

System and Software Requirements Definition (SSRD)

Data Mining of Digital Library Usage Data

Team 07

Clients

Jewel Ward

Team Members

Project Manager: Maxim Krivokon
Developer: Bo Lee
Developer: Genesan Kim
Developer: Vu Nguyen
IV&Ver: Shing-Cheung Chan
IV&Ver: Marie Chi
IV&Ver: Kristine Guevara

<Feb 9th 2005>

Version History

Date	Author	Version	Changes made
9/28/04	Maxim Krivokon	1.0	<ul style="list-style-type: none"> Initial draft
9/30/04	Maxim Krivokon	1.1	<ul style="list-style-type: none"> Incorporated client's suggestions Incorporated suggestions of team members
10/01/04	Maxim Krivokon	1.2	<ul style="list-style-type: none"> Added two new prototype screenshots
10/10/04	Maxim Krivokon	1.3	<ul style="list-style-type: none"> Replaced SR-1 Removed PR-6 Added Level Of Service and Evolution requirements
10/24/04	Maxim Krivokon	1.4	<ul style="list-style-type: none"> Added LR-5 = performance scalability as level of service requirement Elaborated off-nominal requirements in terms of availability and dependability Elaborated user front end to the analyzer and visualization modules Changed sections 4 and 6 to table format
11/14/04	Maxim Krivokon	1.5	<ul style="list-style-type: none"> Removed PR-3, Rephrased PR-4, PR-5. Split SR-1, IR-1 Changes are pending client's approval
11/30/04	Maxim Krivokon	1.6	<ul style="list-style-type: none"> Moved PR-3 to PR-8; Added PR-9 Introduced deployment requirement Removed DBA requirement
12/3/04	Maxim Krivokon	1.7	<ul style="list-style-type: none"> Introduced GNU compilation tools requirement Introduced corresponding deployment requirement Merged IR-1 and IR-2 Removed LR-4 Removed ER-2, 3 (because of stand alone architecture)
1/30/05	Bo Lee	2.0	<ul style="list-style-type: none"> [RLCA REVISION] 5. Level of Service (L.O.S.) Requirements revised to be consistent with OCD 4.4 Level of Service (L.O.S.) Goals. (LR-1 – LR-5) Updated Table of tables for Table 32 – 35. Cover credits updated to 577b team. Entry for SR-11: Remove Analysis Report inserted.
2/8/05	Bo Lee	2.0.1	<ul style="list-style-type: none"> [RLCA REVISION] Apache log file -> log file (Digital Archive usage log)
2/9/05	Bo Lee	2.1	<ul style="list-style-type: none"> [RLCA REVISION] Updated References Updated Section 5 Table 33, 35, 36 (from informal peer review)

Table of Contents

SYSTEM AND SOFTWARE REQUIREMENTS DEFINITION (SSRD)	I
VERSION HISTORY	II
TABLE OF CONTENTS	IV
TABLE OF TABLES	V
1. Introduction	1
1.1 Purpose of the System and Software Requirements Definition Document.....	1
1.2 References	2
1.3 Change Summary	3
2. Project Requirements	4
2.1 Budget and Schedule	4
2.2 Development Requirements	5
2.3 Deployment Requirements	10
2.4 Transition Requirements.....	10
2.5 Support Environment Requirements	10
3. Capability Requirements	11
3.1 System Definition.....	11
3.2 System Requirements	12
4. System Interface Requirements.....	18
4.1 User Interface Standards Requirements	18
4.2 Hardware Interface Requirements	20
4.3 Communications Interface Requirements	20
4.4 Other Software Interface Requirements	20
5. Level of Service (L.O.S.) Requirements	21
6. Evolution Requirements.....	25
6.1 Capability Evolution Requirements.....	25
6.2 Interface Evolution Requirements.....	25
6.3 Technology Evolution Requirements	25
6.4 Environment and Workload Evolution Requirements.....	25
6.5 Level of Service Evolution Requirements.....	26
7. Common Definition Language for Requirements.....	27

Table of Tables

<i>Table 1 Table 1 Changes Summary</i>	3
<i>Table 2 PR-1: Limited schedule</i>	4
<i>Table 3 PR-2: No budget</i>	4
<i>Table 4 PR-3: Use MS Project for scheduling and progress tracking</i>	5
<i>Table 5 PR-4: Use GNU compilation tools</i>	5
<i>Table 6 PR-5: Use efficient development language with 3d visualization capabilities</i>	6
<i>Table 7 PR-6: Use UML for modeling and design</i>	6
<i>Table 8 PR-7: Use free components</i>	7
<i>Table 9 PR-8: Target operating system is Mac OS X</i>	7
<i>Table 10 PR-9: Use visualization component that provides interactive 3d interface</i>	8
<i>Table 11 PR-10: Use data storage component that does not require maintenance</i>	8
<i>Table 12 PR-11: Input data format compatibility</i>	9
<i>Table 13 PR-12: Well documented code</i>	9
<i>Table 14 PR-13: Distribution consists of source code and binaries for Mac OS X and Linux</i>	10
<i>Table 15 PR-14: Maintainer training requirements</i>	10
<i>Table 16 SR-1: Import usage data from Log files</i>	12
<i>Table 17 SR-2: Remove usage data imported from the specified log file</i>	12
<i>Table 18 SR-3: Relationship generation</i>	13
<i>Table 19 SR-4: Generate collection structure tree</i>	14
<i>Table 20 SR-5: Graph node statistics</i>	15
<i>Table 21 SR-6: Visualization</i>	15
<i>Table 22 SR-7: Omit mal-formatted retrieval records</i>	16
<i>Table 23 SR-8: Omit usage log file with 0 valid records</i>	16
<i>Table 24 SR-9: Do not create analysis report in case of abortion</i>	16

Table 25 SR-10: Do not visualize analysis report of corrupted format.....17

Table 26 IR-1: Manage usage data18

Table 27 IR-2: Generate new analysis report.....18

Table 28 IR-3: Abort report generation18

Table 29 IR-4: Manage analysis reports19

Table 30 IR-5: Visualize analysis report.....19

Table 31 IR-6: log format support.....20

Table 32 LR-1: System Dependability - Stable data import/export21

Table 33 LR-2 : Usability – User-friendly interface for viewing item relationships and updating data22

Table 34 LR-3: Usability – Maximizing the usability of host resources.....23

Table 35 LR-4: Performance – Organizing data meaningfully for users23

Table 36 LR-5: Performance- data of current scale.....24

Table 37 ER-1: Advanced analysis.....25

Table 38 ER-2: Interface with other usage log formats.....25

Table 39 ER-3: Accommodating collection growth.....25

Table 40 ER-4: Workload growth should affect system performance in linear proportion.....26

1. Introduction

1.1 Purpose of the System and Software Requirements Definition Document

The purpose of this document is to express the common vision among the project stakeholders in form of specific statements on what the Digital Archive (DA) Usage Data-Mining system should be, what it should do, how well it should do it and how it should be implemented and deployed. These specific statements establish clear objectives for future development process and guide the choices of architecture and third party components. For example, user interface requirements that have been developed as a result of prototyping effort and that are described in this document determine the standalone architecture of the system, and C++ as the main development environment. Another purpose of this document is to draw a clear line between optional and required features of the system in terms of functionality and quality of service, thus, defining the scope of this project more specifically. Namely this document distinguishes basic and advanced analysis capabilities that will be developed in two stages given successful transition between them.

This document extends and brings details to the requirements and constraints defined in the Operational Concept Description. As such it defines specifically the system capabilities and features that provide necessary contributions according to the system's Results Chain diagram presented in OCD. For example this document contains interface requirements providing for visualization of Digital Archive collection structure. The requirements defined in this document express agreements reached among the project stakeholders during regular meetings and Win-Win negotiations. Changes introduced to the project requirements by new agreements among stakeholders should be reflected as changes to this document.

1.2 References

“Data Mining of Digital Library Usage Data” Project Description

http://sunset.usc.edu/classes/cs577a_2004/projects/description/project7.htm

MBASE Guidelines version 2.4.1

http://sunset.usc.edu/classes/cs577a_2004/guidelines/MBASE_Guidelines_v2.4.1.pdf

MBASE Version 2.4.1 Templates for SSRD (version 1a)

http://cse.usc.edu/classes/cs577a_2004/guidelines/MBASEtemplates/SSRD_Templatev1a.doc

MBASE Electronic Process Guide

<http://cse.usc.edu/research/MBASE/EPG>

Fall 2004 CS 577a Team 7 Easy Win-win report

http://seacliff.usc.edu/~team7/LCO/EWW_LCO_F04a_T07.doc

Fall 2004 CS 577a Project #7 LCA version of OCD

http://seacliff.usc.edu/~team7/LCO/OCD_LCO_F04a_T07_V5.3

USC Copyright Agreement

http://cse.usc.edu/classes/cs577a_2004/forms/CopyrightAgreementV1.html

LCO ARB Prototype Demo

http://seacliff.usc.edu/~team7/LCO/Prototypes/Prototype_V01.19.zip

LCO ARB Presentation

http://seacliff.usc.edu/~team7/LCO/ARB_LCO_F04_T07.ppt

LCA ARB Presentation

http://seacliff.usc.edu/~team7/LCA/ARB_LCA_F04_T07.ppt

Dr. Bollen, Johan presentations

http://www.cs.odu.edu/~jbollen/presentations/facstaff_02_28_03.pdf

<http://www.cs.odu.edu/~jbollen/presentations/ecdl02.pdf>

<http://www.cs.odu.edu/~jbollen/presentations/aisti04.pdf>

Negotiations with the client

http://seacliff.usc.edu/~team7/CMN/CMN_09_16_F04a_T07.doc

http://seacliff.usc.edu/~team7/CMN/EMAIL_11_23_F04_T07.doc

http://seacliff.usc.edu/~team7/CMN/EMAIL_11_24_F04_T07.doc

1.3 Change Summary

Table 1 Table 1 Changes Summary

Version	Changes Made
1.0	Initial draft.
1.1	Complete the rest of Operational Concept Descriptions (OCD) document
1.2	Added two new prototype screenshots
1.3	Replaced SR-1, Removed PR-6, Added Level Of Service and Evolution requirements
1.4	Added LR-5 = performance scalability as level of service requirement Elaborated off-nominal requirements in terms of availability and dependability Elaborated user front end to the analyzer and visualization modules Changed sections 4 and 6 to table format
1.5	Removed PR-3, Rephrased PR-3, PR-5. Split SR-1, IR-1 Changes are pending client's approval
1.6	Moved PR-3 to PR-8; Added PR-9 Introduced deployment requirement Removed DBA requirement
1.7	Introduced GNU compilation tools requirement Introduced corresponding deployment requirement Merged IR-1 and IR-2 Removed LR-4 Removed ER-2, 3 (because of stand alone architecture)
2.0	Revised 5. Level of Service (L.O.S.) Requirements to be consistent with OCD 4.4 Level of Service (L.O.S.) Goals. (LR-1 – LR-5)

2. Project Requirements

This section contains the mandates which were agreed on during Win-Win and other negotiations with the client and completion of which will determine project's success. These mandates are expressed as constraints on budget, schedule, specific features and capabilities, development and transition processes.

2.1 Budget and Schedule

Table 2 PR-1: Limited schedule

Project requirement:	PR-1: Limited schedule
Description:	Development of LCO and LCA packages has to be completed by the end of the Fall semester. Initial Operational Capability should be finished at the end of the Spring semester.
Measurable:	LCO package is scheduled for delivery on 10/25/2004 and LCA package on 12/06/2004. IOC should be delivered by the middle of May 2005. All should be delivered on time.
Achievable:	The project uses Schedule as Independent Variable process to achieve the deliverables and comply with deadlines. Development proceeds according to a defined schedule with several intermediate deliverables.
Relevant:	Realizes: OCD 2.4.1, PG-1, Win condition 1.1
Specific:	Specifies deliverables and dates
Priority:	High

Table 3 PR-2: No budget

Project requirement:	PR-2: No budget
Description:	The design and development efforts will not be financed by the client and will be accomplished by students for credit.
Measurable:	The final deliverables of the project will be turned in as an assignment for a class and at the same time made available to the client.
Achievable:	Students are motivated to satisfy the customer and complete the project in order to pass the class.
Relevant:	Realizes: OCD 2.4.3, Win condition 1.2
Specific:	Specifies what parts of the project will be affected by the lack of budget and how uncompensated development is motivated
Priority:	High

2.2 Development Requirements

2.2.1 Tools Requirements

Table 4 PR-3: Use MS Project for scheduling and progress tracking

Project requirement:	PR-3: Use MS Project for scheduling and progress tracking
Description:	MS Project should be used for creating a schedule for the project's main tasks, monitoring the project's progress, evaluation of alternative project plans and for tracking of individual performance.
Measurable:	Produced schedules, reports, evaluations should comply with standards / formats established by MS Project documentation
Achievable:	All team members have accomplished MS Project tutorial and a related homework assignment. LCP document already contains planning for inception and elaboration phases that was done using MS Project.
Relevant:	Realizes Win condition 1.6. Using MS Project will provide for more productive project management, and will allow producing documentation artifacts that are compliant with industry accepted standards.
Specific:	Addresses reasons to use MS Project
Priority:	Medium

Table 5 PR-4: Use GNU compilation tools

Project requirement:	PR-4: Use GNU compilation tools
Description:	Configuration and compilation of the system source code should be automated using GNU tools. GNU Make of version 3.80 and up should be used. Use of other tools such as autoconf, automake is optional.
Measurable:	The produced code is compiled by running the corresponding GNU utility program.
Achievable:	All open source components to be used by the system are developed using GNU tools. Several team members have experience using them.
Relevant:	Ensures ease of system installation. Realizes win condition 3.3
Specific:	Addresses how the compilation should be automated
Priority:	Medium

2.2.2 Language Requirements

Table 6 PR-5: Use efficient development language with 3d visualization capabilities

Project requirement:	PR-5: Use efficient development language with 3d visualization capabilities
Description:	Major system capabilities should be implemented using efficient, compiled language that provides extensive 3d visualization and graphical user interface capabilities.
Measurable:	Examples invoking 3d visualization and GUI capabilities of the language should be available. The language should have a compiler that produces binary executable code.
Achievable:	C++ is fast, compiled language that has extensive 3d graphics Application Programming Interfaces (API) (e.g. "OpenGL", "DirectX"). Several team members have development experience in C++. H3viewer visualization library is implemented in C++.
Relevant:	Realizes OCD 2.4.5, 2.4.6.
Specific:	Addresses which features of development language are most important.
Priority:	High

Table 7 PR-6: Use UML for modeling and design

Project requirement:	PR-6: Use UML for modeling and design
Description:	System models and diagrams should be designed in compliance with definitions of UML language.
Measurable:	Produced artifacts can be compared with examples from UML specifications
Achievable:	UML tutorials and specifications are widely available
Relevant:	Realizes Win condition 1.6
Specific:	Addresses the modeling standards that should be followed
Priority:	Medium

2.2.3 Computer Hardware Requirements

The system is a regular user application that does not access hardware directly and does not utilize any special hardware devices. The system will use interfaces for hardware access provided by the host operating system.

2.2.4 Computer Hardware Resource Utilization Requirements

The proposed system will be using hardware resources through interfaces provided by the host operating system and therefore will be limited in the amount of resources it can use. No extraordinary hardware resource usage is expected.

2.2.5 Computer Software Requirements

Table 8 PR-7: Use free components

Project requirement:	PR-7: Use free components
Description:	Third-party software components reused for developing the system must be open source or freeware.
Measurable:	For each third-party component its license should provide for free non-commercial use in distributed or modified form.
Achievable:	Availability of the following open source components has been confirmed: implementations of graph clustering algorithms, graph visualization libraries, RDBMS.
Relevant:	Realizes PG-1 and Win condition 1.2
Specific:	Specifies which functionality can be implemented with open source components.
Priority:	High

Table 9 PR-8: Target operating system is Mac OS X

Project requirement:	PR-8: Target operating system is Mac OS X
Description:	The developed system should be operational on Mac OS X as of version 10.3.6
Measurable:	Developed system provides functionality that satisfies all system requirements when run on the Mac OS X.
Achievable:	Mac OS X is a Unix based operating system and thus provides equivalent compiling and system programming capabilities. Prototyping and evaluation confirmed that identified third-party components are compatible with Unix based systems.
Relevant:	Realizes Win condition 1.3. Realizes additional agreements with the client (see references)
Specific:	Specifies main operating system
Priority:	High

Table 10 PR-9: Use visualization component that provides interactive 3d interface

Project requirement:	PR-9: Use visualization component that provides interactive 3d interface.
Description:	The project should use visualization component that provides interactive 3d graph visualization capabilities.
Measurable:	Examples of three dimensional visual representations of data should be available for the component used by the project. Interface provided by the component accepts user input and dynamically changes visualization behavior based on that.
Achievable:	H3Viewer library provides 3d hyperbolic interactive visualization of arbitrary input graphs. Sample visualization tool based on H3Viewer library was demonstrated at the LCO Architecture Review Board (ARB). Tailored graphical interface using H3Viewer was presented at the LCA ARB.
Relevant:	Realizes: OCD 2.4.5, 2.4.6
Specific:	Specifies requirements for visualization platform to be used
Priority:	High

Table 11 PR-10: Use data storage component that does not require maintenance.

Project requirement:	PR-10: Use data storage that does not require maintenance.
Description:	The system should use data storage component that does not require additional maintenance except regular user input.
Measurable:	System installation does not require additional setup of data storage. The system stores / retrieves data seamlessly during operation not requiring any assistance for those actions.
Achievable:	Data stored in files does not require maintenance, since file system is maintained by the operating system. Database components that do not require database server installation are available (e.g. SQLite).
Relevant:	Ensures ease of system use. Realizes win condition 3.3
Specific:	Addresses which component of the system is critical for usability
Priority:	High

2.2.6 Computer Communication Requirements

The proposed system is a standalone application. All data necessary for performing usage analysis will be provided by the user and will be accessed by the system from the local computer. Thus no communication with other computers will be necessary.

2.2.7 Standards Compliance Requirements

Table 12 PR-11: Input data format compatibility

Project requirement:	PR-11: Input data format compatibility
Description:	The developed system must be able to parse usage logs of the format that is used by the Digital Library web server – .
Measurable:	The system is able to parse usage log file of specified format and store the retrieved data to the system's data storage.
Achievable:	Parsing facilities are widely available in C++ and other languages. The team has access to sample log files.
Relevant:	Realizes PG-2
Specific:	Specifies the format of input data that must be supported
Priority:	High

Table 13 PR-12: Well documented code

Project requirement:	PR-12: Well documented code
Description:	The code developed or tailored by the project team should be reasonably documented to provide for ease of future development and maintenance.
Measurable:	Adequacy of documentation will be determined through customer reviews of each major deliverable.
Achievable:	Customer reviews will ensure proper level of documentation detail
Relevant:	Realizes Win condition 1.7
Specific:	Addresses the mechanism of achieving proper documentation
Priority:	High

2.3 Deployment Requirements

Table 14 PR-13: Distribution consists of source code and binaries for Mac OS X and Linux

Project requirement:	PR-13: Distribution consists of source code and binaries for Mac OS X and Linux
Description:	The system distribution package should include system source that can be configured and compiled on Mac OS X using GNU compilation tools. The distribution should also include two binaries executable on Mac OS X and Linux.
Measurable:	Delivered source code compiles on Mac OS X and produces a functional binary. Distributed binaries are fully functional.
Achievable:	The system will be developed using GNU compilation tools and therefore final development snapshot will be transformed into distribution package.
Relevant:	Ensures ease of system use. Realizes win condition 3.3
Specific:	Addresses contents of distributable package
Priority:	Medium

2.4 Transition Requirements

Table 15 PR-14: Maintainer training requirements

Project requirement:	PR-14: Maintainer training requirements
Description:	Potential maintainer for the system should have experience designing and debugging programs in C++. Also the maintainer should have enough background knowledge about operations of the Digital Archive and rationale behind this project.
Measurable:	C++ experience can be established from previous experience. Background knowledge can be evaluated during interview
Achievable:	Among Digital Archive information technology staff there are employees who have sufficient C++ experience. Archive and project background knowledge can be provided through educational presentations
Relevant:	Realizes Win condition 1.9
Specific:	Specifies requirements for maintainer
Priority:	Medium

2.5 Support Environment Requirements

The system is expected to be stable and will not require extensive support.

3. Capability Requirements

3.1 System Definition

The system defined in this document is a Digital Archive usage data-mining tool with visualization capabilities. It is a standalone application operational with minimal dependencies. The proposed system will have only one class of users and therefore one operational mode. (OCD Figure 9)

The system takes usage log files as input, parses and normalizes usage records contained in the input files and stores the records for future use by main functionality.

The two major modules of the system are:

- analyzer – based on the selected date range and analysis options, applies analysis and statistics heuristics to the usage records and saves the produced results.
- visualization – presents graphically and interactively the analysis results

Analyzer capabilities:

- generation of weighted relations between items based on their co-retrievals
- relative centrality/impact ranking of items
- clustering of items based on their relations
- construction of structure tree based on the clustering and centrality analysis
- collecting usage statistics for each item
- generation of user relationships (pending successful development of item relationship analysis)

Visualization capabilities:

- 3d hyperbolic view of the structure tree
- navigation capabilities within the graph view: rotation, panning
- display usage information associated with the selected item
- display item list for a selected leaf node

3.2 System Requirements

3.2.1 Nominal requirements

Table 16 SR-1: Import usage data from log files

System requirement:	SR-1: Import usage data from log files
Description:	Provided with a usage log file of format specified by the database of Digital Archive (ORACLE). The proposed system should be able to parse the retrieval records contained in the file and store them in normalized form in the system's database.
Measurable:	System is able to insert corresponding retrieval records into the database given a sample usage log file.
Achievable:	Parsing and regular expressions API is available in C++. Normalization is done through id replacement and merging of related records.
Relevant:	Realizes: OCD 2.4.4, Win condition 2.1, SC-3
Specific:	Addresses what kind of normalization is done
Priority:	High

Table 17 SR-2: Remove usage data imported from the specified log file

System requirement:	SR-2: Remove usage data imported from the specified log file
Description:	The system should provide ability to remove retrieval records that were imported previously from the specified file.
Measurable:	After this action search in the database for records contained in the corresponding file should return no results.
Achievable:	Deletion based on a key can be implemented in one SQL query
Relevant:	Realizes: OCD 2.4.4, Win condition 2.1, SC-3
Specific:	Addresses achievability
Priority:	High

Table 18 SR-3: Relationship generation

System requirement:	SR-3: Relationship generation
Description:	<p>Provided that usage log data was properly imported and analysis options were set the analyzer module performs the analysis of relationships among Digital Archive items based on their co-retrievals. As a result the module produces a graph representation of the DA collection with the following features:</p> <ul style="list-style-type: none"> - a node represents a DA item - a link and its weight represents strength of relatedness between two items <p>The produced graph is stored in the database.</p>
Measurable:	Generated relations are stored in the database and can be reviewed in a form of a table dump.
Achievable:	Examples of this analysis are provided in Johan Bollen's papers (See references)
Relevant:	Realizes: Win condition 2.2 , SC-2
Specific:	Describes how item relations are generated
Priority:	High

Table 19 SR-4: Generate collection structure tree

System requirement:	SR-4: Generate collection structure tree
Description:	Based on the detailed relationship graph collection structure tree should be produced using centrality analysis and graph clustering. For each tree level the item with highest centrality rank in the current graph is selected to be the local root. All of remaining items of the current graph are clustered based on strengths of their relations and resulting sub-graphs are made children of the local root. Then the procedure is repeated for each child sub-graph containing more than N nodes. The resulting tree is stored in the database.
Measurable:	After the generation is complete the tree with the following properties should be stored in the database: - the produced collection structure tree has items as regular nodes and sets of items as leaf nodes. Leaf nodes contain up to N items - for each tree level items in different sub-trees on average have weaker relations than items in the same sub-tree. – all parents of an item in the tree have higher centrality rank. -all children of an item in the tree have lower centrality rank.
Achievable:	Markov Cluster Algorithm has been evaluated and successfully used for prototype demo at the LCO ARB. Centrality ranking is done by simple statistic based on item relations.
Relevant:	Realizes SC-2, Win condition 2.3. Ensures visualization / browsing of the item relationship graph
Specific:	Describes hierarchical structure of clustering
Priority:	Medium

Table 20 SR-5: Graph node statistics

System requirement:	SR-5: Graph node statistics
Description:	The following information should be collected and stored for each node (an item or a set of items) of the collection tree: <ul style="list-style-type: none"> - name, ID (for a set: item with highest centrality rank) - list of items contained and their centrality ranks - number of items contained - sum of centrality ranks for items contained - list of users who have retrieved this item (items) and number of retrievals by each user - number of users - total number of retrievals
Measurable:	Given a node id there should exist a query that retrieves the above information from the database where imported usage records are stored.
Achievable:	All of the above information can be retrieved from imported usage records through simple counting and statistics.
Relevant:	Realizes SC-2.
Specific:	Lists information that is collected and stored for each node
Priority:	High

Table 21 SR-6: Visualization

System requirement:	SR-6: Visualization
Description:	Provided that the analysis was done and results stored in the database the visualization module should be able to load them and provide a 3d hyperbolic view of the collection structure tree. The view should provide the following capabilities: <ul style="list-style-type: none"> - navigation of the tree (rotation / panning) - display usage statistics associated with a selected node
Measurable:	The collection structure tree and associated statistics is presented according to specifications and prototype examples
Achievable:	Major visualization capabilities were demonstrated by the prototype at the LCO ARB. Extended graphical user interface allowing display of node statistics was demonstrated by the prototype at the LCA ARB.
Relevant:	Realizes: Win condition 3.1, 3.4, SC-3
Specific:	Specifies the required visualization capabilities
Priority:	High

3.2.2 Off-nominal requirements

Table 22 SR-7: Omit mal-formatted retrieval records

System requirement:	SR-7: Omit mal-formatted retrieval records
Description:	If in provided usage log file a retrieval record is met that does not comply with log format, it should be skipped and noted as such. When data import is done the system should present the user with result summary which specifies the number of skipped records.
Measurable:	Importing log file with n mal-formatted entries should produce result summary with n skipped records.
Achievable:	log format is specified in corresponding documentation
Relevant:	Addresses possible usage log data corruption
Specific:	Details actions of the system
Priority:	High

Table 23 SR-8: Omit usage log file with 0 valid records

System requirement:	SR-8: Omit usage log file with 0 valid records
Description:	If the provided usage log file contains no valid retrieval records, the system should stop the import action and should not create any database records. User should be given corresponding action result summary.
Measurable:	Database is not modified after importing mal-formatted file
Achievable:	log format is specified in corresponding documentation.
Relevant:	Addresses possible usage log data corruption
Specific:	Details actions of the system in case of mal-formatted file
Priority:	Medium

Table 24 SR-9: Do not create analysis report in case of abortion

System requirement:	SR-9: Do not create analysis report in case of abortion
Description:	If the analysis stops prematurely due to user request or some system / algorithm error, then analysis report should not be created and database should not be changed.
Measurable:	If program is aborted during the report generation the data source for visualization module remains unchanged – i.e. list of reports available for visualization stays the same.
Achievable:	Intermediate analysis results could be stored in memory or in a temporary database table, and transferred to original storage when analysis process is finished.
Relevant:	Provides for data consistency
Specific:	Addresses how data consistency is enforced
Priority:	Medium

Table 25 SR-10: Do not visualize analysis report of corrupted format

System requirement:	SR-10: Do not visualize analysis report of corrupted format
Description:	If in the analysis results of the chosen report are of corrupted format then visualization module should stop visualization action and should provide user with corresponding error message.
Measurable:	Visualization stops if corrupted analysis report is chosen
Achievable:	Format of analysis results could be explicitly defined in terms of field data types, data ranges and regular expressions.
Relevant:	Addresses possibility of analysis data corruption
Specific:	Details how format for analysis results could be defined
Priority:	Medium

Table 26-1 SR-11: Remove analysis report

System requirement:	SR-11: Remove analysis report
Description:	If there are one ore more analysis data for a particular analysis report that the system generated before, user should be able to select and remove all the analysis data that was generated for that report.
Measurable:	
Achievable:	
Relevant:	
Specific:	
Priority:	Medium

4. System Interface Requirements

4.1 User Interface Standards Requirements

4.1.1 Graphical User Interface Standards

Table 26 IR-1: Manage usage data

Interface requirement:	IR-1: Manage usage data
Description:	In this screen user should be presented with a list of filenames from which usage data was already imported. Along with a filename system should show how many retrieval records were imported and date-time of the import. User should be given an option to select a filename and remove from the database usage data that was imported from that file. Also user should be given an option to import new usage data by selecting a source log file on the local disk.
Relevant:	Realizes: Win condition 2.2 , SC-1
Priority:	High

Table 27 IR-2: Generate new analysis report

Interface requirement:	IR-2: Generate new analysis report
Description:	The form should have the following fields: <ul style="list-style-type: none"> - analysis report name - analysis period start and end dates (might be in form of a calendar with selectable dates) - reinforcement function – single choice list - button to start the analysis
Relevant:	Realizes: Win condition 2.1, SC-3
Priority:	High

Table 28 IR-3: Abort report generation

Interface requirement:	IR-3: Abort report generation
Description:	During the report generation the system should provide a progress bar / indicator displaying estimated time left and a button to stop generation. If user aborts generation the system should leave the database in the same state as before the generation.
Relevant:	Provides user ability to cancel the generation if it's too time / resource consuming, or it was started with wrong parameters.
Priority:	High

Table 29 IR-4: Manage analysis reports

Interface requirement:	IR-4: Manage analysis reports
Description:	<p>In this view user should be presented with a list of generated analysis reports. Each report might be identified by the following attributes: name, author, date. Also upon selection other information associated with report might be displayed: analysis parameters, analysis period, number of nodes, number of edges, size etc.</p> <p>User should be able to visualize or delete any report of interest. Thus might be done through buttons available upon selection of a report. Visualization might be available through double-click on a report.</p>
Relevant:	Outlines report management view
Priority:	Medium

Table 30 IR-5: Visualize analysis report

Interface requirement:	IR-5: Visualize analysis report
Description:	<p>After opening analysis report user should be provided with the following:</p> <ul style="list-style-type: none"> - 3d hyperbolic view of the collection structure tree - navigation of the tree using mouse: rotation pivoted at the center of the current view, panning (left, right, up, down). - centering of the view on a clicked item and orienting a tree such that all item ancestors are on the left, and all descendants on the right - displaying associated usage statistics for selected item
Relevant:	Realizes: Win condition 3.4
Priority:	High

4.1.2 Command-Line Interface Requirements

The proposed system will be equipped with graphical user interface.

4.1.3 Diagnostics Requirements

User does not require any specific debugging information to be provided. Thus debugging capability will be determined during the development phase and tailored to the needs of the development team and system maintainer.

4.2 Hardware Interface Requirements

No hardware devices are accessed directly by the system

4.3 Communications Interface Requirements

The proposed system is a standalone application that does communicate with remote computers.

4.4 Other Software Interface Requirements

Table 31 IR-6: log format support

Interface requirement:	IR-6: log format support
Description:	The analyzer module should support the logging format that is used by the web server.
Relevant:	Necessary for SR-1
Priority:	High

5. Level of Service (L.O.S.) Requirements

Table 32LR-1: System Dependability - Stable data import/export

Level of Service:	LR-1: System Dependability - Stable data import/export
Description:	The system should be able to perform importing/exporting data as many and frequent as desired without crash or unexpected restart.
Measurable:	System dependability should be measured as the rate of successful importation and exportation over all importation and exportation which is logged on system performance log. And the integrity of data processing (No loss of data)
Achievable	Valid format check before and after the importing/exporting process. Robust error handling mechanism that suggests alternative way to achieve a task when the first way failed. Keep error log.
Relevant:	SR-1: Import usage data from ORACLE-formatted file. SR-2: Remove usage data from the specified log file SR-7: Omit mal-formatted retrieval records. SR-8: Omit usage log file with 0 valid records. SR-9: Do not create analysis report in case of abortion OCD 4.4 LS-1: Dependability
Specific:	SR-1 the system should import data from ORACLE-formatted file. For SR-2, removing previously imported data from the system so that other set of data can be imported. Format check for SR-7 and file integrity check for SR-8 In case of user abortion, analysis report is not created and the data remain unchanged. (SR-9)
Desired level	The system should not crash while importing and exporting data with the successful importation and exportation rate of 95% of all times when running constantly with the 10 average times of importing and exporting a day for 5 days. No loss of data on unexpected crash while exporting. No program crash with any fatal error on importing/exporting modules.
Accepted level	The system should not crash while importing and exporting data with the successful importation and exportation rate of 90% of all times when running constantly with the 10 average times of importing and exporting a day for a day. No loss of data 90% of time on unexpected crash while exporting.
Priority:	High

Table 33 LR-2 : Usability – User-friendly interface for viewing item relationships and updating data

Level of Service:	Usability – User-friendly interface for viewing item relationships and updating data
Description:	Provide user-friendly interfaces for viewing item relationships and updating data
Measurable:	User will provide usability feedback Correctness in visualizing the item relationships as a graph. Visualization of the item relationship as a tree in 3D. Data integrity when loading several data files consequently.
Achievable	Internal and external usability test. Algorithm verification mechanism for correctness in item relationships. Using Graphic API for 3D display.
Relevant:	SR-2: Remove usage data imported from the specified log file. SR-3: Relationship generation SR-4: Generate collection structure tree. SR-6: Visualization. OCD 4.4 LS-2: Usability
Specific:	Removing the previously loaded data to load another set of data (SR-2). In SR-3, once relationships are generated, a graph is produced, and the graph is in the form of tree structure (SR-4). All the generated relationships and related data is displayed in 3D hyperbolic view so as to enable users to traverse the relationship graph just by clicking (SR-6).
Desired level	Usability feedback with overall score 8 points out of 10. 99% correctness in visualizing the relationships. Flawless 3D view. No loss or corrupted data.
Desired level	Usability feedback with overall score 6 points out of 10. 90% correctness in visualizing the relationships. No loss or corrupted data 90% of time.
Priority:	High

Table 34 LR-3: Usability – Maximizing the usability of host resources

L.O.S. Requirement:	LR-3: Usability – Maximizing the host resources
Description:	Maximize the host resources by which all the computationally / time heavy system procedures which do not require user participation should be done in background allowing parallel usage of the host computer for light / routine tasks (internet, email, word editing etc.)
Measurable:	Usability of a host while performing a heavy system task can be tested experimentally, and measured by number of routine tasks user can perform per unit time.
Achievable:	The proposed system will not interfere with standard OS task-switching policies. Non-busy waits might be introduced to release host resources for other tasks.
Relevant:	OCD 4.4 LG-2: Usability
Specific:	It conforms to OCD 4.4 LG-2 Usability providing better usability of host resources.
Desired level:	User can proceed with routine tasks with minor performance degradation
Accepted level:	User can accomplish main routine tasks with reasonable performance
Priority:	Medium

Table 35 -4: Performance – Organizing data meaningfully for users

L.O.S. Requirement:	LR-4: Performance – Organizing data meaningfully for users.
Description:	Data should be organized in a way that is meaningful to both archive staff and researchers.
Measurable:	Archive staff will provide usability feedback Data successfully apply to research publications
Achievable:	Export file in the format that is conventional among digital library related people.
Relevant:	SR-5: Graph node statistics OCD 4.4 LG-3: Performance
Specific:	By storing meta-data for each node in the resulted graph, allows for relevant data to be generated in any type of format
Priority:	Medium

Table 36 LR-5: Performance- data of current scale

L.O.S. Requirement:	LR-5: Performance - data of current scale
Description:	The system should be able to produce analysis reports and visualize them for input data of size and complexity as for the Digital Archive collection in the current state.
Measurable:	The system should be able to generate, cluster and visualize for graphs with desired number of nodes.
Achievable:	H3viewer visualization framework is claimed to be able to visualize graphs of up to 300000 nodes. This fact and possibilities for increasing the number of nodes will have to be researched.
Relevant:	SR-3: Relationship generation SR-4: Generate collection structure tree SR-5: Graph node statistics SR-6: Visualization OCD 4.4 LG-3: Performance
Specific:	The number of relationship generated (SR-3), the size of structure tree (SR-4), the statistics of each node in the resulted graph (SR-5), and the scale of visualization are all dependent on the capability of the H3viewer framework. Current scale of data set is, hence, equivalent to the number of possible nodes, which is 300000 (SR-6).
Desired level:	The system should be able to handle graphs with up to 300000 nodes.
Accepted level:	System is able to handle most of the analysis report generation requests submitted by the user. (constrained analysis period might reduce number of nodes in the graph and its complexity)
Priority:	High

6. Evolution Requirements

6.1 Capability Evolution Requirements

Table 37 ER-1: Advanced analysis

Interface requirement:	ER-1: Advanced analysis
Description:	Based on successful implementation of basic analysis functionality additional types of analysis will be implemented: - extracting of user relationships based on similarities in their retrieval patterns - determining the most 'important' / influential users of the collection
Relevant:	Realizes: Win condition 5.2
Priority:	Medium

6.2 Interface Evolution Requirements

Table 38 ER-2: Interface with other usage log formats

Interface requirement:	ER-2: Interface with other usage log formats
Description:	The system should be designed in modular fashion so that it would be easy to write additional module that would parse usage log files of different formats but containing all necessary information
Relevant:	Provides for system extensibility, and future reuse
Priority:	Medium

6.3 Technology Evolution Requirements

The standards and interfaces that the system is relying on are relatively static (Apache server log format, SQL) and even if changed require minimal tailoring.

6.4 Environment and Workload Evolution Requirements

Table 39 ER-3: Accommodating collection growth

Interface requirement:	ER-3: Accommodating collection growth
Description:	The system should accommodate collection growth that is

	most probable in near future. With current Digital Archive collection of about 300000 system should be able to function properly with collection increase of 10%.
Relevant:	Collection is predicted to grow.
Priority:	Medium

6.5 Level of Service Evolution Requirements

Table 40 ER-4: Workload growth should affect system performance in linear proportion

Interface requirement:	ER-4: Workload growth should affect system performance in linear proportion
Description:	Projected workload growth of 10% should deteriorate system performance in linear fashion – that is by 10% * some scalar. System should not exhibit exponential or worse performance deterioration, since that will make it less usable.
Relevant:	Realizes Win Condition 3.3, LG-4, LG-5
Priority:	Medium

7. Common Definition Language for Requirements

Digital Archive (DA)

Collection of electronic and multi-media documents, such as images, photos, audios, and videos

SQLite

a small C library that implements a self-contained, embeddable, zero-configuration SQL database engine.

GNU

An on-going project of the Free Software Foundation to create a complete, freely distributable, POSIX-compliant computing environment.

Operating System (OS)

Software that shares a computer system's resources (processor, memory, disk space, network bandwidth, and so on) between users and the application programs they run.

Application Program Interface (API)

the interface (calling conventions) application programs use for accessing services provided by some lower-level module (such as the operating system or JVM)

Architecture Review Board (ARB)

developer or property owner controlled committees responsible for the interpretation and application of property covenants to proposed construction and remodeling projects in a given community

Apache

An open source Web server originally formed by taking all the "patches" (fixes) to the NCSA Web server and making a new server out of it