Framework: Outline

F1: Summary description of the application

F2: User Characteristics

F3: Product context

F4: Task List

F5: Software system

F6: Context diagram

F7: Application diagram

F8: Software system attributes

F9: Installation requirements

F10: Performance requirements

F1: Archivists’ Toolkit: Summary Description

The purpose of this software specification is to describe the features and functions to be included in the first release of the Archivists’ Toolkit application.

The Archivists’ Toolkit is to be an open source database application that will support description of archival materials, and, moreover, will increase archival processing efficiency, lower the costs of most archival description, and promote standardization for description of archival resources. The application will:

- satisfy and integrate key functions in the archival descriptive cycle, including accessioning, registering source information for archival resources, providing topical and name access points for subjects and creators of resources, tracking location of archival resources and producing finding aids, collection guides, METS digital objects, and various administrative reports.

- allow small to medium-sized archives and special collections (without necessarily excluding large archival repositories) to ingest extant metadata about collections in the form of EAD finding aids and MARC records into a centralized, integrated database, and store and process it along with information about digital assets, accessions, and sources;

- use a relational database management system with an Applet based, Swing client user interface to support core functions of archival administration and description; and

- be deployable in a range of archival repositories from historical societies, college archives, museum archives, commercial archives, and other archives specializing in non-textual materials (sound or video archives, for example) on the one hand, and, on the other, in archives with a single staff member or in multi-repository consortia such as the Five Colleges, Inc. or the Online Archive of California.

The application described in this software specification is designed to support essential functions of the record cycle, to promote community descriptive standards, to allow for increased efficiency in archival processing, and, very importantly, to be customizable to local work settings. In addition, it is our objective that the Archivists Toolkit will be released in a state that is easy to install and use, with sufficient supporting documentation for end users and for tracking and addressing bugs.

Besides smart application design and programming, meeting these objectives requires 1) substantial input from the archival community; 2) documentary control of all aspects of the project's work; 3) open communication between the Software

****

Specification Team and the Software Design Team, 4) extensive testing of all application components; and 4) clear and easy-to-use documentation to support operation of the application.

****

F1a: Benefits of the AT Application:

- Accommodate data relating to a broad range of archival functions in a single, affordable, dependable tool. This integration supports quick and efficient retrieval of information and allows a single system to produce both descriptive and administrative reports supporting the management of archives, as well as their use by the research community.

- Facilitate assignment of more archival descriptive tasks to lower staff levels, thereby also decreasing the cost of archival description

- Reduce need for training by automating certain procedures such as EAD and METS encoding.

- Allow for customization of the application to local work settings (rather than require modification of work processes to the application)

- Increase efficiency of archival processing and lower processing costs, thereby helping to diminish processing backlogs

- Improve control over archival holdings, thereby helping to expose “hidden” resources in a more timely fashion, enabling better identification of materials, and providing a foundation for more systematic processing and preservation strategies

- Promote community-based standards, by building of the application on archival content standards and automating the output of metadata such as EAD, MARC, MODS, DC, OAI, etc.

****

F1b: General Features of the AT Application:

[A complete list and description of the archival tasks supported by the AT application is provided in the task list in section F4 and in the specification for each functional area.]

Open Source: The application will be made available as open source, non-proprietary software under an Open Source Initiative (OSI) Certified License.

Platforms/ configuration: The application will be capable of being installed on Windows, MAC OS X, and Linux operating systems as either a stand alone or networked application. Design should not preclude implementing the application in a clustered machine, fault tolerant environment.

Multiple users: The application will support multiple users in a networked installation.

Intuitive User Interface: The application will be relatively easy to use and navigate from functional area to functional area. Use of the application should require minimal training at most staff levels, and it must absolutely be usable by the lowest levels of staff.

Performance: The application will be quickly responsive during all phases of use: data entry, navigation, record operation, and searching.

Secure: The application will be made reasonably secure against system failure, vandalism, or accident. Some effort will be made to minimize the opportunity for data entry error.

Integrated: To streamline processing and support efficient retrieval of information, the application will be based on the entire life cycle of archival materials, beginning with initial contacts with the sources of materials and encompassing all work carried out on materials once they are received in a repository: accessioning; arrangement and description for collections, items, and surrogates; application of controlled access terms for creators and subjects, registration of sources, and location tracking. Sharing records for names and subjects will be possible in a multi-repository implementation of the application.

Standardized: The application will be designed to promote standards such as DA:CS compliant description and to produce access tools compliant with the EAD and METS standards.

Data consistency: Ability to restrict or force use of certain names and subjects to minimize variance of terms and promote compliance with nationally established forms of the terms.

Output neutral: The stored archival information will conform to DA:CS, the SAA sponsored standard for archival description, but a particular output for the information will not be presumed. The information should be capable of being purposed for a wide range of outputs.

Flexible: Adaptable to various work settings because of its customizable modular design.

Customizable: Provide for modification of data entry templates, including presentation of the template, elements contained on the template, elements that are required to complete the documentation task, and sequence of the elements on the template; and modification of reports in respect to branding of the reports, exclusion of data elements from the report, and sequencing of data elements in the report.

Modular: Dedicated modules representative of core tasks represented in the application. Repositories will have the option of not using some of the modules.

****

F1c: Functional Areas

Project Management: The application will support processing project management by allowing templates, available vocabularies, and reports to be customized for any definite processing project.

Accessioning: Ability to create and edit accession records and to produce accession records, lists of accessions for a specified time period, lists of processed and unprocessed accessions, and processing plans.

Resource description: Ability to describe archival items, collections, and surrogates and their component parts where extant at all appropriate levels.

Names: Ability to create records for names (personal, family, or corporate) and link them to accession, resource, or surrogate records to indicate creators of the material, subjects of the materials, or sources of the materials. Name records are designed to be EAC compliant.

Subjects: Ability to create records for subjects (topic, geographic, genre, name) and link them to accession, resource, or surrogate records to indicate what the materials are about.

Location: Ability to create records to indicate stack location of archival materials, to link location records to accession and resource description records, and to produce shelf lists and resource location lists.

Online help: The application will integrate definitions, references to DA:CS rules, and other help messages into data entry templates, making them readily accessible during data entry

Record auditing: When a record is created and when it is edited, the application will automatically enter and store the name of the operator who created / edited the record and the date the record was created / edited. The creation information is captured once and stored forever. The edit information is captured for the last edit and stored only until the next time the same record is edited.

****

Standardized Reports: Descriptive reports compliant with EAD, METS, and other metadata standards, as well as various reports to support management of an archival collection.

Searching: Ability to search for single resources by a resource number, title, creator, or subject; to search for locations of materials to support on site paging of materials for researchers, and to search across all resources for contents satisfying a researcher’s query.

****

F1d: Deliverables:

The following items will be produced as part of the first release of the Archivists' Toolkit:

- An Archivists' Toolkit release that:
  - embodies the features listed above
  - is modular in design with customizable input templates and output formats, and that supports the critical life cycle functions of tracking sources, of accessioning, of description of collections and items, and of location;
  - supports the ingest of legacy EAD (ver. 1.0 and 2002) finding aids and MARC 21 catalog records for archival resources (not for component parts of resources)
  - supports output for EAD, METS, Dublin Core, and OAI Harvesting;
  - is delivered as an open source application; and
  - is released for Linux, Mac OS X and Windows servers in online and in media formats

- Documentation sufficient for installing and using the application;

- A software specification describing all entities, relationships, and procedures in the prototype, including plans for future development and / or needs for modification and maintenance;

- Data acquired from partners for building and testing the prototype;

- An evaluative project report indicating areas of success and areas needing additional work;

- A final financial report

****

F-2: User Characteristics

A: Profiles

Users of the Archivists' Toolkit will assume a variety of roles, some of which are delineated below. These roles may be represented by different staff members working together in a single institution, or may be embodied in a single archivist working as a "lone arranger."

Generally speaking, archival workers are trained for archives management and not for managing the operation of computer machines and networks. Thus, many archival workers may require technical assistance to set up a network or install a database application. Given this fact, it is important to strive to make the AT application as easy to install and use as possible.

Archival management

The archival manager is responsible for the archives' entire operation. She is interested in facilitating the productivity of her staff, of being able to respond quickly to demands from her administrators, and in maintaining confidentiality and security of information contained in the archival collections and the descriptions of them. Accessing information quickly and easily is a primary concern, whether that information is in the form of statistical reports or in the form of a list of accessions given by a specific donor. In terms of technical experience, the manager may be a novice or an advanced user.

System administration

The system administrator oversees the installation and maintenance of the application, and troubleshoots any problems. She may be a technically savvy member of the repository staff or a member of an external IT department, and therefore may or may not interact with the application on a day to day basis. Her technical skills are generally in high demand, and she does not have a great deal of time to troubleshoot and apply patches to the application. In general, the application should be designed in a manner that minimizes as much as possible the need for system administration support, as many archival repositories may be lacking these kinds of skills.

Project management

The project manager oversees and participates in processing operations. She establishes parameters for data entry, and is concerned with producing a variety of descriptive products, from accession records to EAD encoded finding aids. In recent years, the demand to produce a variety of descriptive products (paper finding aids, MARC, EAD, METS) as well as digital surrogates has created additional layers of work to the task of arranging and describing the repository's holdings. At the same time, interest in reducing backlogs is driving repositories to process more quickly. The project manager's technical skills range from mid-level to advanced user.

Data entry

The data entry staff enters data values at all levels of description. She is responsible for accurate and timely entry of information into the database. She may be involved in processing activities, or may only be responsible for inputting data gathered or composed by others. She may produce descriptive reports and finding aids, but these will be reviewed and approved by the project manager or another staff member. The data inputter's technical skills could range from basic computer literacy to advanced use, and she may or may not have archival experience. She may be an archivist, a paraprofessional, a student, or a volunteer. A fundamental objective of the AT application is to allow for a greater amount of data entry work to be done by lower staff levels, e.g., students in an academic setting.

Reference service

The reference archivist assists researchers in identifying, accessing, and using relevant materials from the repository's holdings. In this capacity, she may utilize descriptive reports output by the system or search the system directly. The need to access a variety of legacy systems in order to cover all bases determines her ability to respond quickly to the needs of researchers. She is responsible for pulling materials for patrons, and therefore requires accurate and easy to retrieve location information. Her technical skills may range from novice to advanced user.

The five roles are to be supported by three user interfaces provided in the application: 1) Application / Processing Project Interface (archival management, project management, system administration), 2) Data entry (data inputting), and 3) Search (reference).

****

B: Working environment

The Archivists' Toolkit will be designed to support a range of working environments, primarily focusing on small to medium sized repositories with one to a dozen employees. Managing and processing archival materials can be an ongoing task, where information in even "fully processed" collections may need to be updated or amplified, for example due to accretions or digitization. Accessioning and processing a group of materials may involve a single person, or up to five different staff members. In some cases, the repository may be part of a consortium that shares electronic resources.

Currently, most of these repositories are using a variety of different systems to support the range of archival tasks. For example, accessions information may be kept in a database while finding aids are produced with word processing software. Both current and legacy systems may also include analog tools such as card catalogs and paper files. Windows operating systems are most frequently used, though some repositories also use Macs. Occasionally the Linux operating system may be deployed in a respository.

C: Needs

Archival management

- Inexpensive, dependable, and easy to use system.
- Quick and easy retrieval of information, including the capability to capture and retrieve information about relationships among donors and materials.
- The ability to customize the system to support existing workflow and maximize efficiency.
- System security that allows control over who has access to various types of data.
- Built in administrative and statistical reports.

System administration

- System that is easy to install and maintain.
- Ease of use with intuitive interface and quick performance, to minimize user complaints.
- Dependability and security against loss of data due to system failure, obsolescence, data entry error, vandalism, or accident.
- Complete and accurate system documentation.

****

Project management

- A system that produces a variety of standardized outputs based on the same data.
- Ability to constrain data entry where desirable. For example, to control access terms and forms of titles and names or to establish required data elements.
- System that is simple to use, thereby reducing training time.
- Interface customizable to the repository's workflow.
- Audit trail for records to track who created and modified the data, and when.

Data entry

- Shortcuts for data entry, facilitating the replication of predictable or repetitive data values.
- Ability to easily access lists of controlled access terms and boilerplate text at the point of need.
- Easy to navigate.
- Built in, context-sensitive help screens and well-organized, easy to read manuals.

Reference service

- Descriptive outputs that are easy to use and don't require specialized training for researchers.
- Ability to quickly and easily search a variety of descriptive information.
- Quick access to location information for pulling and re-shelving of materials.

F3: Product Context of the Application

An environmental scan has not revealed any software applications, commercially or locally developed, available today that include the range of functionality planned for the Archivists' Toolkit.

Commercial Systems:

The Research Library Group sponsored development of a similar application in purpose and scope to the Archivists' Toolkit, the Archives and Museum Information System (AMIS), at end of the 1980s and early 1990s. Though no prototype was produced, the project resulted in substantial important documentation of the planning process, especially process diagrams for the various archival tasks to be supported.

A variety of software packages and processing tools have been developed over the past decade in response to the automation needs of archival work. the ADLIB Archive software, CAIRS-ARC, Calm for Archives, Gallery Systems' TMS, Re:Discovery, PastPerfect, Db/Textworks, and Wizard's MAVIS system. (An extensive survey of additional collection and processing applications for archives and museums is available online at http://www.mda.org.uk/softover.htm.)

As good as some of these applications are at some things, all are seriously wanting in one or more significant ways at the time this specification was written. Some, such as Gallery Systems' TMS, are far too expensive for smaller archives to contemplate adopting as a solution. Others lack support for the full range of functionality needed by many archives, or do not support community-based descriptive standards. ADLIB Archive, CAIRS-ARC, and Calm for Archives, for example, lack support for appraisal and de-accessioning. PastPerfect, Re:Discovery, and Db/Text do not have built in capabilities for EAD export. Some tools, meanwhile, do not support collection management adequately, choosing instead to focus exclusively on end-user access.

In addition to issues of cost and functionality, there is a more fundamental problem with the existing commercial solutions: many are proprietary or closed-source products. This is true for many "home grown" solutions, as well, inasmuch as they are based on proprietary database management applications and other software programs. As proponents of open source software have noted elsewhere, the closed source nature of commercial off-the-shelf software limits end-users' ability to customize the product for their local environment, and also raises doubts about the long-term sustainability of the software.

****

Locally developed systems

Lacking commercial alternatives, many repositories have elected to develop their own system. Some of these are database applications, but many are not, relying on the use of macros and other kinds of strategies for automating archival work. These systems are important for helping to solve certain kinds of problems and reveal effective strategies. But more often than not, locally developed systems are oriented to serving the immediate needs of the repository in which they are developed, they are often singular in purpose and do not integrate or support a range of archival functions, and, finally, they are not designed to promote all the standards pertinent to describing archives.

This document is not the place to review locally developed systems in detail. Here, however, is a list of several systems that are important because they have influenced the specification of the Archivists' Toolkit to some degree or another.

ARIADNE: A relational database application developed using FoxPro database software. The database was developed to support processing work in UCSD's Mandeville Special Collection Library. It was particularly successful in supporting EAD production. The application, although still in use, has not been updated and is targeted for replacement by the Archivists' Toolkit. Contact: Steve Coy or Brad Westbrook, UCSD Libraries.

DAMD: A greatly modified version of GenDB built using FileMakerPro database software. The application was partly modified to support the archival information in a museum setting. The development team has secured an IMLS grant to fund developing of the tool for general deployment among museums. Contact: Rick Rinehart, Berkeley Museum of Art.

DIANA An Access database application developed by the Philadelphia Museum of Art. The application supported delivery of EAD and METS, as well as a broad range of management functions. This may be one of the better, more complete precursor applications for the Archivists' Toolkit.

GenDB: Developed in the libraries at the University of California, Berkeley. Envisioned as a EAD and MOA2 (METS) production tool, it has yet to have been implemented to support EAD production, but it has been generally employed for METS production. It is the one tool known to have a Web based interface. Contact: Mary Elings, UC Berkeley Libraries.

****

KB: A database strategy deployed at UCSD and the San Diego Super Computer Center. The strategy is based on an attribute / value system for presenting metadata in RDF-like manner. UCSD has been using KB for a range of database services. It has experimented with moving the ARIADNE database to KB. The results of that experiment are not yet known. Contact: Chris Frymann, UCSD Libraries.

MySQL: UC Davis developed a MySQL database application to support the description of archives and the production of EAD finding aids. The effort had limited success, as curators were reported to be reluctant to use it and library administration ceased supporting its development. Contact: Jared Campbell, UC Davis Libraries.

ZeroDB An application developed by NYU to support archival processing in its three institutional repositories. Contact: Brian Stevens, NYU Libraries.

****

F4: Tasks Supported by the Archivists' Toolkit

0.0 Prerequisites

1.0 Installation

2.0 Application / Project Management

3.0 Manage Records

4.0 Audit Trail

5.0 Record Navigation

6.0 Generate Report

7.0 Search Database

The application elements (functional areas, inputs, outputs, etc) are referenced in parentheses after the primary task description, e.g., "(I-A-Ia) refers to the first accession input element.

****

0.0 Prerequisites for installing the Archivists' Toolkit application

0.1 Operating System:

Windows 2000 or more recent version
MAC OS X
Linux Red Hat

0.2: Download and successfully install any ancillary software, such as the MySQL database server needed to run MySQL.

****

1.0 Installing the AT application

Download or acquire and then install appropriate version for local operating system (0.1) and network capability. Versions will be available on a server via ftp and via media (e.g., CD) that can be requested from the product team

Installation interface will give users choice between the two following installation types:

1.1: Network version (for either a single repository network or multi-repository network)

1.2: Desktop, non-network version

A newly installed application, network or desktop, will have one predefined account set for the Application Manager.

The application should be a web-based application that, once installed, can have access restricted to a specified user ID / password statements and / or IP addresses. The application will be not be openly available on the web.

Appropriate default security protocols will need to be established by each repository for the network version.

Also, all customizable features (user assignments, input templates, and report forms) will be set to the application standard defaults.

The application should be completely operational following installation and not require modifications of any settings, although it will be highly advisable for a repository using the application to customize several of the default settings.

1.3: Repository Identification Record

The user is prompted on initial install to complete the repository identification record. This can also be done later from project management interface.

This record is a base level record that includes the name of the repository, its address, telephone numbers, and institution code(s). Some of this data is used in the production of access instruments listed in 6.1.

****

The Repository Identification Record also contains information about the format for the institution's accession numbers and resource identification numbers. In each case, the repository has up to four parts that it can customize for use in its accession/resource identification number. Each part can be designated as a string, date, or number. In addition, one of the four parts of the accession number can be designated as a number and set to increment automatically to the next number, providing the operator with the next available accession number when he/she enters a new accession record. Finally, the project manager can also establish standardized punctuation to be used between different parts of the accession/resource identification number. This punctuation won't appear in the data entry screen, but will be used to format these numbers in all outputs. Different punctuation characters can be used in different places in the same number.

1.4: Staff Permissions Record

The user is prompted on initial install to define user permissions; this can also be done later from the project management interface.

If permissions and access levels are not established, then the product defaults shall prevail: the application can be used by anyone and at any level.

1.4.1: Permit access for each staff based on userid and password or IP address.

Userid and password is the protocol for assigning access to the four available levels. Logon identifications are assigned to pre-established permission levels.

Userid also supports the audit function of the application. Staff should be able to access the application from different machines and IP addresses, making userid and password the more flexible access "certificate".

It should also be possible to install "Search" access to machine IP addresses. In such a case, the only accessible interface would be the "Search" interface. Users would not be able to access other parts of the application at that machine unless the complete application is launched and a valid userid and password is provided.

1.4.2: Assign one of four levels of access to each staff user based on userid and password.

Level two and level three assignments should be capable of being specified for each project definition enacted in the application. The level one assignment applies to the whole of the database.

1.4.2:1:Level one: Application management

Can implement the repository defaults. Can manage the contents of the database, such as purging records for marked for deletion. Can ingest records. And can perform all functions at level two and three.

1.4.2.2:Level two: Processing project management

Can implement secondary parameters for processing projects. Can produce all reports. Can perform all data entry functions. Cannot ingest records. Cannot alter repository defaults or modify the contents of the database, i.e., purge records.

1.4.2.3:Level three: Data entry

Restricted wholly to data entry and printing individual records, resource descriptions, and hierarchical lists of records in a given resource. Can navigate through records. Cannot ingest records. Cannot modify applicable project definition.

Data entry access should be capable of being further restricted to read and/or write access for accessions, description, donor, creator, and subjects record types. For example, a permanent staff member may have data entry privileges with read and write access to each of these record types, meaning that they can view and create/edit any accession, description, donor, creator, or subject record. The repository may want to restrict data entry privileges for a student employee, however. For example, the repository may not want to allow read or write access to donor records. In addition, the repository may wish to allow the student to read from the creator and subject records so that they can add these terms to an accession or description record, but may not want that student to be able to add creator names or subject terms. Therefore, the student would have read but not write access to these record types.

1.4.2.4: Level four: Search (Public Service)

Restricted to the search interface for querying the database to support use of a repositories collections. Can be restricted to userid / password or to an IP address for a dedicated machine. (This function is distinct from the "locate record" function to assist project managers and data entry staff to navigate through the application's contents.)

1.4.3: Assign default initial interface

Assigned per IP address or login ID.

The project management interface should be the default initial interface for project managers. This interface should have access to the data entry interface, report generation interface, and search interface.

The data entry interface should be the default initial interface for data entry staff. This interface should have access to selected parts of the report generation interface and to the search interface. It should not have access to the project management interface.

The search interface should be default interface for staff / machines that only access the database to search its contents. The search interface shall not have access to the project management, data entry, or report generation interfaces.

1.5: Launch application

1.5.1: Interface: Opening Screen of Application (Full Access)

Application / Project Management
Create / Edit Record (leads to 1.5.2)
Generate Report
Search Database (leads to 1.5.3)

1.5.2: Interface: Opening Screen of Application (Access Restricted to Data Entry)

Ingest Records (function may be restricted to specified staff)
Create New Record
Edit Record

****

1.5.3: Interface: Opening Screen of Application (Access Restricted to Search)

Resource description
Resource location
Subject search
Cross resource search

****

2.0: Application / Processing project management

Application / Processing project management refers primarily to the ability to customize the “out of the box” defaults to reflect the needs of 1) a consortium, 2) a repository, or 3) distinctive processing projects within a repository. Processing project management functions serve to distinguish processing projects and processing parameters of one repository from another in a multi-repository network installation, and, additionally, processing parameters for one project from another project in either a multi-repository network installation or a single repository network installation. The project management functions in the AT also serve to streamline processing work flow by allowing screens to be tailored to a repository’s work practices, to secure the database against unwanted use, and to enforce entry of consistent and limited data values.

Customization of the “out of the box” defaults is not required to install and use the application. The application will install and be wholly operational without customization of the “out of the box” defaults: unrestricted permissions and generic data entry templates and report forms. Each repository using the application will have the option to accept the defaults issued with the application or to modify most of the defaults to reflect the needs and preferences of the using repository.

The application can be modified or customized in three key respects:

1) Data input templates can be modified.

   The color schemes of the templates can be altered. Any of the elements on the “out of the box” template, except the required fields for that template, can be removed from the template. The sequence and layout of the data elements on the template can be altered.

2) Vocabulary lists for names and subjects can be constrained.

   Names and subjects show up in an A-Z list when a name or subject is being applied to a resource description at any level. Entry of new terms to either list can be restricted to specified individuals. Entry of terms applicable to a given processing project can be restricted. Thus, when applying terms to a resource, only the terms permitted for application to the resource will display in the pick lists for names and subjects.

****

3) Reports can be altered.

   Formatting of the report, such as font type and size, bolding, footers, etc. can be modified. Content of the report can be modified either by altering the sequence in which the content is displayed for printed or by removing some data elements from the report. Titles, headers, and logos that represent local branding can be added to the report.

None of these modifications alter the underlying structure of the application. That structure can not be modified.

Customization of the product defaults amounts to defining a set of processing parameters to govern processing projects. One set of distinct processing parameters is a project definition, and it is represented in the toolkit by a project definition record and associated templates and report forms.

A processing definition can be assigned to one of three levels:

1) The consortial level is the highest level.

   A consortial level means a single implementation of the AT serving two or more distinct repositories. The processing definition governs all processing projects for all repositories participating in the consortium. It is expected that the consortial processing definition would stipulate certain features of outputs, but would likely not address the input templates or vocabularies. For instance, a processing definition for the Five Colleges participants might stipulate that "Five Colleges" appears in all public access instruments (EAD, catalog records, etc.).

2) The repository level is the second level (the highest level in a single implementation serving a single repository).

   The repository level definition is assigned to the repository identifier, its institutional code. The repository definition governs all processing projects for the specified repository. In a consortial implementation, repositories may not share the same processing definition, since it would not then be possible to distinguish the resources of repository A from those of repository B. It is likely that the repository level processing definition would add information to outputs that are specific to the repository. For instance, Smith College might want to stipulate that "Smith College" appears on all if its public access instruments, in addition to whatever the consortium stipulates. In addition, a repository might want to contour the input templates in a manner more congruent with its work practices and tastes, as well as constrain the population and use of the name and subject vocabularies. It is expected that a repository level processing definition would apply to all processing projects for a certain repository, unless a more specific processing definition exists.

3) The project level is the third level (second level in a single implementation serving a single repository).

   The project level processing definition is assigned to the resource ID for an archival resource within a repository setting. The project level processing definition may be assigned to one or more archival resources, Each resource it is assigned to is governed by the processing parameters specified in the project definition. It is expected that the processing level definition will most typically be used for constraining data element choices and vocabulary choices for different classes of archival resources. For instance, a repository may want to distinguish and restrict the kinds of subject headings applicable to its literary collections from those applicable to its university archives collections. But the project level processing definition can also be used to modify for a class of archival resources the appearance of input templates and of outputs. For example, Smith College might want the public access instruments for archival resources in its Women Writers Collection to carry the phrase “Women Writers Collection,” as well as the phrases “Five Colleges” and “Smith College.”

   Project definition records can be created prior to a project commencing and can exist for an indefinite period of time in the application before being assigned to a processing project. Project definitions can only be assigned to consortium members, to a repository, or to archival resources in a single repository. Thus, all accessions for a single archival resource must be governed by the same processing definition. Any modification of a processing definition applies to all archival resources and accessions assigned to the project definition.

The following areas of the application will be customizable at the repository level and then at the project level.

Interface: Application / Project Management Interface

Edit Repository Identification Record
Modify User ID and Password List

****

Create / Edit Project Definitions
Purge Database

2.1: Customize data entry templates

2.1.1: Remove elements from a template

A label and its corresponding input frame are inexorably bound together. Together they represent a data element that can be captured and stored in the database and processed in some fashion.

The product default template will include all data elements possible for any given template.

A repository shall be able to remove from a template any data element for which it does not want to collect data. Removing a label will remove the associated input frame, and vice versa.

The data element is removed from the template only, and not from the underlying tables.

A repository shall not be able to add data elements to those already contained in the product defaults. The underlying tables cannot be augmented.

2.1.2: Relabel data elements

A repository shall be able to modify the label for any default data element it chooses to retain on an input template. For instance, for a file heading, one repository might prefer the term "title," another the term "heading," and another the term "contents."

A label for one data element cannot be associated with a different input frame.

2.1.3: Resequence data elements

The product default will have the all the data elements available to that template sequenced according to the data flow diagram.

A repository can modify the sequence to better reflect its workflow.

Data elements and their labels must be united so that label and element are always moved together.

****

2.1.4: Modify presentation elements

A repository shall be able to modify the color scheme and font sizes for a template.

2.1.5: All modifications to the templates should be executable for repository defaults and for project definitions.

2.2: Specify applicable vocabulary

The default shall be that the pick lists for names as subjects and topics as subjects shall be operable but not constrained to permissible choices. That, is terms can be entered and they will display in the pick lists, but there will be no restrictions against any user entering any additional terms.

These lists, in effect, are comprehensive lists of unique subjects (names and topics) used in an instantiation of the database.

Upon entering a subject term (name or topic) for a resource description, the pick list will open to the term in the list most matching what was typed or is desired. The data entry staff can then select the term from the list to be applied to the resource or finish typing in the new term which will be added to the list once the record is saved and then available for subsequent use.

In some situations, it will be desirable to constrain the list of applicable terms, allowing only specified terms to be applied to certain resources and / or allowing only certain staff members to add terms to the name and subject lists. Typically, this will be done for project definitions where only a subset of terms is required for the descriptive work.

2.2.1: Restrict the names that can be applied as subjects to the description of a resource within a project definition. Permissible values are to be listed alphabetically in a pick list

2.2.2: Restrict the topical terms that can be applied as subject to the description of a resource within a project definition. Permissible values are to be listed alphabetically in a pick list.

2.3: Modify report forms

Default report forms will be capable of being modified in three ways, presumably through the use of XSLT modification tools.

****

2.3.1: Change the presentation of the report including background and font colors, page orientation, and local branding.

2.3.2: Change the layout and sequence of data elements

2.3.3: Remove elements from the report

2.4: Identify errant, duplicate, and deleted database records (all types) and purge from the database.

****

3.0 Manage Records (e.g., created, edit, navigate, delete)

The key purpose of the application is to facilitate the description of archival resources in a manner that satisfies key archival functions.

Interface: Opening Choices

Ingest Records (function may be restricted to specified staff)
Create New Record
Edit Record

3.1: Ingest Records

Ingest should be supported either file by file (providing a file identifier) or by batches (providing a directory identifier for a directory containing files all of the same type).

The ingest function should be capable of being restricted to staff having highest level of access to the application.

The ingest function is governed by source / target data maps built into the application. These mappings are observable but cannot be modified.

3.1.1: MARC 21 parent records (mapped to resource description records)

3.1.2: EAD finding aids (v. 1 or 2002; mapped to corresponding levels of descriptive resource records) expressed as XML

3.1.3: Other archival information (accession database, e.g.)

It is not certain release 1 will be have this functionality

3.1.4: Ingest report

Any ingest of a record results in the generation of the following reports

3.1.4a:Error report. **(O-I-1)**

XML file of errors for why a record could not be ingested. The report is printed to screen and stored in a directory, accessible by an external text editor.

3.1.4.b:Completion report. **(O-I-2)**

XML file indicating the type of record(s) ingested, the total number of records ingested and the number of corresponding records created in the application, broken down by hierarchical level.

3.2: Create a Record

Create a new record to be created from the various record types permissible. A record supports a certain archival function or part of a function, and, in the AT, it is best represented by an input template. The input template indicates what values are permissible.

The AT application contains the following types of records: accession, location, name, subject, archival resource description (resource record, series/subseries record, file record, item record), surrogate description (surrogate record, child record, sibling record), project definition record, and repository id record. In practice, some of these record types are very similar, often distinguished only by a level or type characterization, whereas other record types are widely divergent.

Selecting a record type loads an empty template for that record type. The loaded template reflects customized presentation, label, sequence, and constrained data values, if and only if they have been established.

A new record should inherit some of the data values from the context in which it is opened. For instance, a sub-series record opened in the context of a resource description should inherit values such as resource id, series title, etc. And a new top level resource description record should inherit some information from the related accession records, should they exist.

A new record should be validated for unique assignment of values within its context. For example, a sub-series record should have a unique sub-series identifier for its context.

All records having the same resource ID should automatically be related or linked together.

3.2.1: Create an accession record (linked to resource record by resource ID, many to one) **(F-A-1)**

3.2.2: Create a location record (linked to resource record by resource ID, or in absence of that, to accession record, many to one.). **(F-L-1)**

3.2.3 Create a name record (linkable to resource ID, to resource componet record, or to accession record, one to none or one to many.) **(F-N-1)**

Three types of name records are supported, each determined by the primary role type of the name.

The most basic type, and probably most frequent, is the name record role equals subject. This record can be linked to an accession record, any level of resource description record, and any level of surrogate record. The data elements are sufficient for providing a name subject heading and indicating if it's a personal or corporate name and the authority of the construction, e.g. LCNAF.

The second type is the name record role equals creator. This name record can be linked to an accession record, any level of resource record, and any level of surrogate record. Typically, the name record role equals creator will be applied to top level resource records and less frequently to series and sub-series records and other component records. The name record role equals creator includes additional data elements for biographical / historical data. The name record role equals creator should be capable of being output into an EAC record.

The third type of name is role equals source. This name record can be linked to accession records and ultimately to the top level resource record. The record identifies the person or agency responsible for "transferring" the resource(s) to the repository; identifying elements include name, address, etc.. When a contact person is designated, the record will include the name of a contact person, as well as the means for contacting the person.

In some cases, the source and creator of the resource(s) will be same, and that name will also be a subject for the resource.

The same name record can be applied to multiple resources.

The typical process should be to

A: choose option to create a name record

B: chose name record from index of stored names or created new name record

C: chose role from stored roles for the name or assign additional role (only three roles are possible)

D: establish link to appropriate record.

In sum, each unique name in the database can have only one of three roles assigned to it. However, each name record can be related multiple times to accession, resource, and surrogate records.

3.2.4 Create a subject record (linkable to resource ID, to resource component record, or to accession record, one to none or one to many) **(F-S-1)**

The subject record should indicate the type of subject heading (topic, geographic, genre) and the authoritative source for the term (AAT, LCSH, local, etc.).

3.2.5: Create a descriptive resource record **(F-Dr-1)**.

A resource record is the essential record for identifying an archival entity that is fully processed and described. There may be only one resource record per archival entity, and each resource record must have a unique resource id for that entity.

A resource record can be set for an item or for a collection . Item records can be subsequently rearranged into a collection.

A resource record for a collection may have linked to it multiple child records delineating the component parts (subdivision(s), file, item) of a collection. Component records may not be created in the application unless linked to an existing resource record for a collection.

3.2.5.1: Create a subdivision record (must be linked to a resource record only)

3.2.5.2: Create a lower subdivision record (extends for nine levels; must be linked to subdivision record. All subdivision records must resolve to a subdivision record and then to a resource record for a collection

3.2.5.3: Create a file record (must be linked to either a subdivision, or resource record for a collection.)

3.2.5.4: Create an item record (can link to a file, subdivision, or resource record for a collection.)

****

3.2.6: Create a surrogate record (must be linked to a resource description record or component record of any level). **(F-Ds-1)**

A surrogate record is for an object, digital or analog, that is captured in a different format than the item described in the resource description.

The surrogate record contains descriptive metadata for the surrogate, and that descriptive metadata may be more or less granular than the descriptive metadata for the object described in the resource or component record. There will be an option to inherit in the surrogate record the descriptive metadata as it appears in the resource or component record.

The surrogate record also contains technical, rights, and structural metadata for the surrogate and its component parts, if such exist.

There must be one top level surrogate record.

Each top level record can have multiple child records.

Each component surrogate record can have multiple child records.

Each component surrogate record can have multiple sibling records.

The relationship can be expressed as being physical (pages) and / or logical (date entries). The expression of relationships for a set of surrogate records constitutes the structural map(s) for that surrogate.

3.2.6.1: Create a component surrogate record (linkable to surrogate record)

3.2.6.2: Create a component record for surrogate component record (next level down; linkable to next highest component record)

3.2.6.3: Create a sibling record for a component surrogate record (same level, links to next highest child record)

3.2.6.4: Create a file identification for a component surrogate record

3.2.6.5: Create core technical metadata for a component surrogate record

3.2.6.6: Create extended technical metadator a component surrogate record

****

3.3: Modify hierarchy (see 5.2.3 below)

3.4: Edit a record **(F-A-1,F-L-1, F-N-1, F-Dr-1, F-S-1, F-Ds-1)**

Any record of any type should be capable of being located by a primary element such as resource id, accession number, creator, or title.

The current values of the record should load into the appropriate record template.

Each field value stored as part of that record should be capable of being modified.

The modified record can then be saved.

3.5: Add additional fields to a project record definition (enabled for permission levels one and two only, and not for permission level three)

3.6: Save records **(P-O-2)**

Each record type should have a save function associated with it.

Selecting the save function first determines if the fields required (per the product defaults, the consortial defaults, the repository defaults, or the project defaults) have been completed.

If a required field is not completed, a message is sent to the screen indicating that the record cannot be saved and listing / or highlighting the fields that need to be completed.

If required fields are all completed, the entry person is asked if they want to save the record (or to save the changes in the case of a previously saved but revised record).

If user responds affirmatively, the record is saved, and the user is prompted to either start a new record of the same type or to return to the opening create / edit record menu.

As part of the save record process, the audit fields are completed or revised depending on if the record is a new record or a revised record. (See 4.0 below)

3.7: Cancel records **(P-O-3)**

Each record type (accession, location, resource, etc.) should have a cancel record function associated with it.

The cancel function should be operative after a new record has been started or after a previous record is modified.

Selecting the cancel function returns a prompt asking if the entry person wants to clear the record or remove the changes (if it is for a revised record).

Answering yes returns an empty record template in the case of a new record and, in the case of the revised record, the record as it was when first loaded for editing.

Answering no returns the record as it was before the cancel function was selected.

3.8: Print records (print screen) **(P-O-5)**

Every record type should have a print record function associated with it. The record should be capable of being printed without data values present and at any time after data values have been added to part or all of the record.

Selecting the print record function prints to the default printer the record as it appears on the screen

3.9 Add link from record to other record **(P-O-6)**

Some records can be linked to each other. For instance, an accession record should ultimately be linked to the resource record to which it belongs. Also name records and subject records can be linked to accession records, resource records and component records, and surrogate records.

Choosing to link a record should produce a window listing all the records that can be linked. For instance, if choosing to link a name record, a list of available names would be shown. The operator would select the name from the list and a link would be created from the “home” record to the name record.

Resource records should be listed by resource ID and title, names by name in alphabetical order, and subjects by subject in alphabetical order.

****

The list window should also include a command button for creating a new type of record of the kind represented in the window (i.e., resource record, name record, subject record). If the operator does not find the desired value in the list, he/she would then create a new record.

The application should indicate that the link has been made successfully and the primary value of the linked record (resource id and title, name, subject) should appear in the template of the record to which it is linked. That is, the resource id and title from the resource record to which an accession record is linked should appear on the accession record template.

3.11 Remove link from record to other record **(P-O-7)**

It may be desired to break links at times. This command allows the operator to disassociate or unlink two records.

The operator will select to the remove link command for a kind of record linked to the home record. A window will open listing all the records of that type linked to the home record. For instance, if the command is to unlink a subject record from an accession record, the list would contain all the subject records, by subject heading, linked to the accession record. The operator would then select the record to be disassociated from the accession record and execute the remove link command.

Successful execution of the command should be acknowledged by an application message to the effect that "the link to record nnnn has been broken" and by the removal of the reference to the record from the home record template.

Removing a link between two records does not result in either record being marked for deletion.

4.0: Audit Trail (automated task for every record created in the database, except the repository identification record**) (F-Au-1)**

4.1: Record record id, including institutional code for repository creating the record and department name. **(I-Au-1, I-Au-2, I-Au-3)**

4.2: Record creator and date of creation for each new record (accession, location, name, subject, resource and component, surrogate) saved in the application. **(I-Au-4, I-Au-5)**

4.3: Record name of person who last edited the record and the date the record was last edited in the application. **(I-Au-6, I-Au-7)**

In 4.1-3 the date / time should be in standard yyyy-mm-ddT00:00:00 format. The name should be derived from the roster of application users, based on the login script for the session in which the record was created. The institutional name and department name should be inferred from the user's identity, assuming one user ID is assigned to only one institution / department.

****

5.0: Record Navigation

All records in the application should be capable of being sorted by certain criteria, grouped, and then navigated as a whole.

In addition, at any time during record creation / editing, the data entry person should be able to navigate to the records forming the context in which the record creation / editing is being done. That is, the data entry person should be able to move to the previous or next sibling record, or to the first or last record in the list of sibling records. The data entry person should also be able to move to the record up one level or down one level.

5.1: Sort / locate records **(P-N-1, P-N-2)**

Group records according to some identifying characteristic: resource number, title, creator, etc. and sort by expressed sequence (e.g., hierarchy, numeric, alphabetic).

5.1.1: Move to bottom of record group (last record in file according to sort criteria) **(P-N-3)**

5.1.2: Move to top of record group (first record in file according to sort criteria) **(P-N-4)**

5.1.3: Move to previous sibling record (at same level) **(P-N-5)**

5.1.4: Move to next sibling record (at same level) **(P-N-6)**

5.1.5: Move to parent record (up one level) **(P-N-7)**

5.1.6: Move to child record (down one level) **(P-N-8)**

5.2: Display record hierarchy for navigating the records in a certain context

Whenever the data entry person is working within a resource record hierarchy or a surrogate record hierarchy, a visual representation of the hierarchy should be displayed. The display should represent each record in the group and its hierarchical relationship to every other record (parent, child, sibling).

It is not necessary to display at the same time the hierarchy of a set of resource records and the hierarchy of each attached set of surrogate records.

****

The record in the hierarchical display corresponding to the record open in the window should be highlighted or visually distinguished from other records in the hierarchical display.

5.2.1: Collapse / expand record nodes

5.2.2: Move to other record in group using hierarchical display

Clicking on another record in the hierarchical display opens that record, replacing the record that was previously opened.

The highlight in the hierarchical display shifts to the newly opened record.

5.2.3: Modify hierarchy, with auto re-numbering

5.2.3.1: Use drag and drop function to modify the hierarchy of the record set.

5.2.3.1.1: Allow sequence of nodes to be changed (e.g., change sequence of subdivisions).

5.2.3.1.2: Allow a node to be related to a different node (e.g., reassign subdivision to a different subdivision) .

5.2.3.1.3: Allow partial contents of a node to be reassigned to a different node (e.g. reassign some of the items for a particular subdivision to a different subdivision).

6.0 Generate a Report

The basic purpose of the application is to provide outputs that simplify the task of building access instruments for a repository's archival resources and that enhance management of those resources.

All reports ship with a default format. Repositories implementing the AT application will be able to customize that format in several key respects:

A: Modify presentation formatting, including background, font size and colors, page orientation, and local branding (headers and footers).

B: Remove data elements from the output

C: Change the layout and sequence of data elements

6.1: Repository guides

An overview of a repository's holdings

6.1.1:Collections Guide **(D-O-12)**

A list of all the collections, typically sorted by resource ID or creator / title. Includes resource ID, creator's name, title, dates, physical extent, and abstract (if extant). Subject descriptors might also be added to the description.

6.1.2:Brief subject guide **(O-S-4)**

A list of each unique subject descriptor (name, topic, geographic, genre) used in the database with a list of all the resources to which the descriptor is applied. Resource entry includes resource ID, creator name, and title.

6.2: Finding aid

A finding aid is a complete, multi-level description of an archival aggregation. In the AT application, the multi-level description conforms to the ISAD-G standard. Repositories should be able to produce two kinds of finding aids:

6.2.1: EAD finding aid (EAD 2002 compliant) **(D-O-6)**

6.2.2: Printed finding aid (stored outside the application, but viewable using the print preview function) **(D-O-7)**

6.3: Catalog record

A catalog record is a brief description of a resource, composed according to a certain data format, that is then searchable with other catalog records of the same data format.

6.3.1: MARC XML record **(D-O-8)**

Not a full or minimum level standard MARC XML record. Rather an incomplete record for resource or surrogate description, focusing on the 1xx-7xx MARC fields primarily The MARC XML record will need to be completed outside of the AT application for it to be operational within the local MARC database.

6.3.2: MODS record **(D-O-11)**

A top level resource record or surrogate record expressed in the MODS XML format. Again, this record is focused on the top level resource record or surrogate record.

6.3.3: Dublin Core record **(D-O-9)**

A top level resource record or surrogate record expressed in the Dublin Core format

6.3.4: OAI record **(D-O-13)**

An abbreviated version of the Dublin Core record, intended to support OAI harvesting of a finding aid or a surrogate.

6.3.5: METS record **(D-O-10)**

A METS instance wedding descriptive and structural metadata for a surrogate with a file reference to the surrogate. METS will be produced in two varieties:

6.3.5.1: METS record with MODS descriptive metadata

6.3.5.2: METS record with Dublin Core descriptive metadata

6.3.6: EAC archival authority record **(O-N-1)**

****

6.4: Single event management report

6.4.1: Receipt for accession / resource **(O-A-2)**

6.4.2: Complete accession record **(O-A-1)**

6.4.3:Accession processing plan **(O-A-3)**

6.4.4: Donor record **(O-N-2)**

6.4.5: Location record(s) for a specified resource **(O-L-1)**

6.5: Cumulative management reports

A cumulative report is a list of entities matching a certain criterion or a summary of certain kinds of activities over a given period of time.

6.5.1 Repository production report **(O-A-4)**

The total number of accessions acquired, processed, and cataloged during a specified time period, as well as the total extent represented by the production. Time increments should be calibrated to the month.

6.5.2: A list of accessions and their total physical extent acquired in a specified period of time. **(O-A-5)**

6.5.3: A list of accessions processed, and their total physical extent, in a specified period of time **(O-A-6)**

6.5.4: A list of resources or surrogates that are restricted. **(D-O-16)**

The list includes resource / surrogate id, creator, title, date of resource, and the terms of the restriction.

6.5.5: A list of un-cataloged resources

The list includes resource id and title

6.5.6: A list of resources not having completed "property" agreements. A property agreement is a document (transfer statement, purchase agreement, gift agreement, deposit agreement) indicating that material property and the associated rights rightfully belong to the repository.

****

6.5.7: A list of resources and their intellectual property status.

The list should be sorted first by whether the intellectual property 1) has been acquired by the repository, 2) remains with a known third party, 3) is in the public domain, or 4) is unknown. The resources should be listed by either resource ID or alphabetically by title within each section.

6.5.8: A list of all sources of archival resources in alphabetical order **(O-N-3)**

6.5.9: Shelf list: a list of stack locations, in stack order, that have resources located there **(O-L-2)**

6.5.10:Resource location: a list of resources, in resource id order, and their stack locations (the opposite of 6.2.2.8) **(O-L-3)**

6.5.11: Repository collection profile: a summary statement of the total number of resources, the total physical extent represented by those resources, the total number of catalog records reflected in the database, etc.

6.6: Application report

A report reflecting some aspect of the application that is useful for operation of the application in the context of the archival repository.

6.6.1: Application profile

A report indicating the total number of records in the application, with a break out of number of records per record type (e.g., accession records, name records, etc.), the registered users for the application, and the total disk space occupied by the collection

6.6.2: List of registered users and permissions

6.6.3: Database growth report

List of total number of records and number of records per record type added to database during a specified period of time. Time should be capable of being expressed as month/year, that is December 2002 to November 2003. Supplements repository profile.

****

7.0 Search (Public Service)

The Archivists' Toolkit is designed to support archival processing. But with little effort it can serve a modest public service function; it can be searched by a repository's public service staff to identify resources and component parts of resources that onsite users might wish to consult. In some instances, the results of such searches will be more useful than those of MARC or EAD databases. For example, neither MARC catalogs or EAD database currently allow for cross-collection browse lists, that is a list of all folder or item headings in all collections matching a search query.

Typically such searches would be mediated by public service staff. But it is quite possible and reasonable that a researcher could conduct the search without staff mediation and without risk to the contents of the database.

The application should initialize to a search interface on machines where the application is installed or accessed for the sole reason of searching. Access permissions should be based on the IP address of the machine and not the userid / passwords of users.

The search interface should not allow entry into the data entry or project management interfaces, or even indicate such functions exist.

Interface: Search (restricted to only search functions)

Resource description
Resource location
Subject search
Cross resource search

7.1: Resource description search

Search for a known resource using either the resource ID, the creator name, or the title. Display complete description for the resource. This will probably look like an online finding aid in the case of aggregated resources and like a MARC record for item resources.

7.2: Resource location search

Search for the location of a known / requested resource using either the resource ID, the creator name, or the title. Allow for searching by creator or title keyword, thereby retrieving a browse list of matching resources. Display is a list of all the locations for a selected resource.

7.3: Subject search

Search for all resources to which a certain subject descriptor (name, topic, geographic term, genre, project name) has been applied. Selecting an item in the list should open the resource description display in 7.1.

7.4: Cross resource search

Search titles in sub-records (series, subseries, file, item) for all resources. Allow scoping by subject term and use of keywords. Display should be a browse list of all matching titles, sorted in order of resource ID and then appropriate order within the resource hierarchy. Selecting a title in the browse list opens the corresponding record for that title. From that record display, the user should be able to go to the entire resource description display in 7.1.

****

F5: Application's Software System

The application shall be an open source, ACID compliant, web-based database application installable on Windows, MAC OS X, and Linux operating platforms.

Ideally, the application should be easily portable from one DBMS platform to another, as some repositories may prefer to implement the application using a commercial DBMS platform such as Oracle.

Software Specification Team Recommendation

Based on cursory examinations of the MySQL database software (http://dev.mysql.com) and the PostgreSQL (http://www.postgresql.org/) database application, the two more obvious open source database candidates, the Software Specification Team recommends using MySQL for the Archivists' Toolkit application. The Software Specification Team's recommendation is based on

- the language in the project proposal,
- data culled from partner repositories about technical skills present in the repositories, and
- the team's assumptions about the number of users and kind of use to be supported by the application.

According to the project proposal, the Archivists' Toolkit will be used by archives of various collection sizes and staff configurations. Some archives, particularly small archives, may not have dedicated technical stuff, thus making it very important that the application be easy to install, configure to local settings, and maintain over time. MySQL appears to meet these criteria better than its competitor PostgreSQL.

Surveys of the partner repositories revealed that the prospective number of collections the application will have to support varies from 21 to 1200. (How many records comprise a collection varies widely from collection to collection. And most repositories, because the focus is on the finding aid as a unit of description and not on component parts, are unable to determine how many records comprise the descriptions of their collections.)

The Software Specification team projects that the maximum number of collections added to the application in a year would likely not exceed 1000 and would more probably be less than 100. Thus, it is possible to envision an installation of the application in a consortial setting that would have to support a few million records in total and that would have to distinguish the records of one institution from those of another. In addition, the Software Specification team conjectures that the application, were it implemented in a consortial setting, might have to support as many as 50 simultaneous users. But that vision may be extreme. More likely is the need to support 20 simultaneous users.

Finally, the Software Specification team assumes that the repositories will be using relatively current hardware and software. That is to say, the Software Specification team deems it unnecessary to support more than two generations of an operating system (e.g., Windows XP and Windows 98) or machines with less than 128 MB of RAM and 20 GB of fixed disk storage space.

MySQL database satisfies all the conditions listed above. In sum, it:

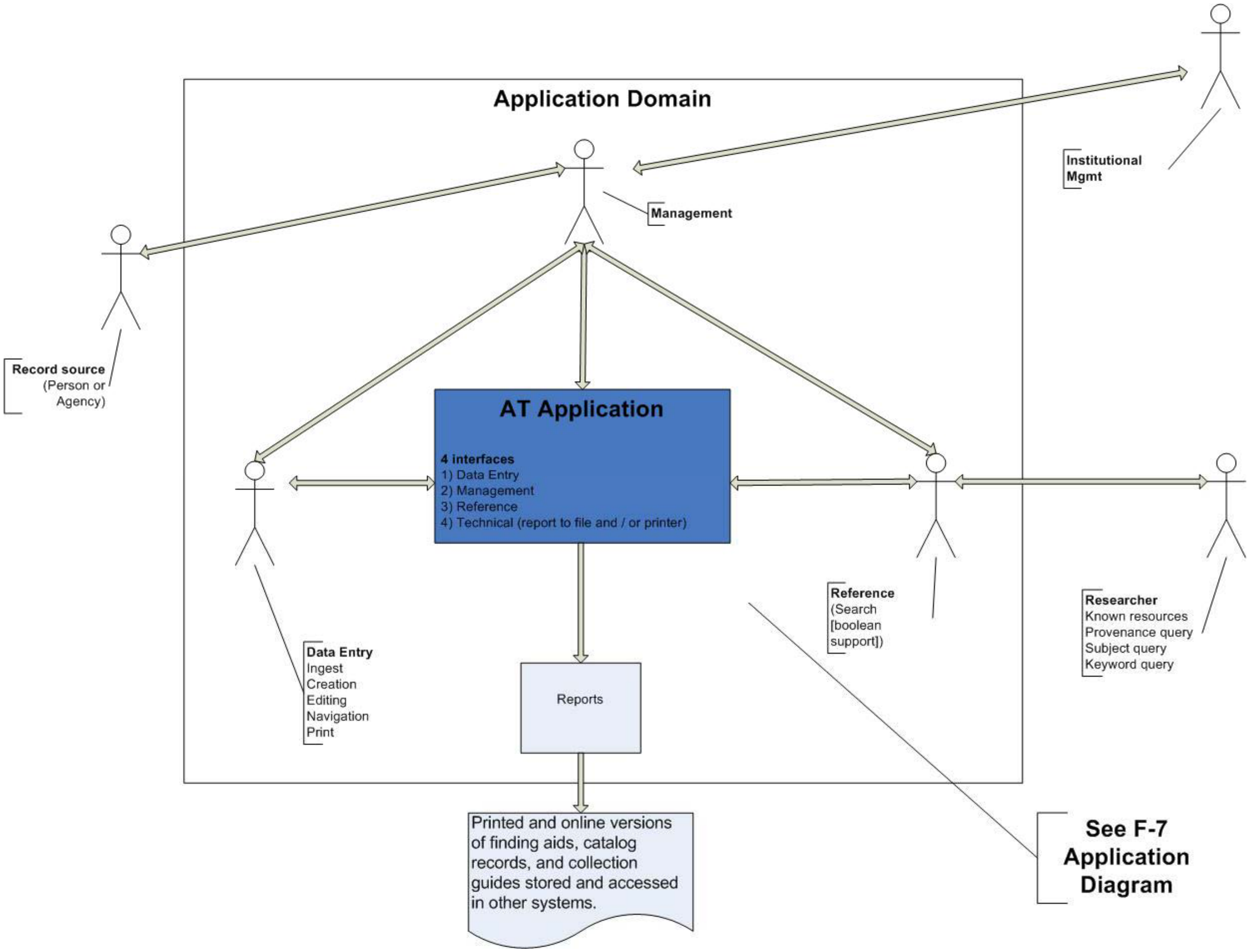
- is open-source with a wide community of users and committers.
- can be easily installed and maintained under any major OS; and
- has a user-friendly administrative interface; and
- moreover, the current version (4.1) has full transaction support, has an advanced row-level locking mechanism, and provides mechanisms to insure data integrity.

In addition, a design tool exists, DBDesigner 4.0 (http://www.fabforce.net/dbdesigner4/) and is optimized for designing and creating MySQL databases. This tool is free.

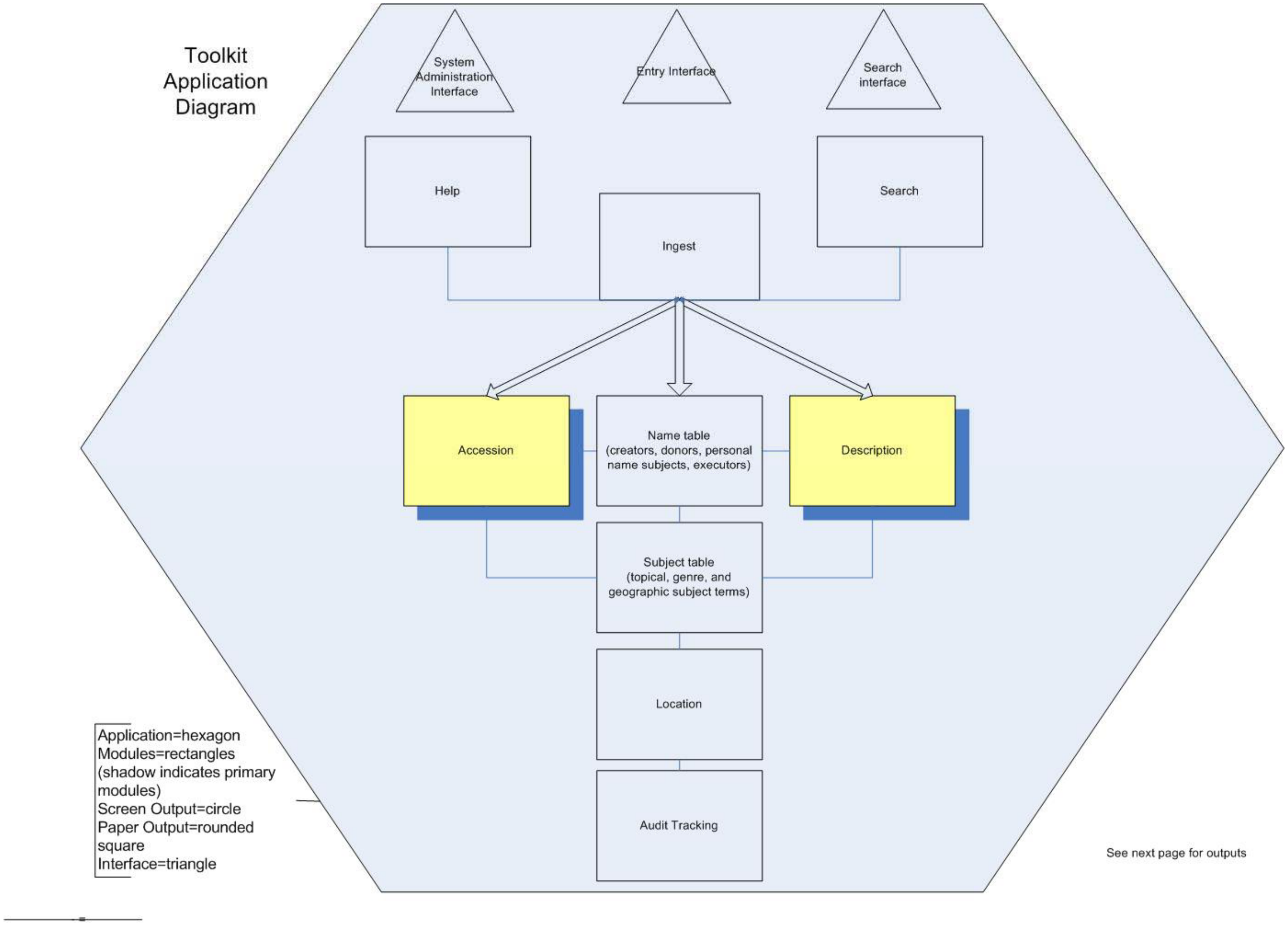
That said, the current 4.1 version of MySQL lacks some useful features, such as views, triggers, and stored procedures. These deficiencies place greater burden on the application server versus the database server, and it decreases the project team's ability to separate different layers of the application. Still, MySQL version 4.1 supports text searching, which might be a more valuable feature and an advantageous tradeoff, keeping in mind the kind of data the application will store and process.

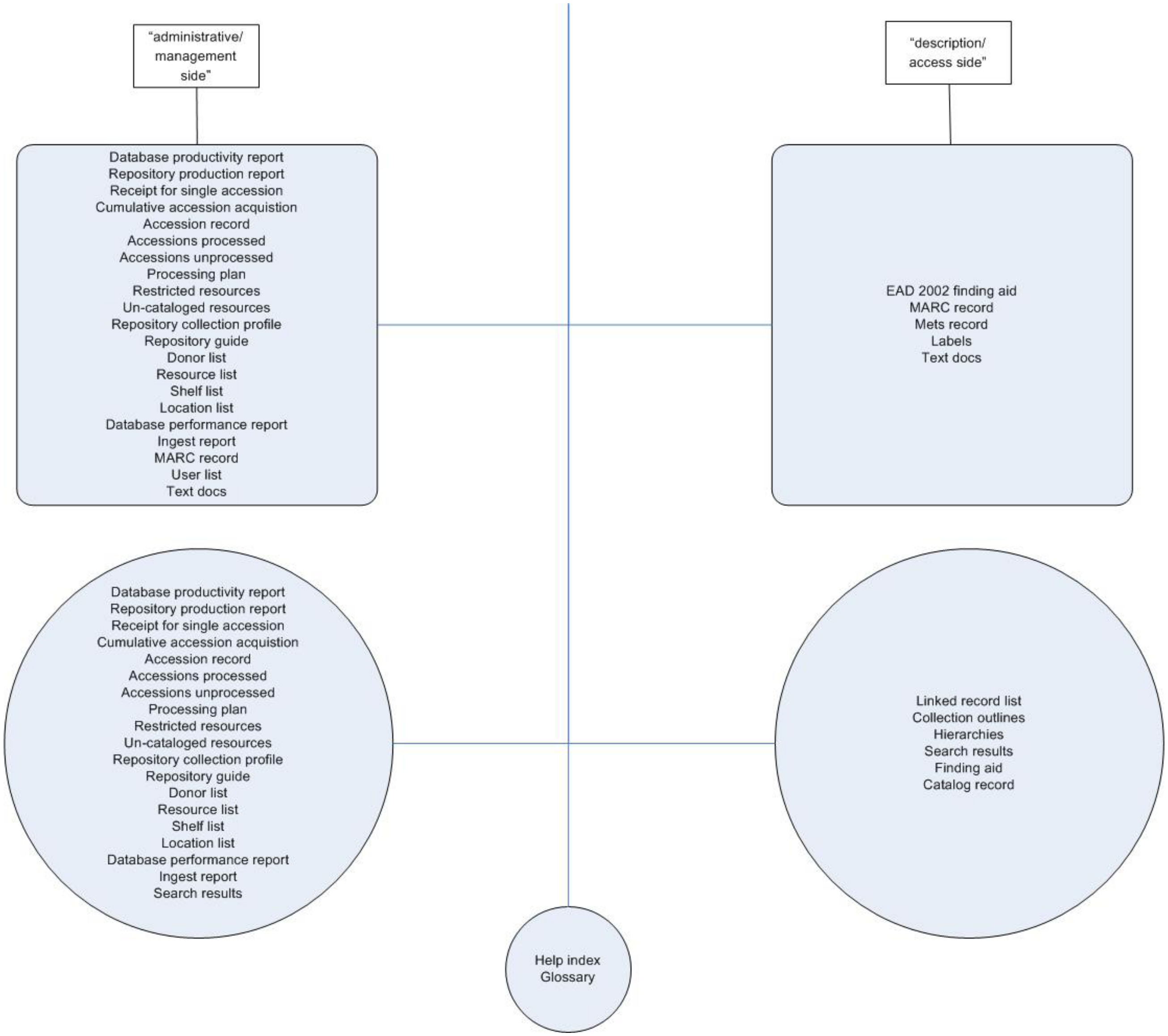
The other open source database software which was also examined for the project is PostgreSQL. Its production version has more features than MySQL, but PostgreSQL is written for Unix-like operating systems. Making it work within the Windows environment requires rather advanced system administration skills, which are presumed to be lacking in many archival repository settings that would use the application. Moreover, the performance of the PostgreSQL software within a Windows system is considerably slower than it is within a Unix-like operating system. Finally, the PostgreSQL Windows version exists only in a beta version as of now. It would be imprudent to base development of a production capable application on beta stage software.

F6: Context Diagram


Application Domain
Institutional Mgmt
Management
Record source (Person or Agency)
AT Application
4 interfaces
1) Data Entry
2) Management
3) Reference
4) Technical (report to file and / or printer)
Data Entry
Ingest
Creation
Editing
Navigation
Print
Reports
Reference (Search [boolean support])
Researcher
Known resources
Provenance query
Subject query
Keyword query
Printed and online versions of finding aids, catalog records, and collection guides stored and accessed in other systems.
See F-7 Application Diagram

F7: Application Diagram


Toolkit Application Diagram
System Administration Interface
Entry Interface
Search interface
Help
Ingest
Search
Accession
Name table (creators, donors, personal name subjects, executors)
Description
Subject table (topical, genre, and geographic subject terms)
Location
Audit Tracking
Application=hexagon
Modules=rectangles
(shadow indicates primary modules)
Screen Output=circle
Paper Output=rounded square
Interface=triangle
See next page for outputs



"administrative/ management side"

Database productivity report
Repository production report
Receipt for single accession
Cumulative accession acquistion
Accession record
Accessions processed
Accessions unprocessed
Processing plan
Restricted resources
Un-cataloged resources
Repository collection profile
Repository guide
Donor list
Resource list
Shelf list
Location list
Database performance report
Ingest report
MARC record
User list
Text docs

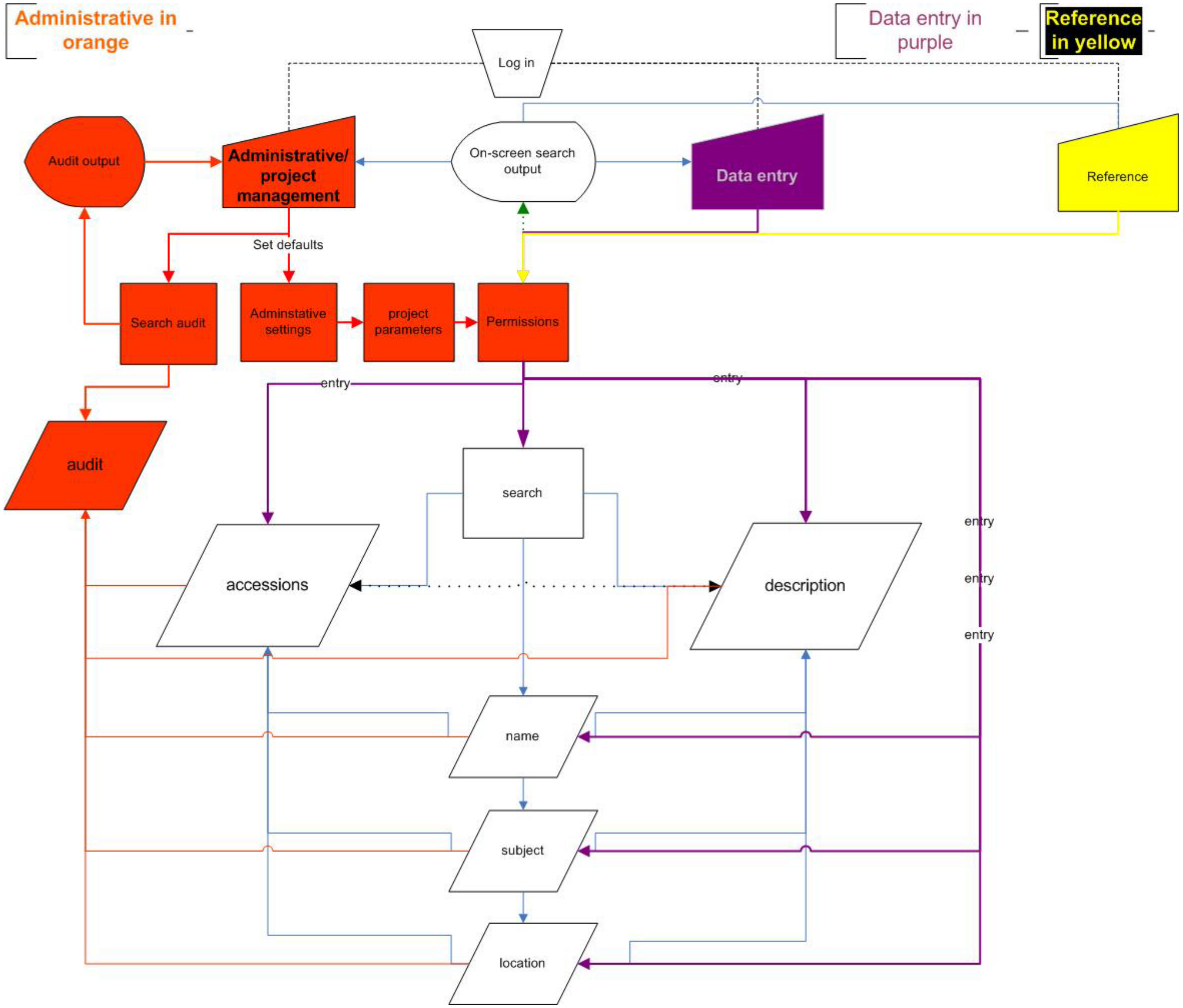
"description/ access side"

EAD 2002 finding aid
MARC record
Mets record
Labels
Text docs

Database productivity report
Repository production report
Receipt for single accession
Cumulative accession acquistion
Accession record
Accessions processed
Accessions unprocessed
Processing plan
Restricted resources
Un-cataloged resources
Repository collection profile
Repository guide
Donor list
Resource list
Shelf list
Location list
Database performance report
Ingest report
Search results

Linked record list
Collection outlines
Hierarchies
Search results
Finding aid
Catalog record

Help index
Glossary


Administrative in orange
Data entry in purple
Reference in yellow
Log in
Audit output
Administrative/ project management
On-screen search output
Data entry
Reference
Set defaults
Search audit
Adminstative settings
project parameters
Permissions
entry
entry
audit
search
accessions
description
entry
entry
entry
name
subject
location

F8: Application's Software System Attributes

Interfaces

There will be three user interfaces.

The first is the application / processing project management interface. This is the "out of the box" default interface. It provides access to the entire application, including the two other user interfaces.

The second is the data entry interface. It can be configured to be the default initial interface. The data entry interface restricts access to the basic data entry functional areas, i.e., creating / editing / deleting accession, resource, name, subject, and location records. Project management functions are not available from the data entry interface and some data entry areas may, by the repository's choice, not be available to all data entry staff. The search interface is available from the data entry interface.

The third is the search interface. It too can be configured to be the default interface of the application. The search interface restricts access to a few basic search queries and query reports. Neither project management nor data entry functions are available from the search interface.

Record management (creation, editing, deletion) and report generation are the two primary functions supported by the application. Project management and searching are lesser functions.

Any record management screen should always have present the operational and navigational functions specified at other points in this document.

The operational functions are (parenthetical references are to the data dictionary): create record (P-O-1), save record (P-O-2), cancel record (P-O-3), delete record (P-O-4), print record (P-O-5), add link to record (P-O-6), and remove link to record (P-O-7).

The navigational functions are: locate record (P-N-1), sort record (P-N-2), last record (P-N-3), first record (P-N-4), previous record (P-N-5), next record (P-N-6), parent record (P-N-7), child record (P-N-8). Sometimes the commands for parent and child record might actually reference the record type that is the parent or child and not the relationship. In other words, from an accession record, the parent record might be referenced as the resource record and not as the more generic "parent" record.

Hardware interfaces:

The application should be integrated to work with a computer's fixed disk for storing database records and reports generated by the application.

The application should be configurable to work with printers connected to the computer.

Software interfaces:

The application should be configurable to interact with the print manager available in the operating system on which the application is installed.

The application need not interact with any other software applications directly, but it should support cutting and pasting across applications, e.g., cutting data from Word documents and pasting it into fields in the application's database records.

Memory:

Support machines with 128MB or more of RAM. (Given the project time of release, it might be reasonable to state this as 512MB or more of RAM.)

Operation

All operations are to be user initiated, using the appropriate user interface and access protocols.

F9: Application's Installation Requirements

Capable of being installed on a Windows (Windows 2000 or XP version), MAC OS X, or Linux operating system.

Configurable to a single desk top installation or to a network (multiple users and multiple repository) installation.

Installs "out of the box" completely operational with only the application's baseline requirements in effect.

Operation requires completion of a repository identification record. Data in this record is used to complete part of the audit function and to provide certain information, e.g., institutional codes, in some of the reports supported by the application.

Access constrainable by userid / password logon scripts or by IP address.

Specification of additional requirements, such as processing project definitions, is to be done by the repository using the forms provided by the application. These additions are not required for the application to be operational.

F10: Application's Performance Requirements

A networked version of the application should support up to 20 simultaneous users.

Users will typically enter alphanumeric data into database fields.

Data entry will be done using the keyboard for the most part. Some data will be cut from another application and pasted into the database field. Some data will also be ingested from legacy sources.

The amount of data to be managed by the application will vary greatly from record to record and, certainly from repository to repository. Some records, such as accession records, resource records, surrogate records, and name records need to support inclusion of large quantities of textual data.

The amount of data added to the application annually will also vary greatly from repository to repository. It is possible to state that an archival repository might add 50 collections to the database annually; however, it is impossible to translate the collections into a usable estimate of database records.

Navigation across records and record management functions should be executed instantaneously.

Report generation should be as quick as possible, depending on the size and complexity of the report.

Searching should be optimized to be as quick as possible. Again the speed of searching will depend on the size of the database to be searched and the complexity of the display for the results.