maintain, and provide access to e-resources in accordance with their business and license terms. Such a system should support the service requirements of e-resources while building on existing investments in library technology, through seamless interaction and efficient sharing of data with traditional MARC-based online catalogs, Web portals, federated searching tools, local resolution services, local authentication and access-management systems, and traditional library-management functions. Whether implemented as a stand-alone system or as part of an existing library-management system, the ERM functions should complement, rather than duplicate, the capabilities of other systems deployed by the library.

This document outlines the broad areas of functionality that are required in such an ERM system. It is not intended, at this stage of development, to be a detailed description of every function needed, but rather to provide an overview of the main functions required of the system. The examples under each functional area are representative and illustrative, rather than exhaustive.

Guiding Principles

The ERM system should offer an integrated environment that supports both management and access, without maintaining duplicate systems. The system should offer a capacity for global updating and flexible addition of fields. It should offer the ability to hide fields and records from public view and have a single point of maintenance for each data element. It should support interoperation and dynamic data sharing with existing OPACs, Web portals, library-management systems, and link resolution services. It should offer users consistent information, regardless of the path they take in seeking it. Finally, the ERMS should, over time, support the ability to store, access, search, and generate reports of the information that it contains.

FUNCTIONAL REQUIREMENTS

General Requirements

[1] This document is based on specifications originally developed by Ivy Anderson, Harvard University, Ellen Duranceau, MIT, and Robin Wendler, Harvard University.

The ability to support complex, multiple-to-multiple relationships that accurately map how resources, business terms, licenses, and other components interrelate is an essential building block for supporting the functionality described throughout this document. The companion Entity Relationship Diagram for Electronic Resource Management (see Appendix C) presents a schematic representation of these relationships.
For all areas of functionality described in this document, it must be possible to

1. Identify what bibliographic entities (electronic and print) are covered by or provided through a given license, set of business terms, package, or online interface platform (hereinafter "interface");

2. Associate the characteristics of a given license, set of business terms, package, or interface with all the bibliographic entities (electronic and print) to which they apply;

3. Offer security features to control staff views and maintenance rights
    3.1. hide confidential information from certain staff
    3.2. restrict the ability to add, update, or delete certain data (field by field) to designated staff
    3.3. provide the ability to construct simplified data views by hiding unused fields; and

4. Perform ad hoc queries and generate reports across a broad range of fields and table values.

Resource-Discovery Requirements

While adequate facilities for searching and browsing for e-resources already exist in OPACs and portals at many libraries (through searches of author, title, alternate titles, cross-references, subjects, keywords, and other descriptors), facilities to cull e-resources for resource-discovery purposes are lacking at other institutions. Ideally, the ERM system should provide a user interface for libraries that require one as well as a means of integrating appropriate license data into the results of searches within externally deployed systems.

The ideal system should also provide Web developers database table- or view-level access and the ability to bypass the native interface to query and present data to the user in any number of custom-configurable ways. An ERMS implementation should include a dedicated user interface that is able to capture appropriate data in other systems or to share its information with elements in other supported systems to create unified user displays, including the capabilities described below.

It should be possible to:

5. Make resources available through or pass on information about resources to OPACs and Web-presentation services using traditional resource discovery methods, including the ability to search and browse by author, title, alternate titles, cross-references, subject, and keywords, facilitate the generation of dynamic, database-

driven Web pages that constitute pathfinders, course- or discipline-specific pages, or lists of selected e-resources in a given subject area and of a given type (e.g., e-journals, article databases) while supporting fully customizable presentation templates that are not limited to the branding and look and feel of the native interface;

6. Offer contextual presentation of relevant license information to the user at the point of access, regardless of the access path taken through any supported tool,

    6.1. make apparent to users whether or not they are authorized for access under the license
    6.2. display relevant permitted uses, use restrictions, and special requirements such as:
        6.2.1. permission to use in course packs, course Web sites, and distance education
        6.2.2. unusual prohibitions, such as record download limits and time of day restrictions
        6.2.3. citation requirements;

7. Support site-defined and auxiliary descriptive data. Such data might include:

    7.1. locally defined fields for descriptive needs not covered in existing MARC records
    7.2. a free-form note
    7.3. locally defined lists and descriptors;

8. Make available information about and provide access to other versions of the resource being viewed, including links to catalog holdings information for physical manifestations and direct links to all accessible electronic versions;

9. Make available information about issues particular to the online interface, such as inaccessible or nonsubscribed portions not marked as such at the site, unusual log-on and log-off requirements, and navigation or accessibility features;

10. Flag resources as unavailable in real time, with an optional explanatory note (see also Administrative and Management Functions); and

11. Offer advisory notices of planned downtime and other time-sensitive information.

Bibliographic Management Requirements

In many libraries, both traditional and auxiliary bibliographic data may be distributed among a variety of systems, including the library-management system, a federated searching portal, and a link resolution service. Nevertheless, to the extent possible, each unique data element should have to be maintained only once. Updates in one system should be automatically reflected in corresponding systems, either through the dynamic sharing of data or, where redundant storage cannot be eliminated, by propagation to other systems. Loading of data from external systems should also be supported.

Specifically, it should be possible to:

12. Provide a single point of maintenance for bibliographic and auxiliary descriptive data that can be exchanged or shared between the OPAC, portal lists, federated search tools, local resolution services, and other bibliographic systems and services; and

13. Import subscription-management data from external providers using standard software and developing protocols,
    13.1. lists of titles, ISSNs, and uniform resource identifiers (URIs) delivered in Excel or delimited format
    13.2. changes to titles and dates of coverage in aggregated databases communicated via any XML-based or standards-based data transfer protocol, such as ONIX for serials
    13.3. holdings updates supplied by a publisher or subscription agent to facilitate electronic check-in.

Access-Management Requirements

All libraries rely on authentication and access-management systems that are external to the systems and tools described in this document. These external systems may be as straightforward as reliance on the remote-authorization mechanism of an online provider (via IP addresses or user names and passwords) or as complex as a locally developed access-management service that assigns persistent identifiers to resources, passes connection requests to a system that validates users according to a local authentication scheme (e.g., Kerberos), and routes valid users through a proxy server. Institutions with complex local environments can be expected to have customized systems and tools with which to perform these functions, and the ERM system must interoperate with them.

The ERMS should accommodate both simple and complex environments with a disparate range of needs. To accomplish this, the following generalized requirements are necessary:

- Management of basic access-related information, such as URIs, user IDs and passwords, and lists of institutional IP addresses;
- Support for the creation of persistent URIs and for additional data elements required to support complex local access management services, such as proxy servers; and
- A set of export functions by which an ERMS can communicate its information to a local system or service.

Specifically, it should be possible to:

14. Store and maintain access URIs and make these actionable for end users, according to local requirements:

14.1. store vendor-supplied primary and secondary URIs (e.g., for mirror sites) used for access to the resource

14.2. support the creation, storage, and updating of persistent URIs and/or integration with external systems for managing persistent identifiers

14.3. support authentication and access systems (such as proxy servers or statistics-generating scripts), allowing for URIs to be constructed on the fly on the basis of stored data elements

14.4. generate notifications and/or exports of URI information to appropriate linked or external systems according to local requirements (e.g., notification or export to cataloging and information technology departments or systems)

14.5. provide seamless functional integration with external systems that record this information (e.g., through a shared pointer);

15. Integrate proxy server/access management with other functionality

15.1. provide a means for seamless integration of proxy server access for all or selected users and all or selected resources;

16. Store lists of IP addresses used to register access to specific resources and provide automated e-mail notification to online providers when IP addresses are updated,

16.1. support the creation and maintenance of multiple lists of IP addresses that can be associated or disassociated with one or more licensed locations and linked to one or more bibliographic entities

16.2. for a given resource or online provider, indicate whether IP addresses are/were registered online and record the registration URI

16.3. send automated e-mail notifications to vendors and providers when IP addresses are updated and record the date on which notifications are sent. Include the ability to record an acknowledgment date. This implies the ability to designate a vendor or provider contact address for IP address-notification purposes;

17. Store one or more user IDs and passwords and provide the ability to generate secure screen displays of this information for authorized users and staff, with associated text or for JavaScript autosubmission; and

18. Implement access restrictions,

18.1. record authorized user categories and authorized sites, including the ability to associate specific actions with those elements, such as:

18.1.1. generating staff and user displays

18.1.2. implementing access controls

18.1.3. exporting information to a local access-management system (technical system in use at your institution such as an authentication system and/or proxy server (see also Resource Administrative and Management Functions).

STAFF REQUIREMENTS

Staff Interface

An ERM system requires a staff interface that enables library staff to efficiently carry out the work described in the sections that follow. The interface should be organized into views that are optimized for particular areas of staff activity or interest, such as resource acquisition, troubleshooting, license administration, or administration and statistics. Details of appropriate interface design are not addressed in these requirements; these are left to the art and discretion of system developers.

Through this interface, staff should be able to:

19. Search, browse, and retrieve records by attributes unique to e-resources, such as license, vendor, interface, record status, licensed site, consortium, and library selector or other local contact individual;
20. View the full range of information appropriate to the staff member's security profile and functional role;
21. To the extent possible, link to other relevant information stored in library management systems, portals, or related systems; and
22. View records that are hidden from the public.

Selection and Evaluation Process

During the selection and evaluation process, it is typically necessary for a decentralized group of individuals to coordinate a complex and iterative series of steps. The ERM system should provide support for recording actions at each of these steps and be capable of performing specified actions upon completion of a given task and of sending alerts when anticipated actions do not occur.

A typical selection and evaluation process might involve the following steps and require the following reminders and notifications, all built from stored action dates and locally definable status fields:

- Build request record for trial;
- Flag resource as trial;
- Notify interested parties that trial is live and provide the access instructions and expiration date (triggered by trial start date and e-mail address[es] input into request record);
- Allow staff (and, optionally, users) to access the trial via stored URL;
- Send reminder to those notified of a trial that expiration date is near (triggered by trial end date and e-mail address[es] input into request record);
- Record opinions of key players and final purchase decision, allowing for both central and local input of opinions and funding commitments into notes fields in request record;

- Prompt licensing/acquisitions contact to report purchase decision (triggered by decision date input into request record). The most common options are “Approve,” “Reject,” or “Put on Hold”; and

- For resources that are approved for purchase, additional statuses and actions would be required, including
  - notify licensing contact to obtain and negotiate license, and remind this contact if license is not completed by a given deadline (triggered by stored licensing contact, notification of approved status, and deadline input at time of approval)
  - notify access contact if status has not changed to “live” by deadline (triggered by inputting likely access date into system at time of approval)
  - notify cataloging and selector/product sponsor, as well as other interested parties, when access is available (triggered, for example, by changing status to “Live” and by stored e-mail address[es] related to this purchase).

The companion Electronic Resource Management Workflow Flowchart provides a detailed diagrammatic view of workflow decision and action points (see Appendix B).

To support these processes, it should be possible to:

23. Create provisional records for resources that may or may not be permanently acquired and track the selection and evaluation process through acquisition or rejection;

24. Assign locally definable status fields to request records and associate particular actions with those statuses. Examples of such fields include “New Request,” “On Trial,” “Trial Expired/Decision Pending,” “Approved,” “Rejected,” and “On Hold”;

25. Assign locally definable fields to request records for license routing and status and associate particular actions with those fields. Site should be able, for example, to define a list of individuals from whom sign-off is required, send reminders to license reviewers, and record approval or rejection status, with notes;

26. Assign multiple local contact individuals and site-defined roles to both request records and permanent resource records;

27. Send e-mail notifications to individuals designated as local or licensing contacts;

28. .Make trial resources available in a secure manner through the library’s resource tool of choice to authorized users (if wanted) and staff
    28.1. store trial URIs and passwords and make these available, securely if necessary
    28.2. flag items unambiguously as trial resources, with associated public and staff notes
    28.3. record a trial expiration date and, optionally, send an alert to designated recipients n days prior to trial expiration;

29. Record a decision due date and, optionally, send an alert to designated recipients n days prior to decision date;

30. Establish a site-defined routing workflow for resources that are approved for purchase. For example, it should be possible to send notifications to designated staff or departments or to place resources in a queue for further action by those units to trigger actions such as the placing of an order, completion of cataloging, and implementation of access management by designated staff; and

31. Purge rejected records from the system, sequester into a history archive, or retain such records with notes about the decision process (including a link to written evaluations if wanted), at the library's discretion.

Resource Administration and Management Functions

The functions described in this section pertain to activities required to fulfill license obligations and to administer and support resources that have been acquired. Some of the data and functions pertain to the license itself, while others pertain to the management of individual resources. The system should be able to record data at the most efficient level required to avoid redundancy, relying on the relationships among entities to make the information available at other applicable levels (e.g., individual resource, package, or interface).

It should be possible to:

32. Store license rights and terms for reference, reporting, and control of services

    32.1. for services including, but not limited to, interlibrary loans, reserves, distance education, course Web sites, and course packs,

        32.1.1. identify whether a given title may be used for the service and under what conditions

        32.1.2. generate reports of all materials that may or may not be used for the service with notes about under what conditions, and

    32.2. include a mechanism for adding new services, allowing staff to specify the name of the service and associated actions such as public displays, reports, and alerts;

33. Record the categories of users and sites that are authorized for access to a given resource and generate displays for end users and staff on the basis of this information (see also Access Management Requirements);

34. Support breach investigation and cure activities and other activities that may be required to fulfill license obligations

34.1. record the cure period for breach and generate alerts at library-specified intervals on the basis of this information when a breach has been reported

34.2. provide a breach-incident log in which to record reports of alleged breaching activity, including date and source of initial report, open/closed status, resolution date, and actions taken. This should have the capability of generating reports on commonalities of breaches (e.g., type of breach, department showing regular breaches)

34.3. include a mechanism for adding fields in which to record unusual compliance requirements that the library may want to track, including implementation status and date and the ability to generate reports or alerts on the basis of this information. Examples might include a requirement to post notices to end users or to destroy locally held copies upon termination (see also Termination Activities);

35. Record additional license terms and metadata for contract management and auditing purposes, such as license commencement date, duration of agreement, confidentiality provisions, and other site-definable key terms;

36. Provide the ability to display or link to an online version of a license agreement;

37. Manage the library's archival rights in electronic content,

37.1. record whether permanent rights to the year level exist in a given resource and, if so, their source, dates of coverage, manner of execution, and the applicable license through which rights are provided. It should be possible to associate multiple archival field clusters with a given resource to track successive rights with multiple providers

37.2. retain archival versions of license terms and conditions if desired, so that preexisting terms of agreement can be mapped to the bibliographic entities and dates for which such terms applied;

38. Support the administration of e-resources,

38.1. store administrative URIs, IDs and passwords, and associated notes and make them available to authorized staff

38.2. store subscriber numbers used to register online journals that are tied to print

38.3. provide the following capabilities for configuration options, including but not limited to features such as institutional branding, hooks to holdings, Z39.50 and OpenURL support,

38.3.1. identify whether a given title supports the feature and whether it has been implemented, with associated notes

38.3.2. generate reports of all materials that do or do not support the specified feature, including implementation status

38.3.3. include a mechanism for adding new features, allowing staff to specify the name of the feature, implementation status, notes, and associated actions such as report generation,

38.4. record the number of licensed concurrent users for a given resource, interface, or pooled user group (including pooled consortium users)

38.5. record cataloging-related data such as,

38.5.1. the availability and quality of MARC records for package items that include individual entities, including the status of loading or prioritization and additional notes

38.5.2. the person or unit responsible for cataloging, if applicable

38.5.3. related specifications, such as specific entries or other data to be included in cataloging records;

39. Support library instructional activities,

39.1. store information about training accounts (URIs, IDs and/or passwords) and other user instruction arrangements and make these securely available to staff arranging for training classes and (optionally) to end users

39.2. record information about and/or provide links to available documentation for staff and end users

39.3. record information about and/or provide links to training classes;

40. Support management of and access to usage statistics,

40.1. indicate whether usage statistics are provided, and record associated data such as frequency, delivery method, and available format(s)

40.2. record URIs and IDs and passwords for access to online statistics

40.3. store or provide information about and/or links to locally stored data

40.4. if local storage of usage statistics is supported, provide the ability to download locally stored data into a spreadsheet for manipulation purposes

40.5. generate staff displays for access to usage data

40.6. provide a note field for special information pertaining to usage statistics (e.g., missing time periods or data errors)

40.7. generate reports or notifications indicating when usage statistics should be available, based on the frequency parameter established in 40.1;

41. Support the troubleshooting of access and performance-related problems,

41.1. record the amount or percentage of allowable downtime provided in the license agreement for performance monitoring purposes

41.2. store information about the provider's normal maintenance window, and include the ability to display this information to the end user

41.3. store information on locally defined notes for local performance monitoring Web sites/programs

41.4. display URI of server status provided by vendor

41.5. display to staff who perform troubleshooting elements such as URI and proxy information, hardware and software requirements, number of licensed concurrent users, subscription expiration dates, local contact information (site-defined), and vendor contact information

41.6. provide the ability to flag resources as temporarily unavailable, with both staff and public notes. It must be possible to apply the flag to individual titles, to all titles in a given package, or to all titles that use a given online interface (see End-User Requirements)

41.7. integrate a support incident log into the system to record and track problems, including date and source of initial report, category and description of problem, open/closed status, resolution date, and a record of actions taken,

41.7.1. provide the ability to route open problem reports to appropriate local contacts for further action including email capability for routing of problem/troubleshooting reports

41.7.2. generate alerts for unresolved problems at site-defined intervals

41.7.3. generate incident history reports on demand by resource, package, interface, and vendor for performance monitoring and auditing purposes,

41.8. [Desirable]: calculate the duration of downtime incidents based on data recorded in the log, and generate an alert if downtime exceeds the amount or percentage of allowable downtime provided for in the license agreement; and

42. Facilitate communication with vendors,

42.1. record multiple vendor contacts, including name, title, e-mail address, phone, and fax numbers, including the ability to assign site-defined contact roles. It should be possible to assign multiple roles to a single contact (technical support, customer support, sales, billing support)

42.2. provide the ability to generate standard notifications of IP addresses and changes to vendors

42.3. record official contract notice address and associated requirements (e.g., delivery requirements)

42.4. provide the ability to generate email messages to designated technical support contacts from data in the incident log.

Business Functions

Many of the business activities described here are related to, and may already be accomplished through, functionality that exists in a traditional library learning management system. Although item 44 in this section affirms the general requirement to support traditional acquisitions-related functions, such as ordering, fund accounting, and the ability to commit and expend budgets and to generate budget reports, it does not attempt to describe these requirements in detail, on the assumption that full acquisitions functionality will continue to reside within a library management system. These specifications concentrate instead on the functions and elements that are uniquely required to manage e-resources.

It should be possible to:

43. Make complex business information available to staff,
  43.1. store a description of the pricing model applicable to the resource
  43.2. know what license agreement is applicable to a given set of business terms
  43.3. know what print resources are subscribed to that are part of a package
  43.4. where there are restrictions on cancellation of print subscriptions
    43.4.1. record this information
    43.4.2. if the system is integrated with a library management system that supports print resources, warn (or block with override by authorized staff) attempts to cancel print subscriptions
  43.5. where price caps exist for multiyear agreements
    43.5.1. record this information
    43.5.2. calculate renewal invoices to ensure that price caps are not exceeded, where possible
  43.6. if the system is integrated with a library management system that supports print resources, prompt the library to evaluate retention of associated print subscriptions at renewal;

44. Facilitate the acquisitions process,

  44.1. perform traditional acquisitions functions such as:
    44.1.1. fund accounting
      44.1.1.1. commit, expend, and update budgets
      44.1.1.2. Produce budget and expenditure reports
    44.1.2. ordering
      44.1.2.1. ability to assign and issue purchase orders in any standardized format or protocol. For example, print, e-mail, or EDI transactional transmissions such as X.12, EDIFACT, or other XML-based suites
      44.1.2.2. purchase orders should accommodate site-defined data elements unique to e-resources, such as IP addresses and activation instructions
    44.1.3. invoice payment
  44.2. support cost sharing among departments and fund lines based on fixed numbers, percentages of some characteristic such as use, total budget or population, or special formulas
  44.3. support the ability to pay from one fund and charge costs against multiple funds belonging to distinct budget or administrative units according to the above algorithms;

45. Facilitate cooperation with consortial partners in a license agreement,

45.1. for resources acquired through a consortium, record the name of the consortium, relevant notes, and, optionally, the names of other participating institutions or the materials available as a result of participation of consortial partners
45.2. store name and contact information for key consortial contacts;

46. Facilitate the renewal process,

46.1. record the expiration date and update this date automatically upon renewal
46.2. record the advance notice period for renewal (in the form n days prior to expiration), and calculate a renewal notification date based on this information
46.3. generate a report or send an alert to designated recipients n days prior to renewal-notification date, with associated note. It should be possible to assign a system default period that can be overridden for a given resource (similar to existing claim functions)
46.4. record whether renewal is automatic or explicit and generate an action report and/or alert N days prior to renewal or notification date for resources requiring explicit renewal
46.5. record renewal action or decision and date
46.6. provide pointers or links to written evaluations associated with renewal decisions; and

47. Facilitate termination actions and decisions,

47.1. record whether termination by the library during the contract term is permitted, and if so, record the advance-notice period and applicable conditions (e.g., breach, specific nonperformance, any)
47.2. record whether termination by the licensor during the contract term is permitted, and if so, record the advance-notice period and applicable conditions (e.g., breach, specific nonperformance, any)
47.3. store the termination date and reason for termination
47.4. provide pointers and/or links to written evaluations associated with termination
47.5. record license conditions to be fulfilled upon termination (such as the requirement to destroy locally held copies) and generate alerts based on this information (see also Resource Administrative and Management Functions)
47.6. provide the ability to archive licensing information for noncurrent licenses whose terms remain applicable to previously licensed material
47.7. provide the ability to manage, record, and report on permanent or perpetual access rights following termination of a current agreement.